# A LOAD-CELL BASED IN-BED BODY MOTION DETECTION AND CLASSIFICATION SYSTEM

By

Musaab Adil Alaziz

A Dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Yanyong Zhang

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

October, 2017

## ABSTRACT OF THE DISSERTATION

# A load-cell based in-bed body motion detection and classification system

**by** Musaab Adil Alaziz

Dissertation Director:

Yanyong Zhang

The basic necessity of sleep in our life is critically important to ensure our well-being. Sufficient sleep of good quality is highly desired in order to have enough energy to live. One of the main factors to measure sleep quality is the amount of body motion during sleep. In-bed motion detection is an important technique that can enable an array of applications, among which are sleep monitoring and abnormal movement detection. When detection is combined with classification, it can be used to detect, notify, and recognize specific events, enabling us to focus on critical tasks.

In this study, we present a low-cost, low-overhead, and highly robust system for in-bed movement detection and classification that uses low-end load cells. By observing the forces sensed by the load cells, placed under each bed leg, we can detect many different types of movements, and further classify them as big or small depending on magnitude of the force changes on the load cells. We have designed three different features, which we refer to as Log-Peak, Energy-Peak, and ZeroX-Valley, that can effectively extract body movement signals from load cell data that is collected through

wireless links in an energy-efficient manner. After establishing feature values, we employ a simple threshold-based algorithm to detect and classify movements. We have conducted a thorough evaluation, that involves collecting data from 30 subjects who perform 27 pre-defined movements in an experiment. By comparing our detection and classification results against the ground truth captured by a video camera, we show the Log-Peak strategy can detect these 27 types of movements at an error rate of 6.3% while classifying them as big or small movements at an error rate of 4.2%.

In the second part of this dissertation, we set out to achieve much finer body motion classification. Towards this goal, we define 9 classes of movements, and design a machine learning algorithm using Support Vector Machine (SVM) and Random Forest techniques to classify a movement into one of these 9 classes. In this way, we can find out which body parts are involved in every movement. For every movement, we have extracted 24 features and used them in our model. This movement classification system was evaluated on data collected from 40 subjects who performed 35 predefined movements in each experiment. The accuracy of our model is not the same for all classes of movements. On average, it correctly classifies 90% of movements. This model can be used conveniently for long-term home monitoring.

To improve the classification accuracy, we investigate more machine learning techniques. We use Random Forest and XGBoost as additional classification tools. We apply multiple tree topologies for each technique to reach their best results. After examining various combinations, we achieve the final classification accuracy of 91.5%.

Lastly, another in-bed motion detection system is built. We use a geophone sensor to detect body motions in bed, which we call *MotionPhone*. *MotionPhone* is more accurate in detecting motion but not efficient for classification purposes. We thus believe combining these two systems can give us better results. Both systems are unobtrusive, low-cost, and private, which can thus enable a large array of important applications.

# Acknowledgements

During my PhD journey, there are many people I would like to thank for their help and support. First of all, I would like to thank my advisor Professor Yanyong Zhang for her support, guidance, and the time she spent to teach me how to accomplish academic achievements. During my years as a graduate student, she gave me space to grow into as an independent researcher, aiming high, and not giving up. My Ph.D. journey would not be so enlightened, and fruitful without her.

I would like to express my sincere gratitude and appreciation to Dr. Richard E. Howard. I was fortunate to have him as my mentor during my research work at WINLAB. He added significant great values to my work with his knowledge, especially on physics, wireless and hardware. He never gets tired from helping me even if we work the whole day. He spends hours and hours and tries his best to explain things and makes problems easier to solve. I can say, he is the light, the hope, and the blessing during my Ph.D journey.

I would like to thank Professors Wade Trappe, Richard Martin, Xiaodong Lin, and Pei Zhang for their thoughtful comments and discussions throughout my research.

I would like to give a special thank to my teammate, Zhenhua Jia, who has been there always for me. He is one of the smartest and kindest students I have ever met. Our working together has made things greater and problems easier to solve.

This work would not be possible without all the subjects that participated in the studies. I would like to thank all subjects from WINLAB and my friends for their help and patience. I also would like to thank all WINLAB graduate students, especially those who participated in the study. Thanks to all people at WINLAB, students, staff, and faculty members for the great years and the company I have had.

I would like to thank my sponsor in my country, The Higher Committee For Education Development in Iraq, for their help and support.

I would like to express my earnest gratitude to my parents and my siblings, for their tremendous support and encouragement. In particular, I would thank my mother and father for their unconditional love and belief they have in me, from my early childhood, thank them for their support and help, and for their kindness and praying for me. Without them, I would never reach the level I am in now.

Finally, I owe immense gratitude and thanks to my wife and kids. Special thanks to my wife, for her support, patience, and endurance to put up with my often hectic life style during my Ph.D. Her presence, assistance, and strength gave me the drive to accomplish my Ph.D.

For all I mentioned above, you made the years I have spent in my Ph.D. program amazing and unforgettable. I couldn't complete this work without your help and support. Thank you for everything.

# Dedication

*To my Mother and Father,*

*to my Siblings,*

*to my lovely Wife,*

*and to my Kids...*

# Table of Contents

# List of Tables

# List of Figures

# Chapter 1
# Introduction

In-bed mobility measurement is a very important factor when monitoring patients or people during sleep is required. Monitoring a person's body movements during sleep can also enable an array of applications, ranging from sleep monitoring to abnormal body movements detection, such as restless legs.

A large fraction of a person's life is spent resting and sleeping. We generally consider times when we are not awake to be safe periods, free from danger and health risks. Actuality, this is not true: every year, roughly one in eight human deaths occur while people are sleeping. Many of these deaths are related to chronic health conditions, often unknown to the person. Movement during sleep can be a sign of disrupted sleep since it is associated wakefulness that effects on sleep quality. As less motion or movement during sleep, as better quality of sleep because its associated with sleep depth [2]. People may still feel fatigue and cannot concentrate even if they sleep for a good amount of time. This can be caused by a bad quality of sleep, such as having frequent periods of wakefulness during sleep. Changes in the pattern of motion activities in bed can reflect various abnormal physiological and neurological conditions. Some motor disturbances are triggered by sleep such as restless legs syndrome (RLS) and periodic limb movements during sleep (PLMS). Detecting these movements can help as a diagnostic tool. Therefore, the assessment of body movement is used as important indicator of sleep quality and depth [3].

When mobility detection is mixed with motion classification algorithms it yields to further monitoring to some particular events detection, notification and recognition.

Motion classification system can enable focusing on more critical tasks, especially at hospitals or clinics. The need for smart systems that provide monitoring options is increasing year over year as the cost of health care continues. Especially when these systems can be used at homes as well as hospitals. As a result, we believe unobtrusive, low-cost, wireless, and easy to install sensor system is the best approach for in-bed body motion detection and classification. In this dissertation, our goal is to provide a system that can accurately detect and classify any motion during sleep or in-bed resting time.

## 1.1 In-Bed Movement

Sleep state is highly connected to the major changes in motor activity [4, 5]. As lower motor activity level, as deeper sleeping state. Intermittent wakefulness during sleep is related to high activity levels and arousal that are associated with movement [6]. For that, sleep quality can be reduced by increasing mobility in bed, which can be a sign of disrupted sleep [7].

Movement in bed, by itself, can be considered as an indicator of health problems. Many illnesses, like flu, depression, or pain, can be reflected by the pattern or amount of motor activity [4, 5]. Depressed patients, according to [8, 9], show increased motion activity at night. Patients of sleep apnea also have shown an increased in motor activity at night resulting from disrupted sleep [10]. Also, abnormal movements during day and nighttime, that may adversely affect sleep, can present many neurological disorders [5, 7]. For example, in Parkinson's disease, normal body movements may be repressed by motor daytime symptoms that persist during sleep, such as a decreased ability to start and continue movements, and impaired ability to adjust body position. Sleep quality is worsen by these symptoms, and can cause discomfort and pain [7, 11]. Restless legs syndrome (RLS) and periodic limb movements during sleep (PLMS) are also another examples of motor disturbances that are triggered by sleep. With restless leg syndrome

(RLS), patients report feelings of discomfort in the legs, and they feel compelled to move (for example, tossing and turning in bed) to relief the discomfort [12]. Sleep is disrupted by such symptoms and they cause daytime tiredness and sleepiness [7]. At least 80% of patients with RLS have PLMS and may provoke frequent arousal or even awakenings. PLMS are involuntary, repetitive movements, and most typically seen in the lower limbs but sometimes seen in the arms [7,12]. These motor disturbances, most of the time, are ignored by the sleeping patient for a long time. According to that, the need of monitoring in-bed movement during sleep or resting time is increasing year over year.

## 1.2 Techniques for Evaluating Motor Activity

Activity monitoring, or actigraphy, is the simplest means of measuring movement. It is the most direct and specific technique for quantifying and recording movements [5]. Other ways of activity monitoring are performed through obtaining information about the nature of the movement from the patient or overnight polysomnograph recording [7]. Understanding the nature of the movements is the main factor to do the assessment of nocturnal motor disturbances. Diagnostic, in general, is based on information provided by patients. The main factors used for evaluating motor activities are the type of movements, frequency, and duration [13,14].

Additional techniques may include overnight polysomnograph recording. The gold standard to evaluate and study abnormal motor events occurring during sleep is Video-polysomnography (VPSG) [5, 7]. It combines simultaneous audiovisual monitoring and recording with the traditional PSG recording of the patient in the sleep laboratory. Continuous recordings of several physiological measures should be done in PSG, including brain waves (electroencephalography), electrical activity of muscles, eye movement (electro-oculogram), breathing rate, blood pressure, blood oxygen saturation, and heart rhythm. Application of additional leads is required to other parts

of the body (for example,arms and legs) if there is a specific motor complaint [5, 7]. It requires at least a full nights stay in a sleep laboratory attended by properly trained technicians [7, 15].

Expensive techniques are required for long-term assessment and behavior therapy. Most of these techniques are inconvenient in the way that they need some equipment or devices to be attached to a person's body. Moreover, additional information should be provided by patients to make more accurate evaluation. Therefore, more convenient ways are highly required to match the improvement in technology in nowadays.

## 1.3 Proposed Work

We propose an accurate, robust, low-cost, and easy-to-use in-bed body movement monitoring system, which is centered around low-end load cell sensors. The system consists of both hardware and software components. Its hardware components include load cell sensors, an amplifier, a power control circuit, and a wireless communication unit (which consists of an A-to-D converter); software components involve interpolation, normalization, filtration, feature extraction, and detection and classification. Our system can detect many different types of body movements, ranging from turning over to tiny hand movement, and can classify these movements. It can simultaneously detect and classify movements into 9 classes: turning/rolling right, turning/rolling left, right hand, right leg, left hand, left leg, legs, head, and combined motions. Our classifier model is a multilevel binary decision tree that uses SVM model in each decision step to classify motions into the right branch. 24 features are extracted from each motion to be used for training and testing purposes. Random Forest technique has been used to compute feature importance in each level and select the best features. We added two machine learning techniques to improve the classification accuracy. These techniques are Random Forest and XGBoost. All three techniques are combined in a logical way to have one final result at the end.

## 1.4 Organization

The remainder of this dissertation is organized as follows. In Chapter 2, we summarize the existing bed-mounted body movement monitoring systems, and compare their pros and cons. In Chapter 3, we describe the hardware system design of *MotionScale* system, and its signal processing algorithms. We also present our evaluation setup and experimental results. In Chapter 4, we describe our classifier system, which we call *MotionTree*. we describe *MotionTree*'s signal processing algorithms with the classifier construction. We present our evaluation setup and experimental results in Section 4.4. In Chapter 4 we also describe the using of Random Forest and XGBoost as another classification tools. We also present a combination algorithm to have one final result from all techniques at the end. In Chapter 5 we introduce another system that uses a geophone sensor to detect in-bed movements. Finally, Chapter 6 concludes the dissertation and proposes the future steps.

# Chapter 2
# Related Work

In the past years, many bed motion sensing systems have been developed. Most of these systems used high-cost sensors, complicated signal processing techniques, and wired communication. Also, many of them are only focused on the movements detection without movements discrimination. Moreover, some of these systems require special mattress which may reduce their wide use. In this section, we describe some of the existing in-bed motion detection systems.

## 2.1 Movements Detection Systems:

Kortelainen et al. [16] proposed a movement detection system, including heartbeat and respiration, using a foil pressure sensor placed inside the mattress. It requires special mattress with wired communication and it does not classify the type of movements. Watanabe et al. [17] developed a noninvasive pneumatics-based system that uses an air cushion and a pressure sensor. The air cushion is placed under the mattress while the pressure sensor detects the change of the pressure due to body movements, respiration, and heartbeat. This system needs special cushion with wired communication. Aubert et al. [18] proposed to use an electric foil pressure sensor to detect three vital signs during sleep, namely, heartbeat, respiration, and activity index related to body movements. This sensor needs specific technical installation to be placed in the thorax region under a thin mattress. Nukaya et al. [19] proposed a bed sensing system by using piezoceramic bonded to stainless steel plate sandwiched between the floor and bed legs. This system can sense many human bio-signal, including body movements.

It needs high-cost sensor and it does not classify movements. Yamana et al. [20] developed a non-constraint cardiac vibration, respiration, and body movement monitoring system. It has a 40-kHz ultrasound transmitter and receiver pair. The transmitted signal is reflected on the mattress under-surface, and the received signal is processed to know the information about human vital signs, including body movements. Special, hard, installation is required for this system. Brink et al. [21] proposed a non-contact sensing system of in-bed heartbeat, respiration, and body movement. This system uses four sensors, one in each corner of the bed. Each sensor is composed of two aluminum plates and reflex light barrier in between. The reflex light barrier senses the distance between the plates. This distance changes with the amount of applied force. There is no movement classification in this study and it uses wired communication. Harada et al. [22] proposed a human's body in-bed movement sensing system. It can detect human's existence, posture, articular movement, and respiration. This system uses a special, high-cost, sheet with 210 pressure sensors. Joned et al. [23] proposed a movement Identification system using pressure sensor array. This system needs special handling of bedding and it cannot classify the type of movements. Tamura et al. [24], and [25] proposed systems to detect body movements during sleep by temperature monitoring. The proposed systems consist of 16 temperature sensors. Each system requires special installation for these sensors in the mattress. Hoque et al. [3] proposed a Wireless Identification and Sensing Platform system (WISPs) for monitoring body position and movements during sleep. The WISP tags are attached to the bed mattress to collect accelerometer data from them. Movement and body position can be detected. This system does not give any information about movement's type. Walsh et al. [26] proposed a system composed of a grid of 24 fiber optic based pressure sensors integrated into a foam mat. The proposed system can detect movements without classification. Adami et al. [27] proposed a system to detect and classify in-bed movements. It uses load cell sensors, one under each of the bed's legs. This system is very close to our system but it uses wired communication with complex signal processing techniques.

In [2], [28], and [29], load cell sensor is used to detect movements. All these systems don't give any movement classifications and use wired communication. Spillman et al. [30] proposed a fiber optic system for monitoring patient respiration, heart rate, and movements without classification. Nishyama et al [31] developed a system to monitor respiration and body motions. Pressure sensors based on hetro-core fiber optics are used in this proposed system. This is high-cost system and requires special installation. Hao et al. [32] proposed a system for sleep quality monitoring. This system uses a smart phone with an app called iSleep and the phone should be placed somewhere close to the bed. The built in microphone in the smart-phone is used to detect the required activities such as body movement, couch, and snore. The proposed system does not classify the type of movements. Rofouei et al. [33] proposed non-invasive, wearable neck-cuff system capable of real-time monitoring and visualization of physiological signals. This system is used for sleep quality purposes. It has many sensors housed in a soft neck-worn collar and the data send by Bluetooth to a cell phone which stores the data. It uses accelerometer sensor for body movement detection. This system cannot classify movements and requires some intrusive system to attached to human body. Kaartinen et al. [34] proposed a system for long-term monitoring of movements in bed using static charge sensitive bed (SCSB) sensors. This system can detect body movements, respiratory movements and heartbeat. It does not classify movements and uses wired communication.

## 2.2 Movements Detection and Classification Systems:

Lu et al. [35] and [36] temperature sensors, thermistors, systems are proposed for in-bed movement detection. These systems detect torso and legs movements by using two arrays of 16 thermistors placed under the waist and under the legs. The proposed systems cannot detect head or hand movements. Also, they cannot distinguish the type

of movements. Cheng et al. [37] developed a physical activity detecting Mat (PAD-Mat) system. It has three conductive mats placed under the chest, hip, and legs. These conductive mats are made of conductive fabric to detect physical activities as electrical resistance changes. It can detect upper limit, legs, and body motions without classification. Adami et al. [27] proposed a system uses load cell sensors, one under each of bed's legs to detect and classify in-bed movements. This system can classify movements into big and small only. Adami et al. [38, 39] developed the previous system to have a Gaussian mixture model (GMM) as a classification method. It can classify movements into three classes: major posture shift, small and medium amplitude movements, and legs movements. In [40], load cells are used for periodic leg movements detection. King et al. [41] proposed a system for the measurement of periodic leg movements in sleep. It uses the Actiwatch, a wireless actigraphy device, that has a uni-axial accelerometer as sensing element. This sensor is place on the foot at the base of big toe using a tape to monitor leg's movement. Shino et al. [42] developed a system that can detect body movements and scratching motion in bed. Piezoceramic sensor sensors are employed in this system. This sensor is bounded on a stainless steel plate with a washer under the plate. Four devices are used to be put under each bed leg. It can detect turning and sitting up motions. Accelerometer sensor, angular velocity sensor, ceramic sheet, strain gauge, and microphone are attached to the hand for scratching motion detection. Ren et al. [43] proposed a system based on a pressure sensitive mat of 72 fiber optic pressure sensors to detect in bed movements. The system analyzes the center of pressure progression because of movement to provide prior alarm of falling. It can detect turning movements without classification. Bustamante et al. [44] presented a system that can detect movements and a fall before it happens. Shock sensor, accelerometers on X and Y axes, extensometric gauge, 8 plain pressure sensors, and camera with motion detection option are used. This system can detect three modes, lain down (patient position), seated (bed exist), and exit (out of bed). It can also give information about the position of the patient in which side of the bed. It can predict if the

patient is about to fall from the bed. Aronoff et al. [45] proposed a system for patients movement classification. It uses 6-axis accelerometer that is attached to the patients hospital bed. Frequency-series analysis is used to extract relevant patterns for patient movement and train a classifier to identify patients movement patterns. 12 events can be classified in this system.

We improved a system based on a wireless load cell sensor, that can detect and classify any in-bed motion from all body's parts. It is low-cost, low-overhead, highly robust, and unobtrusive system. It does not need any special designed mattress and can be easily installed at hospital or home without any subject's complain requirements.

# Chapter 3
# MotionScale

## 3.1 Introduction

The ability to accurately monitor a person's body movements during sleep can enable an array of applications, ranging from sleep monitoring to abnormal body movements detection, such as restless legs. A number of bed-mounted sensing systems have been proposed for this purpose, including pressure sensors [16, 22], temperature sensors [36], ultrasound sensors [20], load cell sensors [27] and custom-made sensors [21]. Among these sensors, load cells have been shown to provide a viable solution for several reasons. Firstly, load cells are very affordable and readily available. Secondly, deploying a load cell based system can be very conveniently done, without interfering with the bed or how it is currently used. Thirdly, load cells (when placed under the bed legs) can easily capture the changes in body weight distribution caused by movements, especially when the movements are rather noticeable. As a result, we believe that load cells could potentially offer a practical approach to on-bed body movement monitoring.

Even though earlier studies point out that low-end load cells can be integrated to beds to detect some large body movements, whether they are able to accurately detect both large and small movements at the same time still remain a question, especially due to their limited sensitivity. In this study, we set out to fill this void by designing and developing an accurate and robust body movement monitoring system based upon low-cost load cells (around $.70 per unit). We refer to this system as *MotionScale* as it can "weigh" the motions on a bed. Moreover, the entire system is considered as a

low-cost system for a big quantity.

With *MotionScale*, we can simultaneously detect both large and small movements and classify these movements. We address these challenges through the following techniques. As far as the hardware design is concerned, we have carefully designed the amplifier circuit so that the circuit can handle a wide range of movements – as large as the whole-body roll over while as small as hand movements. We have also made great effort to minimize the power consumption of the system by turning off the system when it is not needed, e.g., during the daytime. As far as the software design is concerned, we have adopted several signal processing algorithms that can efficiently extract body movement signals. Firstly, we have designed algorithms to deal with frequent packet losses due to wireless interference in the environment. Secondly, our detection and classification algorithms work across different body weights, adopting a uniform threshold value regardless of the user's body weight. Thirdly, we devise three types of features that leverage the redundancy between multiple load cells to infer different in-bed movements. Through these optimization techniques, our experimental results that involve 30 subjects show that we can detect 27 types of body movements with an error rate of 6.3%, and can classify these 27 types of movements into big and small movements with an error rate of 4.2%.

To summarize, we have made the following contributions in this study:

1. We have developed an accurate, robust, low-cost, and easy-to-use in-bed body movement monitoring system, which is centered around low-end load cell sensors. The system consists of both hardware and software components. Its hardware components include load cell sensors, an amplifier, a power control circuit, and a wireless communication unit (which consists of an A-to-D converter); software components involve interpolation, normalization, filtration, feature extraction, and detection and classification.

2. We have built a prototype and used it to instrument an experimental bed. We

have used the experimental bed to collect load cell signals from 30 subjects who make 27 different body movements during each experiment. We have compared the detected body movements against the ground truth observed captured by a video camera, and found that the average error rate is 6.3%.

3. We have also used the same data to classify these 27 body movements into big movements (those that involve the entire body) and small movements (those that only involve one part of the body). We compare the classification results against the ground truth observed by a video camera, and found that the average error rate is 4.2%.

The remainder of the chapter is organized as follows. In Section 3.2, we describe the hardware system design of *MotionScale*, and in Section 3.3, we describe *MotionScale*'s signal processing algorithms. We present our evaluation setup and experimental results in Section 3.4. Finally, we provide concluding remarks in Section 3.5.

## 3.2 System Overview

In-bed body motion detection can facilitate a variety of research in Human Computer Interactions (HCI), smart home, and health-care, such as home environment control, sleep monitoring, etc. Our main goal is to detect in-bed body motions by utilizing low-cost, low-overhead sensing techniques. Toward this end, we devise a motion detection system based on low-cost load cell sensors. The system can be easily integrated to an existing bed by placing the load cell sensors under each bed leg. The basic idea is to observe the electrical resistance changes on each load cell to infer possible body motions on the bed. Intuitively, when a body motion occurs, the body weight distribution changes, causing each load cell's resistance to change accordingly.

In this work, we also focus on utilizing the relative load cell resistance changes to discriminate two types of body movements: *Big Movements* and *Small Movements*. Big Movements usually happen when there is a motion in the body's torso, such as

turning to the left or right, and Small Movements happen when just a small part of the body moves, such as re-positioning the arm or head. Since our system can accurately detect in-bed body motions using load cell sensors, we refer to it as *MotionScale*.

As illustrated in Figure 3.1, in *MotionScale*, each load cell sends its data using a PIP-Tag (the wireless communication protocol described in Section 3.2.2) with a sampling rate of 30 Hz. The base station, which is connected to the USB port of a laptop, conducts the following processing after receiving the data:

1. *Data Interpolation.* We first interpolate the data by applying the spline interpolation technique.

2. *Data Normalization.* We normalize the data by using subject's weight. Weight is also computed by our system. Because the system aims to detect motions, it just focuses on the segment of data that contains large changes or oscillations. To achieve this, the system performs *Local Mean Removal* to remove the constant value in the load cell data by using a sliding window. we determine the segment of data only contains large changes and oscillations.

3. *Data Filtration.* We filter the data by low pass filter with 10 Hz as a cutoff frequency. We remove the high frequency spikes or noise by this filtration.

4. *Feature Extraction.* We investigate three different features in this study, i.e., peaks in log-scaled sum of the square of the data (Log-Peak), peaks in the energy of the sum of the data (Energy-Peak), and valleys in zero crossing of the sum of the data (ZeroX-Valley).

5. *Motion Detection and Classification.* Using these features, we detect body movements and classify these movements as big or small movements using a simple thread-based scheme.

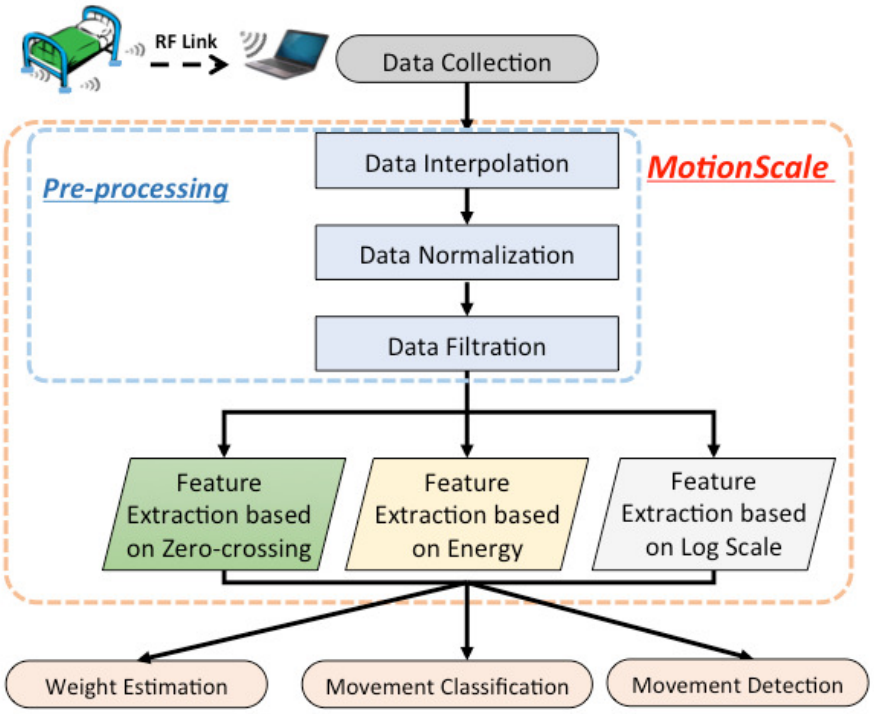



Figure 3.1: Overview of system flow

### 3.2.1 Design Challenges

Building *MotionScale* involves a number of challenges in design and implementation:

**Load Cell Installation.** Installing load cells under bed legs requires some careful consideration; direct installation may cause imbalance in pressure distribution on the surface. To address this challenge, we have designed a docking station for each load cell. This docking station consists of a washer that is placed under the load cell and a metal disk that is placed above with some pasting and cutting operations to get it fit.

**Power Supplies.** Each amplifier circuit need +3V, 0, and -3V. That could be something difficult because we use four load cells. We built 2 power sources with 3V and connect them together to provide all required voltages. We used these 2 power sources to feed all the four amplifier circuits. This cannot be done easily on different places and different beds. We, probably, need separate power supplies for each circuit in a

different environment.

**Packet loss.** Using wireless communication to transmit signal can lead to packet losses due to interference in the environment. For example, in our system, we observe an average packet loss rate of 10%. To address this challenge, we take into consideration the fact that we have multiple load cells in the system and there is sufficient redundancy in the data. Therefore, we use interpolation techniques to overcome the missing data problem, which will be explained in detail in Section 3.3. Moreover, we strive to minimize the packet losses through careful placement of the system, especially the base station.

### 3.2.2 MotionScale Hardware Design

Our *MotionScale* system consists of four major components: a load cell circuit, a differential amplifier circuit, a power switch circuit, and a wireless communication component. The load cell measures the voltage change due to motion. Because the raw voltage change values are usually very small, it is hard to accurately measure them directly. In order to capture such small changes in voltage, we design a differential amplifier circuit to amplify the raw voltage measurements for subsequent processing. In order to reduce its power consumption, our system exploits a power switch circuit that can switch the load cell and amplifier on or off, depending on a control signal from the communication component. In addition, we leverage a RF Transmitter (referred to as PIP-Tag, designed in our group [46]) that converts analog voltage signals to digital values, and sends the digital values to the basestation Unit through low-energy wireless communications. The basestation is connected to a laptop through a USB port, from which we receive data for subsequent processing.

**1) Load Cell Circuit:** Our load cell circuit uses a half Wheatstone bridge, as shown in Figure 3.2 [47]. Specifically, it has two fixed-value resistors (the two resistors on the

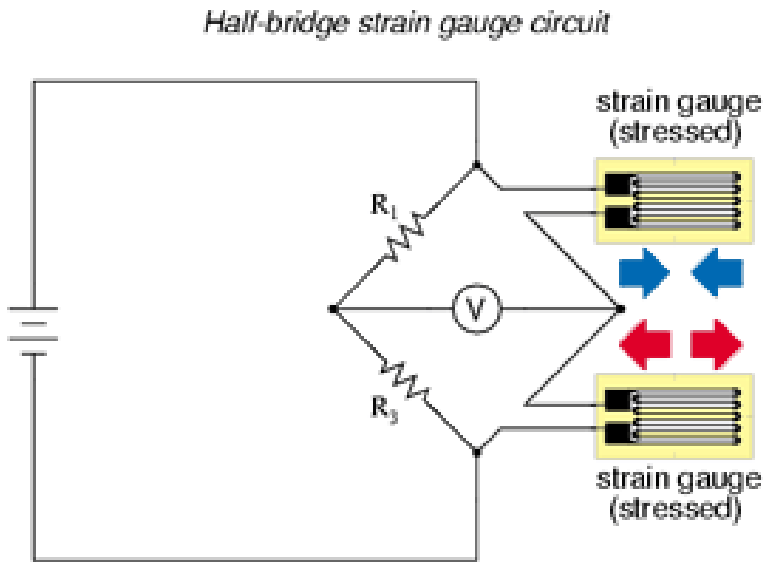



Figure 3.2: Half bridge strain gauge circuit.

left-hand side of the bridge in the figure) and a three-wire load cell (shown on the right-hand side of the bridge in the figure). The voltage between the connection of two fixed resistors and the ground is a fixed value with/ without stress. The three-wire load cell is made of two single strain gauges in series. When the three-wire load cell is stressed, one of the strain gauges is compressed and results in a decreased resistance, and at the same time the other strain gauge is stretched and leads to an increased resistance. Thus, the voltage between the connection of these two strain gauges and the ground increases as a response to the introduction of the external weight.

The output of the load cell circuit is thus the voltage between these two connections, which is linear to the weight value

$$V_{in} = V_{in}^{+} - V_{in}^{-} = (\frac{R_3}{R_3 - R_4} - \frac{R_2}{R_1 - R_2}) * (V_{cc} - V_{ss}). \tag{3.1}$$

In our system, we use Generic YZC-161B load cells with a nominal load of 50 kg, as shown in Figure 3.3, which costs around $.70 and is generally used as a weighing scale.

**2) Differential Amplifier Circuit:** In general, the output of such a load cell is rather small – the maximum voltage change of the load cell is less than 6 mV under stress

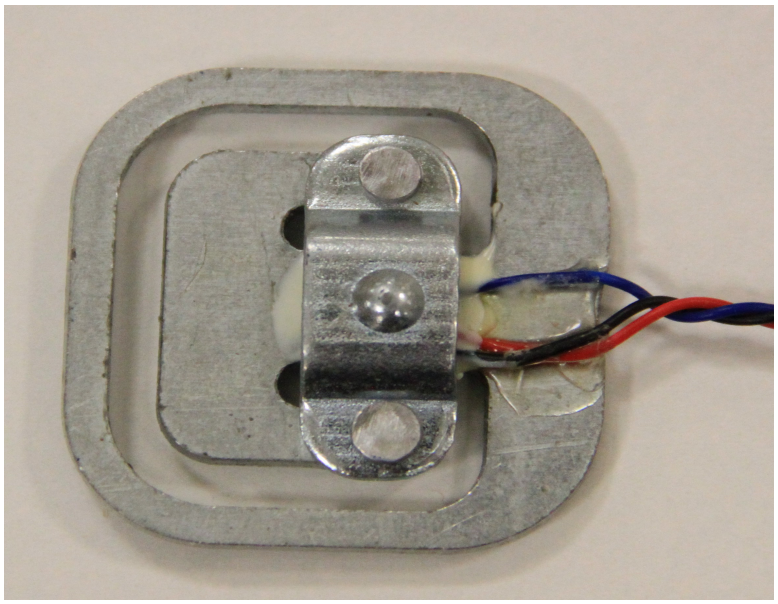


Figure 3.3: The Load Cell.

corresponding to the gravity of 50 kg object/subject (its capacity). This makes it difficult to get the accurate value from the A/D converter. Thus, we use a Differential Amplifier circuit to enhance the signal that we get from the strain gauge. The amplifier model that we use in our project is INA126 [48].

$$V_{out} = G * V_{in} = (5 + \frac{80k\Omega}{R_g}) \tag{3.2}$$

**3) Power Switch Circuit:** We also use a power switch circuit to turn on/off the power supply, which is a simple p-MOS FET and n-MOS FET circuit. The switch is controlled by the transmitter tag, which uses a pin to pull down to the ground to turn off the circuit or pull up to 3V to turn on the circuit. In this way, we can conserve a significant amount of energy when the measurement is not needed.

The whole circuit is shown in Figure 3.4.

**4) Wireless Communication Component:** We use a wireless communication system developed in our group (details can be found in [46, 49]) to convert the analog signal to digital values and then transmit them through wireless links. The system consists a transmitter that we refer to as PIP-Tags which contains a 10-bit A/D converter with the range of 0 to 1.5V. It is low-cost, low-power, and easy to program. The PIP tag has

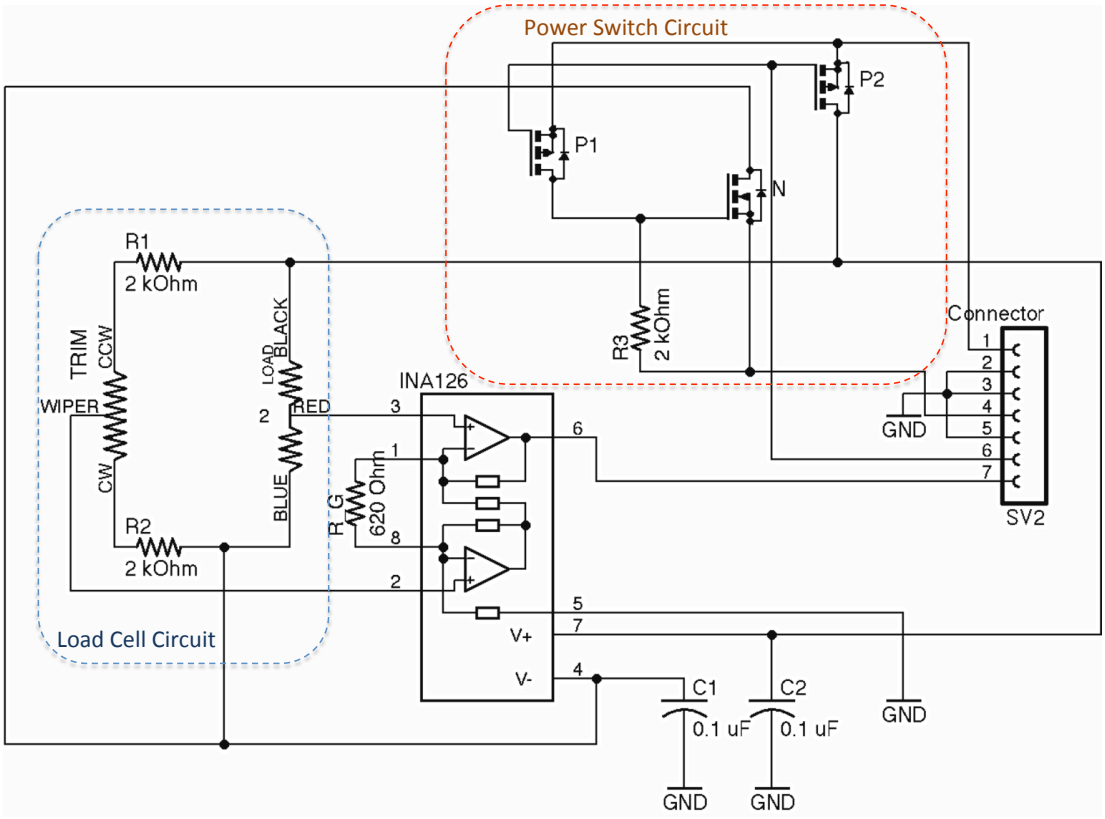



Figure 3.4: The whole electrical circuit

its own processor and radio transceiver. The basestation has the same hardware as the PIP-Tags, with a tuned 900 MHz monopole antenna attached. The basestation is also equipped with a standard USB connection for data transfer to the laptop which runs the signal processing algorithms. In fact, we could have used the Blue-tooth devices in our system, but we would not get the low power consuming system like what we have. Moreover, PIP-Tags work reliably in a system with hundreds of sensors in small space with large number of small packets.

**5) Assembling a *MotionScale* System:** Our *MotionScale* system consists of four load cells, with each load cell placed under a leg of our experimental bed. We have 2 power supplies to provide the +3, 0, and -3V for all of the circuits. There is only one receiver that can be connected to any USB port in a laptop. This receiver collects all the data from the four load cells and transfer the data to a processing unit.

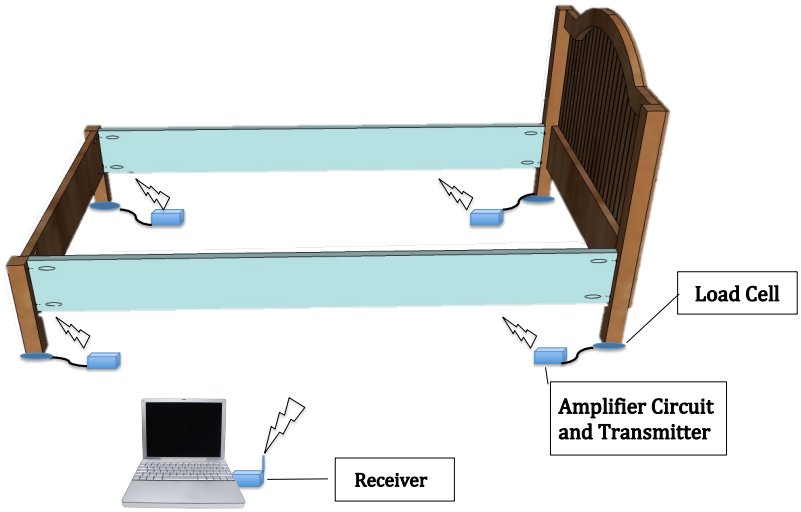



Figure 3.5: The general overview of our system.

## 3.3 MotionScale System Design

In this section, we explain how our system is designed to process the data from load cells to mitigate noise and further detect in-bed motions. The detection results can not only detect motions, distinguish big and small motions, but also can determine when the user lies on the bed, leaves the bed, and moves on the bed, which can facilitate a variety of applications in smart home and health-care.

### 3.3.1 Data Pre-processing

After raw data are collected from load cells, our system first performs a sequence of preprocessing steps to remove noise and determine the important segments that contain the data corresponding to motions on the bed. The Data Pre-processing is in three steps. First, the received data are interpolated to balance the samples that are unevenly distributed in time due to packet losses. Second, we normalize the data by estimating the body weight of the user and remove this constant bias from the data. Third, the system drops the local mean in the data and apply filtration with low pass filter to further remove the high-frequency noise.

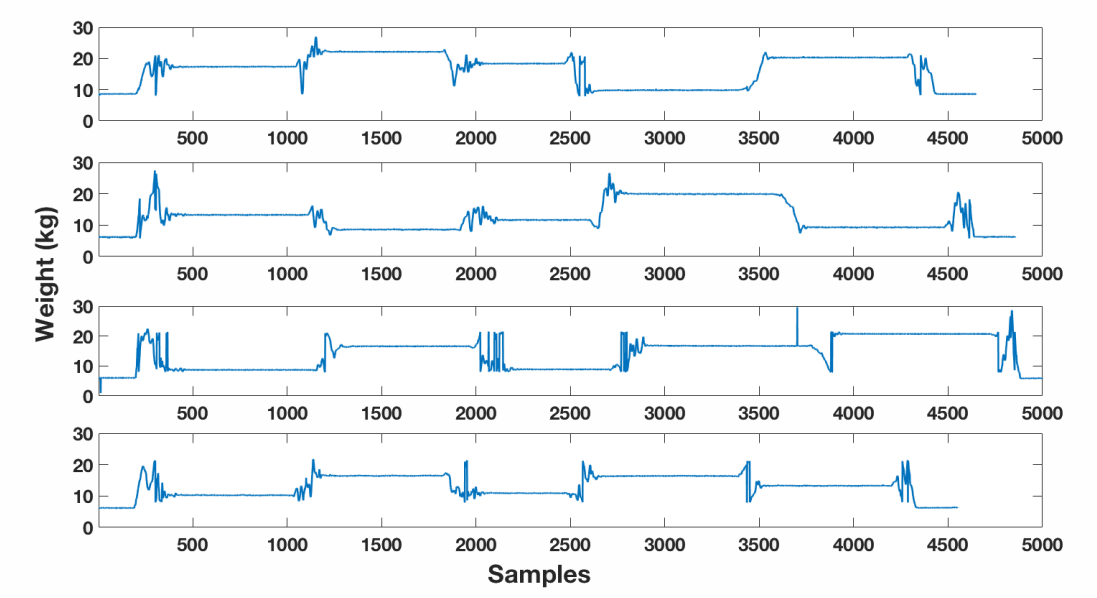
**Data Interpolation.** In our work, multiple load cells use PIP-Tags to transmit

the measured data in real time through wireless communications. Due to the high-noise nature of wireless communications, it is common to find that some data may be missing or have large errors when data packets from different load cells collide or corrupt with each other. To illustrate this, we conduct an experiment by collecting data from four load cells placed under the legs of a bed when a participant is asked to lie on the bed and perform some movements, such as turning left/right. Although the sampling rates are set to the same value on each of the load cells, the total number of data measurements received from four load cells are different. For example, when the sampling rate on load cells is set to $30$ Hz, we find that about $10\%$ of the total measurements from four load cells are missing in a time period of $3$ minutes because of packet collisions. Such inconsistency in data from different load cells would severely impact the motion detection because the data cannot reflect the weight variation on different load cells during the same time period that has the same motion. In order to align the data from different load cells, we apply the spline interpolation technique to the data from different load cells to make sure the data from different load cells have the same length. From our experiments, we find that the frequency of most body movements is less than $4$ Hz [2], and therefore, even after losing about $10\%$ of the packets, we still have sufficient data samples for body movements according to the Nyquist Theorem [50]. Figure 3.6 presents the three-minute data measurements before and after the interpolation. In the original data, we can see that the data from the third row, for example, has about $5200$ samples, which is more than others, and the variations caused by the user's movements are not properly aligned between different rows. After the interpolation, we have equal length and aligned activities in all rows.

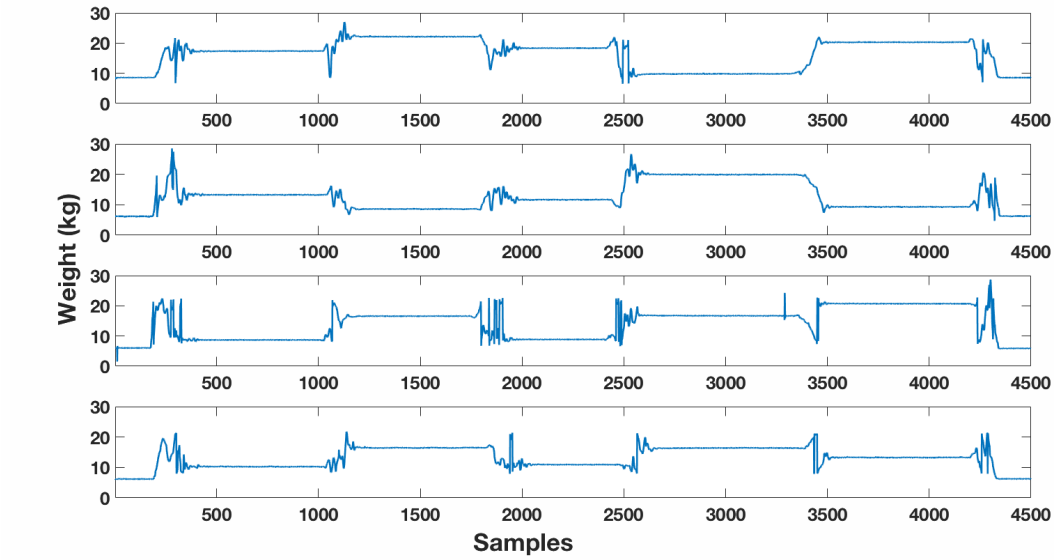
**Calibration and Weight Estimation.** Each load cell circuit and amplifier circuit has a voltage regulator to adjust the amplified voltage. However, that regulator is not exactly the same to all circuits. We propose to use calibration equations to determine the relationship between voltage values and corresponding weight values. Calibration was done separately to each load cell. We applied some known weight to each load

cell and observe the corresponding output voltage, from which we derive the weight calculation parameters for each load cells. This step is a one-time effort, and the derived parameter values can be applied to all subsequent experiments. Our system can estimate the body weight with resolution of about $200$ g.

In the experiment, we first collect weight measurements from the empty bed for $10$ seconds, and then ask the participants to get in the bed and record the weight measurements. Intuitively, to determine the weight of a participant on the bed, we first need to remove the weight of the bed and find the mean of the raw weight data.




(a) The data before Interpolation



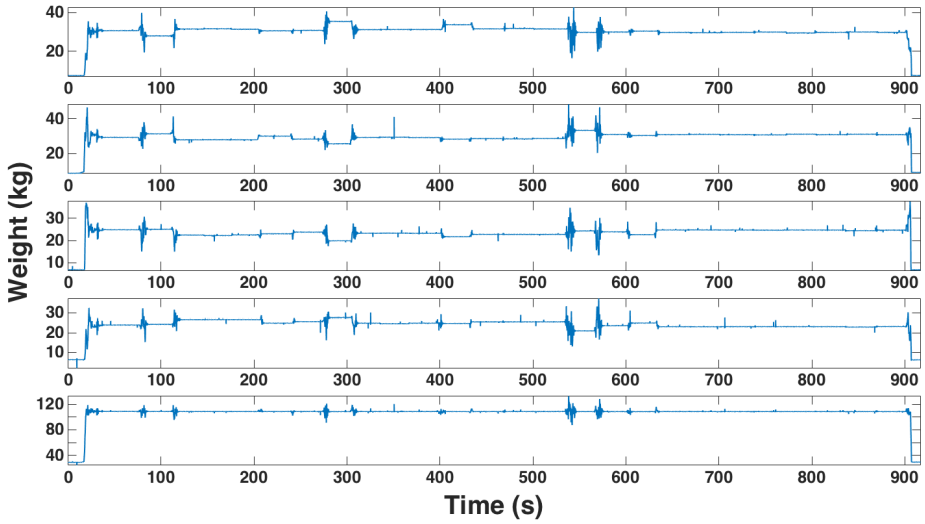

(b) The data after Interpolation

Figure 3.6: The four raw of data before and after doing the interpolation on the data of 3 minutes experiment with some movements of a subject on the bed.

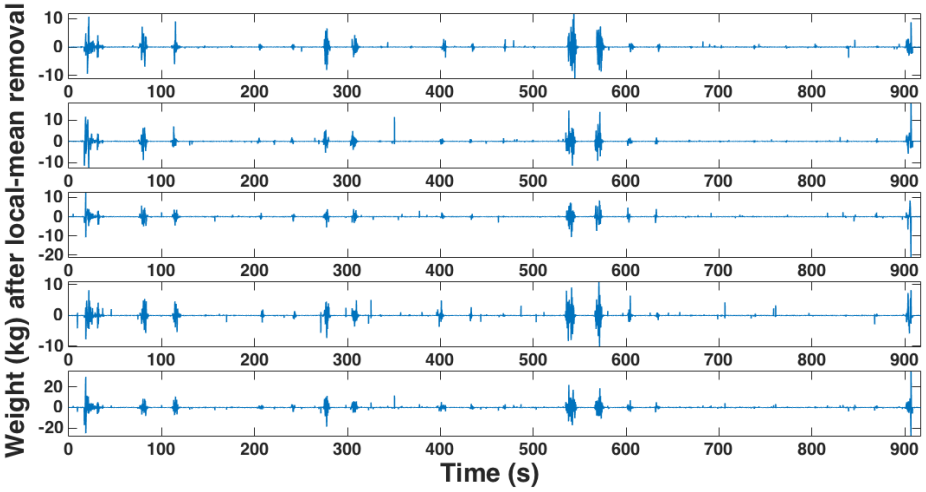
**Mean Removal and Filtration.**

We find that when in-bed movements happen, the weight measurements from load cells have oscillations. However, the oscillations are not obvious in the raw data. After

we remove the local mean from the raw data, the oscillation associated with every movement become more noticeable. In particular, we calculate the mean values in a moving time window of 50 samples. Figure 3.7 shows the weight measurements before and after removing the local means, where the data is collected from one of our participants with 27 in-bed movements. We can see that local mean removal can amplify the oscillation to improve the movement detection.




(a) The data before removing the local mean




(b) The data after removing the local mean

Figure 3.7: The raw of data with their summation before and after removing the local mean. Local means 50 samples.

After that, the data is filtered using a low pass filter of a 10 Hz of cutoff frequency.

### 3.3.2 Feature Extraction

Next, we extract features from the preprocessed data and adopt a simple threshold-based detection/classification strategy. Below, we describe the three features we have explored and the corresponding detection schemes one by one.

**Log-Peak Feature Extraction:** Log-Peak uses the logarithm of a physical quantity instead of the quantity itself, which has the potential to have a good view to show both small values and large ones. Moreover, we find a way to merge the four signals into one and supply it to the log scaling to simplify the further processing. Specifically, we first square every raw data signal collected from each load cell sensor and sum them up to create a new merged signal, which is the summation of their squares. To make the system applicable for all people or subjects, we normalize the merged signal by dividing by the subject's weight. Log (i.e., natural log where log to the base e) is applied to the merged signal. The output of log scaling is very messy and doesn't reflect any good information, therefore we apply a low pass filter with a cutoff frequency 0.2 Hz to the log's output. We get an observable pattern with a clear peak whenever there is an in-bed body movement. Figure 3.8 shows the squares of the four raw signals and their summation in the bottom line. Figure 3.9 shows the log output of the normalized summation result before and after filtration. Threshold is applied to find peaks, which also means to find movements. Also, another threshold is applied to classify these movements as big and small. More details about thresholds and movements classification will be discussed in 3.4.

**Energy-Peak Feature Extraction:** We observe that there exists stronger oscillation with high amplitude in the collected load cell readings if an in-bed movement is performed. It means that signal has more energy in that portion of oscillation. Similar with log-peak feature extraction, we first sum the four collected load cell signals up to create a new merged signal, and use a low pass filter to remove the un-relative high

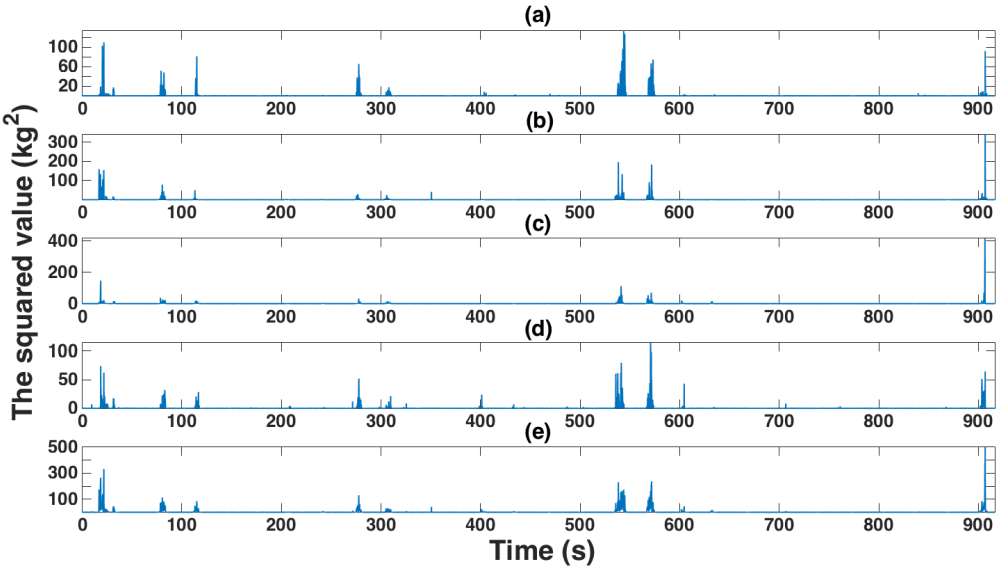



Figure 3.8: (a) Square of the data from load cell 1. (b) Square of the data from load cell 2. (c) Square of the data from load cell 3. (d) Square of the data from load cell 4. (e) The summation of all squares.

frequency components. Then we extract the energy in every 2 seconds window. That window size came from the fact that body's motion cannot be more than 2-3 Hz. So we just pick a size that can cover all possible movements. Extracted energy features will give a peak whenever there is a movement. The height of the peak depends on the strength of movement. The stronger movement results in higher energy peak. All energy widows are normalized with highest value window. Peak detection is applied with some threshold values to find all movements in the data. Also, another threshold is used to classify movements as big or small. Figure 3.10 shows the filtered summation signal and its energy graph for 2 seconds window. We can see the peaks whenever we have movements.

**Zero-Crossing (ZeroX-Valley) Feature Extraction:** We use the same input signal used in the Energy-Peak feature extraction, which is the filtered summation of load cells' readings with local mean removal. There is stronger oscillation in the signal with high amplitude if any in-bed movement is performed.

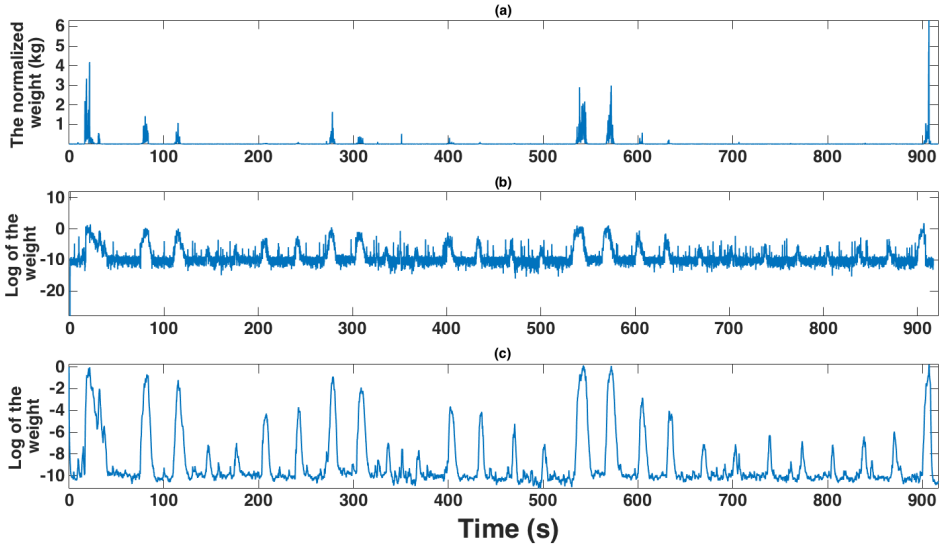



Figure 3.9: (a) The Normalized summation of the squares. (b) The log result of the normalized summation before filtration. (c) The log result of the normalized summation after filtration (using 0.2 Hz low pass filter).

This portion will cross the zero axis less than the surrounding low amplitude parts of signal. So, we use the same size of 2 seconds window used before, and compute the ZX rate in each one. As a result, we see that low value ZX rate window is always connected to the part where we have motion. We used that to get a graph for ZX and try to find valleys. Thresholds also applied here to find these valleys, which also means find movements. Another threshold is applied to distinguish between movements, big or small. As previous, more about choosing these threshold and movement classification will be explained in 3.4. Figure 3.11 shows the input filtered summation signal (after removing the local mean) with its ZX rate graph per 2 seconds window. We have Valley with every movement.

## 3.4 Performance Evaluation

In this section, we first describe the experimental methodology, and then present the evaluation results. In this study, we have carefully evaluated *MotionScale* for its performance in body motion detection and classification.

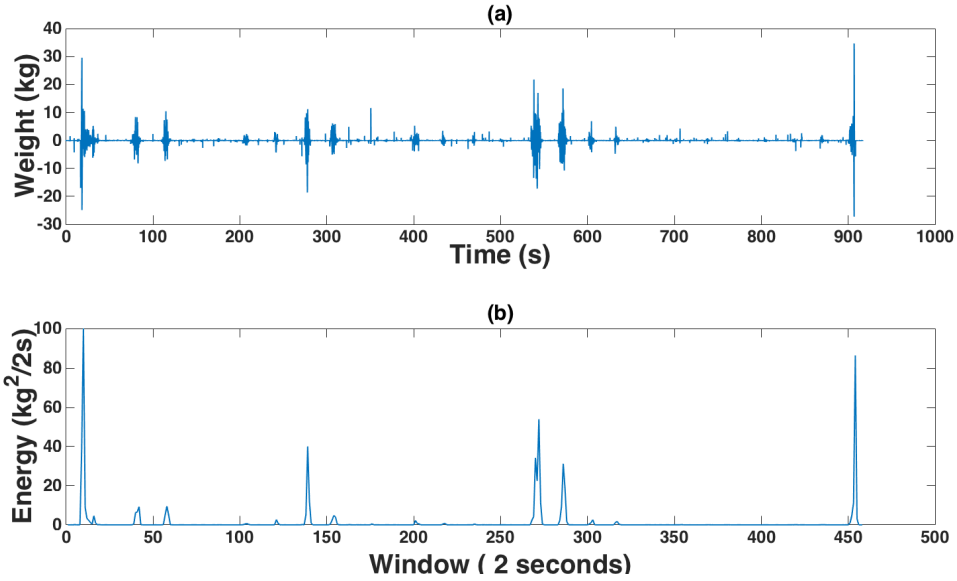



Figure 3.10: (a) The summation of load cells data after dropping the local mean and filtered with 10 Hz low pass filter. (b) The Energy for signal (a) computed for 2 seconds window.

### 3.4.1 Experimental Methodology

The experiments are conducted on a twin size bed in a university laboratory with 30 healthy subjects (22 males and 8 females, age ranging from 22 to 42 years old) over a three-month time period [1]. A common innerspring mattress with dimension of 90cm (width)×185cm (length)×20cm (height) is on the bed, and the *MotionScale* prototype is mounted under the four legs. During the experiments, we ask each subject to perform 27 pre-defined in-bed movements with 20 seconds quiet period after each movement. Among all 27 pre-defined movements, there are 8 large movements involving the entire body (e.g., getting in/off bed, turning left, turning right or rolling over), and 19 small movements that only involve parts of the body (e.g., head, arms and legs). More specifically, 6 of the 19 small movements are leg movements, and the rest are arm and head movements.

We record all the data using the same prototype and laptop to avoid any possible bias in readings. A camera is mounted on a tripod 1.5 meter away from the bed to

[1] Our studies were approved by the Institutional Review Board (IRB) of our institution.

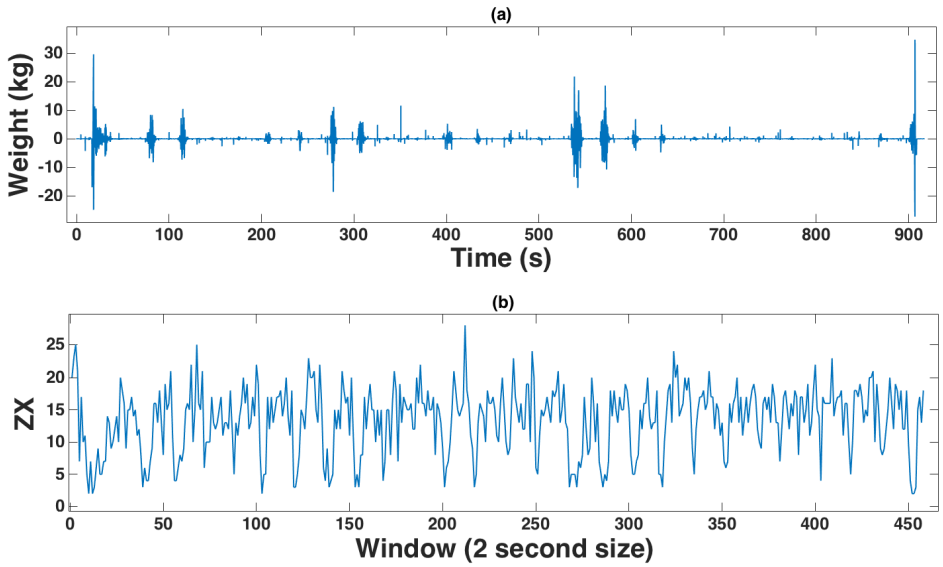



Figure 3.11: (a) The summation of load cells data after dropping the local mean and filtration with 10 Hz low pass filter. (b) The ZX rate computed for 2 seconds window.

record videos for the ground truth recording. In order to select a suitable threshold for motion detection and classification, as we discussed in Section 3.3, we randomly choose 10 subjects' data-sets for the training purpose. Additionally, we repeat this process 100 times to find the most suitable threshold.

### 3.4.2 Performance of Motion Detection

**Comparison of Three Features:** We first compare the performance of the three features extracted from the collected load cell data, i.e., Log Peak, Energy Peak, and Zero-X Valley. In order to conduct a fair comparison, we report each feature's performance using the best threshold value for that feature. From our experiments, we observe that each feature presents an obvious peak or valley whenever there is a motion. These peaks/valleys are very different in amplitudes and widths (even for the

same movement) among the three features, which suggests that we need to find a general threshold (i.e., height of peak) to detect the 27 performed in-bed movements[2] [51]. In order to find the best threshold value for each feature, we apply different threshold values on data collected from 10 randomly selected subjects for a total of 100 times and choose the one that gives the best performance. Specifically, we identify a range of values for each feature's threshold – the peak value threshold for Log-Peak is varied from -12 to 0 in 30 steps, the peak value threshold in Energy-Peak is varied from 0.01 to 0.31 in 30 steps, and the valley value threshold for ZeroX-Valley is varied from 3 to 18 in 30 steps.

We have a total of 30 subjects, and for each of the 100 tests, we randomly choose 10 subjects as training subjects and use the remaining 20 subjects as test subjects. For each test subject, our detection algorithm detects $n$ movements, and the detection error rate is thus calculated as $\left|\frac{27-n}{27}\right|$ where 27 is the number of known movements in each experiment. Figure 5.11 reports the detection error rate distribution of the 100 experiments for each feature. It is very clear that Log-Peak is the best among the three features, delivering a detection error rate of 6%. We therefore believe that *MotionScale* is a viable movement detection system during sleep.

[2]To prevent the inference from other noise, we apply a threshold to the minimum distance between two neighboring peaks based on the time interval between each two consecutive movements. Specifically, since there is a 20 second quiet period between two consecutive movements in our experiments, we set the threshold of the minimum distance between two neighboring peaks as 20 seconds as well. In real life we can use different periods, such as 5 or 10 seconds.

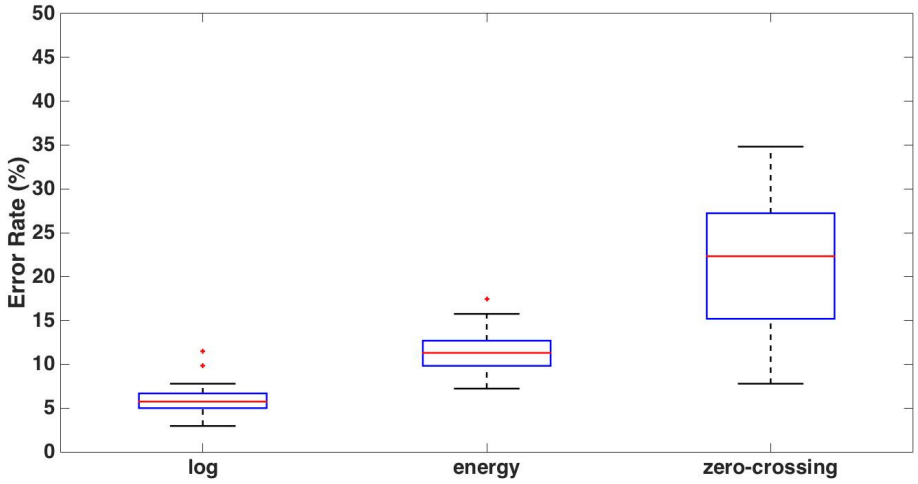



Figure 3.12: The error rate for testing phase.

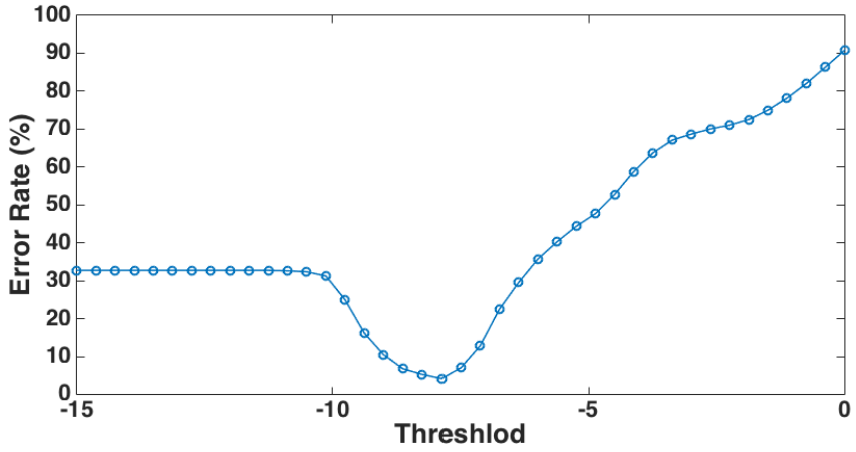
**Study of the Impact of Parameters:** We next study the impact of threshold values on the performance of different detection strategies, and report the results in Figures 5.12(a)-(c) respectively. All three curves exhibit a "U" shape, meaning that there is an optimal value for each threshold. When the threshold is properly chosen (around the optimal value), the corresponding strategy only detects the peaks/valleys caused by valid body motions and ignores the peaks/valleys caused by noise. In this way, we achieve the lowest error rate. When the threshold is too small (starting from the left hand to the bottom area of the U shape), the corresponding strategy detects noise in the environment as body motions, leading to a higher error rate. When the threshold is too large, the corresponding strategy misses legitimate body motions by treating them as noise, resulting in a higher error rate as well. This suggests that by having a suitable training dataset, we are able to learn the optimal threshold values that can minimize the detection error rate for *MotionScale*.

### 3.4.3 Performance of Movement Classification

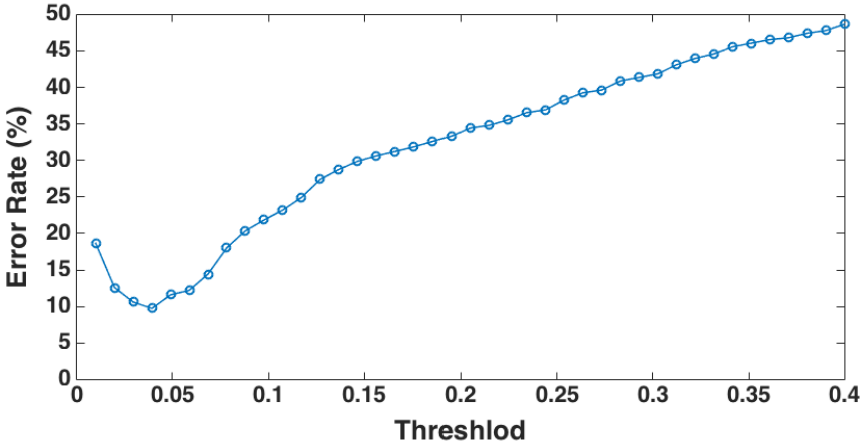
In the second part of the evaluation, we attempt to classify each detected movement as either a big movement or a small movement. In the experiments, we ask the subjects

to perform both big and small movements. In a big movement, the subject moves her entire body from one position to another, or moves the most part of the body; while in a small movement the subject moves only one part of her body, such as arms, legs, or head. The rationale behind discriminating these two types of movements is that big movements normally possess higher energy and longer duration than small ones. We thus, expect higher peaks (or lower valleys) for these motions in our peak detection system. Figure 3.14 shows all the movements that are detected for a subject using the three strategies. In the figure, we label all the big movements using red dashed circles.

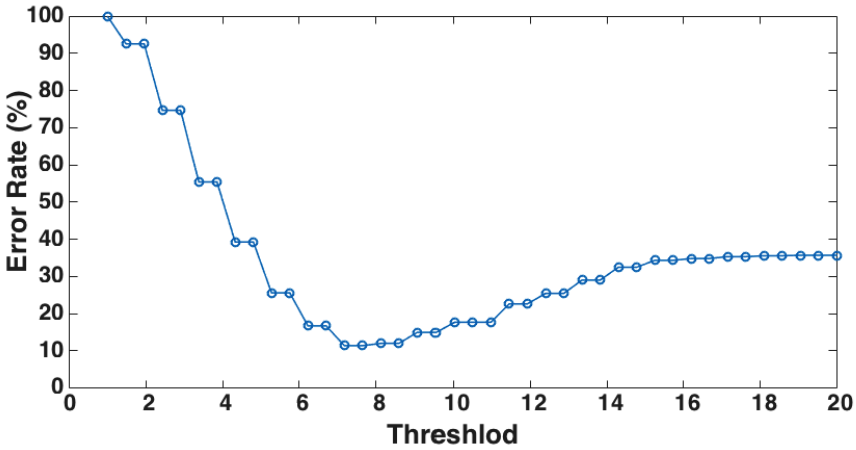
Figure 3.15 shows the classification error rate distribution over 100 experiments, and in each experiment we randomly choose 10 subjects' data as training data and use the remaining as test data. Here, we vary the threshold value for Log-Peak threshold from -5 to -1, the threshold for Energy-Peak from 0.1 to 10, the threshold is from -10 to -5, each in 80 steps. It is obvious from Figure 3.15 that Log-Peak is the best strategy to distinguish between small and big movements, with a mean classification error rate of 4.27%. Across all the data, we find that Log-Peak exhibits the largest gap between small and big movements.




(a) The error rate of variation the threshold in Log Peak strategy

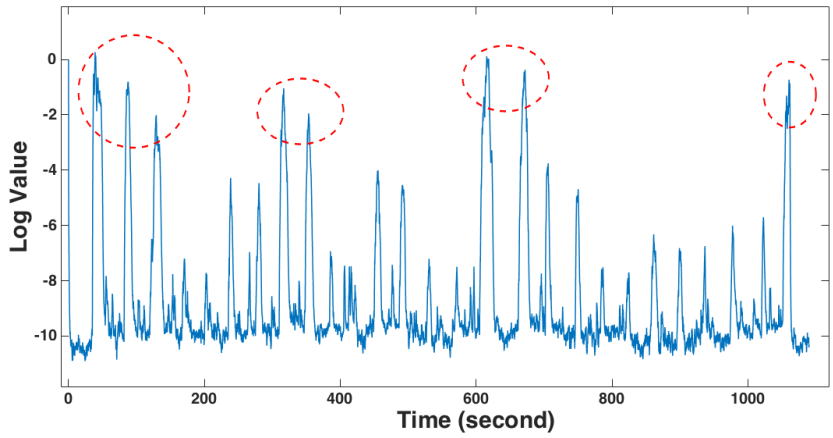



(b) The error rate of variation the threshold in Energy Peak strategy

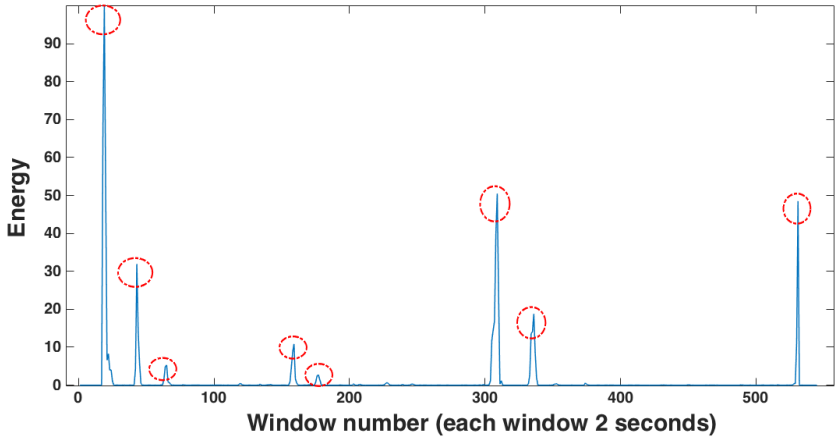



(c) The error rate of variation the threshold in ZX strategy

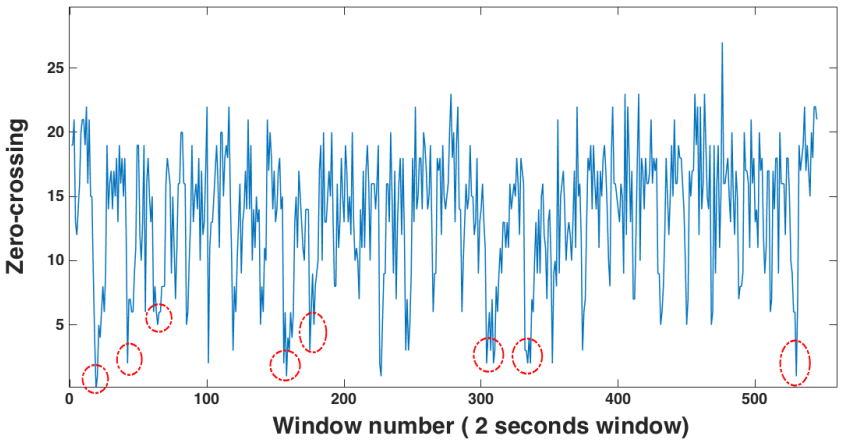
Figure 3.13: The error rate of the three strategies when we varied the threshold. All the 30 subjects are tested here.




(a) The Log scale of the data with big movements on red circles




(b) The Energy of the data with big movements on red circles




(c) The Zero-Crossing of the data with big movements on red circles

Figure 3.14: Data with big movements.

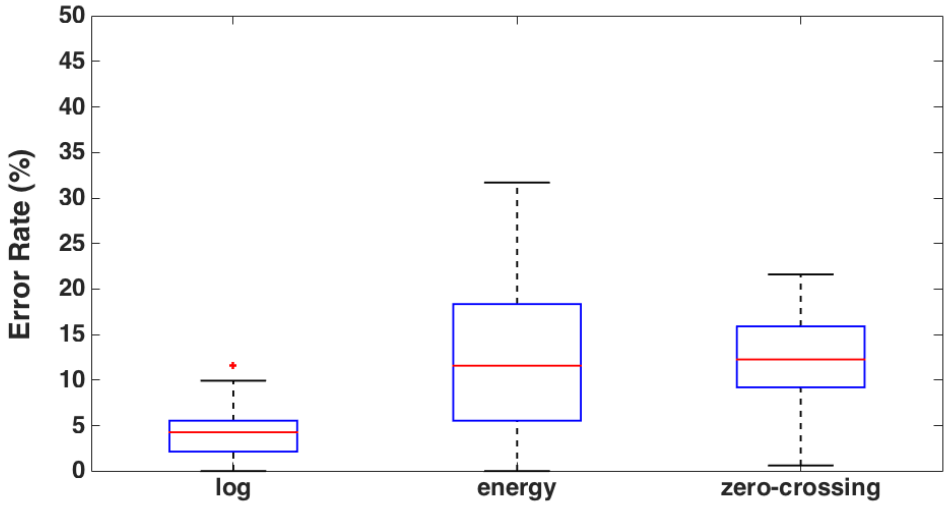



Figure 3.15: The Cross-Validation error rates for the three strategies: Log, Energy, and ZX.

Figure 3.16 shows the ROC curves of the three strategies in classifying big and small movements. We obtain the ROC curves with the threshold value that gives the same false positive rate and false negative rate. These results also indicate that Log-Peak has the best classification performance.

Figure 3.17 shows how Log-Peak's classification algorithm fares with varying threshold values. The results show that Log-Peak reaches the minimum classification error of 5% when the threshold is -2.9. Finally, Table 3.1 summarizes the results with best threshold values and error rates.

| Strategy | Movement Detection | | Movement Classification | |
|---|---|---|---|---|
| | Best Threshold | Error Rate | Best Threshold | Error Rate |
| Log | -7.6 | 6.3% | -3.16 | 4.2% |
| Energy | 0.0445 | 12.4% | 2.2 | 11.5% |
| ZX | 11 | 18.1% | 2.99 | 12.2% |

Table 3.1: Best Thresholds and their associated error rate to detect all movements and the big and small movements.

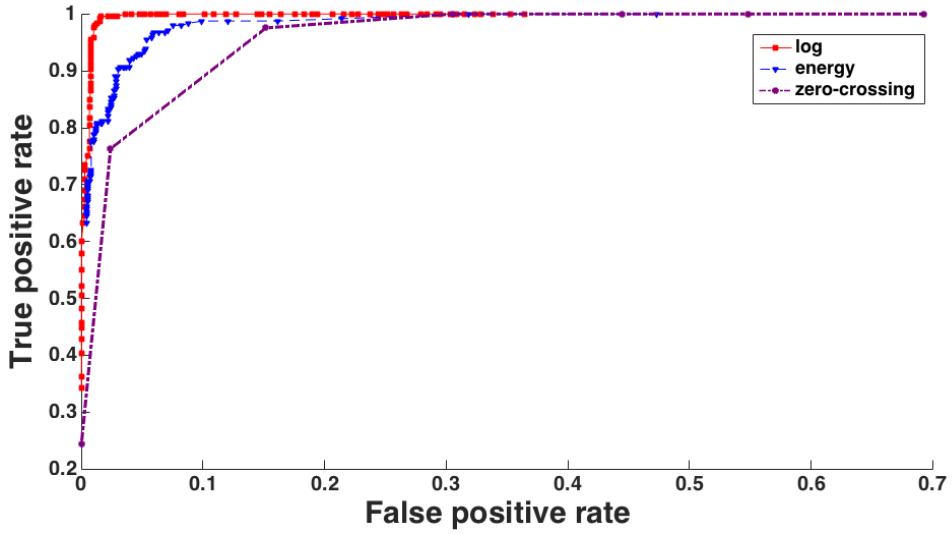



Figure 3.16: The ROC curve for the three strategies applied on 30 subjects.

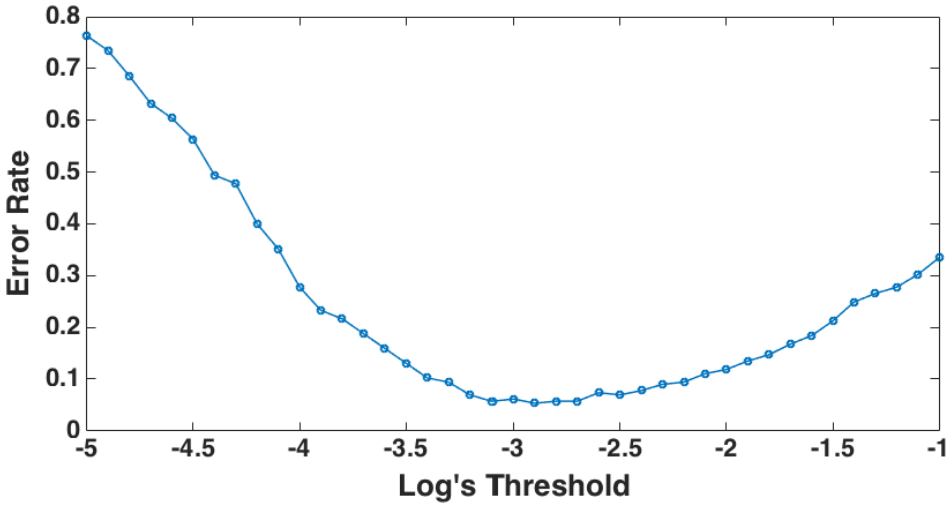



Figure 3.17: The error rate of Log Peak applied on 30 subjects.

## 3.5 Concluding Remarks

In this chapter, we propose a low-cost, low-overhead, and highly robust system for in-bed movement detection and classification, the MotionScale, which can facilitate many smart home and health-care applications. This system utilizes four low-end load cell sensors installed under the legs of a bed to capture the weight distribution on the bed and further accurately determine movements on bed. Compared to existing solutions, MotionScale can use more low-cost hardware to achieve comparable results, and it is very easy to apply in our lives unobtrusively. By utilizing the load cell based system,

MotionScale can detect different types of in-bed body movements with different scales, ranging from parts of body (e.g., arm, head) movements to whole body movements (e.g., turn over, get off bed). To evaluate our system, we build a prototype with off-the-shelf low-cost load cells and PIP-tags and extensively experiment the prototype with $30$ participants over three-month time period. The results show that by utilizing our three main strategies, Log-Peak, Energy-Peak, and ZeroX-Valley, the MotionScale can effectively extract body movement signals from load cell data and detect in-bed movements with a low error rate of $6.3\%$, and classify them to big or small movements with an error rate of $4.2\%$. We have fond that these features are really distinguishable and we got very good results with thresholds.

# Chapter 4
# MotionTree

## 4.1 Introduction

Body motion measurement during sleep is an important consideration for a person's well-being. In fact, in-bed body movements monitoring can enable many applications, such as sleep monitoring, abnormal body movements detection, etc. In-bed motions can be sensed using different kinds of sensors; e.g., temperature sensors [36], pressure sensors [43], accelerometer sensors [41], load cell sensors [39, 40, 52], geophone sensors [53], and custom-made sensors [37, 42]. When motion detection is combined with motion classification, it can be used to detect, notify, and recognize specific events, enabling us to focus on critical tasks. The need for smart systems that provide monitoring capabilities is increasing year over year as the cost of health care continues to rise, especially when these systems can be used at homes as well as hospitals. As a result, we believe unobtrusive, low-cost, wireless, and easy to install sensor system is the best approach for in-bed body motion detection and classification. In this study, our goal is to provide a system that can accurately detect and classify any motion during sleep or in-bed resting time.

Among all proposed sensors, load cells have been noticed to achieve efficient solutions for several reasons. Firstly, load cells are low-cost off-the-shelf, very affordable, and readily available. Secondly, easily to be deployed in a very convenient, and unobtrusive way without any subject's complain requirements. Thirdly, load cell can be placed under bed legs to easily capture the change in the distribution of the weight on the bed that caused by any motion. Fourthly, a set of features can be extracted from

load cells signals to be used for efficient motion classification approach. For that, in our earlier work [52] we used load cell as sensing unit in building a system that can detect in-bed body motions and classify them into big or small movements only. In this study we improved the previous system (that we call MotionScale) to offer a practical approach to much finer in-bed body motion classification. So, in addition to the load cell sensor, we have to have a good classifier model that can use the set of extracted features to give the right class for each movement.

In this study, we try to have a mixture of low-cost, unobtrusive, robust, easy to install hardware device with an efficient and accurate classifier to have a practical in-bed movement classification system. We developed a system based on low-cost wireless load cell sensors and a binary decision tree based on Support Vector Machine as a classifier. We refer to this system as *MotionTree* as it can classify movements using a binary decision tree.

With *MotionTree*, we can simultaneously detect and classify movements into 9 classes – turning/rolling right, turning/rolling left, right hand, right leg, left hand, left leg, legs, head, and combined motions. We tried to cover all possible in-bed motions ranging from large ones like the whole-body roll/turn to small ones like hand movements. Our classifier model is a multilevel binary decision tree that uses SVM model in each decision step to classify motions into the right branch. Twenty four features are extracted from each motion to be used for training and testing purposes. We have also made a great effort to boost the overall accuracy of our system by selecting the best features that can be used for each classification step. The Random Forest technique has been used to compute feature importance in each level. This system can work across different body weights and does not require additional training when a new subject is introduced. Through these techniques, our experimental results that involve 40 subjects show that we can classify in-bed motions into 9-classes of movements with an average accuracy rate of 90%.

In addition to our SVM tree, we use here two more techniques: Random Forest,

and XGBoost. In this chapter, we apply Random Forest in the same way we did for SVM, using similar tree design in *MotionTree* but with RF as classifier instead of SVM. Moreover, we use XGboost as another classifier. We then combine all the above techniques, in one final result. So, whenever we have a movement, we have 3 output from the above 3 techniques: SVM, RF, and XGBoost. We try here to combine all these 3 outputs in one decision to improve the final accuracy of our system.

To summarize, we have made the following contributions in this study:

1. We have built a *MotionTree* prototype and used it to instrument an experimental bed. We have used the experimental bed to collect load cell signals from 40 subjects who make 35 pre-defined different body movements during each experiment. We have detected all motions with the beginning and end moments for each one. 24 features have been extracted from each motion to be used in training and testing phase.

2. We have used the extracted features to train all SVM models used in the binary decision tree. We select the best features for each model by applying Random Forest technique. We classify the collected motions into 9-classes that cover all possible in-bed movements. We compare the classification results against the ground truth observed by a video camera, and found that the average accuracy rate is 90%.

3. We have designed a binary decision tree that uses 8 SVM classification models instead of using one step multi-class SVM classifier. This step improves the whole accuracy from 65% (in the flat SVM classifier) to 90% (in our *MotionTree*).

4. We have used Random Forest in the same way we used SVM to have another classifier. Three tree topologies were used: Flat, Small-Tree, and RF *MotionTree*. We achieved the average classification accuracy of 90.9%.

5. XGBoost is used as another classifier in this chapter. We use flat topology only to achieve a classification accuracy of 90.5%.

6. To have one result, we combine all previous techniques using a logical combination algorithm. We achieve the final classification accuracy of 91.5%.

The remainder of the chapter is organized as follows. In Section 4.2, we describe the hardware system design of *MotionTree*, and in Section 4.3, we describe *MotionTree*'s signal processing algorithms with the classifier construction. We present our evaluation setup and experimental results in Section 4.4. In Section 4.5, we describe our classifier that is based on Random Forest technique, and in Section 4.6, we describe the classifier that is based on using XGBoost technique. We present our combinational results in Section 4.7. Finally, we provide concluding remarks in Section 4.8.

## 4.2 Overview of *MotionTree*

In-bed body motion classification is of great importance that can enable a large array of applications in domains such as Human Computer Interactions (HCI), remote monitoring, health-care, sleep monitoring, smart home, etc. With the ability to detect and classify body motions, many critical health issues can be detected at an early stage to prevent serious health problems. The main goal here, in our system, is to detect in-bed body motions and classify them by using a low-cost, non-invasive sensing system and the machine learning algorithm.

### 4.2.1 Hardware Design

Figure 4.1 shows a load cell under bed leg with amplifier circuit and the wireless transmitter. We collect the data through the base station that has the same hardware as the transmitter, with a tuned 900 MHz monopole antenna attached. The base station is connected to USB port to the dedicated laptop that runs all required data processing

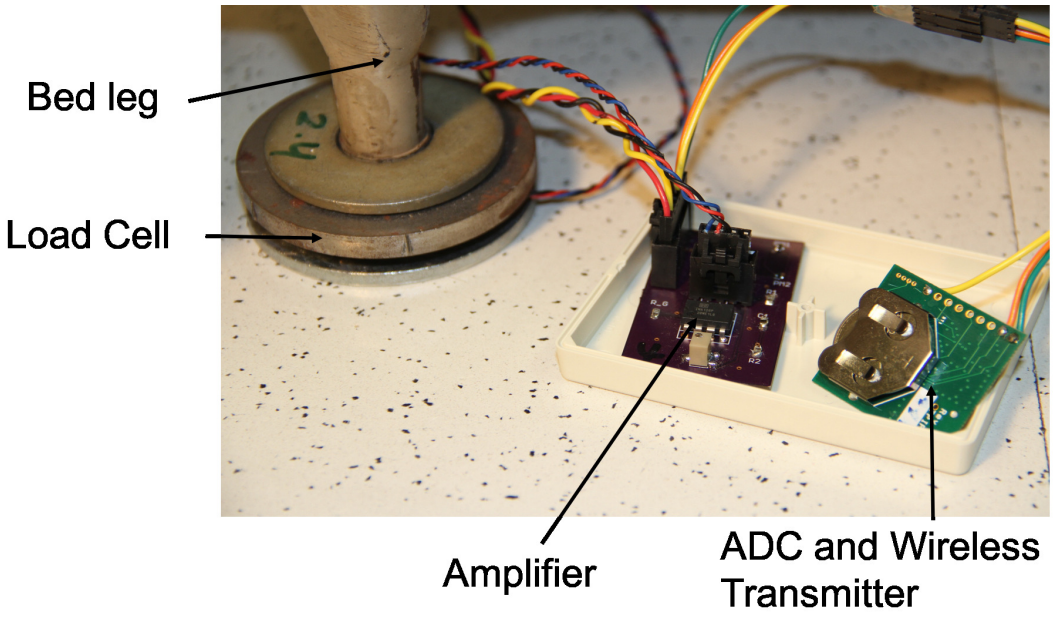



Figure 4.1: Load cell under bed leg with amplifier and transmitter devices

steps. More details about Hardware are described in Section 3.2.2.

### 4.2.2 System Overview

For the purpose of classifying each motion, we first establish the problem design space by defining 9 classes of in-bed body movements: (1) turning/rolling right, (2) turning/rolling left, (3) right hand, (4) right leg, (5) left hand, (6) left leg, (7) legs, (8) head, and (9) combined motions. We define this design space because it almost covers all possible in-bed movements. Also, some movements that cannot be clearly defined, we included them in some of our defined classes. For example, rolling over is very hard to be distinguished from turning to one side of the body, and therefore we lump these two motions together.

As illustrated in Figure 4.2, we collect the measurements from the base station, and conduct the following data processing techniques:

1. *Data Pre-Processing.* Prior to motion classification, we conduct several signal

processing steps. We first interpolate the collected data using the spline interpolation method [54,55]. Since our system focuses on the portion of the data when there is a motion, we try to identify those segments with high oscillation values. Towards this end, we apply local mean removal to remove the constant values in load cells data by utilizing a sliding window. All these steps are explained in Section 3.3.

2. *Movement Detection.* We compute the energy for each load cell data and add them up to have one merged signal that is normalized by using the body weight. To detect motions, we apply the log scale on the merged signal and find peaks which means detecting the movements (Section 3.3).

3. *Feature Extraction.* For each motion, we extract 24 different features – the change of the weight in each load cell (4 features), the change of the center of mass in both $x$ and $y$ axes (2 features), the change of the center of mass around its mean in both $x$ and $y$ axes (2 features), the motion trajectory length, the Euclidean distance between the centers of mass before and after the motion, the log peak value (as defined in section 3.3.2), summation for the real, absolute, and square values of the change in the body center of mass on both axes during the movement (6 features), summation for the real, absolute, and square values of the change in the body center of mass on both axes from its initial point (6 features), and movement's duration.

4. *MotionTree Construction.* Considering the inefficiency of using one-level classifiers to discriminate among nine classes[1], we have designed a binary decision tree with multiple levels for accurate motion classification, which we refer to as *MotionTree*. With a total of 9 classes we could have constructed trees with many different topologies, and we have chosen a topology that is intuitive and leads to

[1]We have tried to classify among the 9 classes in parallel and got very low accuracy rates, e.g., 65%, because many movements have mixed features and it is very difficult to have them classified with SVM.

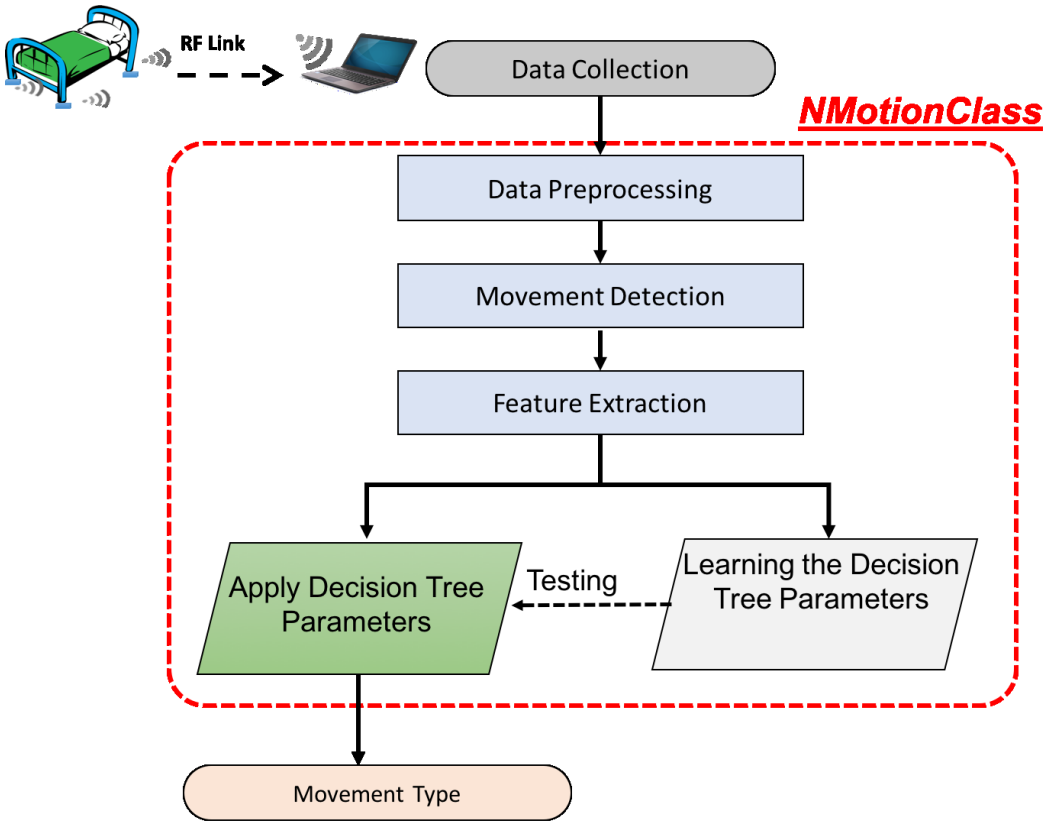



Figure 4.2: Overview of system flow

good classification results.

*MotionTree* adopts multilevel classification where each layer uses Support Vector Machine (SVM) technique [56–59], and a subset of features that are important for that layer. Specifically, we evaluate each feature's importance at every layer using Random Forest technique [60–65].

5. *Motion Classification.* In classification phase, we use the features calculated from the test motion and walk through the *MotionTree* to arrive at the correct motion type.

## 4.3 Motion Classification through *MotionTree*

In this section, we explain how our *MotionTree* model takes the data from load cells and detects and classifies body motions accordingly. The entire process consists of the following steps: motion detection, feature extraction, training and *MotionTree* construction, and classification.

### 4.3.1 Movement Detection

The four output signals from the pre-processing step are used as input to movement detection. We calculate the energy for every raw data signal and sum them up to create a new merged signal, which is the summation of their energy. We have to make our system applicable for a large range of people/subjects, and therefore the merged signal is normalized by dividing by the subject's weight. Our goal here is to find data segments that contain motions (i.e. high energy). Moreover, to successfully detect both big and small movements, we find that the linear view/scale is a poor fit, but the log scale works much better as it can simultaneously show both small and large values. We then apply natural log to the merged signal and apply a low-pass filter of a cutoff frequency 0.2 Hz to the log output to have clear signal. As a result, we can observe clear peaks whenever there is a motion. We then apply suitable thresholds to locate the peaks which correspond to body movements. We assume the beginning of the motion starts at a small time window (e.g., 20 seconds) before the peak, and the end of the motion is at time window of 20 seconds after the peak. So, we select the minimum points during these windows to represent the beginning and the end of the motion. We believe that the duration of 40 seconds covers most of the common body movements. Figure 4.3 shows the log output of a portion of the merged signal, where we use red triangles to mark the detected peaks, orange squares to mark the beginning of the motion, and red stars to mark the end of every motion.

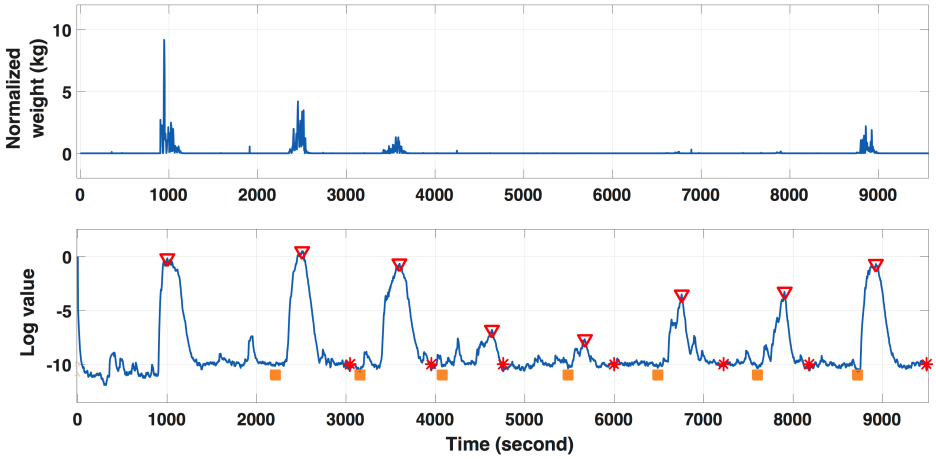



Figure 4.3: Normalized summation of four load cells signals' energy and its log result after filtration (using 0.2 Hz low-pass filter). Red triangles to show the detected movements. Orange squares and red stars to show the beginning and end of every movement

### 4.3.2 Feature Extraction

From the signals during a detected movement period (between the beginning and end of a movement) – which we refer to as *motion data* or *motion signal* – we calculate 24 features to characterize the movement.

First, we use raw load cell signals to estimate the exact location of the body's center of mass. Figure 4.4 shows how we represent the positions of the load cells and the subject in the Cartesian system. Using this representation, we can calculate the following features that depend on the distribution of body weight on the four load cells.

1. *The change of the weight on load cell* $i$, $i \in [1-4]$, $\Delta W_i$. This feature represents the change in the weight on load cell $i$, from the beginning and the end of a motion.

$$\Delta W_i = W_i(t_{end}) - W_i(t_0), \tag{4.1}$$

where $W_i(t_{end})$ and $W_i(t_0)$ correspond to the measurements from load cell $i$, at the end of the movement, and at the beginning of the movement, respectively.

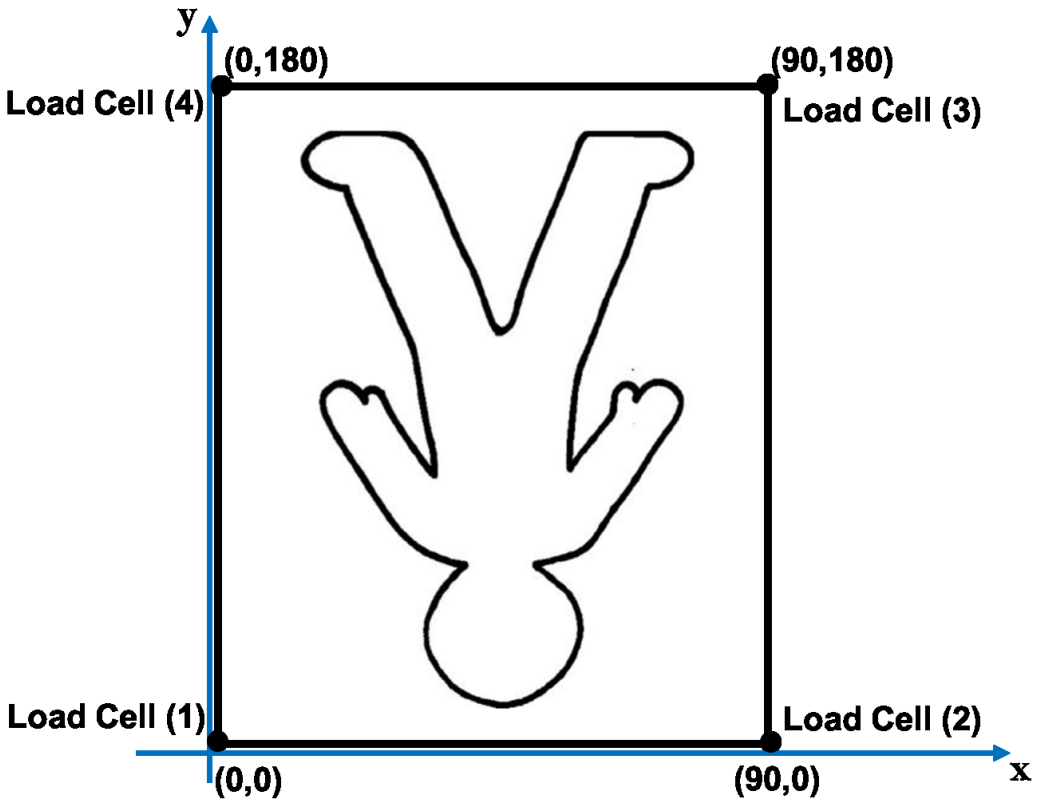



Figure 4.4: The bed coordinates representation in a Cartesian system.

2. *The change of the location of the body center mass*, given by $\Delta X_{cm}$ and $\Delta Y_{cm}$. This set of features reflect how much the center of mass moves after each movement in both directions (the left-right direction denoted by x-axis, and up-down direction by y-axis). To calculate this set of features we use the law of levers to find the coordinates of the body center of mass at a given time:

$$X_{cm}(t) = X_{max}(\frac{\Delta W_2 + \Delta W_3}{\sum_{i=1}^{4} \Delta W_i}), \tag{4.2}$$

$$Y_{cm}(t) = Y_{max}(\frac{\Delta W_3 + \Delta W_4}{\sum_{i=1}^{4} \Delta W_i}). \tag{4.3}$$

where $X_{max}$ and $Y_{max}$ are the maximum width and length of the bed – $90$cm and $180$cm in Figure 4.4.

As a result, we calculate this set of features as follows:

$$\Delta X_{cm} = X_{cm}(t_{end}) - X_{cm}(t_0), \tag{4.4}$$

and

$$\Delta Y_{cm} = Y_{cm}(t_{end}) - Y_{cm}(t_0). \tag{4.5}$$

3. *The change of the body center of mass from its mean on both axes*, which we refer to as X variance $X_{var}$ and Y variance $Y_{var}$. This set of features measure how the center of mass changes from its mean value during the motion. They reflect how big the movement is and how it is concerned on some spot or location on bed. $X_{var}$ is useful to help our system in discriminating movements involving left or right body sides, for example this feature has bigger value for left leg motion than left hand. In the same way, $Y_{var}$ helps to distinguish motions involving upper and lower parts. They can be calculated as follows:

$$X_{var} = \frac{\sum_{i=1}^{N}(X_{cm}(i) - X_{cm}(mean))^2}{N}, \tag{4.6}$$

and

$$Y_{var} = \frac{\sum_{i=1}^{N}(Y_{cm}(i) - Y_{cm}(mean))^2}{N}, \tag{4.7}$$

where $X_{cm}(mean)$ and $Y_{cm}(mean)$ denote the estimated mean of the change on the x and y axes over the interval of a movement.

4. *Movement Trajectory length ($TL$).* This feature measures the length of the trajectory traveled by the body center of mass during a movement. The value is estimated by summing the distance between the body center of mass locations for subsequent samples during the movement. Figure 4.5 shows such an example trajectory for a turning-left movement. This feature reflects how big the movement is and how much the body center of mass has moved spatially. Usually its value is small for movements that involve only one body part such as an arm or

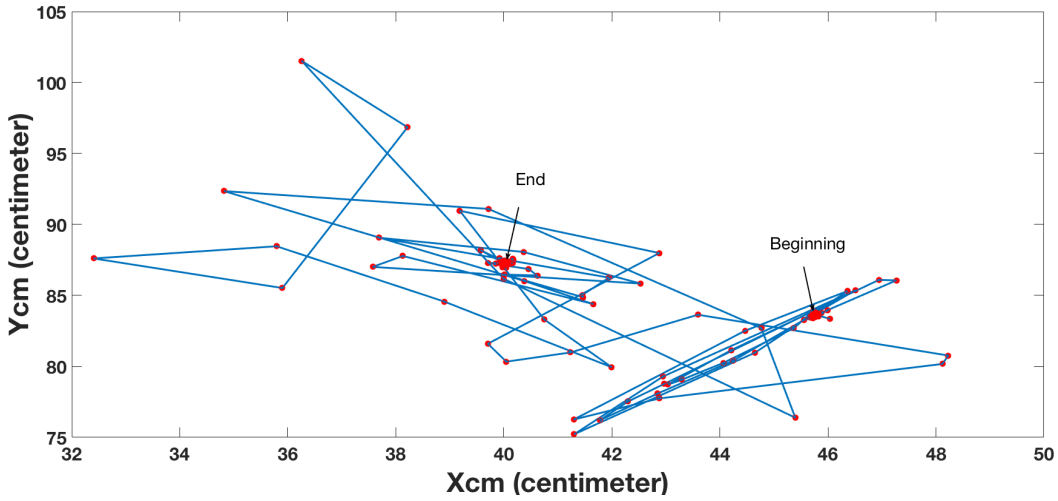



Figure 4.5: The trajectory of the body center of mass during a turn left movement. The positions of the body center of mass at the beginning and at the end are given.

the head, but large for big movements that involve mu body parts or the whole body. $TL$ is calculated as the following:

$$TL = \sum_{i=0}^{N-1} \sqrt{(X_{cm}(i+1) - X_{cm}(i))^2 + (Y_{cm}(i+1) - Y_{cm}(i))^2}, \tag{4.8}$$

where $N$ is the number of samples collected during a body movement.

5. *The Euclidean distance between the beginning and end points of the body center of mass ($D$).* This feature depicts the displacement of the body center of mass due to a body movement, whose value is usually directly affected by the magnitude of the movement. $D$ is calculated as:

$$D = \sqrt{(X_{cm}(t_{end}) - X_{cm}(t_0))^2 + (Y_{cm}(t_{end}) - Y_{cm}(t_0))^2}. \tag{4.9}$$

6. *Log-Peak value ($V_{peak}$).* In our movement detection phase, we use the presence of log peaks to detect body motions, as shown in Figure 4.3. Here, we use the peak value as a feature to describe the movement – higher peaks always associated with bigger movements.

7. *The real summation of each change of center of mass (from the previous CoM position) on both axes during the movement, ($RSX_{cm}$ and $RSY_{cm}$).* These two

features reflect how much the center of mass changes from its previous location. The summation of all these changes or differences are computed here for both axes during the entire movement. They can be calculated as follows:

$$RSX_{cm} = \frac{\sum_{i=1}^{N}(X_{cm}(i+1) - X_{cm}(i))}{N}, \tag{4.10}$$

and

$$RSY_{cm} = \frac{\sum_{i=1}^{N}(Y_{cm}(i+1) - Y_{cm}(i))}{N}, \tag{4.11}$$

where $N$ is the number of samples collected during a body movement.

8. *The real summation of each change of center of mass (from the initial CoM position) on both axes during the movement, ($RSXB_{cm}$ and $RSYB_{cm}$).* These two features reflect how much the center of mass changes from its initial location. The summation of all these changes or differences are computed here for both axes during the entire movement. They can be calculated as follows:

$$RSXB_{cm} = \frac{\sum_{i=1}^{N}(X_{cm}(i) - X_{cm}(t_0))}{N}, \tag{4.12}$$

and

$$RSYB_{cm} = \frac{\sum_{i=1}^{N}(Y_{cm}(i) - Y_{cm}(t_0))}{N}, \tag{4.13}$$

9. *Summation for the Absolute values in the changes of the body center of mass on both axes during the movement, ($ASX_{cm}$ and $ASY_{cm}$).* The summation of all absolute differences between the current and previous value are computed here for both axes during the entire movement. They can be calculated as follows:

$$ASX_{cm} = \frac{\sum_{i=1}^{N}|(X_{cm}(i+1) - X_{cm}(i))|}{N}, \tag{4.14}$$

and

$$ASY_{cm} = \frac{\sum_{i=1}^{N}|(Y_{cm}(i+1) - Y_{cm}(i))|}{N}, \tag{4.15}$$

10. *Summation for the Absolute values in the changes of the body center of mass from its initial point on both axes during the movement, ($ASXB_{cm}$ and $ASYB_{cm}$).* They can be calculated as follows:

$$ASXB_{cm} = \frac{\sum_{i=1}^{N} |(X_{cm}(i) - X_{cm}(t_0))|}{N}, \tag{4.16}$$

and

$$ASYB_{cm} = \frac{\sum_{i=1}^{N} |(Y_{cm}(i) - Y_{cm}(t_0))|}{N}, \tag{4.17}$$

11. *Summation for the Square values of the changes in the body center of mass on both axes during the movement, ($SSX_{cm}$ and $SSY_{cm}$).* The summation of all absolute differences between the current and previous value are computed here for both axes during the entire movement. They can be calculated as follows:

$$SSX_{cm} = \frac{\sum_{i=1}^{N} (X_{cm}(i+1) - X_{cm}(i))^2}{N}, \tag{4.18}$$

and

$$SSY_{cm} = \frac{\sum_{i=1}^{N} (Y_{cm}(i+1) - Y_{cm}(i))^2}{N}, \tag{4.19}$$

12. *Summation for the Square values of the changes in the body center of mass from its initial point on both axes during the movement, ($SSXB_{cm}$ and $SSYB_{cm}$).* They can be calculated as follows:

$$SSXB_{cm} = \frac{\sum_{i=1}^{N} (X_{cm}(i) - X_{cm}(t_0))^2}{N}, \tag{4.20}$$

and

$$SSYB_{cm} = \frac{\sum_{i=1}^{N} (Y_{cm}(i) - Y_{cm}(t_0))^2}{N}, \tag{4.21}$$

13. *Duration of the Movement, ($N$).* The value of this feature is the number of samples observed during the movement.

### 4.3.3 *MotionTree* Construction

The purpose of this study is to build a classification model that uses the above features to classify unknown motions into one of the pre-defined movement classes. We first define the following 9 classes of common body motions: (1) turning/rolling right, (2) turning/rolling left, (3) right hand, (4) right leg, (5) left hand, (6) left leg, (7) both legs, (8) head, and (9) combined motions.

In this study, our main idea is to build a binary decision tree based on Support Vector Machine (SVM). Even though SVM is good for binary classification, as discussed in [66, 67], a flat multi-class SVM classifier yields poor classification results when the number of classes becomes large such as in our case. We have tried to classify among the 9 classes in parallel and got very low classification accuracy rates, 65%, because many movements have mixed features and it is very difficult to have them classified with SVM. To improve this accuracy, we construct a binary classification tree, and at each intermediate tree node, we run a SVM-based binary classification. This tree design moves the classification accuracy from 65% (of the flat classifier) to about 90% (as will see in section (Sec 4.4)).

Our 9 classes come from human body shape and its motion during sleep. Legs, hands and head are 5 classes. Two hands together is another class. The same for two legs together when they move to have one more class. Moving the entire body is a very common motion during sleep. We put that in 2 classes to have turning/rolling right, and turning/rolling left. Based on that, we have tried to design a decision tree that gives the best accuracy and covers all possibilities of body's motions.

The first classification decision (at the root level) is to categorize all the movements into two classes: big movements and small movements. Usually the first two classes of movements (i.e., turning/rolling right and turning/rolling left) are considered as big movements because the entire body is involved. After the first binary classification, we got big and small movements. Further classifying big movements only involves one

more level of binary classification: is the movement turning/rolling left or right. On the other hand, we have more types of small movements (in our design space, seven classes of movements belong to small movements), and several levels of binary classification are required to effectively differentiate among them. First, we use a binary classifier to check if it's a leg movement. If it is a leg movement, then we use two binary classifiers to determine which leg has moved or both (classes 4, 6, and 7). If it is not a leg movement, then we use a binary classifier to check whether it is a head movement (class 8), or a hand movement. We treat the classification of a hand movement in a similar way as the classification of a leg movement – we use three levels of binary classifiers to determine which hand has moved or both (classes 3, 5, and 9). Figure 4.6 shows the structure of the binary decision tree we have designed.

To boost the accuracy for each binary classification step, we select the best features for that step from the entire feature set. To determine the importance of features for each classifier, we use the Random Forest technique in our study. This step helps boost the overall classification accuracy, as we will demonstrate in the performance evaluation section (Sec 4.4). In Section 4.4.3, we also discuss the rationale behind our tree structure, and compare its classification accuracy with several different structures.

## 4.4 Performance Evaluation

In this section, we carefully evaluate how well *MotionTree* can classify in-bed body motions. We first describe our experimental methodology, and then present detailed experimental results.

### 4.4.1 Experimental Methodology

We have conducted experiments in a university laboratory setting on a twin size bed with 40 healthy subjects (26 males and 14 females, age ranging from 22 to 42 years

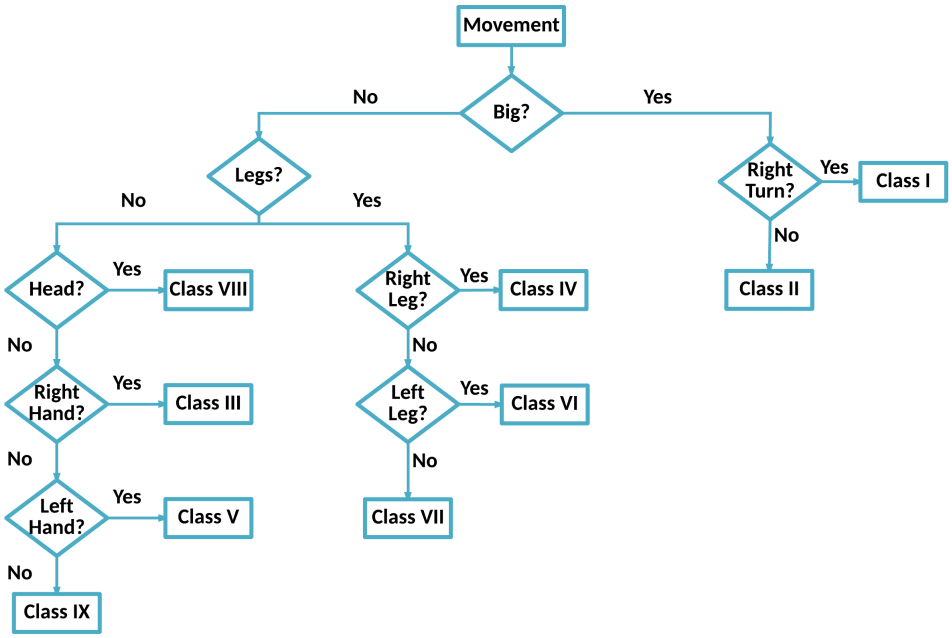



Figure 4.6: Our Binary Decision Tree design that based on SVM to classify motions into 9 classes.

old) over a four-month time period, with 70 experiments[2]. Each of the four load cells is mounted under a leg of the bed with a common innerspring mattress of 90cm (width) $\times$ 185cm (length) $\times$ 20cm (height). During each experiment, we asked the subject to perform 35 pre-defined in-bed movements with a 20-second quiet period following each movement. We have designed these movements to cover a large array of possible in-bed movements ranging from big turning or rolling movements that involve the entire body to small movements that only involve a hand or the head. Table 4.1 shows the group of movements, in our experiments, that associated to each class. Table 4.1 shows the group of movements, in our experiments, that associated to each class.

The same prototype and laptop are uses to record all the data to avoid any possible bias in readings. We used a camera that is mounted on a tripod 1.5 meter away from the bed to record videos for the ground truth recording.

[2]Our studies were approved by the Institutional Review Board (IRB) of our institution.

| Class Number | Class Name | Group of Movements |
|---|---|---|
| I | Turning/Rolling Right | Turn from back to right.<br>Turn from left to back.<br>Roll over from right side. |
| II | Turning/Rolling Left | Turn from back to left.<br>Turn from right to back.<br>Roll over from left side. |
| III | Right hand | Straighten right arm.<br>Put right hand on chest.<br>Put right hand on head.<br>Put right hand close to head. |
| IV | Right Leg | Straighten right leg.<br>Bend right leg. |
| V | Left Hand | Straighten left arm.<br>Put left hand on chest.<br>Put left hand on head.<br>Put left hand close to head. |
| VI | Left Leg | Straighten left leg.<br>Bend left leg. |
| VII | Both Legs | Straighten legs.<br>Cross right leg on left leg.<br>Cross left leg on right leg. |
| VIII | Head | Turn head to the left.<br>Turn head to the right.<br>Turn head from left to the right.<br>Turn head from right to the left.<br>Turn head from right to normal position.<br>Turn head from left to normal position. |
| IX | Combined Motions | Put both hands on chest.<br>Straighten both hands.<br>Move both hands with legs. |

Table 4.1: The set of body movements chosen for the study that associated to each class.

### 4.4.2 Performance of Motion Classification

As mentioned in Section 4.3, we have a model of a binary decision tree that based on SVM. In each decision step, we have a SVM model that classifies movements according to that level. Each level depends on the results from the upper one. Each SVM model is trained separately with the portion of data suitable to that level. Also, each SVM model is trained using the best features to achieve the best accuracy. So, we don't need to use the 24 features in each SVM model. As in Figure 4.6, each decision step, or rhombus, is an SVM model. We have eight models to be trained. We used 60% of the data that is associated to each model for training and 40% for testing. For example, all data is used in the highest level SVM model, to classify as big or small. While in the model below, that classify turn/roll left from turn/roll right, we used the data that belongs to big family only to train that model. For feature selection, we used Random Forest technique to select the best features for the associated level. To know feature's importance, Random Forest permutes the values of this feature across every observation in the data set and measures how much worse the Mean Square Error becomes after the permutation. This is repeated for each feature. The Mean Square Error averaged over all trees, in the Random Forest model, and divided by the standard deviation taken over the trees, for each variable. The larger this value, the more important the variable [60, 68]. In our model, we choose all features with value 0.2 and above to be the best features for training.

To evaluate our system, 500 tests have been done with cross validation of randomly 60% of the data for each level selected for training and 40% for testing. In each iteration, we have never used any portion of the data that picked for training in the testing phase. We calculate the average value of classification accuracy for each decision, i.e. SVM model. Table 4.2 shows the eight decision steps, or SVM models, and their average accuracy with all features and with reduced set of features. We list all features that are **not** used in each level (according to Random Forest). All other features described in 4.3 and are not listed in the table are used for the associated level. SVM model

names are from the decision steps in the *MotionTree* in Figure 4.6.

Since we are dealing with classification problem, we computed three factors for each class to have a complete evaluation of the efficiency in our system. The three factors are: Recall (sensitivity), Precision, and Accuracy. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. Accuracy is the sum of correct classifications divided by the total number of classifications [69].

$$Recall = (\frac{T_p}{T_p + F_n}), \tag{4.22}$$

$$Precision = (\frac{T_p}{T_p + F_p}), \tag{4.23}$$

$$Accuracy = (\frac{T_p + T_n}{T_p + T_n + F_p + F_n}), \tag{4.24}$$

where $T_p$ is the number of true positives, $T_n$ is the number of true negatives, $F_p$ is the number of false positives, and $F_n$ is the number of false negatives [70].

Table 4.3 shows the average *Recall* and the *Precision* for each class by using our *MotionTree* system. It also shows the recall for each class with flat SVM. This table includes the accuracy of the entire tree (*MotionTree*), which is the accuracy of our system, and the accuracy of flat SVM topology.

We can see that classes VII and IX, motions generated by many body parts at the same time, like head and hands or both legs, have lower accuracy because they have close features values with other classes that have motions for the same body parts. That can make their recognition harder than other classes. Moreover, it is very clear that *MotionTree* is better than flat SVM classifier (multi-classes SVM).

| SVM Model Name | Average Accuracy with All Features | Features Not Used According to RF | Average Accuracy with Reduced Features |
|---|---|---|---|
| Big? | 86.8% | $\Delta Y_{cm}$, $RSX_{cm}$, $RSY_{cm}$, $RSYB_{cm}$ | 99.5% |
| Right Turn/Roll? | 85% | $\Delta Y_{cm}$, $Y_{var}$, $V_{peak}$, $ASX_{cm}$, $SSX_{cm}$, $RSY_{cm}$, $ASY_{cm}$, $SSY_{cm}$, $RSYB_{cm}$, $ASYB_{cm}$, $SSYB_{cm}$, $N$ | 93.4% |
| Legs? | 95.8% | - | 95.8% |
| Right Leg? | 93% | $TL$, $V_{peak}$, $RSX_{cm}$, $ASX_{cm}$, $SSX_{cm}$, $RSXB_{cm}$, $ASXB_{cm}$, $SSXB_{cm}$, $ASY_{cm}$, $SSY_{cm}$, $ASYB_{cm}$, $SSYB_{cm}$, $N$ | 96.9% |
| Left Leg? | 93.8% | $TL$, $V_{peak}$, $RSX_{cm}$, $ASX_{cm}$, $SSX_{cm}$, $RSXB_{cm}$, $ASXB_{cm}$, $SSXB_{cm}$, $RSY_{cm}$, $ASY_{cm}$, $SSY_{cm}$, $RSYB_{cm}$, $ASYB_{cm}$, $SSYB_{cm}$, $N$ | 95.3% |
| Head? | 92% | $RSX_{cm}$, $RSXB_{cm}$, $SSY_{cm}$, $N$ | 95.2% |
| Right Hand? | 93.1% | $TL$, $ASX_{cm}$, $SSX_{cm}$, $ASY_{cm}$, $SSY_{cm}$, $N$ | 95.3% |
| Left Hand? | 88% | $RSX_{cm}$, $SSX_{cm}$, $SSY_{cm}$, $N$ | 92.6% |

Table 4.2: SVM models with features that are not used according to feature importance and the average of accuracy for each model.

### 4.4.3 Different Tree Structures

To get the best tree structure, with better accuracy, we should start with the best classifier and make all possible paths as short as possible. Longer paths have more classification decisions, so we expect higher classification errors that will impact the whole accuracy. In order to show that we have the best tree design, we have tested many other structures and evaluated their performance. Figures 4.7, 4.8 and 4.9 show 3 examples of these structures. The tree in Figure 4.7 starts with the 'leg or not' classification. Since this binary classification has less accuracy than 'big or not', this tree has much lower overall accuracy (about 78%). The tree in Figure 4.8 has a different decision sequence, which leads to longer paths in the left side than what we have in our tree

| **Class Number** | **Class Name** | **Recall Using *MotionTree*** | **Precision with *MotionTree*** | **Recall Using Flat SVM** |
|---|---|---|---|---|
| I | Turning/Rolling Right | 91.2% | 92.3% | 18% |
| II | Turning/Rolling Left | 94% | 91% | 14.7% |
| III | Right hand | 91% | 96% | 87% |
| IV | Right Leg | 90% | 95% | 75% |
| V | Left Hand | 89% | 96.3% | 72% |
| VI | Left Leg | 91% | 88% | 79.3% |
| VII | Both Legs | 86% | 72% | 81.2% |
| VIII | Head | 90% | 84% | 69% |
| IX | Combined Motions | 79% | 78.2% | 17% |
| **The Whole Tree Accuracy** | | 90% | | |
| **Flat SVM Accuracy** | | 65% | | |

Table 4.3: Recall and precision for each class with the whole accuracy for the entire system using both our tree (*MotionTree*) and flat SVM classifier.

(Figure 4.6). This structure impacts the entire accuracy of the system badly. It has more questions to some portion of the data and therefore more error happens with each decision (about 80% as classification accuracy). The tree in Figure 4.9 is very similar to our tree. The only difference is switching the leg's decision part with hand's part. Here the problem is the behavior of body's movements. Usually it's easier to differentiate the motions of the lower half of human body from those of the upper half. That is what we have done in our tree. In many times, with using the current set of features, the process of classification mixes between hands and head motions because they are close to each other. So, asking about hands movements in some high decision node would increase the propagation error for the entire system. Therefor the classification accuracy for this tree would go to around 82%. We can see from above examples that our tree (Figure 4.6) can achieve the best possible classification accuracy. Table 4.4 summarizes the accuracy difference among the above structures.

| Tree Structure | Accuracy |
|---|---|
| Tree in Figure 4.7 | 78% |
| Tree in Figure 4.8 | 80% |
| Tree in Figure 4.9 | 82% |
| *MotionTree* in Figure 4.6 | 90% |

Table 4.4: The Accuracy difference among different tree structures.

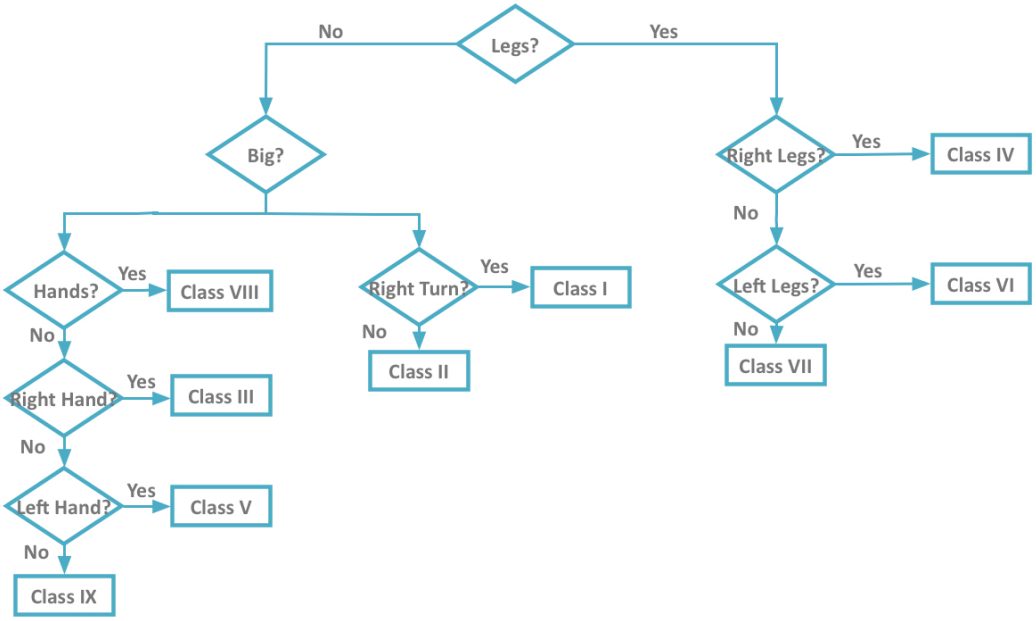



Figure 4.7: A tree starts with legs classification.

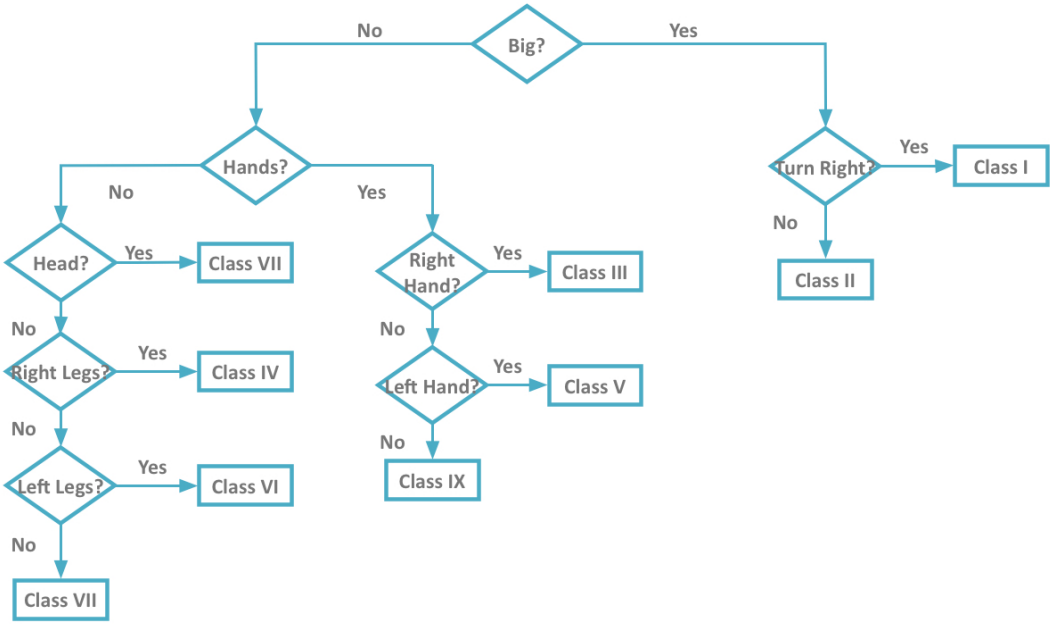



Figure 4.9: A tree with hands before legs as a decision.

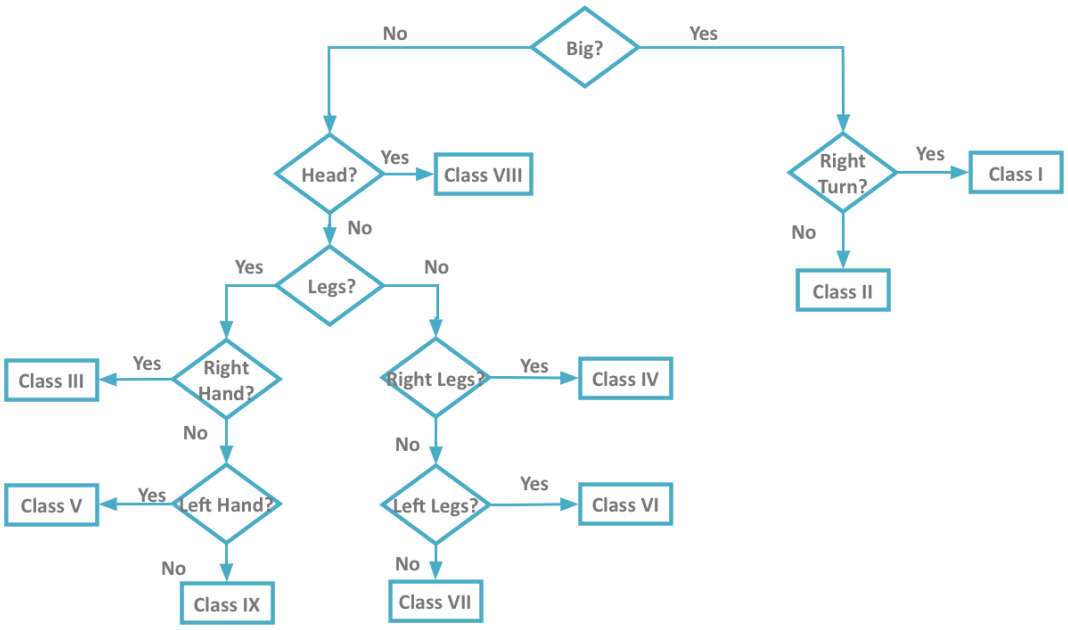



Figure 4.8: A tree with longer paths.

## 4.5 Random Forest

We mentioned Random Forest technique in Section 4.4 but we used it as a feature selection tool only. In this section, we use Random Forest as a classification tool. Random Forest is better than SVM in multi-class classification, so, we apply it in 3 different topologies to see which one is the best. We start with flat topology, where we have one level of classification to classify all movements to 9 classes. Then we apply RF in the same tree design we have for *MotionTree*. Lastly, we apply RF in another tree design, that is smaller than *MotionTree*.

### 4.5.1 Flat Topology

Random Forest is a very useful tool for multi-class classification. We use it here to classify in-bed motions into one of the predefined 9 classes. Figure 4.10 shows the flow of data in the flat topology design. Here, we use the same data, with the same 24 features, in section 4.3.2. In this case, we have one classifier to be trained, instead of 8 in *MotionTree*. To evaluate this classifier, 500 tests have been done with cross

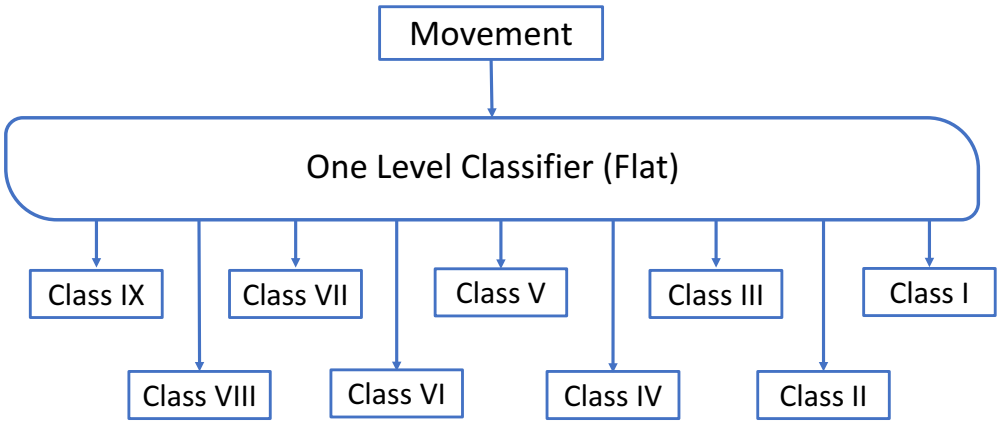



Figure 4.10: One level classifier, Flat Classifier.

validation of randomly 60% of the data for each level selected for training and 40% for testing. This Flat model classifier was able to classify the detected movements into 9 classes with average accuracy of 90.8%. Table 4.5 shows the average recall and the average precision for each class with the whole accuracy to this topology.

| **Class Number** | **Class Name** | **Recall Using Flat RF** | **Precision with Flat RF** |
|---|---|---|---|
| I | Turning/Rolling Right | 92.3% | 91.3% |
| II | Turning/Rolling Left | 91.2% | 91.1% |
| III | Right hand | 93.7% | 94.8% |
| IV | Right Leg | 90% | 96% |
| V | Left Hand | 93.3% | 93.6% |
| VI | Left Leg | 88.2% | 92% |
| VII | Both Legs | 82.3% | 75.7% |
| VIII | Head | 93% | 88% |
| IX | Combined Motions | 79% | 90.6% |
| **Flat RF Accuracy** | | 90.8% | |

Table 4.5: Recall and precision for each class with the whole accuracy for the entire system using Flat RF.

### 4.5.2 *MotionTree* Topology

We apply Random Forest in the tree design for *MotionTree* in Figure 4.6 instead of SVM classifiers. We have here 8 Random Forest classifiers. We use the same data with the same cross validation strategy (500 tests with cross validation of randomly 60% of the data for each classifier selected for training and 40% for testing). The same features in Table 4.2 were not used in the same way described in section 4.4.2. Table 4.6 shows the recognition accuracy for each Random Forest classifier and the accuracy of the entire tree, which is 89.3%. Table 4.7 shows the average recall and the average precision for each class with the whole accuracy for the entire system using RF *MotionTree*. We can see that *MotionTree* design has better accuracy for SVM models. At the same time, the flat topology is much better in Random Forest than in SVM (90.8% in RF to 65% in SVM).

| **RF Model Name** | **Average Accuracy** |
|---|---|
| Big? | 99.4% |
| Right Turn/Roll? | 92.48% |
| Legs? | 96.1% |
| Right Leg? | 95.8% |
| Left Leg? | 93% |
| Head? | 95.6% |
| Right Hand? | 96% |
| Left Hand? | 93.8% |
| **Whole Tree Accuracy** | 89.3% |

Table 4.6: Random Forest models' accuracy and the whole tree accuracy (the tree in Figure 4.6).

| **Class Number** | **Class Name** | **Recall Using RF *MotionTree*** | **Precision with RF *MotionTree*** |
|---|---|---|---|
| I | Turning/Rolling Right | 92.1% | 91.3% |
| II | Turning/Rolling Left | 91.6% | 91.8% |
| III | Right hand | 93.4% | 95% |
| IV | Right Leg | 89% | 96% |
| V | Left Hand | 93% | 89% |
| VI | Left Leg | 89% | 87.5% |
| VII | Both Legs | 72% | 75.2% |
| VIII | Head | 90% | 88% |
| IX | Combined Motions | 77% | 87% |
| **RF *MotionTree* Accuracy** | | 89.3% | |

Table 4.7: Recall and precision for each class with the whole accuracy for the entire system using RF *MotionTree*.

### 4.5.3 Two Level Tree Topology

From the previous two topologies, we can see that Flat, or one level, classification is better in Random Forest than using multi-level tree, like *MotionTree*. That gives us an idea of using another topology in between the previous two. We use here another tree topology but with two levels only. We keep the right side of the tree in *MotionTree* as is and put all the classifiers in the left side in one flat classifier. In this tree we have 3 classifiers. We start first by classifying motions into two large families, big and small. The right side is used to classify big motions into (1) turning/rolling right or (2) turning/rolling left. On the left side, we have one classifier to classify all small motions into the remaining 7 classes: (3) right hand, (4) right leg, (5) left hand, (6) left leg, (7) both legs, (8) head, and (9) combined motions. Figure 4.11 shows the 2-level tree that we improved for Random Forest. To evaluate this new tree design, we use the same data with the same cross validation strategy (500 tests with cross validation of

randomly 60% of the data for each classifier selected for training and 40% for testing). Table 4.8 shows the recognition accuracy for each Random Forest classifier and the accuracy of the entire small tree. We name this tree as *RF-Small-Tree*. This tree has a slightly better recognition accuracy with 90.9%. Table 4.9 shows the average recall and the average precision for each class with the whole accuracy for the entire system using *RF-Small-Tree*.

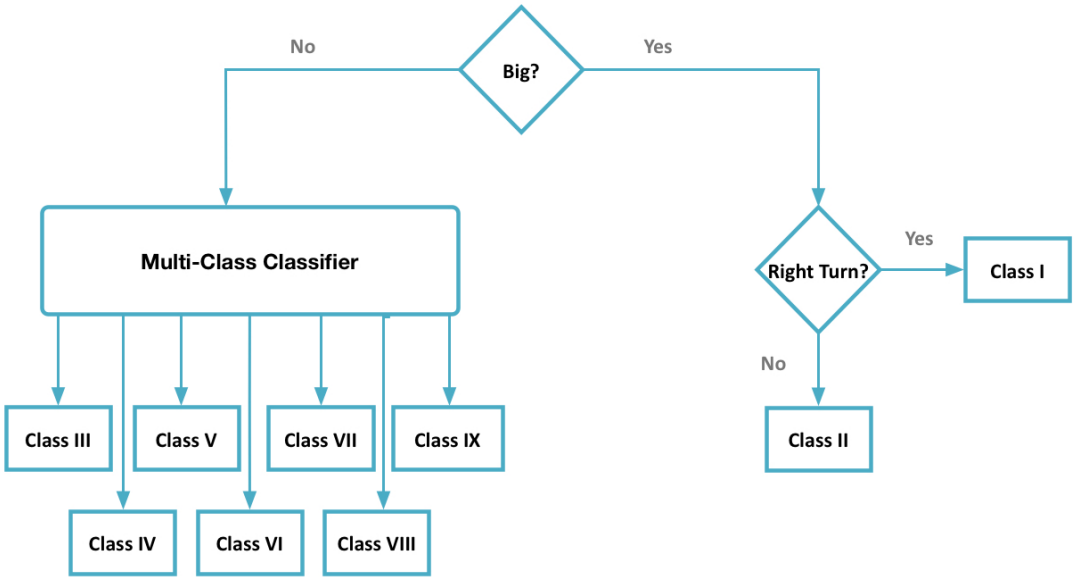



Figure 4.11: Two levels tree for Random Forest.

| RF Model Name | Average Accuracy |
|---|---|
| Big? | 99.4% |
| Right Turn/Roll? | 92.48% |
| Small-Motions-Classifier? | 90.8% |
| **Whole Small Tree Accuracy** | 90.9% |

Table 4.8: Random Forest models' accuracy and the whole accuracy of the 2-levels tree, we call it RF-Small-Tree

| **Class Number** | **Class Name** | **Recall Using *RF-Small-Tree*** | **Precision with *RF-Small-Tree*** |
|---|---|---|---|
| I | Turning/Rolling Right | 92.1% | 91.3% |
| II | Turning/Rolling Left | 91.7% | 92% |
| III | Right hand | 93.69% | 94.7% |
| IV | Right Leg | 90% | 97% |
| V | Left Hand | 93.3% | 93.7% |
| VI | Left Leg | 87.7% | 91.8% |
| VII | Both Legs | 82.5% | 77% |
| VIII | Head | 92.8% | 87.2% |
| IX | Combined Motions | 78.4% | 90.6% |
| ***RF-Small-Tree* Accuracy** | | 90.9% | |

Table 4.9: Recall and precision for each class with the whole accuracy for the entire system using *RF-Small-Tree*.

## 4.6 XGBoost

Gradient Tree Boosting [71] is one of the machine learning techniques that shines in many applications. In this section, we describe XGBoost, a scalable end to-end tree boosting system [72]. XGBoost is short for Extreme Gradient Boosting. XGBoost is an optimized distributed gradient boosting system designed to be highly efficient, flexible and portable. It runs ten times faster than existing popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings [72]. Several important systems and algorithmic optimizations are reasons for this scalability. XGBoost includes: a novel tree learning algorithm for handling sparse data, a theoretically justified weighted quantile sketch procedure, which enables handling instance weights in approximate tree learning, and fast learning due to parallel and distributed computing, which enables quicker model exploration. XGBoost combines these techniques to make an end-to-end system that scales to even larger data with

the least amount of cluster resources [72]. More details about XGBoost can be found in [73].

### 4.6.1 Applying XGBoost in our System

To apply XGBoost technique in our system for classifying in-bed movements, we use the flat topology. We use the same data we have for the same set of features. We have 24 features, which represent every detected motion. XGBoost is capable of finding the best features it needs for its tree. To review, we have the following 24 features: the change of the weight in each load cell (4 features), the change of the center of mass in both $x$ and $y$ axes (2 features), the change of the center of mass around its mean in both $x$ and $y$ axes (2 features), the motion trajectory length, the Euclidean distance between the centers of mass before and after the motion, the log peak value (as defined in section 3.3.2), summation for the real, absolute, and square values of the change in the body center of mass on both axes during the movement (6 features), summation for the real, absolute, and square values of the change in the body center of mass on both axes from its initial point (6 features), and the movement's duration. We explain all these features in section 4.3.2. To show how XGBoost works in our system, we pick randomly 60% of the given data for training and the 40% for testing. XGBoost gives the importance for each feature (of the 24) as shown in Figure 4.12. This is a metric that simply sums up how many times each feature is split on. Depending on this importance, XGBoost updates its features weight on its internal tree with every iteration. We use 250 times as number of rounds the XGBoost tunes its internal weights depending on previous errors.

To evaluate this technique, XGBoost, in our system, we use the same data with same cross validation strategy (500 tests with cross validation of randomly 60% of the data for each classifier selected for training and 40% for testing). The average accuracy for all these 500 trials was 90.5%. Table 4.10 shows the average recall and the average precision for each class with the whole accuracy for the entire system using XGBoost.

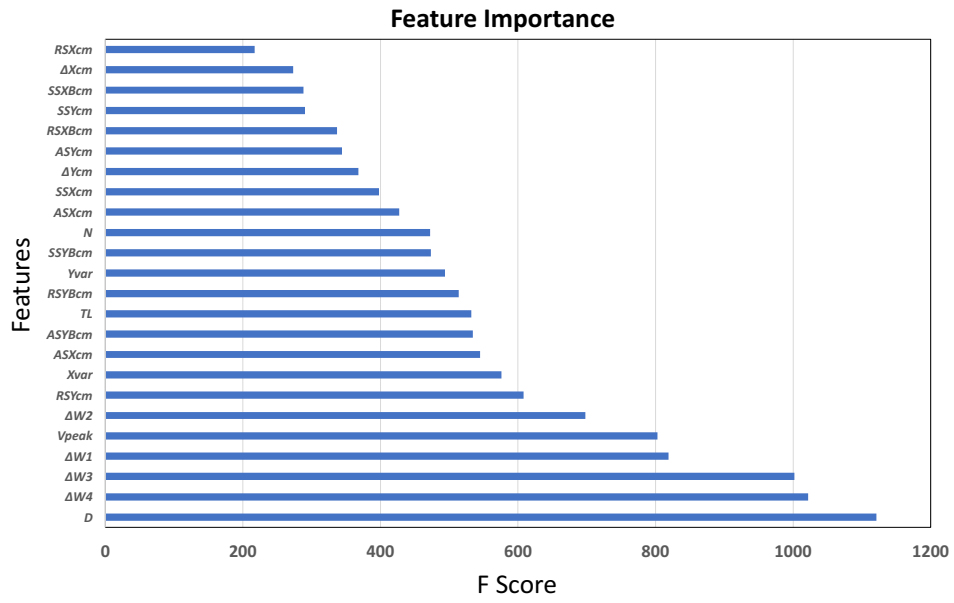



Figure 4.12: Features importance according to XGBoost

| Class Number | Class Name | Recall Using XGBoost | Precision with XGBoost |
|---|---|---|---|
| I | Turning/Rolling Right | 90.6% | 90% |
| II | Turning/Rolling Left | 91% | 90.7% |
| III | Right hand | 93.6% | 94% |
| IV | Right Leg | 90% | 94% |
| V | Left Hand | 93.5% | 94% |
| VI | Left Leg | 87% | 88% |
| VII | Both Legs | 80% | 78% |
| VIII | Head | 92.6% | 87.8% |
| IX | Combined Motions | 77.3% | 89.2% |
| **XGBoost Accuracy** | | 90.5% | |

Table 4.10: Recall and precision for each class with the whole accuracy for the entire system using XGBoost.

## 4.7 Logical Combination Approach

In this study, we use three machine learning techniques to classify all in-bed motions into the predefined 9 classes. We have used all these techniques separately. We try here to have a final decision by combining all the previous 3 techniques, SVM, Random Forest, and XGBoost, together. This combination is done in a logical way. Basically, it says:" If we have 2 out of three with the same result, we choose that result as a final decision." Figure 4.13 shows the main idea of this logical combination. We take *MotionTree*, RF-Small-Tree, and XGBoost outputs for each test and do logical OR relation for them. If 2 or more are the same, then we choose that same output as the final result. In case all the 3 previous techniques are different, we take the output from the RF-Flat topology, since it has a very good recognition accuracy. The idea behind that, is to have an accuracy that is better than all of them. Algorithm 1 explains the logical combination.

In order to evaluate this combination algorithm, we use the same data with same cross validation strategy (500 tests with cross validation of randomly 60% of the data for each classifier selected for training and 40% for testing). Every single element is tested by the four techniques we have. We combine the four outputs as explained in the algorithm. The average accuracy for the combination algorithm is 91.5%, which is better than using any of the previous techniques separately. Table 4.11 shows the average recall and the average precision for each class with the whole accuracy for the entire system using the combination algorithm.

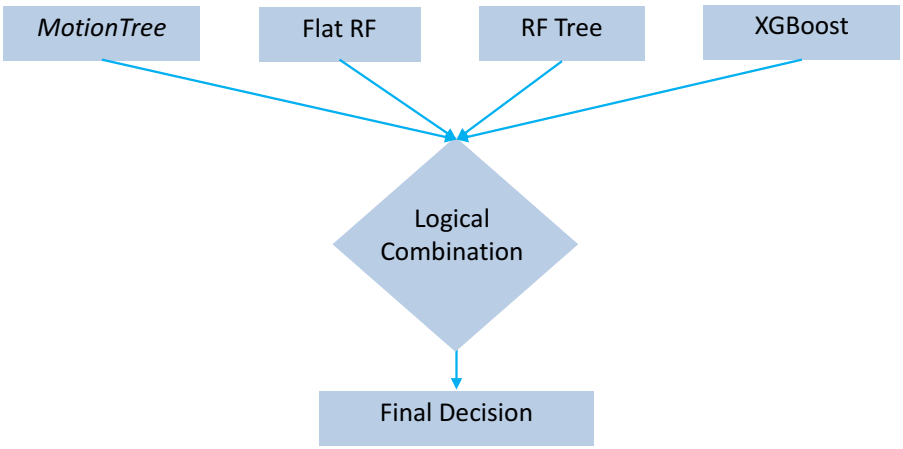



Figure 4.13: The logical combination of all techniques.

**Read Data Element**
initialization;
SVM-Output = *MotionTree* (Data Element);
RF-Tree-Output = RF-Small-Tree(Data Element);
XGBoost-Output = XGBoost(Data Element);
RF-Flat-Output = RF-Flat(Data Element);
**if** *RF-Tree-Output* = *XGBoost-Output* **then**
  Result = RF-Tree-Output;
**else**
  **if** *RF-Tree-Output* = *SVM-Output* **then**
    Result = RF-Tree-Output;
  **else**
    **if** *XGBoost-Output* = *SVM-Output* **then**
      Result = XGBoost-Output;
    **else**
      Result = RF-Flat-Output;
    **end**
  **end**
**end**

**Algorithm 1:** Combination algorithm

| Class Number | Class Name | Recall Using Combination Algorithm | Precision with Combination Algorithm |
|---|---|---|---|
| I | Turning/Rolling Right | 92.3% | 91% |
| II | Turning/Rolling Left | 91.5% | 92% |
| III | Right hand | 94.2% | 95.7% |
| IV | Right Leg | 90.6% | 97.2% |
| V | Left Hand | 93.8% | 95% |
| VI | Left Leg | 89% | 93% |
| VII | Both Legs | 85.4% | 78.6% |
| VIII | Head | 93.4% | 87.8% |
| IX | Combined Motions | 79% | 91% |
| **Combination Algorithm Accuracy** | | 91.5% | |

Table 4.11: Recall and precision for each class with the whole accuracy for the entire system using Combination Algorithm.

## 4.8 Concluding Remarks

In this chapter, we have described the design, implementation, and evaluation of *MotionTree*. It is a low-cost, low-overhead, and highly robust system for in-bed body movement detection and classification that uses low-end load cells and special designed decision tree. Compared to other existing solutions, *MotionTree* uses affordable hardware, and it is very easy to apply in our lives unobtrusively. *MotionTree* uses load cell sensor to detect different kinds of in-bed body movements. It classifies in-bed motions into 9-classes: turning/rolling right, turning/rolling left, right hand, right leg, left hand, left leg, legs, head, and combined motions. 24 features are extracted from each motion to be used for training and testing a special designed binary decision tree based

on SVM. It is a multilevel classification tree to classify the predefined 9-classes. We select the best features for each decision level by applying Random Forest technique to boost the overall classification accuracy. We have built a *MotionTree* prototype and evaluated it extensively in experiments that involved 40 subjects. Each subject has been asked to do 35 different type movements. Our results show that *MotionTree* can detect movements and classify them into 9-classes with an average accuracy of $90\%$.

In addition, we have described the implementation, and evaluation of using another two machine learning techniques in our system. They are Random Forest and XGBoost. We have used Random Forest in three topologies: Flat, multi-level tree (*MotionTree*), and 2-level tree (RF-Small-Tree). XGBoost has been applied in one topology only which is the flat (one-level) topology. All these techniques have been used separately to classify in-bed motions into the predefined 9-classes. We have applied a combination algorithm to combine all results from the previous explained techniques to have our final results. Table 4.12 shows the classification accuracy for all techniques we have used. It also shows that the accuracy of the combination algorithm (91.5%) is better than all of the rest.

| **Technique's Name** | **Average Accuracy** |
|---|---|
| Flat SVM | 65% |
| SVM *MotionTree* | 90% |
| Flat RF | 90.8% |
| RF *MotionTree* | 89.3% |
| RF-Small-Tree | 90.9% |
| XGBoost | 90.5% |
| **Combination Algorithm** | 91.5% |

Table 4.12: The average accuracy for all machine learning techniques we have used with the combination algorithm accuracy

# Chapter 5
# MotionPhone

## 5.1 Introduction

In the recent years, personal well-being management using smart sensors has received a great deal of attention in both academia and industry. As far as one's well being is concerned, an important aspect is the ability to continuously monitor a person's mobility during sleep. Due to that, we introduce another system for in-bed motion detection and classification and try to compare it with our *MotionScale* system. Among the array of motion sensing techniques, sensing bed vibrations caused by movements presents a promising approach because of its accuracy and ease to use, in which the sensor can be attached to any position in the bed without worrying about the subject's sleeping position/posture, as shown in the recent study [53].

The vibration based approach, however, still faces several significant challenges before it can serve as a reliable at-home in-bed motion sensing/monitoring technique. Some of these challenges are accuracy, installation, and cost. Towards this end, we seek to fill this void by proposing a system that is accurate, robust, low cost, and easy to use. Our solution involves the use of a commercial off-the-shelf analog geophone under the mattress to detect and monitor the user's motions during sleep. Just like a geophone can detect pressure waves (i.e. "sounds) in the earth (e.g., [74, 75]), our system can detect the vibration of motions that are propagated through a mattress. Therefore, we refer to this system as *MotionPhone*. Compared to other sensors, the geophone sensor has several advantages, which make it a suitable choice for motion

detection[1]. Firstly, it is highly sensitive to tiny motions – geophones are often used to detect distant motions (such as earthquakes), and can generate a noticeable response to body movements (after going through a normal mattress). Secondly, it is commercially available and rather affordable. Thirdly, deploying a geophone-based system can be very conveniently done, without interfering with the bed or how it is used. As a result, we believe that *MotionPhone* offers a very practical solution to at-home in-bed motion detection during sleep.

With *MotionPhone*, we can simultaneously detect both large and small movements and classify these movements (as in *MotionScale*). We tried to keep the wireless manner that we had in *MotionScale*. We have designed other amplifier circuit that can handle a wide range of movements. As far as the software design is concerned, we have applied the same signal processing algorithms that we used in *MotionScale* to extract body movements. Through these techniques, our experimental results that involve 15 subjects show that we can detect 35 types of body movements with an error rate of 2%, and can classify these 35 types of movements into big and small movements with an error rate of 8%.

To summarize, we have made the following contributions in this study:

1. We have developed an accurate, robust, low-cost, and easy-to-use in-bed motion detection system *MotionPhone*, which is centered around a commercial off-the-shelf analog geophone. The *MotionPhone* system consists of both hardware and software components. Its hardware components include a geophone, an amplifier and a wireless communication unit (which consists of an A-to-D converter); software components involve filtration, feature extraction, and detection and classification.

2. We have built a *MotionPhone* prototype and used it to instrument an experimental bed. We have used the experimental bed to collect the signals of 2 geophone

[1]In this study, we use the term geophone to refer to the analog geophone.

sensors, one at the upper half of the bed and the other at the lower half, from 15 subjects, with 30 experiments. Each subject should make 35 different body movements during each experiment. We have compared the detected body movements against the ground truth observed captured by a video camera, and found that the average error rate is 2%.

3. We have also used the same data to classify these 35 body movements into big movements and small movements Also, we tried to classify movements as big, legs, and hands/head movements. We compare the classification results against the ground truth observed by a video camera, and found that the average error rate is 15%.

The remainder of the chapter is organized as follows. In Section 5.2, we describe the hardware system design of *MotionPhone*, and in Section 5.3, we describe *MotionPhone*'s signal processing algorithms. We present our evaluation setup and experimental results in Section 5.4. Finally, we provide concluding remarks in Section 5.5.

## 5.2 *MotionPhone* System Design

We show the overview of *MotionPhone*in Figure 5.1. In *MotionPhone*, we place two analog geophones under a mattress to capture movements in the environment. We first amplify the raw geophone response, and then convert it to a digital signal. Next, we transmit the digital geophone signal in a wireless manner to the receiver. A series of signal processing steps are done for the received signal to detect movements from the signal. The outcome from the *MotionPhone* system includes detection of body movements during sleep, and classify these movement as big or small.

In this section, we first present the hardware design of *MotionPhone*. Then we discuss the unique challenges we have faced in designing the *MotionPhone* system.

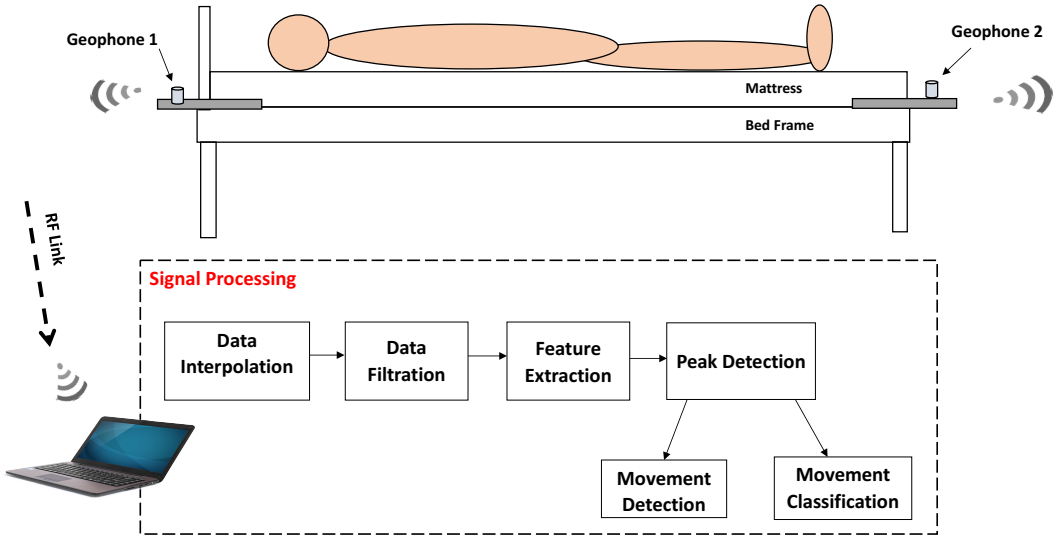



Figure 5.1: Overview of the *MotionPhone* system. An analog geophone is placed under a mattress. The raw geophone signal goes through amplification and A/D conversion to generate a digital signal that is suitable for subsequent signal processing. A series of signal processing methods will then be applied to detect motions in the signal.

### 5.2.1 *MotionPhone* Hardware Design and Prototype

The *MotionPhone* system is centered around the use of a geophone sensor. As shown in Figure 5.2, a geophone consists of a spring-mounted magnet that moves within a wire coil to generate a voltage, which can thus measure the speed of a movement at different frequencies. The use of a powerful magnet and a differentially wound coil gives it low noise and high sensitivity at frequencies 7Hz and above, while being less sensitive to movements with lower frequencies. In our *MotionTree* prototype, we use the SM-24 Geophone Element [76], whose natural frequency is at 10Hz.

The raw geophone signal is first filtered by a hardware bandpass filter in the range from 0.25 to 10kHz, which is then fed to a TI LMV358 amplifier circuit [77]. We have carefully configured the amplifier circuit to ensure the *MotionPhone* is able to sense all different kind of body's movement (ranging from finger's tapping to whole body rolling). For this purpose, we configured the amplification circuit such that the motion

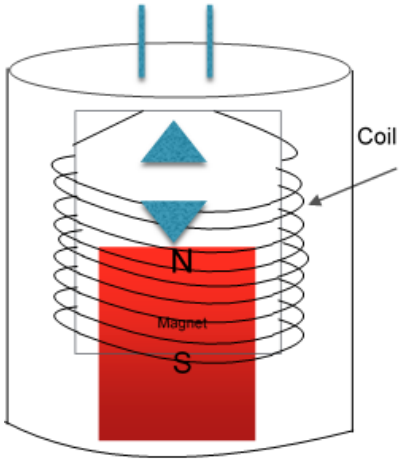



Figure 5.2: The geophone consists of a spring-mounted magnet that is moving within a wire coil to generate electrical signals that measure movements in the environment.

signal's amplitude falls within 0-3V, which is a range determined by the resolution of our ADC.

Figure 5.3 shows the resulting double-stage amplification circuit. Both the first-stage and second-stage amplifying circuit have a RC bandpass filter in the range from 0.25Hz to 10kHz. The gain of the first-stage amplifier is 4 so that we can reduce some noise from the circuit itself. The maximum gain of the second-stage amplifier circuit is 20 and the gain is adjustable by tuning the adjustable resistor $R_7$ shown in Figure 5.3. In total, the maximum gain of this circuit is 80. The amplified signal is based on 3.3V and quantized to 1024 levels (10 bits) using the A/D converter [46] in the PIP-Tag. Our sampling frequency is 30 Hz. For that, we have 30 packets per second to be send from each geophone.

In Figure 5.4, we show the picture of our prototype *MotionPhone* system. We attached the geophone to a piece of wooden lumber and insert the wood under a memory-foam mattress. Lying down on the bed, the user does not feel the geophone at all, and user's sleep will not be interfered in any way.

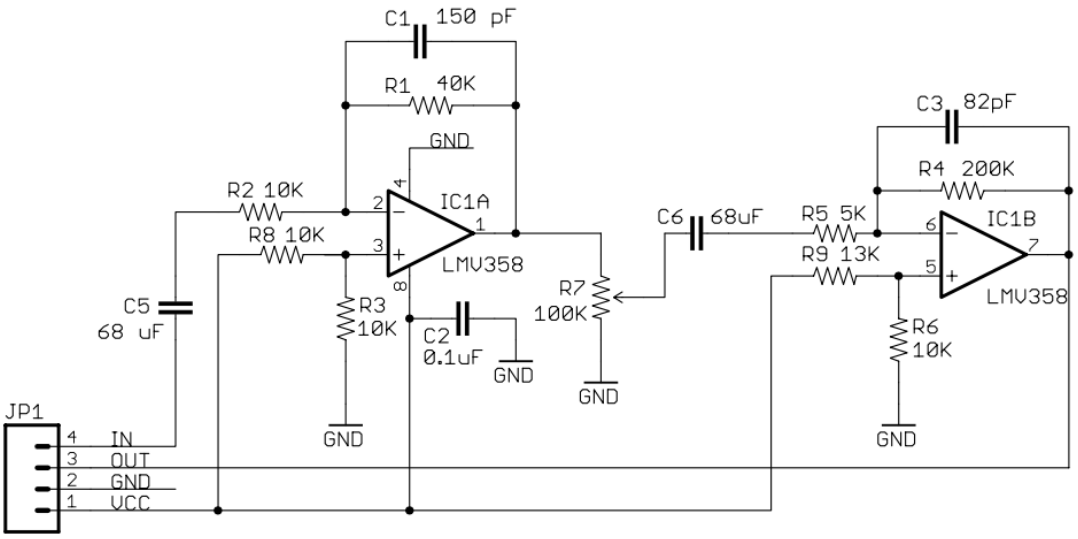



Figure 5.3: The AC amplifier circuit design.

### 5.2.2 Understanding Unique Challenges of the Geophone

Using the geophone to detect a body's movement that propagates through a mattress poses serious challenges to the underlying system design. Below we discuss these challenges.

**Noises Caused by Other Movements.** The first challenge lies in the high sensitivity of the geophone sensor, which is also the very reason why we choose this sensor in the first place. The geophone responds to tiny motions or vibrations in the environment – when placed under the mattress, its response signal shows fluctuation when someone walks in the room or someone closes the door. Thus, we need to differentiate in-bed body's movements from other movements. Examples include other people walking around while the subject is in sleep, fans in the room, pets moving on the bed, etc.

In order to address this challenge, we need to carefully adjust our amplifier to do not amplify the very tiny and weak ones, and extract the right features that reflect the real in-bed movements.

**Insensitivity to Low Frequency Movements.** Secondly, we note that the geophone sensor is only sensitive to signals that are higher than a certain frequency (7Hz in our

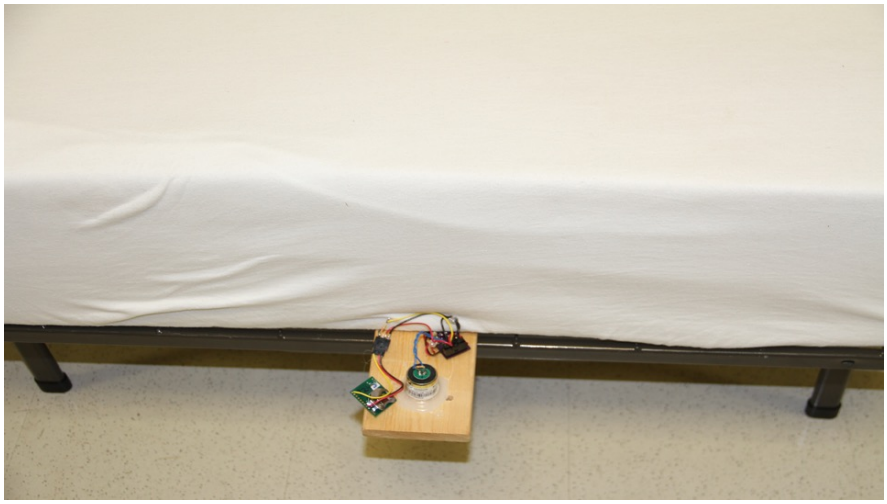


Figure 5.4: The picture of our *MotionPhone* prototype, where the geophone, the amplifier, and the PIP-Tag are attached to a wooden board that is inserted between mattress and bed frame.

case). This can be explained as follows. As Figure 5.5 shows, the geophone response increases quadratically with frequency when the frequency varies within the range from 1Hz to 12Hz if the movement speed is fixed. For example, let us consider a movement at 1m/s, the geophone generates a voltage about 20V when the frequency is at 10Hz, and a voltage of .2V when the frequency is 1Hz, resulting in a 100 times response increase. Hence, the geophone itself works as a second-order high-pass filter, which is hard to detect responses to low-frequency movements.

The fundamental frequency range of body motion signal falls between 0.2Hz and 4Hz. As explained above, the geophone response to movements at 10Hz would be 100 times as strong as the response to movements at 1Hz. After taking a closer look at the body motion signal and the corresponding geophone responses, we notice that on-bed movements have multiple harmonic frequencies and they are strong enough to oscillate the geophone in a very sufficient way to cover all possible movements.

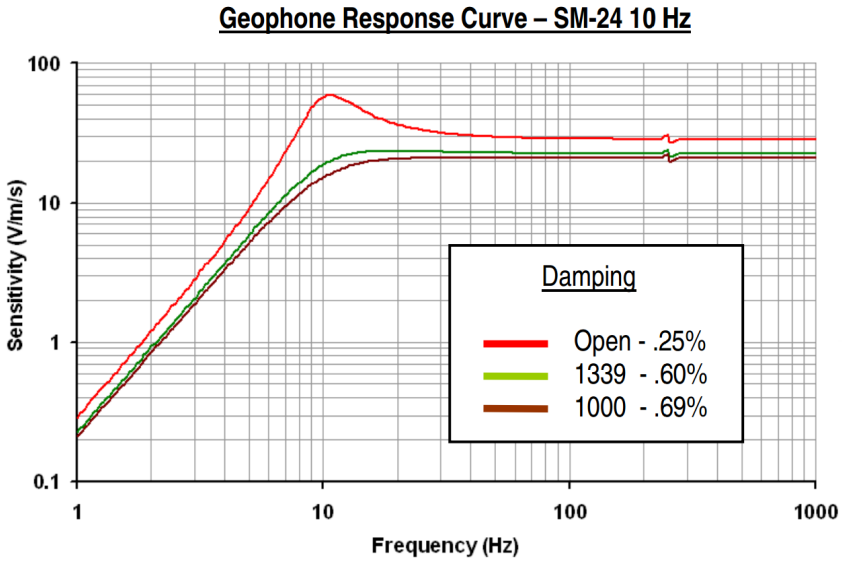



Figure 5.5: Geophone response curve from the data sheet of Geophone SM-24 [1].

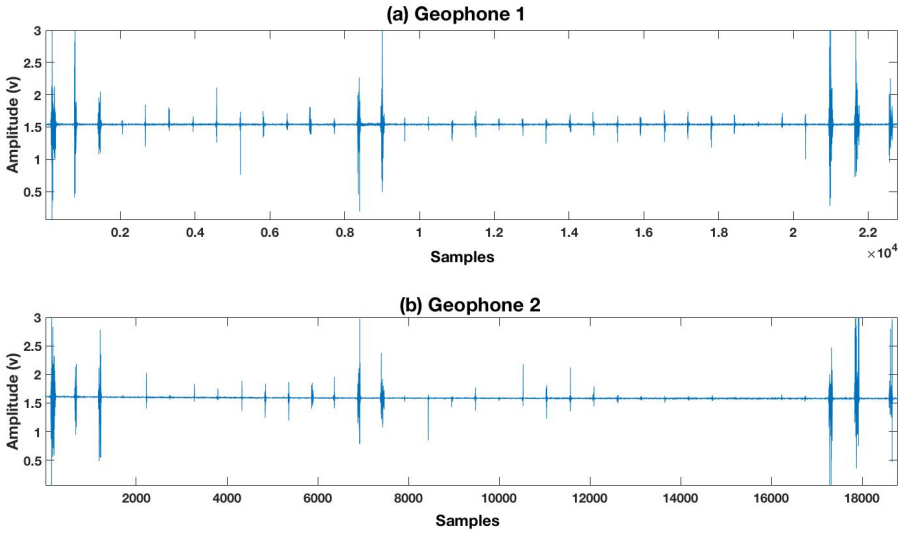
## 5.3 *MotionPhone* System Design

In this section, we explain how our system is designed to process the data from the two geophones to mitigate noise and further detect in-bed motions. Motions detection and classification are the main goals for this system. We basically, used the same steps we have used in Chapter 3

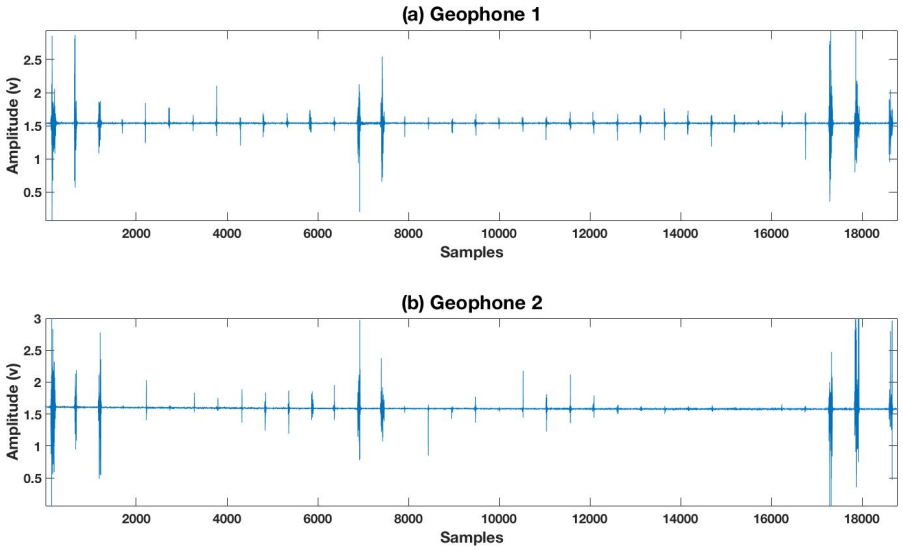
### 5.3.1 Data Pre-processing

After raw data are collected from geophones, our system first performs a sequence of preprocessing steps. We interpolate the collected data using the spline interpolation to have all collected signal in the same volume. Then we drop the local mean of the signal because our set of features, as will be explained later, requires a signal that oscillate around the zero axis. In particular, we calculate the mean values in a moving time window of 50 samples. That also helps to focus on the portion of data when we have a motion, as explained in Chapter 3. After that, the data is filtered using a low pass filter of a 10 Hz of cutoff frequency. Figure 5.6 shows the collected data, before and after interpolation. Figure 5.7 shows the data raw for geophone1, before and after removing

the local mean.




(a) Geophones 1 and 2 data before Interpolation




(b) Geophones 1 and 2 data after Interpolation

Figure 5.6: Geophones raw of data before and after doing the interpolation on the data of 10 minutes experiment with some movements of a subject on the bed.

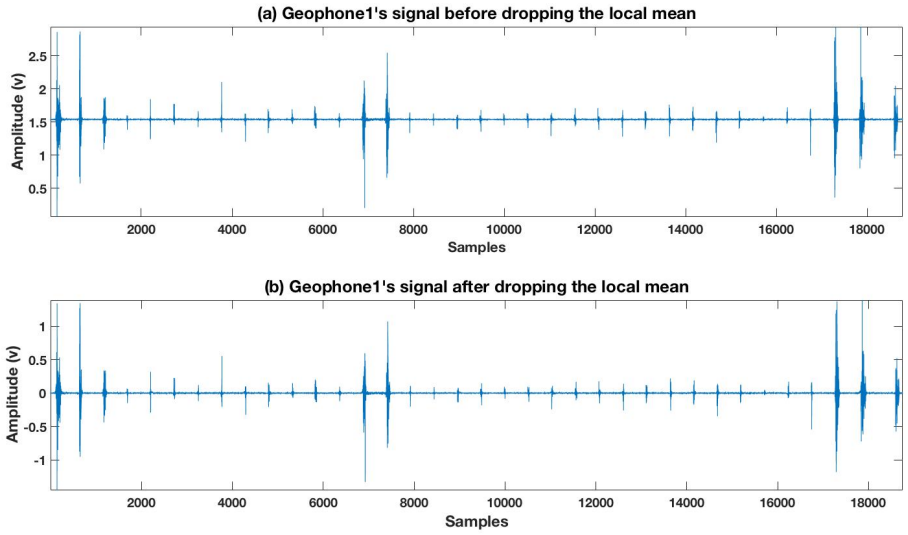



Figure 5.7: The data of geophone1 before and after removing the local mean. Local mean is 50 samples.

### 5.3.2 Feature Extraction

We extract the same features we have in Chapter 3 from the preprocessed data and adopt a simple threshold-based detection/classification strategy. All following operations are done on the two signal from geophones 1 and 2. We will show the graphs and explain the extraction from geophone1's signal only. It is exactly the same for geophone2's signal.

**Energy-Peak Feature Extraction:** We observe that whenever an in-bed movement is performed, there exists stronger oscillation with high amplitude in the collected geophone readings. It means that signal in that portion of oscillation has more energy. We extract the energy in every 2 seconds window (the same strategy we have in Section 3.3.2). We will have a peak whenever we have a movement. Its height depends on the strength of movement. The stronger movement results in higher energy peak. All energy widows are normalized with highest value window. Peak detection is applied with some threshold values to find all movements in the data. Figure 5.8 shows the signal from geophone and its energy graph for 2 seconds window. We can see the peaks

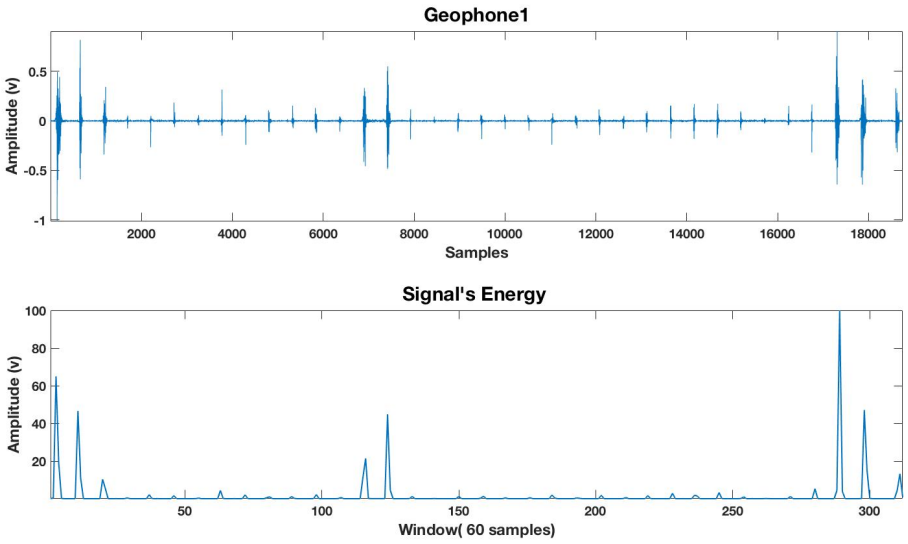



Figure 5.8: The signal from geophone1 and its energy computed for 2 seconds window, or 60 samples.

whenever we have movements.

**Log-Peak Feature Extraction:** To have a good view to show both small and large values, we use Log-Peak. It uses the logarithm of a physical quantity instead of the quantity itself. Specifically, we first square raw data signal collected from each geophone sensor, then log (i.e., natural log where log to the base e) is applied to the squared signal. We then apply a low pass filter with a cutoff frequency 0.2 Hz to the log's output to make it smoother. We notice that we have a clear peak whenever there is an in-bed body movement. Figure 5.9 shows geophone1's signal, its square, and its log output after filtration. Threshold is applied to find peaks, which also means to find movements.

**Zero-Crossing (ZeroX-Valley) Feature Extraction:** Here we tried to find the zero-crossing rate for geophone's signal using the same 2 seconds window. We tried here to repeat the same idea we had in Chapter 3 but the results are not the same. We noticed that geophone signal does not reflect the same behavior, like what we have for load cell, and extracting Zero-Crossing feature would not help for further steps. Figure 5.10 shows geophone1's signal (after removing the local mean) with its ZX rate graph per 2 seconds window. We can see that this feature is not useful here. We can

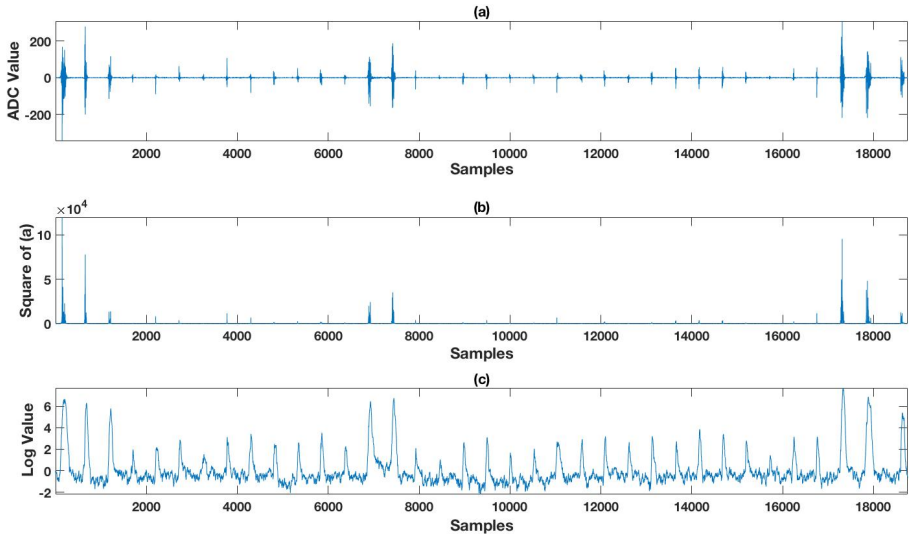



Figure 5.9: (a) Geophone1's signal. (b) Square of the data from geophone 1. (c)The log result of the squared value after filtration (using 0.2 Hz low pass filter).

not see clear valleys or peaks whenever we have movements. So we will not use it as a feature in our detection and classification steps.

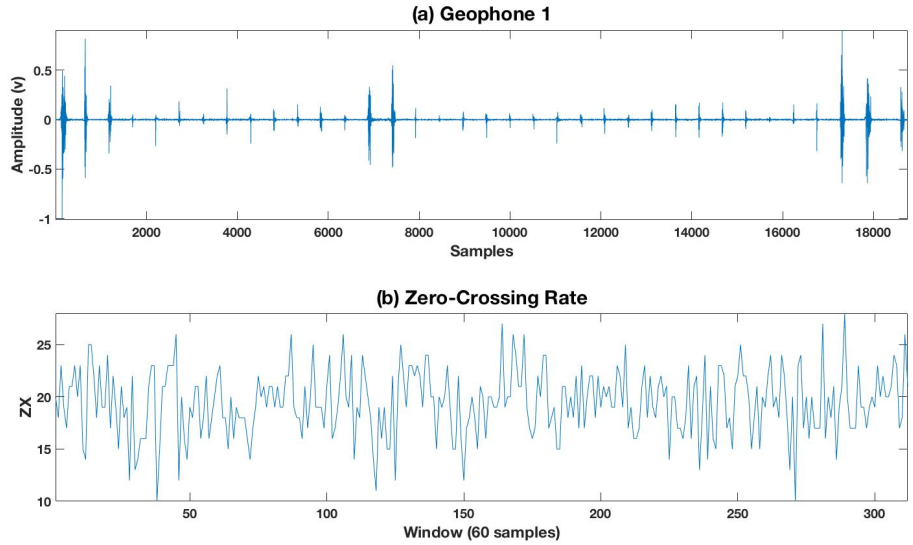



Figure 5.10: (a) Geophone1's signal (b) The ZX rate computed for 2 seconds window, or 60 samples.

## 5.4 Performance Evaluation

In this section, we describe the experimental methodology, and then present the evaluation results for both movement detection and classification.

### 5.4.1 Experimental Methodology

The experiments are conducted on a twin size bed in a university laboratory with 15 healthy subjects (11 males and 4 females, age ranging from 20 to 43 years old) [2]. A common innerspring mattress with dimension of 95cm (width)$\times$190cm (length)$\times$20cm (height) is on the bed. During the experiments, we ask each subject to perform 35 pre-defined in-bed movements with 15 seconds quiet period after each movement. Among all 35 pre-defined movements, there are 8 large movements involving the entire body (e.g., getting in/off bed, turning left, turning right or rolling over), and 27 small movements that only involve parts of the body (e.g., head, arms and legs). More specifically, 8 of the 27 small movements are leg movements, and the rest are arm and head movements.

We record all the data using the same prototype and laptop to avoid any possible bias in readings. Two geophones have been used, geophone1 in the upper half (close to head) and geophone2 in the lower half (close to legs). A camera is mounted on a tripod 1.5 meter away from the bed to record videos for the ground truth recording.

### 5.4.2 Motion Detection

As we mentioned in Section 5.3 about the set of features we use for movement detection, we use only Log-Peak and Energy-Peak to detect motions. In order to conduct a fair comparison, we report each feature's performance using the best threshold value

[2] Our studies were approved by the Institutional Review Board (IRB) of our institution.

for that feature. From our experiments, we observe that each feature presents an obvious peak whenever there is a motion. These peaks are very different in amplitudes and widths, which suggests that we need to find a general threshold (i.e., height of peak) to detect the 35 performed in-bed movements. In order to find the best threshold value for each feature, we apply different threshold values on randomly $50\%$ selected from the collected data for a total of 100 times and choose the one that gives the best performance. The peak value threshold for Log-Peak is varied from -1 to 9 in 100 steps, the peak value threshold in Energy-Peak is varied from 0 to 5 in 100 steps.

We have a total of 15 subjects who did 30 experiments, and for each of the 100 tests, we randomly choose 50% of the data for training and 50% for testing. For each test subject, our detection algorithm detects $n$ movements, and the detection error rate is thus calculated as $\left|\frac{35-n}{35}\right|$ where 35 is the number of known movements in each experiment. Figure 5.11 reports the detection error rate distribution of the 100 experiments for each feature. It is very clear that Log-Peak is better than Energy-Peak, delivering a detection error rate of 2%.

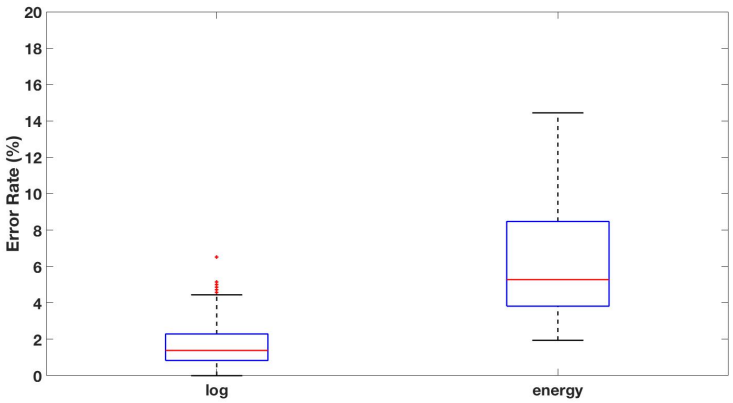
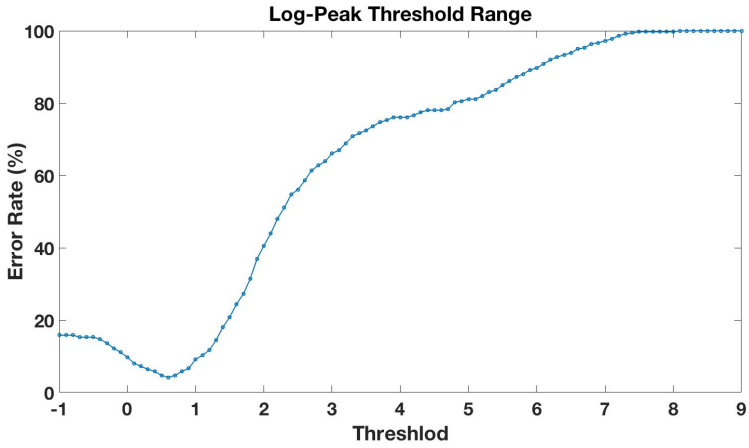
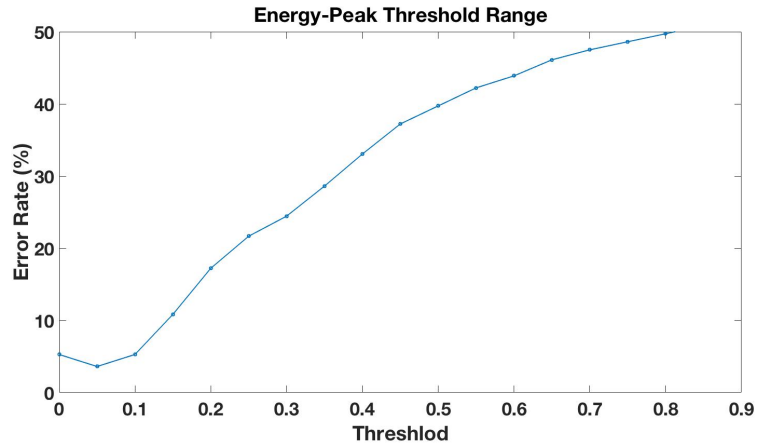



Figure 5.11: The error rate for testing phase.




(a) The error rate of variation the threshold in Log Peak strategy




(b) The error rate of variation the threshold in Energy Peak strategy

Figure 5.12: The error rate of the two strategies when we varied the threshold. All the 15 subjects are tested here.

We next try to find the best threshold values on the performance of different detection strategies, and report the results in Figures 5.12(a)-(b) respectively. These curves exhibit a "U" shape, meaning that there is an optimal value for each threshold. choosing the proper threshold (around the optimal value) helps the corresponding strategy to detect peaks caused by valid body motions and we achieve the lowest error rate. This is the same way that we used in Chapter 3. Figure 5.13 shows the ROC curves of the two strategies in movements detection. This ROC curve shows that Log-Peak has the best detection performance. Table 5.1 shows the detection accuracy for both strategies.

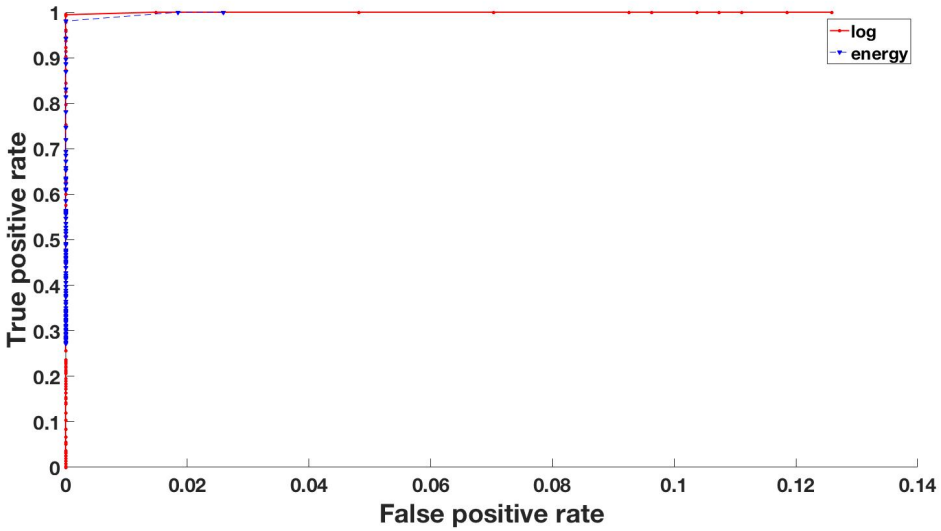



Figure 5.13: The ROC curve for the two strategies applied on 15 subjects.

| Strategy | Movement Detection | |
|---|---|---|
| | Best Threshold | Error rate |
| Log | 0.6 | 2% |
| Energy | 0.05 | 6% |

Table 5.1: Best Thresholds and their associated error rate to detect all movements.

### 5.4.3 Performance of Movement Classification

**Big or Small:** In the second part of the evaluation, we start with classifying each detected movement as either a big movement or a small movement. The definition of big and small is the same that we mentioned in Section 3.4. For that purpose, we use the data from geophones 1 and 2, and apply Log-Peak strategy to detect motions (since it is the best). For each detected motion, we have four features:

1. $Peak1_{Val}$; the value of the log peak from geophone1.
2. $D1$; the duration of motion from geophone 1.
3. $Peak2_{Val}$; the value of the log peak from geophone2.
4. $D2$; the duration of motion from geophone2.

The rationale behind these features is that big movements normally possess higher energy and longer duration than small ones, so we expect higher peaks for big ones. Moreover, as a classifier tool, we use Random Forest technique here. To evaluate our system, 100 tests have been done with cross validation of randomly 50% of the data for training and 50% for testing. In each iteration, we have never used any portion of the data that picked for training in the testing phase. We found that the average value of classification accuracy (Big vs Small) is 98.5%.

**Big, legs, or hands/head:** We try here to classify our 35 movements into 3 classes: big, legs, or hands/heads movements. In other words, we try to classify movements into a big class, which covers motions of the entire body, a legs class, which includes lower half body's movements that are usually from legs, and a head/hands class, which covers upper half body's movements that are from heads or hands. We could not have more classes with deeper meaning like those had in *MotionTree* because the sensor here is very different and does not provide any information about body center of mass. Therefore, we cannot have more features than the four features we explained before.

For these three classes: Big, Legs, and Hands/Head, we use Random Forest classifier with the same four features. We have repeated the same strategy of 100 tests with cross validation of randomly 50% of the data for training and 50% for testing. The average value of recognition accuracy for the above 3 classes is 88%. Table 5.2 shows the classification accuracy for our Random Forest classifier, for both 2 classes recognition (Big or Small), and for 3 classes recognition (Big, Legs, or Hands/Head).

| Recognition Size | Classification Accuracy |
|---|---|
| 2 classes | 98.5% |
| 3 classes | 88% |

Table 5.2: Classification accuracy using RF for 2 classes (Big or Small), and for 3 classes ( Big, Legs, or Hands/Head movements)

As a result, we can see that *MotionPhone* system has a better accuracy in detecting movements than *MotionScale*, with 98% for *MotionPhone* versus 93.8% for *MotionScale*. On the other hand, *MotionPhone* does not provide the same classification accuracy for multiple classes like what we have in *MotionTree* and the combination algorithm, 88% in *MotionPhone* for 3 classes versus 91.5% in the combination algorithm for 9 classes.

## 5.5 Concluding Remarks

In this chapter, we have developed unobtrusive, low-overhead, and highly robust system for in-bed movement detection and classification system. This system, *MotionPhone*, utilizes geophone sensors to capture any in-bed movements. Compared to existing solutions, *MotionPhone* is very sensitive and can use low-cost hardware to achieve comparable results, and it is very easy to apply in our lives unobtrusively. By utilizing geophone sensor, *MotionPhone* can detect different types of in-bed body movements with different scales, ranging from fingers movements to whole body movements (e.g., turn over, get off bed). To evaluate our system, we build a prototype with two geophone sensors, one close to the head and the other close to feet, and PIP-tags to provide wireless connection. We extensively experiment the prototype with $15$ participants over two-month time period. The results show that by utilizing our two main strategies, Log-Peak and Energy-Peak, *MotionPhone* can effectively extract body movement signals from geophone data and detect in-bed movements with a low error rate of $2\%$. We have built a classification approach using Random Forest technique and 4 features were extracted from every detected movements. Our system can classify movements to big or small movements with an error rate of $1.5\%$. In particular, this study provides the first, strong evidence that geophones can be used as a low-cost solution for at-home sleep monitoring.

# Chapter 6
# Conclusion and Future Work

## 6.1 Summary

In conclusion, we propose a system that can efficiently detect and classify in-bed movements. Specifically, we have made the following contributions:

- *MotionScale*: We have proposed an in-bed movements detection system that is based on a wireless load cell as sensing unit. It is a low-cost, unobtrusive, robust and easy to install system. Almost all body motions that happen on a bed can be detected. The results show that by utilizing our three main strategies, Log-Peak, Energy-Peak, and ZeroX-Valley, *MotionScale* can effectively extract body movement signals from load cell data and detect in-bed movements with a low error rate of $6.3\%$, and classify them as big or small movements with an error rate of $4.2\%$.

- *MotionTree*: We have developed a system that classifies in-bed motions into 9-classes: turning/rolling right, turning/rolling left, right hand, right leg, left hand, left leg, legs, head, and combined motions. 24 features are extracted from each movement to be used for training and testing a special designed binary decision tree based on SVM. This system has an average classification rate of 90%.

- *More Machine Learning Techniques*: We have implemented and evaluated another two machine learning techniques in our system. They are Random Forest and XGBoost. We have evaluated these techniques using the same data and features we have in *MotionTree* and classify movements into the 9 predefined

classes. We use three topologies for Random Forest: flat, multi-level tree (*MotionTree*), and 2-level tree (RF-Small-Tree). They have the following average recognition accuracy respectively: 90.8%, 89.3%, and 90.9%. We have used only flat topology for XGBoost to have 90.5% as a classification accuracy. We have applied a combination algorithm to combine all results from the previously explained techniques to have our final results and final accuracy of 91.5%. The combination algorithm has a better accuracy than using any of the previously explained techniques separately.

- *MotionPhone*: We have developed another unobtrusive, low-overhead, and highly robust system for in-bed movement detection and classification system using a geophone sensor. It can detect different types of in-bed body movements with different scales. We have evaluated this system by doing 30 experiments with 15 participants over a two-month time period. Each experiment has 35 movements. By utilizing our two main strategies, Log-Peak and Energy-Peak, *MotionPhone* can effectively detect in-bed movements with a low error rate of 2%. We have used Random Forest technique for classification and 4 features were extracted from every detected movement. Our system can classify movements as big or small movements with an error rate of 1.5%.

## 6.2 Future Work

We have evaluated the performance of our proposed methods using controlled experiments. In the future, we would like to re-evaluate our proposed research methodology in more practical systems and applications. Our system can be tested in monitoring kids 'or toddlers' in-bed movements. It can help to know that a toddler is moving and some action is required to have a better sleep position. Our system can also be evaluated to monitor in-bed movements of the elderly.

As another future step, geophone sensor can be used in parallel with load cell sensor in one system. Since the geophone is perfect in movements detection, it can be added to load cell system to improve the detection accuracy. Moreover, the features that are extracted from geophone sensors can be added to the list of features that are extracted from load cell sensor. This can help machine learning techniques to improve the classification accuracy. In addition, these features can be used to classify in-bed movements into more than the nine classes that we have so far.

# References


[1] "Sparkfun," https://www.sparkfun.com/datasheets/BreakoutBoards/OpAmp_Breakout-v16.pdf.

[2] A. M. Adami, M. Pavel, T. L. Hayes, and C. M. Singer, "Detection of movement in bed using unobtrusive load cell sensors," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 481–490, 2010.

[3] E. Hoque, R. F. Dickerson, and J. A. Stankovic, "Monitoring body positions and movements during sleep using wisps," in *Wireless Health 2010*. ACM, 2010, pp. 44–53.

[4] A. Muzet, "Dynamics of body movements in normal sleep," in *Eighth European Congress on Sleep Research, Szeged, Hungary*, 1986, pp. 232–234.

[5] S. Chokroverty, R. P. Allen, and A. S. Walters, *Sleep and movement disorders*. Oxford University Press, 2013.

[6] J. Hilten, H. Middelkoop, E. Braat, E. Velde, G. Kerkhof, G. Ligthart, A. Wauquier, and H. Kamphuisen, "Nocturnal activity and immobility across aging (50–98 years) in healthy persons," *Journal of the American Geriatrics Society*, vol. 41, no. 8, pp. 837–841, 1993.

[7] A. M. Adami, "Assessment and classification of movements in bed using unobtrusive sensors," 2006.

[8] E. Kronholm, E. Alanen, and M. T. Hyyppä, "Sleep movements and poor sleep in patients with non-specific somatic complaintsi. no first-night effect in poor and good sleepers," *Journal of psychosomatic research*, vol. 31, no. 5, pp. 623–629, 1987.

[9] M. R. Lemke, P. Puhl, and A. Broderick, "Motor activity and perception of sleep in depressed patients," *Journal of psychiatric research*, vol. 33, no. 3, pp. 215–224, 1999.

[10] G. Aubert-Tulkens, C. Culee, K. Harmant-Van Rijckevorsel, and D. Rodenstein, "Ambulatory evaluation of sleep disturbance and therapeutic effects in sleep apnea syndrome by wrist activity monitoring," *American Review of Respiratory Disease*, vol. 136, no. 4, pp. 851–856, 1987.

[11] B. Phillips, "Movement disorders a sleep specialists perspective," *Neurology*, vol. 62, no. 5 suppl 2, pp. S9–S16, 2004.

[12] A. S. Walters, M. S. Aldrich, R. Allen, S. Ancoli-Israel, D. Buchholz, S. Chokroverty, G. Coccagna, C. Earley, B. Ehrenberg, T. Feest *et al.*, "Toward a better definition of the restless legs syndrome," *Movement Disorders*, vol. 10, no. 5, pp. 634–642, 1995.

[13] J. Wilde-Frenz and H. Schulz, "Rate and distribution of body movements during sleep in humans." *Perceptual and motor skills*, 1983.

[14] J. Alihanka, "Sleep movements and associated autonomic nervous activities in young male adults," 1982.

[15] A. Culebras, "Who should be tested in the sleep laboratory?" *Reviews in neurological diseases*, vol. 1, no. 3, pp. 124–132, 2003.

[16] J. M. Kortelainen, M. van Gils, and J. Parkka, "Multichannel bed pressure sensor for sleep monitoring," in *Computing in Cardiology (CinC), 2012*. IEEE, 2012, pp. 313–316.

[17] K. Watanabe, T. Watanabe, H. Watanabe, H. Ando, T. Ishikawa, and K. Kobayashi, "Noninvasive measurement of heartbeat, respiration, snoring and body movements of a subject in bed via a pneumatic method," *Biomedical Engineering, IEEE Transactions on*, vol. 52, no. 12, pp. 2100–2107, 2005.

[18] X. L. Aubert and A. Brauers, "Estimation of vital signs in bed from a single unobtrusive mechanical sensor: Algorithms and real-life evaluation," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 4744–4747.

[19] S. Nukaya, T. Shino, Y. Kurihara, K. Watanabe, and H. Tanaka, "Noninvasive bed sensing of human biosignals via piezoceramic devices sandwiched between the floor and bed," *Sensors Journal, IEEE*, vol. 12, no. 3, pp. 431–438, 2012.

[20] Y. Yamana, S. Tsukamoto, K. Mukai, H. Maki, H. Ogawa, and Y. Yonezawa, "A sensor for monitoring pulse rate, respiration rhythm, and body movement in bed," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 5323–5326.

[21] M. Brink, C. H. Müller, and C. Schierz, "Contact-free measurement of heart rate, respiration rate, and body movements during sleep," *Behavior research methods*, vol. 38, no. 3, pp. 511–521, 2006.

[22] T. Harada, T. Sato, and T. Mori, "Estimation of bed-ridden human's gross and slight movement based on pressure sensors distribution bed," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4. IEEE, 2002, pp. 3795–3800.

[23] M. H. Jones, R. Goubran, and F. Knoefel, "Identifying movement onset times for a bed-based pressure sensor array," in *Medical Measurement and Applications,*

*2006. MeMea 2006. IEEE International Workshop on*. IEEE, 2006, pp. 111–114.

[24] T. Tamura, S. Miyasako, M. Ogawa, T. Togawa, and T. Fujimoto, "Assessment of bed temperature monitoring for detecting body movement during sleep: comparison with simultaneous video image recording and actigraphy," *Medical engineering & physics*, vol. 21, no. 1, pp. 1–8, 1999.

[25] T. Tamura, T. Togawa, and M. Murata, "A bed temperature monitoring system for assessing body movement during sleep," *Clinical Physics and Physiological Measurement*, vol. 9, no. 2, p. 139, 1988.

[26] L. Walsh, E. Moloney, and S. McLoone, "Identification of nocturnal movements during sleep using the non-contact under mattress bed sensor," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 1660–1663.

[27] A. Adami, T. Hayes, M. Pavel, and C. Singer, "Detection and classification of movements in bed using load cells," in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*. IEEE, 2006, pp. 589–592.

[28] A. Adami, T. Hayes, and M. Pavel, "Unobtrusive monitoring of sleep patterns," in *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, vol. 2. IEEE, 2003, pp. 1360–1363.

[29] B. H. Choi, G. S. Chung, J.-S. Lee, D.-U. Jeong, and K. S. Park, "Slow-wave sleep estimation on a load-cell-installed bed: a non-constrained method," *Physiological measurement*, vol. 30, no. 11, p. 1163, 2009.

[30] W. SpillmanJr, M. Mayer, J. Bennett, J. Gong, K. Meissner, B. Davis, R. Claus, A. MuelenaerJr, and X. Xu, "A'smart'bed for non-intrusive monitoring of patient physiological factors," *Measurement Science and Technology*, vol. 15, no. 8, p. 1614, 2004.

[31] M. Nishyama, M. Miyamoto, and K. Watanabe, "Respiration and body movement analysis during sleep in bed using hetero-core fiber optic pressure sensors without constraint to human activity," *Journal of biomedical optics*, vol. 16, no. 1, pp. 017 002–017 002, 2011.

[32] T. Hao, G. Xing, and G. Zhou, "isleep: unobtrusive sleep quality monitoring using smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2013, p. 4.

[33] M. Rofouei, M. Sinclair, R. Bittner, T. Blank, N. Saw, G. DeJean, and J. Heffron, "A non-invasive wearable neck-cuff system for real-time sleep monitoring," in *Body Sensor Networks (BSN), 2011 International Conference on*. IEEE, 2011, pp. 156–161.

[34] J. Kaartinen, I. Kuhlman, and P. Peura, "Long-term monitoring of movements in bed and their relation to subjective sleep quality," *Sleep and Hypnosis*, vol. 5, pp. 145–153, 2003.

[35] T. Tamura, J. Zhou, H. Mizukami, and T. Togawa, "A system for monitoring temperature distribution in bed and its application to the assessment of body movement," *Physiological measurement*, vol. 14, no. 1, p. 33, 1993.

[36] L. Lu, T. Tamura, and T. Togawa, "Detection of body movements during sleep by monitoring of bed temperature," *Physiological measurement*, vol. 20, no. 2, p. 137, 1999.

[37] C.-M. Cheng, Y.-L. Hsu, and C.-M. Young, "Development of a portable device for telemonitoring of physical activities during sleep," *Telemedicine and e-Health*, vol. 14, no. 10, pp. 1044–1056, 2008.

[38] A. M. Adami, A. G. Adami, C. M. Singer, T. L. Hayes, and M. Pavel, "A system for unobtrusive monitoring of mobility in bed," in *Computational Science and Engineering Workshops, 2008. CSEWORKSHOPS'08. 11th IEEE International Conference on*. IEEE, 2008, pp. 13–18.

[39] A. M. Adami, M. Pavel, T. L. Hayes, A. G. Adami, and C. Singer, "A method for classification of movements in bed," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 7881–7884.

[40] A. Adami, A. Adami, T. Hayes, and Z. Beattie, "Using load cells under the bed as a non-contact method for detecting periodic leg movements," *IRBM*, vol. 35, no. 6, pp. 334–340, 2014.

[41] M. A. King, M.-O. Jaffre, E. Morrish, J. M. Shneerson, and I. E. Smith, "The validation of a new actigraphy system for the measurement of periodic leg movements in sleep," *Sleep medicine*, vol. 6, no. 6, pp. 507–513, 2005.

[42] T. Shino, S. Nukaya, Y. Kurihara, K. Watanabe, and H. Tanaka, "Development of unconstrained biosignal bed sensing method using piezoceramics to detect body movement and scratching motion," in *SICE Annual Conference (SICE), 2011 Proceedings of*. IEEE, 2011, pp. 2317–2321.

[43] Z. Ren, T. Grant, R. Goubran, M. El-Tanany, F. Knoefel, H. Sveistrup, M. Bilodeau, and J. Jutai, "Analyzing center of pressure progression during bed exits," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014, pp. 1786–1789.

[44] P. Bustamante, N. Guarretxena, G. Solas, and U. Bilbao, "In-bed patients behaviour monitoring system," in *Biocomputation, Bioinformatics, and Biomedical Technologies, 2008. BIOTECHNO'08. International Conference on*. IEEE, 2008, pp. 1–6.

[45] J. S. Aronoff, S. J. Simske, and J. Rolia, "Classification of patient movement events captured with a 6-axis inertial sensor," in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*. IEEE, 2015, pp. 784–791.

[46] B. Firner, P. Jadhav, Y. Zhang, R. Howard, W. Trappe, and E. Fenson, "Towards continuous asset tracking: Low-power communication and fail-safe presence assurance," in *Sensor, Mesh and Ad Hoc Communications and Networks, 2009. SECON'09. 6th Annual IEEE Communications Society Conference on*. IEEE, 2009, pp. 1–9.

[47] "Half-bridge-strain-gauge-circuit," http://www.circuitstoday.com/strain-gauge.

[48] "Ina126-amp," http://www.ti.com/lit/ds/symlink/ina126.pdf.

[49] B. Firner, S. Medhekar, Y. Zhang, R. Howard, W. Trappe, P. Wolniansky, and E. Fenson, "Pip tags: Hardware design and power optimization," in *Proceedings of the Fifth Workshop on Embedded Networked Sensors (HotEmNets)*, 2008.

[50] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, "Signals and systems, vol. 2," *Prentice-Hall Englewood Cliffs, NJ*, vol. 6, no. 7, p. 10, 1983.

[51] "Peak finding and measurement," http://terpconnect.umd.edu/~toh/spectrum/PeakFindingandMeasurement.htm.

[52] M. Alaziz, Z. Jia, J. Liu, R. Howard, Y. Chen, and Y. Zhang, "Motion scale: A body motion monitoring system using bed-mounted wireless load cells," in *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on*. IEEE, 2016, pp. 183–192.

[53] Z. Jia, M. Alaziz, X. Chi, R. E. Howard, Y. Zhang, P. Zhang, W. Trappe, A. Sivasubramaniam, and N. An, "Hb-phone: A bed-mounted geophone-based heartbeat monitoring system," in *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2016, pp. 1–12.

[54] I. J. Schoenberg and I. J. Schoenberg, *Cardinal spline interpolation*. SIAM, 1973, vol. 12.

[55] H. Späth, *Two dimensional spline interpolation algorithms*. AK Peters, Ltd., 1995.

[56] B. Igelnik, *Efficiency and Scalability Methods for Computational Intellect*. IGI Global, 2013.

[57] L. Wang, *Support vector machines: theory and applications*. Springer Science & Business Media, 2005, vol. 177.

[58] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[59] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[60] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[61] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, "A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data," *BMC bioinformatics*, vol. 10, no. 1, p. 1, 2009.

[62] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC bioinformatics*, vol. 7, no. 1, p. 1, 2006.

[63] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[64] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2008, pp. 313–325.

[65] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[66] G. Madzarov, D. Gjorgjevikj, and I. Chorbev, "A multi-class svm classifier utilizing binary decision tree," *Informatica*, vol. 33, no. 2, 2009.

[67] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[68] "Feature importance," http://www.mathworks.com/help/stats/ensemble-methods.html#zmw57dd0e63096.

[69] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[70] "Precision and recall," https://en.wikipedia.org/wiki/Precision_and_recall.

[71] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[72] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 785–794.

[73] "Introduction to boosted trees," http://homes.cs.washington.edu/~tqchen/pdf/BoostedTree.pdf.

[74] G. P. Succi, D. Clapp, R. Gampert, and G. Prado, "Footstep detection and tracking," in *Aerospace/Defense Sensing, Simulation, and Controls*, 2001.

[75] A. Pakhomov, A. Sicignano, M. Sandy, and E. T. Goldburt, "Single-and three-axis geophone: footstep detection with bearing estimation, localization, and tracking," in *AeroSense 2003*, 2003.

[76] "Geophone sm-24," https://www.sparkfun.com/products/11744.

[77] "Ti lmv358," http://www.ti.com/product/lmv358.