UNIVERSITY OF CALIFORNIA
SANTA CRUZ

PIMAP: A SYSTEM FRAMEWORK FOR PATIENT MONITORING

A dissertation submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE ENGINEERING

by

Sam Mansfield

March 2021

The Dissertation of Sam Mansfield is approved:

Professor Katia Obraczka, chair

Professor Shuvo Roy

Professor Peter Alvaro

Quentin Williams
Interim Vice Provost and Dean of Graduate Studies

Contents

List of Figures

Abstract

Samuel Mansfield, PIMAP: A System Framework For Patient Monitoring

We present PIMAP, a system framework for continuous patient monitoring with a specific focus on preventing pressure injuries (a.k.a. bed sores). Pressure injuries are classified as “never events”, meaning they should never occur in healthcare facilities and yet in the U.S. they affect 2.5 million patients a year at a cost of $2.5 billion. In addition the majority of patients affected are the most vulnerable, the elderly and/or physically disabled.

There are many proposed solutions to prevent pressure injuries, the most promising are patient monitoring based, such as monitoring the pressure of the patient against a mattress, monitoring the motion of the patient, and measuring the health of the patient’s skin. Patient monitoring has the advantage that data can be automatically collected and analyzed without healthcare intervention, providing additional insights that would otherwise have to be calculated manually or ignored.

Through the identification of the most promising techniques and through anecdotal evidence from our collaboration with UCSF we discovered a lack of a reliable way to sense, store, analyze, and visualize novel medical device data. There is no current system that is able to: (1) seamlessly, reliably, and persistently acquire patient monitoring data from various medical devices, (2)

analyze acquired data, and (3) present the results to healthcare personnel in an efficient and user-friendly fashion. Instead there are many one-off solutions that will only work with a specific medical device or commercial systems that only work with a commercial medical device.

From this motivation we present Pressure Injury Monitoring And Prevention (PIMAP), a system framework that presents a standard to sense, store, analyze, and visualize medical sensor data in real-time. The framework abstracts the system so that researchers can focus their efforts on specifics, such as the medical device or analysis without having to develop an entire system.

Acknowledgments

The text of this dissertation includes reprints of sections of previously published material. The majority of Chapter 2 is reprinted from S. Mansfield, K. Obraczka, and S. Roy, "Pressure Injury Prevention: A Survey", IEEE Reviews In Biomedical Engineering, pp. 352–368, 2019, doi: 10.1109/RBME.2019.2927200. The co-authors Katia Obraczka and Shuvo Roy supervised this work. Most of the material in Chapter 3 is under review for publication. The majority of Chapter 4 is reprinted from S. Mansfield, S. Rangarajan, K. Obraczka, H. Lee, D. Young, and S. Roy, "Objective Pressure Injury Risk Assessment Using A Wearable Pressure Sensor", in 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), San Diego, CA, USA, Nov. 2019, pp. 1561–1568, doi: 10.1109/BIBM47256.2019.8982939. The co-authors Katia Obraczka, Shuvo Roy, and David Young supervised this work the co-authors Sachin Rangarajan and Hanmin Lee conducted the data collection for this work. We would also like to acknowledge Eric Vin as a contributor to Chapter 4, primarily for his work creating the PIMAP Sense Sentinel component as well as his work gathering and analyzing data to validate the Sentinel Bandage.

Chapter 1

Introduction

In this work we identify a need for an open source Internet of Things based patient monitoring system to accelerate pressure injury monitoring approaches to automatically assess a patient's risk of forming a pressure injury. We propose a system framework, entitled Pressure Injury Monitoring And Prevention (PIMAP), based on this need and evaluate the system by creating new objective metrics to stratify patients in real-time using novel sensors and novel algorithms.

PIMAP is motivated by the persistence of pressure injuries in healthcare. Pressure injuries are open wounds that develop from a combination of prolonged pressure to an area of the body. Pressure injuries are classified as "never events", meaning pressure injuries should never occur in health care facilities (another example of a never event is wrong-site surgery) and yet there are over 2.5 million patients affected in the U.S. every year at a cost of $11 billion [93].

The exact cause of pressure injuries is not known, but it is understood that through a combination of prolonged closure of capillaries and lymphatic vessels, ischemia, reperfusion, and tissue deformation the affected tissue dies [9] [1] [5]. The healthy patient will not form a pressure injury, but instead it is patients with low-mobility such as the elderly and disabled that are most at risk [34]. In addition pressure injuries impact a patient's quality of life as they cause severe pain and impact the social life of the patient [37].

The standard of care to prevent pressure injuries is to assess the risk of the patient using the Braden Scale [13], a questionnaire filled out by the healthcare facility and from this assessment a periodic turning schedule is created. But it is accepted that the standard of care is not enough to eliminate pressure injuries as the Braden Scale has been shown to not be effective at predicting pressure injuries [134], turning a patient to effectively redistribute the load is not trivial [39] [99], healthcare staff are not able to keep to a periodic turning schedule [102], and the patient's bone structure, which is not visible to the naked eye, can make a patient more at risk [67]. In addition because pressure injuries start their formation from the bone, by the time one can visually see the injury it is past the point of prevention, meaning assessing pressure injury risk is not trivial.

Some of the current solutions to prevent pressure injuries are support surfaces, beds that can redistribute pressure, and nursing guidelines, published by medical professionals that describe day to day practices that reduce pressure injury incidence.

Support surfaces are well studied and to date no surface exists that can eliminate pressure injuries. The most extensive literature survey examined 59 randomized controlled trials and found that support surfaces are better than a "standard" hospital mattress [79], but this same study also found a sheepskin overlay to be better than a "standard" hospital mattress. In addition an expert consensus with 100% agreement concluded that support surfaces cannot replace repositioning [12].

Nursing guidelines are published by the National Pressure Ulcer Advisory Panel (NPUAP) [137] and studies have shown that increasing the amount of nursing care does reduce pressure injury incidence [87] [125]. But, scaling nursing care is most likely not sustainable as it is predicted that there will be a nursing shortage in the U.S. based on the number of nursing graduates and the aging baby boomer generation [128].

Another promising approach that we identified in the literature [75] is patient monitoring that uses sensors, such as pressure sensors or accelerometers, to continuously monitor the patient. Patient monitoring is promising as risk can be assessed objectively without supervision from healthcare staff. In addition a patient can be assessed based on their own features, for example with a full bed pressure system we can assess the amount of pressure at individual locations with respect to the individual, e.g. the sacrum has an abnormally high amount of pressure and we can compare this to other patients.

We found a lack of any patient monitoring system to sense, analyze, store, and visualize medical sensor data [75]. Patient monitoring systems presented in the literature use commercial systems that are limited to what the commercial entity offers or are one-off solutions that are not discussed in detail and are not released for use by other researchers. Ideally patient monitoring systems would be integrated into the Electronic Medical Record (EMR), but the most common EMRs are commercial entities, e.g. Epic, and do not allow new approaches to be integrated unless rigorously clinically tested and even then it is up to the discretion of the EMR.

This dilemma leaves medical researchers in a difficult position to evaluate novel sensors or novel algorithms as the novel approach needs to be rigorously clinically tested to be integrated into the EMR if it is to be used. Using commercial software to test is expensive and limited to what the company offers, which is usually a specific set of sensors or a specific set of analysis and to create a patient monitoring system from scratch is tedious and prone to failure without extensive testing.

We designed, tested, and evaluated PIMAP based on this critical need. PIMAP is designed for medical researchers and clinicians. Medical researchers can leverage this system by focusing their efforts on the medical device or on the algorithm without having to design the monitoring system in between. Clinicians can use PIMAP to automatically visualize data to make decisions in real-time.

In brief the PIMAP system consists of four swappable components: Sense, Store, Analyze, and Visualize. Sense is responsible for data entering the system, such as sensing UDP or BLE packets. Store is responsible for storing and receiving data. Analyze is responsible for converting sensor data into insightful metrics. Visualize is responsible for visualizing the data for clinicians. Each component is not tied to a particular service, for example we leverage Kafka as the Store component, but we can incorporate other services such as InfluxDB or Prometheus. PIMAP is written in Python as it supports a large variety of scientific libraries such as Numpy or Tensorflow in addition to being a simple and elegant high level language. In addition PIMAP can be deployed in a distributed manner to accommodate studies across multiple sites or to scale resources in the cloud.

This dissertation is organized into six chapters. In Chapter 2 we present a literature survey on pressure injury prevention strategies ranging from published clinician guidelines to using sensors to assess a patient's pressure injury risk. In addition we identify a gap in research around a reusable patient-centric monitoring system instead of one-off systems that are never used again or commercial systems that lock you into a product line. In Chapter 3 we present the design, components, and workflows of PIMAP and study the system performance in various distributed configurations in addition to proof of concept ways to use PIMAP such as integrating PIMAP with a sensor network simulator. In Chapter 4 we discuss two of our collaborations with UCSF to integrate and analyze the data from two different novel sensor-based bandages. One of which is a pressure bandage designed for monitoring pressure injury formation. From this pressure bandage data we created an Objective Mobility metric that assesses a patient's movement in bed, a key factor in assessing a patient's risk of forming a pressure injury. We also demonstrate PIMAP running in real-time using our Objective Mobility metric and discuss how this could influence care. In this same chapter we present another collaboration with UCSF using a skin impedance sensor, which was demonstrated in the literature to be able to predict pressure injury formation. We demonstrate and evaluate how we integrated this sensor into PIMAP and discuss how we used PIMAP to validate whether the skin impedance sensor was functioning properly. In Chapter 5 we discuss privacy and security considerations in regards to patient monitoring and in addition discuss PIMAP addresses these concerns. In Chapter 6 we discuss future work with PIMAP including a recent collaboration with UC Davis to present a new metric based on the blood pressure of the patient which was shown to correlate to pressure injury risk and in addition we conclude this work.

Chapter 2

Pressure Injury Prevention: A Survey

Pressure injuries, recently standardized from the term pressure ulcers or decubitus ulcers by the National Pressure Ulcer Advisory Panel [85], are classified colloquially as a “never event”, meaning they should never occur in health care settings and yet in the U.S. there are over 2.5 million patients affected every year at a cost of $11 billion [93]. More than 90% of pressure injuries are a secondary condition, meaning the patient was being treated for a different condition when the pressure injury formed [118].

Biomechanically pressure injuries are caused by prolonged pressure to an area of the body. Through a combination of prolonged closure of capillaries and lymphatic vessels, ischemia, reperfusion, and tissue deformation the affected tissue dies [9] [1] [5]. Typically this occurs at the bony prominences, such as the sacrum or heels in a patient lying down. The result is an open wound that descends to the bone, which must be further treated to avoid infection.

We use the term pressure throughout this paper to refer to pressure applied to the body at any angle in order to account for both pressure and shear force. We do so as the majority of the work surveyed does not make a clear differentiation of the effect of pressure versus shear. We find that this simplifies the discussion while still accurately presenting the material.

Pressure injuries have an impact on quality of life as they cause severe pain, treatments increase discomfort and pain, and impact the social life of the patient [48] [37]. Pressure injuries are generally developed while being treated for a different condition, but the resulting pressure injury can affect treatment options [37].

Pressure injuries do not form in the healthy patient, but are common in patients with low mobility as these patients cannot reposition themselves. Garcia-Fernandez et al. [34] identified 83 risk factors used in various pressure injury scales. Of these 83 risk factors an expert panel determined 23 risk dimensions, meaning multiple risk factors were interrelated and could be grouped together. Of the 23 risk dimensions the five that were considered critical in order of importance are mobility, exposure to moisture/incontinence, mental state/level of consciousness, nutrition/diet, and activity. In other words, patients who are at risk for pressure injuries are already suffering from previous conditions and in addition have to cope with a pressure injury.

In an effort to reduce pressure injury occurrence in the U.S. Medicaid and Medicare Services decided to no longer reimburse "never events" [27]. Through improved and more focused nursing care guidelines the prevalence, a benchmarking metric of hospital-acquired pressure injuries in the U.S. were reduced from 6.2% in 2006 to 3.1% in 2015 [138]. The end goal is to have a prevalence of 0% or very close to it, although as we will discuss in Section 2.2 this is somewhat debated. Also important to note is that although there was a 1% drop in prevalence from 2008 to 2009 when reimbursements stopped the prevalence in 2013, 2014, and 2015 respectively was 3.2%, 3.4%, and 3.1% [138], which may indicate that nursing guidelines and established nursing interventions alone may not be enough to bring down pressure injury below a 3% prevalence.

This work is a literature survey of work on preventing pressure injuries from 2010 to present. Although not the main focus we present some of the fundamental problems in the pressure injury prevention field and some of the landmark pressure injury studies derived from the literature. We do so in order to give context to the work surveyed, but also to make this work a standalone snapshot of how and why the research to prevent pressure injuries has led to the current prevention strategies. Our expertise is in the space of Computer Engineering and Bioengineering and therefore we offer a unique perspective on current and future technological solutions.

There are three recent literature surveys that we are aware of that have some overlap with this work [77] [134] [5]. In 2015 a literature survey on software solutions to prevent pressure injuries was presented [77]. This work identifies approaches that monitor sensor information that can be used to prevent pressure injuries. We also cover pressure injury prevention strategies that monitor sensor information, but in addition we cover new literature up until July of 2018, which includes several randomized controlled trials conducted after the previous work was published, we cover new sensor monitoring strategies not covered in the previous work, we cover pressure injury prevention strategies that are not sensor-based, and we created a taxonomy of all strategies that provides insight into which strategies are the most promising currently and for the future.

In [134] papers were reviewed from 2013-2016 with a focus on different types of skin ulcers and the effectiveness of current technologies that are used in healthcare to prevent pressure injuries. Our work also covers technologies that are currently used in healthcare, although we do not cover different types of skin ulcers, we focus only on pressure injuries. But, in addition we examine technologies that are not currently being used in healthcare and we classify all pressure injury prevention strategies using a taxonomy we created to provide insight into which strategies are the most promising currently and for the future.

The most recent literature survey that has some overlap with our own was published in 2018 [5]. This work explores current technologies that can assess the skin integrity of a patient with a focus on the prevention of diabetic foot ulcers. We also cover technologies that assess the skin integrity of a patient, although there are some technologies, such as Magnetic Resonance Imaging (MRI) and Finite Element Modeling, that we do not cover. But, in addition we cover technologies that can prevent pressure injuries without assessing the skin integrity of a patient that are effective at reducing pressure injuries, such as Inertial Measurement Unit Monitoring and Electrical Stimulation. In addition we also created a taxonomy of all strategies to provide insight into the most promising currently researched strategies and the most promising for the future.

Our work starts with a brief history of modern pressure injury prevention in Section 2.1 to give context to the current practices and research of today. This leads into a discussion on how pressure injuries are currently classified and how that classification relates to “never events” as well as the current research on whether all pressure injuries are preventable in Section 2.2. We then discuss the current research on the biomechanics or pathophysiology of pressure injuries in Section 2.3 to give the reader an understanding of how and why pressure injuries form. We then introduce our taxonomy on pressure injury prevention strategies in Section 2.4 that classifies each technique that we cover based on the commercial availability, clinical results of prevention, time savings, and ability to tune the technique to an individual. In Section 2.5 we cover Active Prevention Strategies, a technique that requires active work from healthcare staff to prevent pressure injuries, such as nursing guidelines and nutrition. We will discuss why we consider each technique an Active Prevention Strategy in each subsection. In Section 2.6 we cover Sensor-Based Risk-Factor Monitoring Strategies, a technique that gathers data about a patient and can present it to healthcare staff when needed, to prevent pressure injuries, such as Pressure and Temperature Monitoring. In Section 2.7 we discuss the most promising current prevention strategies and what we see as the most promising future work based on the work presented. We present our conclusion in Section 2.8.

2.1 Brief History

The first recorded instances of pressure injuries date back thousands of years to ancient Egypt [1]. In the early 19th century Jean-Martin Charcot studied pressure injuries, but attributed their formation to an impaired nervous system [65] [1]. In the early 20th century Dr. William Browning established a pressure injury prevention plan, which resembles the treatment plans of today [65].

In the literature of today the two hour turning of high risk pressure injury patients is commonly mentioned [125] [100] [19] [135] [36]. It is believed that the standard two hour turning cycle used today was established during World War II as this is the time it takes on average to turn 32 patients in a nursing unit for war victims [22].

Around this same time Groth [41] performed animal studies that showed increasing the amount of pressure and the time of pressure increased the damage to the muscle fibers and capillaries [35]. Husain in the 1950s [53] continued this work and established that $100 mmHg$ applied for two hours to the legs of rats and guinea pigs caused permanent damage to the skeletal muscle. Kosiak in the 1960s [61] [60] established that it is more complicated than a simple threshold and instead it is a pressure-time threshold, e.g., a high pressure of $190 mmHg$ for a very short time period will not form a pressure injury, but a low pressure of $70 mmHg$ for a long period of time will form a pressure injury. The first human study was conducted by Reswick and Rogers [116] in the 1970s and they established a $300 mmHg \times hour$ threshold.

The interface pressure, the pressure of the patient against a surface, such as a mattress or chair, became the first well known way to monitor pressure injuries using sensors. But, given these first studies were conducted in the 1940s to 1970s the technology of the time was only able

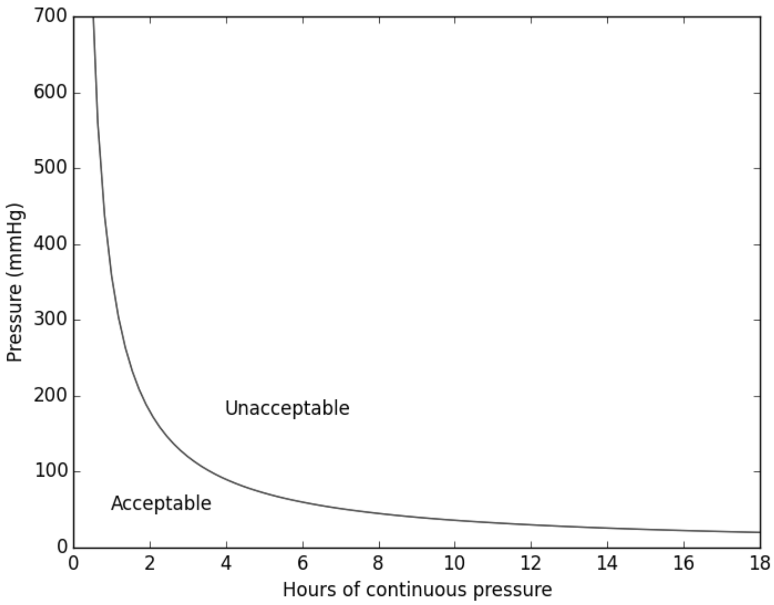


Figure 2.1: Reswick and Rogers Pressure-Time Curve

to use pressure sensors to generate a generic pressure-time curve that could be applied to any patient.

The most established pressure-time curve based on human studies is the Reswick and Rogers Pressure-Time Curve [116], as depicted in Figure 2.1. Reswick and Rogers conducted a study of the interface pressure at the bony prominences in the 1970s using a singular pressure device to study a wide range of patients. The pressure-time curve they found creates a threshold of $300mmHg \times hour$, e.g. if a patient is immobile for one hour the continuous pressure should be less than $300mmHg$, if a patient is immobile for two hours the continuous pressure should be less than $150mmHg$. Reswick and Rogers created the pressure-time curve as a guideline and it was not meant to be used quantitatively.

At the extremes of the time scale the Reswick and Rogers Pressure-Time Curve has received criticism as it "allows" for pressure high enough to rupture organs for very short periods of time and predicts pressure injuries to form during twelve-hour long operations that do not occur [35]. In addition as we will describe in Section 2.3 pressure injuries start their formation in deep tissue and interface pressure is the pressure between the body and mattress or chair. Studies have shown that the deep tissue pressure on the bony prominence cannot always be reduced significantly with cushioning that reduces interface pressure [9].

Another problem with interface pressure is that the same amount of interface pressure does not correspond to the same amount of deep tissue pressure [35]. A study on the seated patient studied six subjects using Magnetic Resonance Images and Finite Element Analysis to determine the amount of strain on the tissue under the ischial turbosities (sit bones) and it was confirmed that the tissue closest to the ischial turbosities had the highest strain, but the amount of strain varied by patient based on the shape of the ischial turbosities and the amount of muscle and fat [67], i.e. a seated patient with a lower interface pressure may be more at risk than a patient with a higher interface pressure.

To date it is established that a pressure-time curve cannot be used to prevent pressure injuries for every patient. To combat this many approaches have been tested and evaluated, some of which take an active role from health care staff such as nursing guidelines or Support Surfaces, as will be discussed in Section 2.5. Other approaches are based on sensor monitoring such as continuous Pressure Monitoring or even measuring the physical WiFi channel as will be discussed in Section 2.6. The various approaches are based on the current understanding of the biomechanics of pressure injury formation, which we will discuss in Section 2.3, but first we will introduce the reader to the classification of pressure injuries and how they relate to the term "never event" as well as the current research on whether all pressure injuries are preventable.

2.2 Never Events And Unavoidable Pressure Injuries

Although colloquially known as "never events", the term used by the National Quality Forum (NQF) in their reports are "serious reportable events" [93]. For consistency we will mention this here, but will continue to use the term "never event".

Never events range from operating on the wrong patient or a serious injury from a patient disappearance [93]. It is often cited that pressure injuries are never events, and in fact we do so in our Introduction, but it is actually only Stage 3, Stage 4, and Unstageable pressure injuries that occur after admission to a healthcare setting that are considered never events.

There are various staging classifications of pressure injuries, one of the more popular in the United States is created by the NPUAP. As a reference the 2016 pressure injury staging from the NPUAP is summarized [85]:

Stage 1 Non-blanchable erythema of intact skin

Stage 2 Partial-thickness skin loss with exposed dermis

Stage 3 Full-thickness skin loss

Stage 4 Full-thickness skin and tissue loss

Unstageable Obscured full-thickness skin and tissue loss

Deep Tissue Persistent non-blanchable deep red, maroon, or purple discoloration

The staging, as stated by the NPUAP, is not meant to be used as a progression, but instead as different types of pressure injuries that can occur. In particular Stage 1 injures are somewhat controversial and are addressed by Berlowitz and Brienza [9] as they note that Stage 1 pressure injuries can occur because of incontinence and do not have any deep tissue injury component.

The NQF mentions in their report [93] that Deep Tissue staged pressure injuries were considered as being never events, but this "would amount to reporting an unconfirmed suspicion."

Some pressure injuries are unavoidable as based on a consensus study by Edsberg et al. [26]. An unavoidable pressure injury is defined as a pressure injury that forms when all preventative measures were correctly assessed and implemented [26]. Although not definitive the study points out that in some cases preventative measures cannot be implemented because the patient is at critically high risk or the prevention would interfere with other conditions of the patient. This is important to consider as this indicates that there may be some percentage of pressure injuries that can never be prevented, but this percentage is not yet determined.

2.3 Biomechanics

As their name implies pressure is the main cause of pressure injuries. From the literature there are five supported reasons why pressure cause their namesake injury. In no particular order they are [9] [1] [5]:

1. Closure of capillaries causing ischemia to the surrounding tissue.

2. Under high pressure, the closure of large vessels causing thrombosis.

3. The accumulation of substances produced by inflammation in response to blood being reintroduced into an ischemic region, known as a reperfusion injury.

4. An accumulation of metabolic waste products from an impaired lymphatic system caused by pressure closing the lymphatic vessels.

5. The pressure deformation of tissue cells.

As continual pressure is applied to the body, almost exclusively from a bed or chair, a combination of the above occur. Internally the pressure has the greatest effect at the bony prominences. This effect was studied analytically and in vivo to reveal that the greatest stress was in the muscle layer next to the bone [9]. This type of injury is called a deep tissue injury.

If we go back and look at the Stage 1 classification of pressure injuries, it is only the skin that is visibly diagnosed. Stage 1 injuries can be a result of deep tissue and studies have shown this, but also can be a Superficial Injury, which is not a result from pressure [9]. Superficial Injuries can be caused by urinary and fecal incontinence, the friction of dragging a patient to be turned, or shear forces tearing blood vessels and will typically occur at the bony prominences [9]. Although Superficial Injuries occur at similar locations and in similar patients these injuries are not a result of pressure and it is argued that they should not be considered a pressure injury as they do not result from pressure [9] [34]. This is an important note as studies will frequently use Stage 1 to indicate the presence of a pressure injury, but this has to be taken with a grain of salt unless otherwise noted, it was most likely not verified to be from deep tissue damage.

Another factor that is often recognized is that an increase in skin temperature correlates to the formation of pressure injuries, but it is believed that this may be from the effects of temperature on ischemia [9]. As temperature rises the metabolic rate increases, which increases the demand of oxygen. In an ischemic region, such as a pressure injury, this increased demand of oxygen will accelerate the damage to the ischemic region. But, to confuse the issue an animal study found that deep tissue injuries happened more frequently at lower temperatures [9], not higher temperatures. The increase or decrease in temperature may be an indicator of pressure injuries, but it is unclear at this time how to use such data.

2.4 Taxonomy Of Pressure Injury Prevention

The current work on preventing pressure injuries fit into one of two categories: Active Prevention Strategies or Sensor-Based Risk-Factor Monitoring. Active Prevention Strategies are approaches that take an active role from a healthcare staff to implement, such as nursing guidelines or nutrition. Sensor-Based Risk-Factor Monitoring are strategies that have the potential to operate without any intervention of healthcare staff, such as Pressure and Temperature Monitoring.

To evaluate the effectiveness/applicability of these techniques in the prevention of pressure injuries we created a rubric as follows:

Commercial Availability Is the application commercially available or is it a prototype/idea?

Clinical Trials Are there clinical trials? And if so do they support the effectiveness?

Time Savings Does the application save time for the healthcare staff?

Tuned To Individual Is the application general or based on the individual patient? For example repositioning a patient every two hours is a general guideline and is not tuned to the individual, whereas Pressure Monitoring is measuring the actual pressure from the patient and decisions can be made based on the individual.

Each category is scored as ↑ , . . . , ↓ , or NA . A ↑ indicates that the rubric category is satisfied, e.g., the application is commercially available. A . . . indicates that it is either mixed or cannot be determined, e.g., a prototype was made. A ↓ indicates that the reverse is shown instead, e.g., the application is not available and is just an idea. NA means it is not applicable, this is specifically for the clinical trials category, if a pressure injury prevention strategy is not clinically tested it will be marked NA . A ↑ in all categories indicates a desirable quality. Each application will be discussed in more detail in the following sections, Table 2.1 is provided as an overview of the gaps in the field and what is to be discussed.

2.5 Active Prevention Strategies

We refer to the following prevention strategies as Active Prevention Strategies as they take an active role of a healthcare staff. We present nursing guidelines in Section 2.5.1 as a reference for all other techniques as nursing guidelines are part of current care and will be needed regardless

Table 2.1: Taxonomy of pressure injury prevention strategies

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Active Prevention Strategies				
Nursing Guidelines	↑	↑	↓	. . .
Support Surfaces	↑	. . .	. . .	. . .
Nutrition	. . .	. . .	↓	. . .
Electrical Stimulation	↑	↑	↓	↓
Sensor-Based Risk-Factor Monitoring				
Pressure	↑	. . .	↑	↑
Temperature and Humidity	. . .	NA	↑	↑
Inertial Measurement Unit	↑	↑	↑	↑
Blood Flow	. . .	NA	↑	↑
Biomarker	↓	NA	↑	↑
Skin Integrity	. . .	NA	↑	↑
Electrocardiography	. . .	NA	↑	↑
Camera	. . .	NA	↑	↑
Ultrasound	. . .	NA	↑	↑
Impulse Radio Ultra Wide Band	. . .	NA	↑	↑
Leaking Coaxial	. . .	NA	↑	↑

of any other additional techniques. Nursing guidelines are an Active Prevention Strategy almost by definition as they are guidelines that the healthcare staff must actively follow. In Section 2.5.2 we will discuss Support Surfaces, mattresses or overlays that actively or passively reduce the interface pressure between the patient and surface. Although the original intent of Support Surfaces may have been to be a set it and forget it technique it is generally accepted now that this is not the case, which we will discuss. We classify Support Surfaces as an Active Prevention Strategy as the general use case is a supplemental tool for healthcare staff. There is no feedback given by the surface and therefore the healthcare staff must rely on their own knowledge and experience on using the surface appropriately. In Section 2.5.3 we will discuss the latest research on nutrition as a way to prevent and increase healing of pressure injuries. Nutrition is an Active Prevention Strategy as the nutrients for the patient must be managed by a healthcare staff. In Section 2.5.4 we will discuss the latest research on Electrical Stimulation, a technique of contracting muscles using electric current, which we categorize as an Active Prevention Strategy as a healthcare staff must actively apply electrodes and verify that muscles are being contracted on every application. In each subsection we will discuss the latest research as well as the taxonomy criteria as it applies to the respective technique: commercial availability, clinical trials, time savings, and whether the technique is tuned to the individual.

2.5.1 Nursing Guidelines/Interventions

Nursing guidelines on preventing pressure injuries are published by the National Pressure Ulcer Advisory Panel (NPUAP) and the European Pressure Ulcer Advisory Panel(EPUAP) [137]. The “Quick Reference Guide” cited here is a 75 page document with extensive information that includes recommendations for care with the level of clinical evidence that supports each recommendation. A selection from the guidelines include risk factors, risk factor assessment, preventative skin care, emerging therapies, nutrition, repositioning, Support Surfaces, medical device related pressure injuries, wound cleaning, pain assessment, wound dressings, special populations, and implementing guidelines. The former is only a selection of the guidelines and each section is covered in detail.

Nursing guidelines/interventions are extensive and because it is not the main focus of our survey we have chosen to discuss a subset of proposed approaches in the literature namely: reposition frequency, risk assessment scales, and how following interventions are correlated with pressure injury incidence reduction/prevention.

It is often noted that two hours is the standard of care repositioning frequency [125] [100] [19] [135] [36], but it is interesting to note that the guideline cited earlier does not advocate this frequency and instead recommends determining a turning schedule based on the individual, making sure to take into account the patient’s comfort. As noted in Section 2.1 it is believed the two hour repositioning frequency comes from World War II clinics as that was the time it would take to turn every patient [22].

Several studies suggest additional nursing care to prevent pressure injuries, such as employing a full-time wound nurse [87] or using a reminder system [125]. Although these studies do present improvement in care the reliance on increasing the demands of the nursing staff may not be scalable in the U.S. as it is expected because of the aging baby boomer generation and the lack of nursing graduates in the U.S. there will be a nursing shortage [128]. In addition, two studies have found that repositioning does not always effectively redistribute load [39] [99], meaning even though healthcare staff may be following all guidelines to rotate patients on a schedule, because there is currently no standard way to objectively measure if a patient is correctly turned, pressure injuries will still form.

Several risk assessment scales have been developed over the years such as the Braden [14], Norton [91], Waterlow [139], and Cubbin and Jackson [72]. All scales are different, but they rely on a series of measurements recorded by a healthcare personnel, such as activity, age, nutrition, and incontinence. From these assessments a score is given and based on the score a patient is assigned a risk designation. Of these scales the Braden scale is the most studied [134], but none of the scales have been shown to be highly effective at predicting pressure injuries [134].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Active Prevention Strategies				
Nursing Guidelines	↑	↑	↓	. . .

Nursing guidelines are published by multiple organizations and therefore are **commercially available**. Evidence based guidelines are published by the NPUAP and recommendations such as repositioning the patient are based on **clinical trials** showing the effectiveness [137], so clinical trials support nursing guidelines. The downside of nursing guidelines is that they are **time** intensive for healthcare staff by necessity and therefore do not save time. Guidelines do request that appropriate changes are made by individual, this relies heavily on the staff expertise, but they can be **tuned to an individual**.

2.5.2 Support Surfaces

In order to stay consistent with the literature we will use the definitions by the NPUAP as part of their Support Surface Standards Initiative [92]. From these definitions a Support Surface is “a specialized device for pressure redistribution designed for management of tissue loads, micro-climate, and/or other therapeutic functions (i.e. any mattresses, integrated bed system, mattress replacement, overlay, or seat cushion, or seat cushion overlay).”

The categories of Support Surfaces are defined by the Support Surface Standards Initiative and are reproduced as follows:

Air Fluidized “A feature of a support surface that provides pressure redistribution via a fluid-like medium created by forcing air through beads as characterized by immersion and envelopment.”

Alternating Pressure “A feature of a support surface that provides pressure redistribution via cyclic changes in loading and unloading as characterized by frequency, duration, amplitude, and rate of change parameters.”

Lateral Rotation “A feature of a support surface that provides rotation about a longitudinal axis as characterized by degree of patient turn, duration, and frequency.”

Low Air Loss “A feature of a support surface that provides a flow of air to assist in managing the heat and humidity (microclimate) of the skin.”

Reactive Support Surface “A powered or non-powered support surface with the capability to change its load distribution properties only in response to applied load.” This category would include Air Fluidized Mattresses/Overlays.

Active Support Surface “A powered support surface with the capability to change its load distribution properties, with or without applied load.” This category would include Alternating Pressure and Lateral Rotation Mattresses/Overlays.

An extensive literature survey on Support Surfaces [79] that examines 59 Randomized Controlled Trials (RCTs) found that the effect of advanced Support Surfaces such as Air Fluidized, Alternating Pressure, Lateral Rotation, Low Air Loss, and Active Support Surfaces have on preventing pressure injuries is minimal. Several studies found these advanced types of mattresses to be better than “standard” mattresses, but “standard” is not well defined. In addition higher-specification foam as well as medical grade sheepskin were found to reduce pressure injuries better than standard mattresses at much lower cost than an advanced Support Surface. The authors conclude that more RCTs should be conducted on Alternating Pressure Mattresses in combination with other technologies such as Low Air Loss, the comfort of the patient should be considered in studies, and the cost effectiveness of the solution should also be considered.

Support Surface studies have been able to show that Active Support Surfaces can lower the peak pressure [81]. But the expert consensus with 100% agreement is that Support Surfaces cannot replace repositioning [12]. In addition evidence based guidelines set forth by the NPUAP also specify when using Support Surfaces patients should still be repositioned [137], although the frequency of repositioning can be adjusted.

The appeal of Support Surfaces is that a patient could be placed on such a surface and pressure injuries would not form, but there is no evidence that this is the case. Instead Support Surfaces are additional tools healthcare staff can use to help prevent pressure injuries, but still requiring active work from the staff.

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Active Prevention Strategies				
Support Surfaces	↑	...	...	...

Support Surfaces are **commercially available** from multiple companies, such as Hill-Rom, with various features. **Clinical trials** are somewhat mixed as they have shown that Support Surfaces are better than hospital mattresses, but repositioning is still required and in the case of Lateral Rotation or Alternating Pressure mattresses or overlays one can imagine they may not be the most comfortable mattress or cost effective. Support Surfaces may be able to save **time** for the clinician, by increasing the time between repositioning, but the amount of time is not known and therefore it is hard to say how much time is really saved. Some types of Support Surfaces are **tuned to the individual** such as Active and Reactive Support Surfaces as they adjust based on the patient, but other types are not.

2.5.3 Nutrition

Malnutrition is associated with the formation of pressure injuries [134] and it is also one of the factors on the Braden Scale [14]. The NPUAP Guide recommends screening patients at risk of pressure injuries to determine if they are malnourished and assessing weight loss, ability to eat independently, and whether the patient is getting appropriate nutrients [137].

Although nutrition is regarded as important to prevent pressure injuries a survey on nutrition in 2014 reviewed Randomized Controlled Trials that evaluate whether nutrition had any effect on pressure injury formation or healing and found no evidence to support nutrition as an effective way to reduce pressure injuries [63].

Another study aimed to address evidence-based nutritional needs of populations at risk for pressure injuries and concluded that nutrition and hydration are important, but future studies are needed to determine what specific supplements are needed [105]. In addition another study concludes that additional energy, protein, zinc, and Vitamins A, C, and E, amino acids arginine and glutamine have been documented to promote wound healing, although the ideal amount of each is not known [119].

There is one large RCT of 200 patients that showed that additional supplementation of arginine, zinc, and antioxidants to a diet that is already high in calories and protein provided improved pressure injury healing [17]. Although this work is significant researchers agree that an additional study is needed to confirm the results [88] [105].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual

Active Prevention Strategies

Nutrition	...	...	↓	...

Nutrition, i.e. food or supplements, are available **commercially**, but a protocol on the amount of nutrition needed is not known. **Clinical trials** are mostly favorable, but there is still some debate on what exact supplements are needed. To make sure that patients are eating as they should be and eating the required nutrients requires more work and **time** for healthcare staff. Nutrition can be **tuned to the individual**, but again this does take time for the staff to assess and administer and relies on the expertise of the healthcare staff.

2.5.4 Electrical Stimulation

Electrical Stimulation (ES) is a technique to contract the muscles, to simulate the natural movements that are made by a healthy individual when sitting or lying down. This method requires electrodes attached to the skin. A current is then passed through, which in turn contracts the muscles. The contraction of the gluteus muscles relieves pressure around the ischial turbosities and produces elevation in tissue oxygenation [129]. ES has also been tested and found to be a safe method of treatment [2].

The frequency of treatment to prevent pressure injuries is currently not known, but a RCT was conducted using two different types of Electrical Stimulation and found increased blood flow and wound area reduction when compared to the control group [101]. Other work has found that

Electrical Stimulation has a positive effect on the healing of pressure injuries for patients with spinal-cord injuries [28]. In addition to the frequency of treatment the method of application is also not established, for instance the electrode configuration and waveforms applied differ in various studies, but ES has been shown with moderate evidence at its effectiveness at pressure injury prevention [57].

A wirelessly controlled ES device using a smartphone accompanied by a cloud based application to track history and provide analysis of the therapy was developed in [90] as a potential way to make ES easier to apply.

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Active Prevention Strategies				
Electrical Stimulation	↑	↑	↓	↓

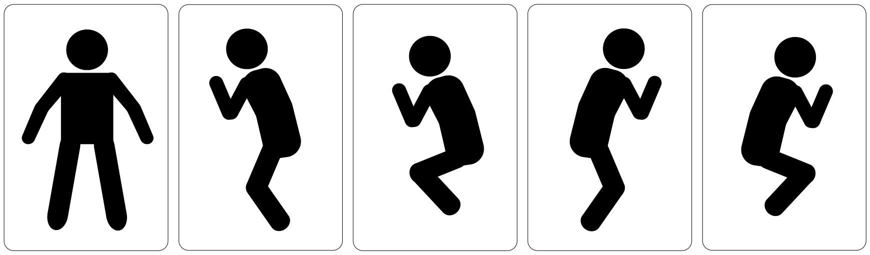
Electrical stimulation devices are available **commercially**. **Clinical trials** have shown that Electrical Stimulation is an effective treatment of pressure injuries. Electrical Stimulation is somewhat **time** intensive to apply as the electrodes must be placed properly and healthcare staff must also verify that the therapy is functioning properly. ES is not **tuned to the individual** as it uses the same protocol for every patient, this may change in the future, but as of now current research therapies are not based on feedback of the individual besides verifying that the muscles are contracting.

2.6 Sensor-Based Risk-Factor Monitoring

In the following sections we will discuss pressure injury prevention strategies that have the potential to monitor a patient without intervention of healthcare staff. These strategies comprise of a sensor component and a software component that monitors the sensor information and displays the data in a way that is beneficial to the healthcare provider. For example Pressure Monitoring, which will be discussed in Section 2.6.1 is a popular Sensor-Based Risk-Factor Monitoring technique as commercial pressure overlays can be placed on top of a mattress and can automatically monitor and display the interface pressure of a patient against a surface without any active part of the healthcare staff. The displayed pressure map would otherwise be unknown and allows staff to make a more informed decision when repositioning a patient.

When considering devices that will be in contact with a patient's skin, which is common in this section, it is important to note that the device itself can cause a pressure injury. This phenomenon is highlighted at the end of Section 2.6.

In Section 2.6.2 we will discuss Temperature and Humidity Monitoring, which we classify as a Sensor-Based Risk-Factor Monitoring technique as temperature and humidity can be sensed and monitored similarly to pressure by providing a map of the temperature and humidity of the body against a surface. In Section 2.6.3 we will discuss approaches to prevent pressure injuries using Inertial Measurement Units (IMUs), which measure orientation and acceleration, to measure the amount of movement a patient makes in bed or when seated. We classify this as a Sensor-Based Risk-Factor Monitoring technique as IMU data can be sampled and the movements of the patient, including the time of last repositioning can be relayed without any active role of healthcare staff. In Section 2.6.4 we will discuss methods to monitor blood flow. As of this writing Blood Flow Monitoring is limited to a certain area of the body and is still mostly a manual process, but we classify this technique as Sensor-Based Risk-Factor Monitoring as it has the potential to be an automatic process. In Section 2.6.5 we will discuss biomarkers that can be tracked to predict pressure injuries. Biomarker tracking is currently a manual process, but it has the potential to be automated in the future, so we classify this technique as Sensor-Based Risk-Factor Monitoring. In Section 2.6.6 we will discuss Skin Integrity Monitoring that can monitor the skin integrity at a certain location of a patient without any active role of healthcare staff, meaning the system itself will relay to the healthcare staff whether an area is at risk. In Section 2.6.7 we will discuss techniques to monitor Electrocardiography (ECG) of a patient automatically and use this to detect the movements of a patient. In Section 2.6.8 we will discuss using a camera to automatically assess the movements of a patient. In Section 2.6.9 we will discuss using ultrasound as a way to assess an area for pressure injuries. Ultrasound as of this writing still relies on the healthcare staff, but with added software it could potentially automatically assess whether an area is at risk of a pressure injury. In Section 2.6.10 we will discuss Impulse Radio Ultra Wide Band (IR-UWB) as a technique to automatically assess the movement of a patient. In Section 2.6.11 we discuss Leaking Coaxial Cable, a technique that monitors the physical WiFi channel, to automatically detect the movement of a patient. In each



(a) Supine (b) Supine (c) Supine (d) Supine (e) Supine

Figure 2.2: Five Common Postures

subsection we will discuss the latest research as well as the taxonomy criteria as it applies to the respective technique: commercial availability, clinical trials, time savings, and whether the technique is tuned to the individual. In regards to the taxonomy each category of the taxonomy is evaluated on a complete system that both senses and presents information to healthcare staff, not just based on the senors themselves. For instance Electrocardiography machines are commercially available, but Electrocardiography Monitoring systems are not.

A common technique in this section is Posture Detection. In this work we use the term posture to describe the position of a patient on a mattress, e.g. left side, right side, or supine. The number of postures vary by paper, for instance some identify three postures [51] and others identify eight [107]. We display several of the more common postures in Figure 2.2.

Posture Detection is an important technique as it gives additional meaning to sensor data as the sensor data can be tracked per posture of the patient over time as well as tracking how often a patient is in each posture, which is also helpful to track turning schedules. In general these techniques either use machine learning techniques to predict a posture from past data or use geometric data to identify a posture.

Device-Acquired Pressure Injuries

It is important to note that patients with medical devices are at higher risk of developing pressure injuries [137]. If a patient has a medical device, as listed in Table 2.2, they are 2.4 times more likely to develop a pressure injury [11]. Of all patients that form a pressure injury two reports publish 34.5% [11] and 32.8% [20] of the pressure injuries formed were from medical devices, verified visually, e.g. the redness of the pressure injury on the skin is in the shape of the medical device. We note these facts as it will be relevant to future applications to prevent pressure

Table 2.2: Medical device listing from [11].

Anti-embolic stockings
Cervical collars
Endotracheal tubes/commercial endotracheal tube holders
Face masks for non-invasive positive pressure ventilation
Faecal containment devices
Nasal cannulas
Pulse oximetry probes
Radial artery catheters
Sequetial compression devices
Splints and braces
Urinary catheters

injuries. If a device is to sit on the skin of a patient the design of the device must take into consideration that the device itself could contribute to pressure injury formation.

2.6.1 Pressure

Interface pressure sensing is the most extensively studied technique to monitor pressure injuries. In a literature survey of software solutions to prevent pressure injuries, including work up until 2013, out of 36 studies surveyed, 26 used pressure sensing [77]. Pressure based approaches break down into two main categories: Continuous Bedside Pressure Mapping (CBPM) and Posture Detection.

Continuous Bedside Pressure Mapping

Continuous Bedside Pressure Mapping (CBPM) uses a matrix of pressure sensors that is placed on top of a mattress. The pressure value at each location is displayed, usually on a tablet. The typical image that is displayed makes it very easy to see the outline of the patient's body as well as currently what areas of the body are experiencing the most pressure. CBPM systems are offered commercially from companies such as Wellsense, Tekscan, Vista-medical, Xsensor, Novel Electronics, and Sensor Products.

CBPM's biggest benefit is to help healthcare workers when positioning patients so that pressure is actually relieved as the CBPM system visually displays the current pressure distribution of the patient on the measured surface. As mentioned in Section 2.5.1, one study found that standard repositioning does not properly position patients to relieve pressure [39], but using CBPM it was found that healthcare staff were able to more effectively reposition patients [42] and reduce peak pressure [122]. Another study found that using CBPM the average time to turn a patient post alarm was reduced from 120 minutes to 44 minutes [102].

Two studies conducted controlled trials using CBPM and found that the CBPM group had a lower incidence of pressure injuries [126] [7], although in both studies the authors concede that more evidence is needed. A Randomized Controlled Trial was conducted using CBPM and found that there was no reduction in the CBPM group [43].

To address the concern that using CBPM a patient would not be able to be tracked when moved to a new hospital bed a study focused on a CBPM system that uses wireless pressure and temperature sensors attached to the patient's body by using Near Field Communication (NFC) [44]. The sensors were attached to high-risk areas of the body and powered through an antenna in the bed. A monitoring study was done on a sleeping patient and the sensors were verified to accurately collect data.

Pressure Posture Detection Algorithms

Pressure Posture Detection was recognized as an important area of research to monitor pressure injuries as detecting the current posture of the patient allows for pressure tracking as the patient moves as opposed to an average or peak value.

Pressure Posture Detection algorithms work by reading a matrix of pressure sensors roughly the size of a mattress. The data read from a matrix of pressure sensors is generally referred to as a pressure map, as each pressure value is mapped to an x, y coordinate.

So far all work we are aware of in this area are conducted on small sample sizes of roughly 3-15 individuals and as far as we are aware have never been tested in a clinical setting. Recent work tends to focus on the accuracy of detecting the posture of a given algorithm.

Lower density custom pressure overlays were used in several works to classify postures, but with relatively low accuracy. In [51] the authors use a custom low density pressure sensor overlay, using rows of pressure sensors, and classify three postures with 78.7% accuracy by using the probability of a posture based on the distribution of pressures on the custom pressure sensor. An extension of this work using two new custom sensor layouts and using Principal Component Analysis (PCA) and Support Vector Machines (SVMs) to extract features of each posture in addition to using the probability of the pressure distributions were able to classify six postures with 83.5% accuracy [50]. To improve the accuracy of lower density pressure overlays several techniques were developed using an additional camera [52] [111]. With the addition of a camera one work was able to achieve 94% accuracy of 9 postures [52].

Higher density pressure-based Posture Detection techniques rely on a combination of image processing techniques, to pre-process the pressure map, and machine learning techniques to identify the posture. Several works use commercial pressure overlays that provide high density pressure maps. A somewhat earlier work using a commercial pressure overlay [142] was able to detect five postures with 98% accuracy. A subsequent work focused on higher speed classification and was able to classify 97% accuracy identifying eight postures [107].

A custom high density pressure overlay was used in two works [69] [140]. The algorithms in the work cited are designed for sleep monitoring, but these algorithms are also applicable for pressure injuries. The first work [69] was able to detect six postures with 83% accuracy. The second work [140] improved on this and were able to classify six postures with 91% accuracy.

Limb-identification is a subset of techniques in the category of Posture Detection that can track the individual limbs in addition to the posture allowing for pressure tracking of individual parts of the body as the patient turns. In addition to the benefits of Posture Detection, Limb-identification can warn if an individual part of the body is at risk, not just a certain posture. All Limb-identification algorithms found use high density pressure overlays.

An algorithm was developed to detect high pressure areas in [32] using a predefined skeleton template with 86% accuracy. Another algorithm was able to classify limbs with 92% accuracy in three different postures [96] by using clustering based on a predefined body map. Another work used pictorial structures of the body to detect limbs with 90% accuracy [68]. An algorithm that requires no predefined template was developed in [106] and was able to achieve 93% accuracy. Another technique developed to be fast and also does not require a predefined template was able to achieve 94% accuracy [108].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Pressure	↑	...	↑	↑

Pressure overlays that can continuously monitor pressure and display the pressure visually to healthcare staff are **commercially available**, such as Continuous Bedside Pressure Monitoring systems. **Clinical trials** are mixed with pressure based systems, some studies have shown a reduction in incidence while other studies did not. Pressure Monitoring does save **time** for staff, as an assessment of the pressure distribution is automatically displayed and with the addition of Posture Detection algorithms pressure-based systems have the potential to pinpoint which areas of the body are at high-risk, which otherwise would have to be done manually by healthcare staff. Pressure Monitoring is **tuned to the individual** as pressure is measured directly from the patient.

2.6.2 Temperature And Humidity

Temperature is often associated as a risk factor of pressure injuries, but as mentioned in Section 2.3 it isn't clear how temperature relates to pressure injury formation. Humidity or moisture of the skin of the patient is also a risk factor, but the relation of humidity to pressure injury formation is not understood. In Section 2.3 we mention that incontinent moisture can lead to Superficial Skin Injury, which are not related to Deep Tissue Injury and is therefore not by definition a pressure injury. Other factors such as an increase in moisture can lead to skin breakdown and a decrease in moisture can lead to cracking of the skin [143].

One study used a thermal camera to manually take pictures of the heels of patients, the idea being that the difference in temperature between the heels can be an indication of pressure injury formation [8]. Another study manually measured temperature, humidity (they call it moisture), and pressure and found that the difference in temperature between the affected area and the skin around the navel could indicate a pressure injury formation [143].

Most studies monitor temperature in addition to other factors, most notably pressure. One study used both a matrix of pressure sensors in addition to a matrix of temperature sensors to view both in real time [30]. In another study a wearable device was created that can measure pressure, temperature, and humidity [80]. A battery-free wireless sensor was developed in [44] that measures both pressure and temperature and can be placed on various parts of the body.

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual

Sensor-Based Risk-Factor Monitoring

Temperature and Humidity	. . .	NA	↑	↑

Temperature and Humidity Monitoring systems have been developed in prototypes, but are not **commercially available**. No rigorous **clinical trials** have been tested with temperature and humidity systems. Prototype systems able to display real-time temperature and humidity information would be able to save **time** for healthcare staff as information would automatically be collected and displayed. Temperature and humidity would be measured directly from the patient and therefore would be **tuned to the individual**.

2.6.3 Inertial Measurement Unit

An Inertial Measurement Unit (IMU) is a device that reports movement. An IMU typically consists of accelerometers, that report acceleration, gyroscopes, that report angular velocity and orientation, and sometimes magnetometers, that report the magnetic field, which can be used to identify headings (e.g. North, South, East, West). IMU data is used to track steps in smart phones and also geographical location on airplanes.

There are several works that use IMU data to detect the posture of a patient in bed or just to monitor mobility as this can be an indicator of a patient at risk of pressure injuries. There are commercially available systems to monitor IMU data, one being Leaf Healthcare.

Posture detection using IMUs were used in a study using Wireless Identification and Sensing Platforms (WISPs) attached to a mattress to infer the posture of the patient [49]. Each WISP has an accelerometer that transmits data. Using this technique the authors were able to achieve 93% accuracy of classifying five postures. Another study used a single accelerometer to classify three postures with 99% accuracy [141]. Another study used three wearable[62] IMUs to detect four postures with 99.5% accuracy and eight postures with 93% accuracy.

Several works researched systems to monitor accelerometer data. One work investigated a system to monitor wearable accelerometer data and track it over time [114]. Another work investigated a system to monitor accelerometer and some pressure data in a mattress [45]. A real-time system that detects the posture of the patient using an accelerometer and takes a picture every time the posture changes was developed in [94].

A Randomized Controlled Trial was conducted using a wearable IMU and scheduled turning [100]. The trial found that there were significantly fewer pressure injuries in the intervention group and turning compliance was significantly higher in the intervention group. This study used a commercially available system by Leaf Healthcare. The authors note that a limitation of the system is that the device is placed on the trunk of the patient, meaning only trunk turning is detected, so extremities, such as the heels of the patient, are not monitored for compliance [100].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Inertial Measurement Unit	↑	↑	↑	↑

Inertial Measurement Unit (IMU) systems are **commercially available**. **Clinical trials** show that fewer pressure injuries occur when using an IMU system. IMU systems save healthcare staff **time** as they measure and report the activity of patients automatically. Activity is tracked based on the patient and is therefore **tuned to the individual**.

2.6.4 Blood Flow

As discussed previously, it is currently believed that ischemia, the lack of blood flow to the underlying tissue, and reperfusion, the reintroduction of blood to an ischemic region, are two causes of pressure injuries. One way to monitor ischemia and reperfusion can be by measuring blood flow to an area of the body. One such study monitored blood flow at the heel by using infrared sensors and were able to detect noticeable changes when the heel was under pressure [3].

Another study designed an optical probe that can be used to get continuous diffuse correlation spectroscopy and diffuse near-infrared spectroscopy to measure blood flow in a patient study [23]. They found that these may be useful methods in predicting pressure injuries.

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Blood Flow	...	NA	↑	↑

Blood flow systems have only been developed in prototype and are not **commercially available**. No **clinical trials** have been conducted using Blood Flow Monitoring. A potential blood flow system that can monitor blood flow and present this data to healthcare staff is not currently developed, but potentially would save healthcare staff **time**. Blood flow is monitored based on the patient and is therefore **tuned to the individual**.

2.6.5 Biomarker

Biomarkers are measurable biochemical substances that can be used to predict an event, in this case a pressure injury. Some studies show that sweat lactate and Cytokines can be tracked to detect skin breakdown, which would be indicative of a Stage 1 pressure injury [5]. To detect deeper level pressure injuries C-reactive protein (CRP) can be monitored in blood [5]. Currently these monitoring strategies are not developed as a system, but instead can be tested from blood or sweat. Another study found that serum albumin had an inverse relationship to pressure injury formation, so the lower the serum albumin the higher likelihood of pressure injury formation [123]. This finding is in agreement with previous studies as low serum albumin is an indicator of malnutrition and as discussed in Section 2.5.3 it is accepted that malnutrition is associated with pressure injury formation.

A flexible wearable biochemical sensing device that analyzes sweat was developed and tested in a healthy patient population [33]. The device can communicate analysis over Bluetooth. The applications mentioned in the paper are not specifically for pressure injuries, but as sweat lactate is a biomarker for pressure injuries this is a promising biochemical sensor that could be used for pressure injury prevention.

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Biomarker	↓	NA	↑	↑

Biomarker work is purely based on testing blood or sweat from the body and does not currently use a biochemical sensor. In the future a system that could automatically monitor biomarkers may be possible, but as of now these systems are not **commercially available**. Currently an automatic biomarker system is not developed so no **clinical trials** have been run. A biomarker system would potentially save healthcare staff **time**. Biomarkers would be measured directly from the patient and would therefore be **tuned to the individual**.

2.6.6 Skin Integrity

Skin Integrity Monitoring is a technique to monitor the skin for water loss, pH, moisture, elasticity, and color. Although these techniques offer promise the variation between patients and ambient conditions are currently not studied in depth enough at this point in time to be a consistent way to monitor or identify pressure injuries [5]. In addition these types of sensors are ideal for measuring a specific site to test for pressure injury formation as opposed to predicting a pressure injury.

A new bandage-based sensor used to measure the skin integrity via skin impedance spectroscopy was developed and was able to test in rat models that the integrity of the skin had a direct correlation with impedance [132] and may be a possible way to detect pressure injuries. The idea behind this technology is that the skin acts as a capacitor, with the skin layer acting as the dielectric. As the skin breaks down the dielectric layer's permittivity changes thereby giving a change in measurement.

Additionally a commercial company Bruin Electronics makes a hand held device called the SEM Scanner that can run impedance spectroscopy on the skin that can also detect skin damage [82], which can be used to detect pressure injuries.

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Skin Integrity	. . .	NA	↑	↑

Skin Integrity Monitoring sensors and devices are commercially available, but a system to monitor the sensor is not **commercially available**. It is important to note that our criteria for commercial availability implies the ability to operate without healthcare staff intervention. As such we classify handheld devices, whose operation requires healthcare staff, as ”neutral” regarding their commercial availability. No **clinical trials** were run using Skin Integrity Monitoring. Such a system to monitor the skin would save **time** for healthcare staff as the system could automatically measure the skin integrity and present the data to the healthcare staff. The measurements would come directly from the patient and therefore would be **tuned to the individual**.

2.6.7 Electrocardiography

There are a few works that monitor Electrocardiography (ECG) to detect the posture of a patient. An Electrocardiogram uses electrodes to monitor the electrical activity of the heart. ECG Monitoring is more applicable to sleep monitoring as it has been found to detect sleep apnea [58], but the posture classification aspect of the technique could be applied to pressure injuries.

One study used a custom ECG monitor overlay and applied a machine learning technique to classify the posture of a patient. The study found they were able to achieve very high accuracy at 98.4% of four postures [64].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Electrocardio -graphy	. . .	NA	↑	↑

An Electrocardiography (ECG) Monitoring system was developed in prototype, but is not **commercially available**. No **clinical trials** using ECG Monitoring have been run. Such a system would save healthcare staff **time** as ECG data would be monitored and presented automatically to healthcare staff. The data would be measured directly from the patient and is therefore **tuned to the individual**.

2.6.8 Camera

Image processing techniques are very common, such as facial recognition built into many smartphones. A camera is a non-invasive way to monitor a patient using image processing techniques to monitor the posture and mobility of the patient. The issue of privacy is often brought up when using a camera and the most frequent approach to avoid privacy issues is by using techniques to remove the details of the image, so that the outline of the body can be determined, but not the face or any other recognizable features.

In [18] the authors develop a system to monitor patients using a depth camera, to block out features, with the goal of classifying postures and monitoring activity. The system can also notify healthcare staff if repositioning is required. In [40] the authors classify three postures based on a depth camera with an accuracy of 94%. Another work also used a depth camera and was able to classify 10 postures with 93% accuracy, but when a quilt was laid on top of the subject, the accuracy was reduced to 89% [66].

Several works use a camera in addition to other sensors. In one work a camera with a polarizer is used in addition to two types of infrared cameras to monitor the size of a pressure ulcer [70]. The following works were mentioned in Section 2.6.1, but in addition to pressure they also use a camera to improve their posture classification [111] [52].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Camera	...	NA	↑	↑

Camera Monitoring systems to measure the activity of a patient are in prototype, but are not **commercially available**. **Clinical trials** have not been conducted on Camera Monitoring systems. A system would be able to automatically measure the movements of a patient and present the information to the healthcare staff, thereby saving **time**. The movements of a patient would be based on the individual, so that activity is **tuned to the individual**.

2.6.9 Ultrasound

Ultrasound waves are sound waves above the range of human hearing. Several studies have found high frequency ultrasound imaging, using the reflected ultrasound waves to construct an image of the underlying tissue and muscle, a possible way to monitor pressure injuries. A study confirmed that when examining pressure injuries ultrasound imaging was able to visualize the damage beneath the skin [46], but the authors admit that ultrasound imaging needs a certain level of skill to assess the images. Another study concluded similarly that ultrasound is a promising technology, but more work needs to be done on interpreting scans [73].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Ultrasound	. . .	NA	↑	↑

Ultrasound devices exist commercially, but a system to automatically use ultrasound to monitor for pressure injuries does not exist **commercially**. **Clinical trials** have not been run on an Ultrasound Monitoring system. Such as system would save healthcare staff **time** as it would automatically assess whether a patient has a pressure injury. Ultrasound measurements would be directly from the patient and could assess the **individual**.

2.6.10 Impulse Radio Ultra Wide Band

Impulse Radio Ultra Wide Band (IR-UWB) radar is a technique that uses the time of reflection of an electromagnetic pulse to extract information about the reflected surface, which has been used to detect heart rate and respiration rate [89]. A study found that in addition they were able to use IR-UWB to detect four postures with 89% accuracy [89].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Impulse Radio Ultra Wide Band	. . .	NA	↑	↑

An Impulse Radio Ultra Wide Band (IR-UWB) system to measure the activity of a patient exists in prototype, but is not **commercially available**. **Clinical trials** have not been run for such a system. An IR-UWB Monitoring system would save healthcare staff **time** as it would automatically measure and assess the activity of a patient. The system would also be **tuned to the individual** as activity would be based directly on measurements from the patient.

2.6.11 Leaking Coaxial

There is one approach at the time of writing that uses a "Leaking Coaxial Cable", i.e. reading the physical characteristics of a coaxial cable, a type of transmission cable, through a WiFi router. The subtle movements of a patient cause enough of a change when reading from the coaxial cable that a study was conducted to explore the ability to predict the posture of the patient, although the accuracy is not reported [124].

Taxonomy Criteria

Application	Commercial Availability	Clinical Trials	Time Savings	Tuned To Individual
Sensor-Based Risk-Factor Monitoring				
Leaking Coaxial	. . .	NA	↑	↑

A Leaking Coaxial system was developed in prototype, but is not **commercially available**. **Clinical trials** have not been run on such a system. A Leaking Coaxial system would save healthcare staff **time** as it would be able to automatically assess the activity of a patient. The system would also be **tuned to the individual** as measurements would be directly from the patient.

2.7 Discussion

Based on our taxonomy, as seen in Table 2.1, we find the most promising currently researched techniques to prevent pressure injuries are Electrical Stimulation, Pressure Monitoring, and Inertial Measurement Unit Monitoring. We find the most promising strategy for the future to be Biomarker Monitoring. Each technique in isolation has its own advantages which we will discuss, but a system that implements all techniques may be the most effective at reducing pressure injuries. It is also important to note that these techniques are supplementary to nursing guidelines, a mainstay of pressure injury prevention.

We find many Sensor-Based Risk-Factor Monitoring techniques promising, but they are not implemented as a viable system in a clinical setting, e.g. the algorithm is tested on pre-collected data and does not discuss ways to present this data back to the healthcare staff. Some examples of such techniques are Blood Flow Monitoring (Section 2.6.4), as blood flow is a key component of pressure injury formation, Skin Integrity Monitoring (Section 2.6.6), as the integrity of the skin can be an indicator of a pressure injury formation, and Camera Monitoring (Section 2.6.8), as cameras are easy and cheap to deploy. But, all of the mentioned techniques do not have a clinical system to gather the data, analyze the data, and present it back to healthcare staff. We think this shows the need for a clinical software system that can use custom sensors, custom analysis algorithms while keeping data secure, and present data back to the healthcare staff in a customizable way. Such software would lower the barrier of entry of implementing a sensor-based technique designed for a clinical setting.

Electrical Stimulation uses electric current to stimulate muscles, which requires placing electrodes around a high-risk pressure injury area with healthcare staff supervision. From our taxonomy Electrical Stimulation is commercially available and backed by clinical trials, but it is a time-intensive strategy as it requires the nursing staff to put on the device and verify it is working properly as well as continually monitor the device throughout a session. In order for Electrical Stimulation to be applied preventively it would have to be applied at every high-risk area of the body, which would increase the time intensive nature of this technique. Electrical Stimulation offers the best fully autonomous solution, as potentially a system could be developed that could continuously stimulate the muscles of a patient, which could potentially eliminate a pressure injury from forming. But, such a system would have to be rigorously safety tested and also developed in such a way that it does not interfere with the patient's quality of life.

Pressure Monitoring uses a matrix of pressure sensors to continuously monitor the interface pressure between a patient and a surface, typically a mattress or a chair. From our taxonomy Pressure Monitoring is commercially available, saves healthcare staff time, and can be tuned to the individual, but more clinical trials are needed to validate the technique as one trial showed a reduction in pressure injuries, while another trial showed no reduction. Pressure Monitoring is the most intuitive technique to prevent pressure injuries because it is well established, as discussed in Section 2.3, that pressure is the primary cause of pressure injuries. The current Pressure Monitoring systems available are limited to displaying the real-time pressure (Continuous Bedside Pressure Monitoring, CBPM) and cannot track the pressure as the patient changes posture or orientation. In addition to detecting buildups in pressure this technique also offers the advantage of studying pressure distributions over time.

One limitation to a CBPM system is that it cannot track the amount of pressure as the patient rotates, but there are several algorithms that can be used to avoid this limitation. A class of algorithms to process pressure map data are Limb-identification algorithms that can identify the individual parts of the body in various positions. This allows for the tracking of pressure per body part as the patient moves. Limb-identification algorithms are not currently tested in a clinical setting, but offer a promising way of detecting at-risk areas automatically, potentially saving healthcare staff having to periodically manually check the patient. A potential Pressure Monitoring system with Limb-identification offers the best non-invasive way to pinpoint areas of the body that are at risk of developing a pressure injury. The sensors used to monitor pressure are commercially available, it is only the software that needs to be developed in a commercial system.

Inertial Measurement Unit (IMU) Monitoring monitors the movements of a patient. From our taxonomy IMU Monitoring is commercially available, backed by clinical trials, saves time for healthcare staff, and can be tuned to the individual. IMU Monitoring is also inexpensive and can be built very small, the entire device including the ability to communicate over a network can be the size of a square inch button. When fixed to a patient an IMU Monitoring system can track how long a patient was in a fixed position. As the patient moves the IMU device tracks the movement and can display to healthcare staff how long a patient was in a certain posture and what the recommended next posture should be to allow reperfusion. If you had to choose a pressure injury prevention system today IMU Monitoring is the best option as it is commercially available and clinically verified.

Biomarker Monitoring tracks specific biochemicals in the blood and sweat to determine skin breakdown. From our taxonomy Biomarker Monitoring can save healthcare staff time and can be tuned to the individual, but is not commercially available and no prototype Biomarker Monitoring system exists and therefore has not been clinically verified, but the biochemicals to track are researched, but currently such a system would have to rely on manually testing the blood and sweat of a patient. If a specialized biochemical sensor is developed to test skin breakdown automatically, this may also be a promising solution. Biomarker Monitoring offers the most promise as it would directly detect pressure injury formation from the patient's biochemistry, but an established biochemical sensor for this application still needs to be researched and developed.

We see one of the biggest barriers for promising Sensor-Based Risk-Factor Monitoring techniques is the lack of a monitoring system that can collect sensor data, store and analyze the data securely, and present the data back to healthcare staff in a customizable way. Currently the only Sensor-Based Risk-Factor Monitoring clinical trials reviewed in this work, CBPM and IMU Monitoring, are commercial systems that are made exclusively for the respective technique. We see the need for a software system that can be modified easily to use different types of sensors, as well as different types of analysis, in a secure manner. This would allow for more clinical trials and more sensors to be used to reduce pressure injuries at a lower barrier of entry.

2.8 Conclusion

Pressure injuries are currently an ongoing obstacle in healthcare. Pressure injuries are classified as "never events" as they should never occur and yet they are still present. In addition they are costly to treat, but more importantly they impact the quality of life of the patient as they are painful and impact the social life of the patient. Additionally current accepted nursing guidelines and interventions may not be enough to eliminate pressure injuries from occurring as they are time intensive and in the U.S. it is predicted there will be a nursing shortage in the future.

Modern pressure injury research primarily started after World War II and began with studies focusing on a pressure-time threshold that would be able to predict the formation of a pressure injury. Today it is accepted that no such singular threshold exists as it is highly dependent on the individual patient. In addition it is believed that some pressure injuries may be unavoidable even when using all of the accepted practices of prevention, but the percentage of unavoidable pressure injuries is not established.

It is currently understood that the biomechanics that cause pressure injuries are from pressure closing capillaries causing tissue to become ischemic, when under high pressure the closure of larger vessels causing thrombosis, the inflammation caused by the introduction of blood into an ischemic region, the closure of lymphatic vessels causing a buildup of metabolic waste products, and the deformation of tissue cells.

We surveyed the literature to find the latest research on preventing pressure injuries. Based on our findings the current research on prevention of pressure injuries can be broken down into Active Prevention Strategies and Sensor-Based Risk-Factor Monitoring. Active Prevention Strategies require an active role from healthcare staff and will most likely be a mainstay of current practice. Sensor-Based Risk-Factor Monitoring uses a variety of different types of sensors and a software platform that monitors the current condition of a patient and presents this data in an intelligible way to the healthcare staff, saving healthcare staff time as they do not need to manually go through this process.

To evaluate the current techniques we created a taxonomy that evaluates every category of technique based on its commercial availability, support of clinical trials, healthcare staff time savings, and whether the technique can be tuned to an individual. We note that not all techniques are mutually exclusive, for instance nursing guidelines are a mainstay of pressure injury prevention regardless of technique, but additional techniques, such as Pressure Monitoring may be able to reduce the time it takes for healthcare staff to follow nursing guidelines as part of the process is handled automatically.

Based on our findings the most promising techniques currently researched that have the most benefit in addition to nursing guidelines are Electrical Stimulation, Pressure Monitoring, and IMU Monitoring. The most promising future strategy is Biomarker Monitoring.

In addition many of the Sensor-Based Risk-Factor Monitoring techniques are promising, but they are not tested in a clinical setting making them hard to determine whether they will actually work. We believe there is an opportunity for a software system that can easily monitor sensor data, store the sensor data in a secure server, and present it back to healthcare providers. Such a system will lower the barrier of entry to test new Sensor-Based Risk-Factor Monitoring techniques that can monitor data that would otherwise have to be collected manually by healthcare staff thereby saving staff time.

Chapter 3

PIMAP: An Autonomous, Continuous, Real-Time Patient Monitoring System

In this chapter we present a novel framework for autonomous, continuous, real-time patient monitoring that can be used in different settings, e.g. hospital ICUs, clinics, skilled nursing facilities, and homes. Our research is motivated by our collaboration with UC San Francisco's medical researchers and their longstanding work on understanding and treating complex wounds, in particular pressure injuries.

In Chapter 2 we discussed the state-of-the-art in pressure injury prevention and we found that sensor-based patient monitoring is a promising approach to assess risk in an objective and patient-centric fashion. The generated objective metrics can then be used by healthcare clinicians to focus efforts on the highest-risk patients. However, patient monitoring systems presented in the literature are either commercial systems that are limited to what the commercial entity offers or are one-off solutions that are not discussed in detail and are not released for use by other researchers. In addition visualization of the sensor data is usually not discussed at all even though this is a key aspect of using the sensed data.

This dilemma leaves medical researchers in a difficult position to evaluate novel sensors or novel algorithms as a commercial system will not be able support a novel sensor/algorithm and creating a patient monitoring system from scratch is tedious and prone to failure. We designed, tested, and evaluated the Pressure Injury Monitoring And Prevention (PIMAP) system framework based on this critical need. PIMAP is designed for medical researchers and clinicians. Medical researchers can leverage this system framework by focusing their efforts on the medical device integration or on the algorithm, which would have to be done regardless, without having to design the monitoring system in between. Clinicians can use PIMAP to automatically visualize data to make decisions in real-time. PIMAP is not a diagnostic system and is instead a tool for medical researchers to accelerate research and a tool for clinicians to view analyzed sensor data.

To address this need we studied the average use case in any patient monitoring scenario agnostic of what technologies are used and claim that a patient monitoring framework must contain the following four components: *Sense* which collects sensor data, *Store* which stores sensed data as well as analytics, *Analyze* which processes sensor data using different algorithms, and *Visualize* which visualizes sensed data and resulting analytics.

The case study and driving application for PIMAP is pressure injuries, but the system model is generalizable to all patient monitoring use cases. PIMAP can be utilized by researchers to monitor any condition as it can incorporate any new sensor and any type of analysis, while leveraging existing sensing, storage, analysis, and visualization technologies. In addition PIMAP is designed with a network in mind and can be deployed in a distributed manner at different layers of the network such as the Edge and/or Cloud.

The power of PIMAP is twofold. One, any new sensing, storage, analytics, and visualization technology can be incorporated without putting the burden on the medical researcher to create an entire new system for this purpose. Two, PIMAP enables clinicians to view new and novel sensing/metrics in real-time.

The remainder of this paper is organized as follows. In Section 3.1 we discuss the related work and how our contribution fits into existing research. In Section 3.2 we discuss the primary design concepts of PIMAP. In Section 3.3 we discuss the four PIMAP components. In Section 3.3.5 we discuss the three PIMAP workflows. In Section 3.4 we discuss our decisions when implementing PIMAP. In Section 3.5 we discuss how we evaluated our implementation of PIMAP. In Section 3.6 we discuss privacy and security concerns. Finally in Section 3.8 we discuss our future work regarding PIMAP and conclude this chapter.

3.1 Related Work

Due to its interdisciplinary nature, our work can be classified under different areas in the research literature including: connected health, wireless body area networks (WBANs), ubiquitous healthcare, remote monitoring, ehealth, patient monitoring, internet of healthcare things (IoHT), mHealth, and telemedicine. In this section, we highlight related works that focus on general patient monitoring frameworks. The survey presented in [75] provides a more complete and detailed description of related work specific to pressure injury prevention.

In order to investigate whether the 6LoWPAN specification, which is an adaptation layer that allows IPv6 packets to be sent over the low-power physical and MAC layer 802.15.4 standard, is able to achieve the necessary throughput to support medical applications, a system consisting of Sensor Units, Patient Unit, Remote Processing Unit, and Server is proposed in [133]. Sensor Units send sensor data, the Patient Unit acts as a bridge between the Sensor Units and the Internet, the Remote Processing Unit processes sensor data, and the Server stores the data and restricts access to the data. While the paper concludes that 6LoWPAN is able to achieve the necessary throughput for common case medical applications, the proposed setup is not released for use by other researchers and does not include visualization capabilities. Since the system was put together with a specific goal, it is not clear whether it could be extended to accommodate different sensing technologies and analytics.

A system application to store historical common biometric sensor data that can be analyzed offline was proposed to address the need to perform historical analysis on medical sensor data [6]. The framework relies on proprietary medical devices to gather the sensor data in the clinic and Kafka [56] to store the data for historical analysis. The analysis presented was performed by downloading data after collection from the data storage server.

Several smartphone solutions have been proposed in the literature. For example HealthSense is a smarthphone application developed to facilitate clinical trials [21]. It uses a web portal to setup a trial, enroll participants, create survey questions, compute, and store data. The focus is on a clinical workflow tool that can support both questionnaires as well as common sensor information that can be gathered from a smartphone.

An Android interconnection layer entitled TIROL was proposed [131] to collect medical sensor data from a myriad of sensors. The author's highlight that no health data protocol or standard has prevailed and typically each vendor has its own method of generating data. TIROL was developed to address this issue and was designed such that any standard can be implemented, but is abstracted by the interconnection layer so that the overarching application is agnostic to the protocol used to gather the data.

A medical sensor data collection application entitled $p^2Health$ uses a smartphone app, ModMedApp, to collect data from vendor servers, Bluetooth or Ant+ sensor devices, and questionnaires. ModMedApp stores the collected data into a cloud-based server that clinicians and patients can interact with [86].

The brain scan community appears to be the furthest ahead in terms of developing open source software to analyze different aspects of brain activity. In one work the authors propose an analysis software, MNE Scan, that can both gather and analyze brain activity data in real time [29]. The authors mention that there is other bran scan software available in addition to their own that is open source and capable of real-time analysis. The software framework is based on sensor plugins, algorithm plugins, and a display manager. The software is able to gather and display data in real-time.

We did attempt to adapt MNE Scan to meet the needs of pressure injury monitoring at one point in time, but the software is very specific to brain scan monitoring and also based on our investigation is not network based, meaning the software is run locally on one computer that can physically connect to the brain scan device. There is a plugin per device connection.

Another work in the brain scan community discussed some of the difficulties of connecting data related to neurological analysis [117]. The authors mention that the brain scan community has software that can collect the data, but connecting all of this data across multiple sites is still elusive and a working group to develop an architecture to do so is in development.

Another specific application example is in the use of ventilator data. A paper was published on a system framework to automatically collect and analyze ventilator data [113]. The framework relies on a very specific workflow and consequently it is not clear how easy it would be to adapt this system to other types of sensor data or other workflows, for example across multiple sites, yet still allows for semi-automated collection and analysis of data with intentional human in the loop design.

As our focus is on pressure injury prevention we have selected three notable systems that show interesting results in reducing pressure injuries using sensor information. However they all use proprietary software. One work used a bed sized array of pressure sensors to continuously read and display pressure information to clinicians and found that patients were more effectively repositioned [42], but the same group conducted a randomized controlled trial using the same sensor and software and found that there was no pressure injury reduction in the group using this sensor [43].

A randomized controlled trial using a wearable inertial measurement unit, a sensor that detects movement, and scheduled turning found that patients using this sensor and associated software had significantly fewer pressure injuries and turning compliance was higher [100]. Turning compliance is how close the clinicians were able to turn patients to a desired periodic schedule.

Our work is complimentary to the existing literature and to the best of our knowledge PIMAP is the only open source patient monitoring system that operates autonomously, continuously, and integrates sensing, data collection, storage, data analysis, and visualization in a single system. It allows different sensors, off-the-shelf and custom, to seamlessly connect to the system and can integrate various analytics and visualizations. Additionally, PIMAP is designed to be deployed in both centralized and distributed configurations in order to cater to the needs of different deployment settings, including edge-, cloud-, and hybrid deployments.

3.2 PIMAP Design Concepts

There are a wide variety of applications for patient monitoring from services as non-critical as location monitoring to critical applications such as pressure injury monitoring. Nevertheless, PIMAP's premise is that the typical data flow for most patient monitoring applications is essentially the same with a few variations.

Data generally starts from a sensor device, for example a GPS sensor or a pressure sensor, and is sampled periodically. This data is then typically stored somewhere for future analysis. Data is then read from this storage and analyzed as the raw data is often hard to interpret, and then stored back into storage. Finally the analyzed data is read from storage and visualized, which may be for clinicians or for a report to correlate this information with the condition being monitored. This general workflow is illustrated in Figure 3.1.

There are variations to this data flow such as not using storage and instead going from

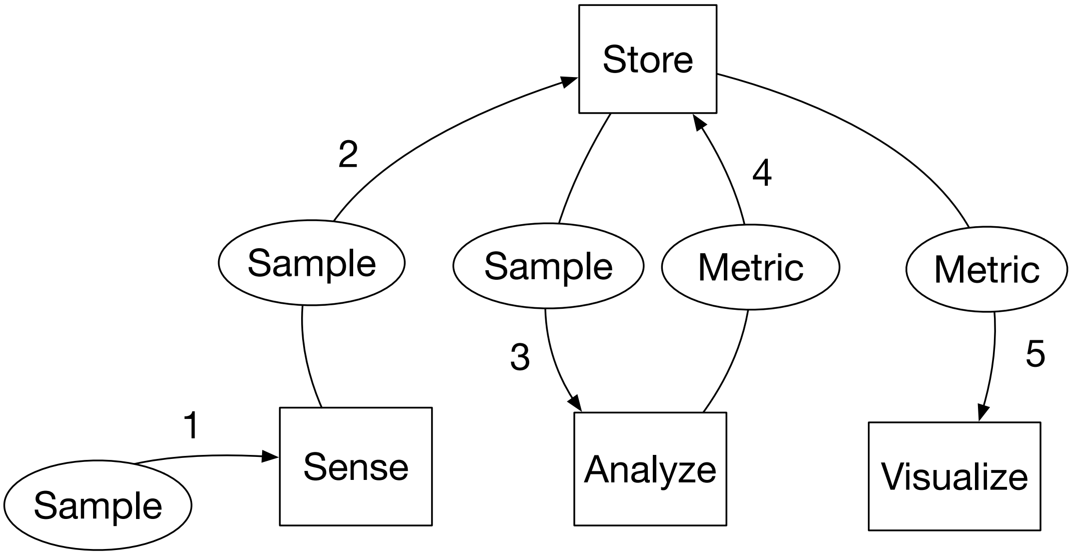


Figure 3.1: General Data Flow For Patient Monitoring

the sensed data to analysis and analyzed data to visualization. However, it is generally better practice to store the data for historical analysis. Otherwise this data is lost and the entire experiment must be redone to perform new analysis.

PIMAP is designed with these concepts in mind and concentrates on four components appropriately named based on the general data flow discussed prior: PIMAP-Sense, PIMAP-Store, PIMAP-Analyze, and PIMAP-Visualize. All data passed between components we define to be lists of PIMAP-samples, PIMAP-metrics, and/or PIMAP-commands, which are all self-contained, meaning all data needed to process a PIMAP data type is contained within itself.

Given the sensitive nature of patient data, PIMAP is designed with both security and privacy in mind. Mechanisms to ensure secure and privacy-preserving operation are discussed in Section 3.6.

3.2.1 The Three PIMAP Data Types

We define three types of data that are passed through the PIMAP system: PIMAP-samples, PIMAP-metrics, and PIMAP-commands. All three pieces of data are self-contained, which means that they can be interpreted on their own and they do not need some sort of exchange or handshake to interpret.

Typically PIMAP-samples contain raw sensor data. For example a pressure sensing device could send data to a PIMAP-Sense component, e.g. via UDP, at which point PIMAP-Sense creates a PIMAP-sample that contains the raw pressure values. A PIMAP-metric is data generated by analyzing PIMAP-samples. A PIMAP-command can be used to actuate on a sensing device, e.g. to change its sampling frequency.

PIMAP-Sample A PIMAP-sample has five identifiers:
sample_type, *patient_id*, *device_id*, *timestamp*, and *sample*. The *sample_type* identifies the type of sensing device, this is used in the storage process in order to streamline analysis. The *sample_type* could be inferred, but by simply providing it we avoid unnecessary complexity and errors.

The *patient_id* and *device_id*, respectively identify which patient is being monitored as there may be multiple patients and which device generated the sample as there may be multiple devices per patient.

The *timestamp* is how we identify the samples in time and is the time that the sample was generated. This assumes that *timestamps* have finer resolution than the number of samples generated, but based on experience this is a reasonable assumption for the majority of applications.

The *sample* is the actual data being sampled. For example if it is a pressure sensing device that is generating the data the sample would be the pressure sensor values.

PIMAP-Metric A PIMAP-metric is very similar to a PIMAP-sample and also has five identifiers:
metric_type, *patient_id*, *device_id*, *timestamp*, and *metric*. In general a PIMAP-metric is created from one or more PIMAP-samples. The *patient_id*, *device_id*, and *timestamp* are all generated from a the PIMAP-sample(s) used to generate the PIMAP-metric. This would typically occur by the PIMAP-Analyze component, which takes as input PIMAP-samples and generates PIMAP-metrics.

The *metric_type* identifies the type of analysis performed that generated this metric. For example, it could be a simple average or a more complex analysis such as Objective Mobility [76], a metric that calculates the mobility of a patient using pressure data from a wearable sensor.

The *metric* is the actual data generated by the analytics algorithm. Using the Objective Mobility Analysis example, the metric would be the number of movements per minute.

PIMAP-Command A PIMAP-command is used to send commands to a sensing device to change some aspect of that device, such as its sampling rate. A PIMAP-command has five identifiers:
command_type, *patient_id*, *device_id*, *timestamp*, and *command.*

The *command_type*, *patient_id*, and *device_id* are the unique identifier of the destination of the PIMAP-command. The *timestamp* is the time the PIMAP-command was generated and the *command* is the actual command with parameters to change. The PIMAP-command is the least developed of the PIMAP data types. We use the PIMAP-command in our experimentation in Section 3.5.4, but the PIMAP-command is still in exploratory development.

3.3 PIMAP System Framework

Examining the general data flow in a patient monitoring application we develop four main components as mentioned earlier in Section 3.2: PIMAP-Sense, PIMAP-Store, PIMAP-Analyze, and PIMAP-Visualize. In this section we define the four main PIMAP components. Each of these components are abstractions and are not associated with a specific technology. For example PIMAP-Sense could be sensing UDP packets or Bluetooth packets. Our goal is to enable PIMAP to integrate different underlying technologies in a seamless fashion while still maintaining the same structure and functionality.

If it is not entirely clear why we would want to treat these components as abstract objects we will provide an example. The PIMAP-Store is probably the simplest of these objects from an interaction perspective. If you write to it the data is stored and you can read back any data that was previously written. But, in practice a data-store is not trivial to implement, the simplest data-store could be a text file, but it is not immediately clear in what format data should be written and how to retrieve data quickly. What if we want to switch to using a database? By treating the data-store as an object, which at this point in time is a well-established practice, all that we need to know is that we write either PIMAP-samples or PIMAP-metrics and we read PIMAP-samples or PIMAP-metrics. The implementation can change, but the interface will not.

To make this even more concrete for the current implementation we rely on Kafka [56], which is a distributed data-store based on the publish-subscribe model, but when interacting with the PIMAP-Store the developer does not need to know how to interact with Kafka as this is all abstracted when using the PIMAP-Store object.

3.3.1 PIMAP-Sense

PIMAP-Sense is how PIMAP interacts with sensor devices. For example if a new novel sensor is created and is sampled via some sort of microcontroller as is typical, in all cases this data must be sent somewhere. One option is to save it locally and then transfer this data to a computer for post-analysis. But, a more favorable practice is to send the data to a computer in real-time over a network or physical wire for real-time analysis. PIMAP-Sense is the component of PIMAP that gathers and/or senses this sensor data.

PIMAP-Sense can accommodate a variety of sensing devices communicating over different network protocol standards. For example an actual implementation of PIMAP-Sense is PIMAP-Sense-UDP, which on each call reads UDP packets that are sent to its interface. One can imagine running a clinical trial with several devices on several different patients sending data via UDP to an endpoint. Using PIMAP PIMAP-Sense-UDP is this endpoint. It is even possible to have many different PIMAP-Sense components running in parallel and in many different locations. In addition PIMAP-Sense can send PIMAP-commands back to the sensor device (this is in exploratory development).

The core interaction of PIMAP-Sense is that a call or query to PIMAP-Sense, PIMAP-Sense.sense() returns a list of PIMAP-samples sent to this endpoint and subsequent calls return any new PIMAP-Samples sent. A PIMAP-command is sent using PIMAP-Sense.send(list of PIMAP-commands).

3.3.2 PIMAP-Store

PIMAP-Store is responsible for data access in the PIMAP system. PIMAP-Store can store PIMAP-samples, PIMAP-metrics, or even PIMAP-commands. To store PIMAP-data only the PIMAP-data itself needs to be provided to PIMAP-Store, but to retrieve data either the *sample_type*, *metric_type*, *command_type* must be provided. PIMAP itself is basically a publish subscribe model as we are dealing with time series data. So when data is retrieved from PIMAP-Store data is returned in order of the timestamps.

To store data, it is simply PIMAP-Store.store(list of PIMAP-samples/metrics) and to retrieve PIMAP-Store.retrieve(*sample_type*, *metric_type*, or *command_type*), which returns a list of PIMAP-samples or PIMAP-metrics respectively. It is not guaranteed that all data will be retrieved in one call, but instead to retrieve all data PIMAP-Store.retrieve() should be continuously called until PIMAP-Store.retrieve() returns an empty list.

PIMAP-Store is not coupled to a particular technology and any implementation of PIMAP-Store must adhere to the simple guidelines as provided. This may make some data-store technologies more or less difficult to implement.

3.3.3 PIMAP-Analyze

PIMAP-Analyze creates PIMAP-metrics from PIMAP-samples. Analysis is often specifically associated with a specific type of PIMAP-sample. This is unavoidable as there are very few types of analysis that can be performed on all types of data that is also useful. For example Objective Mobility analysis is performed on Pressure Bandage PIMAP-samples. It is nonsensical to perform Objective Mobility Analysis on other types of PIMAP-samples.

From experience with using PIMAP-Analyze there are several different common ways to convert PIMAP-samples to PIMAP-metrics. The most straightforward is a one to one conversion, meaning one PIMAP-sample can be analyzed and used to create a PIMAP-metric. We say this is the most straightforward because in this scenario PIMAP-Analyze has no memory, if you feed in ten PIMAP-samples you will get ten PIMAP-metrics.

Another type of analysis that is also somewhat common is a many to one ratio of PIMAP-samples to PIMAP-metrics, for example if one is analyzing the amount of movements per minute of many PIMAP-samples. In this type of analysis a history or state is kept and then using a time window PIMAP-metrics can be generated using multiple PIMAP-samples.

A less common scenario, but one that we employed in our analysis, is a many PIMAP-sample to many PIMAP-metric ratio with a time delay. We employed this strategy when analyzing the gradient. You need multiple PIMAP-samples to calculate the gradient, but when calculated each PIMAP-sample used in calculation has a gradient value. In this analysis a time window is used to gather PIMAP-samples and then PIMAP-metrics are generated when this threshold is reached.

The varieties of analysis that can be supported is rich and the only guideline we enforce for the PIMAP system is that PIMAP-metrics must be generated, but this can be a one-to-one relationship, many-to-one, or many-to-many with a time delay. And there may be other types of analysis that we have not yet discovered that are also supported.

The primary interaction with PIMAP-Analyze is
PIMAP-Analyze.analyze(list of PIMAP-samples), which returns a list of analyzed PIMAP-metrics.

3.3.4 PIMAP-Visualize

PIMAP-Visualize is the component of PIMAP that gives feedback to the clinician/researcher/developer. PIMAP-Visualize takes in either PIMAP-samples or PIMAP-metrics and based on instantiation displays the data. A common type of visualization is displaying the time series data, where the x-axis is the *timestamp* and the y-axis is the data being displayed. But, other types of visualization are also supported, such as heat maps.

From experience it tends to be more useful to display PIMAP-metrics than PIMAP-samples as the reason why PIMAP-metrics are generated is because the PIMAP-samples are often hard to interpret, but an example of where this may not be the case is data such as room temperature.

Visualizing time series data is common to most types of data, but there may be cases where specialized visualization is necessary and this is supported with PIMAP-Visualize. For example to visualize a graph one could use PIMAP-Visualize-Graph and to visualize a heat map one could use PIMAP-Visualize-Heat-Map. The only guideline is that a PIMAP-Visualize component must take in a list of PIMAP-samples or PIMAP-metrics and visualize these in some way. The main interaction of PIMAP-Visualize is PIMAP-Visualize.visualize(list of PIMAP-samples/metrics).

3.3.5 The Three Core PIMAP Workflows

The interaction among PIMAP's components are governed by three main PIMAP workflows:

1. Sense PIMAP-samples and store the PIMAP-samples (sense and store).

2. Retrieve PIMAP-samples, analyze the PIMAP-Samples, and store the respective PIMAP-metrics generated by PIMAP-Analyze (retrieve, analyze, and store).

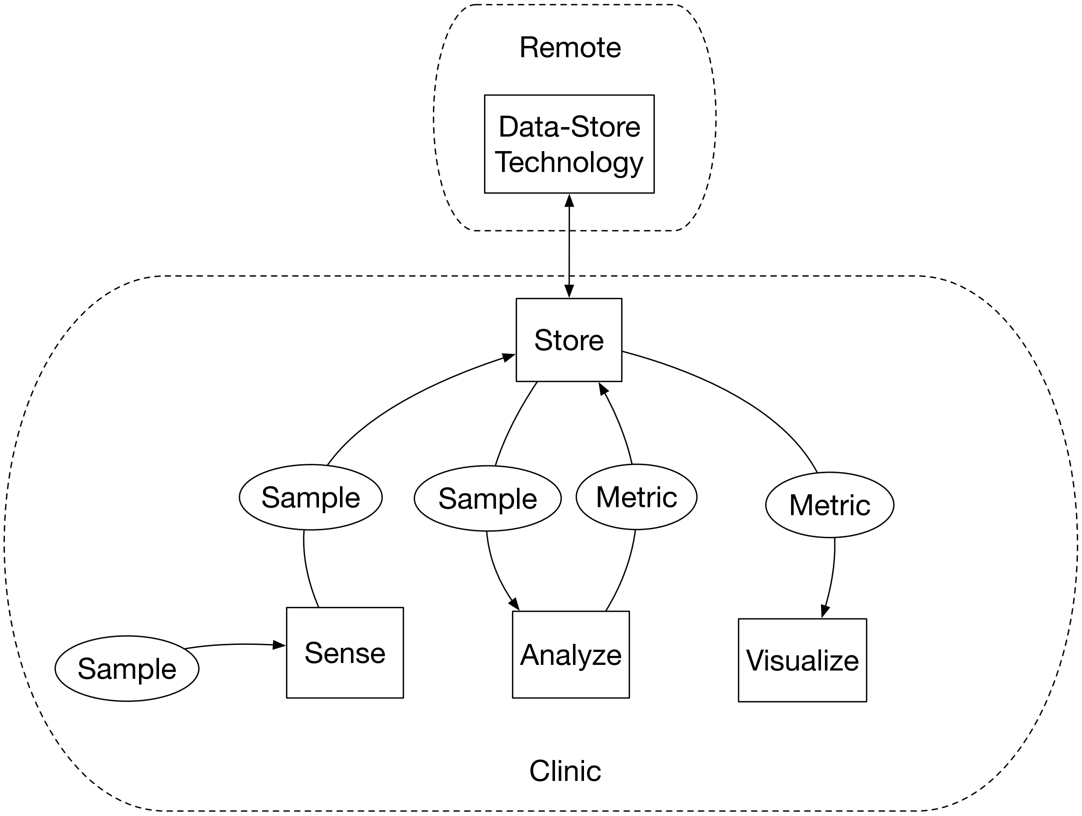


Figure 3.2: Clinic Sense Data Flow Scenario

3. Retrieve PIMAP-metrics and visualize the PIMAP-metrics (retrieve and visualize).

The three workflows are separate, but interlinked, and can be run in parallel and in multiple distributed configurations. For example we present two scenarios, which we entitle Clinic Sense and Cloud Sense to demonstrate two of many scenarios in which PIMAP can be configured.

The Clinic Sense scenario, illustrated in Figure 3.2, has one location, the clinic, which could be a hospital room. In this scenario one or more sensor devices send sensor data to a PIMAP-Sense component located in the clinic. In this scenario the sense and store workflow, the retrieve, analyze, and store workflow, as well as the retrieve and visualize workflow are all located in the clinic. In this scenario PIMAP is run entirely in the clinic, but this makes no assumptions about the underlying technologies at work, for example although PIMAP-Store is running in the clinic, the underlying data-store technology, let us say Kafka, could be running remotely.

The Cloud Sense scenario, illustrated in Figure 3.3, has two general locations, the clinic and the Cloud and are accessible to each other via a network. In this scenario one or more

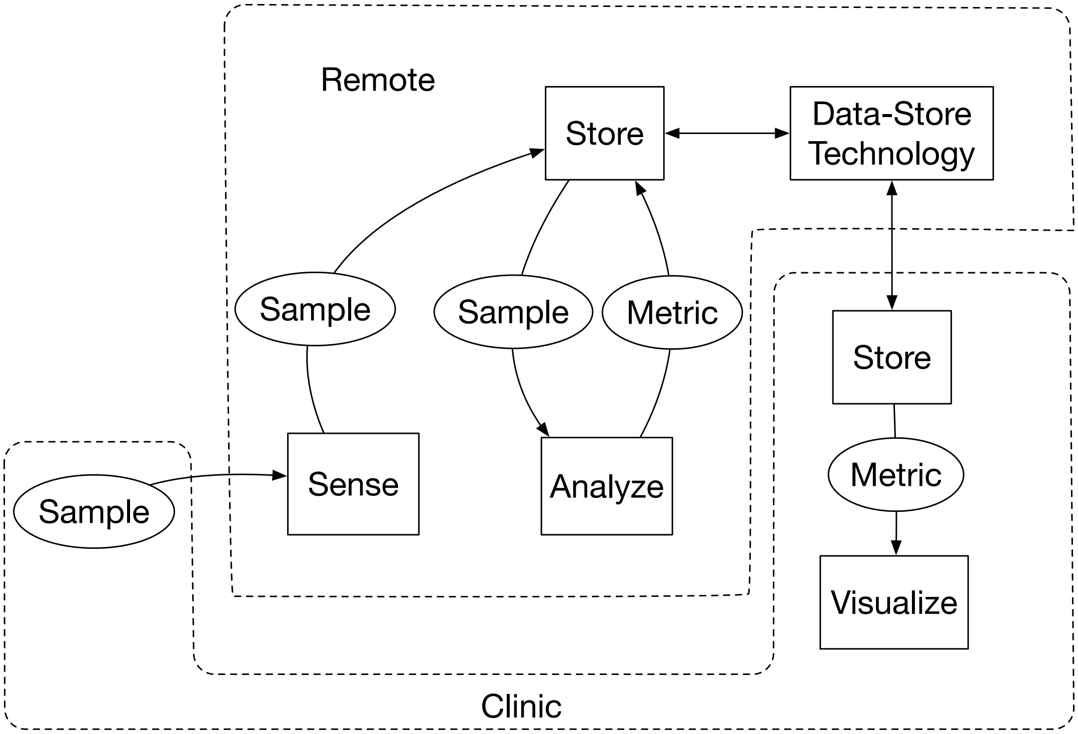


Figure 3.3: Cloud Sense Data Flow Scenario

sensor devices are located in the clinic and send data to a PIMAP-Sense component. The sense and store workflow, the retrieve, analyze, and store workflow are located in the Cloud, and the retrieve and visualize workflow is located in the clinic. This scenario allows for more resources and power to be applied to sensing, analyzing, and storing/retrieving data by utilizing the Cloud. The visualize component would be in the clinic so that clinicians can see the patient's data. To address data security and privacy issues in this data flow scenario, PIMAP-Sense would incorporate data security and integrity mechanisms such as encryption

In Figure 3.4 we present simplified Python style pseudocode of the three core PIMAP workflows.

3.4 PIMAP Implementation Decisions

PIMAP was designed to be easily extensible to accommodate different sensing devices, analytics, and visualization methods. As proof-of-concept, we implemented the following PIMAP component instances in Python: PIMAP-Sense-UDP, PIMAP-Store-Kafka, PIMAP-Analyze-

```
while Running:
  pimap_samples = Sense.sense()
  Store.store(pimap_samples)
```

(a) Sense and Store

```
while Running:
  pimap_samples = Store.retrieve()
  pimap_metrics = Analyze.analyze(pimap_samples)
  store.store(pimap_metrics)
```

(b) Retrieve, Analyze, and Store

```
while Running:
  pimap_metrics = Store.retrieve()
  Visualize.visualize(pimap_metrics)
```

(c) Retrieve and Visualize

Figure 3.4: The Three Core PIMAP Workflows

Objective-Mobility, and PIMAP-Visualize-Plt-Graph, each of which is described in detail below. PIMAP components can adapt to application requirements, e.g. to accommodate both low and high throughput scenarios as well as server and network load. In Section 3.5 we will evaluate how PIMAP performs under the different conditions and workloads.

To enable PIMAP to dynamically adapt to different application requirements, system conditions, and evaluate these limitations we developed a profiling methodology. We profile PIMAP by analyzing the throughput each component can handle in isolation. The profile is not an absolute limit, but instead an estimate. In the discussion of the implementation decisions made for each component we will also discuss our methods on profiling each component. All throughputs are reported in PIMAP-samples/metrics per second.

In addition we realize there is a benefit for the PIMAP system to monitor itself so that it can adapt to different situations. For this reason each component has a parameter, system-samples, that can be set to generate PIMAP-samples that report on information relevant to each component.

3.4.1 PIMAP-Sense-UDP

User Datagram Protocol (UDP) is part of the Internet protocol standard and therefore we leverage existing tools to make the PIMAP-Sense-UDP component. UDP is a non-reliable

Table 3.1: PIMAP profile of three computers

	iMac 2010	System 76 Oryx Pro	Raspberry Pi 2
PIMAP-Sense-UDP.sense()	3,555	49,000	1,217
PIMAP-Store-Kafka.store()	38,780	100,200	3,527
PIMAP-Store-Kafka.retrieve()	102,700	90,840	4,275
PIMAP-Analyze-Objective-Mobility.analyze() In	3,588	10,000	15.20
PIMAP-Analyze-Objective-Mobility.analyze() Out	7,192	20,010	35.00
PIMAP-Visualize-Plt-Graph.visualize()	6,653	1,329	338.0
	PIMAP-samples/metrics a second		

communication protocol and therefore is not appropriate for scenarios where every piece of data is critically important. We recognize this and in future development plan to support reliable communication protocols such as TCP. The PIMAP-Sense-UDP component is a multi-process server that listens on a given host and port. The amount of processes is user configurable, but for evaluation purposes we use three server processes as we did not find a benefit to increasing this number on the systems we profiled. We demonstrate this configuration's adaptivity in Section 3.5.1.

To profile PIMAP-Sense-UDP we run one process that sends PIMAP-samples as quickly as possible. In a separate process we initialize PIMAP-Sense-UDP to output system-samples that report the throughput and call PIMAP-Sense-UDP.sense() as quickly as possible. We monitor the generated system-samples and average the reported throughput over the length of profiling. The results of profiling can be seen in Table 3.1.

3.4.2 PIMAP-Store-Kafka

We leverage Kafka [56] in our initial PIMAP-Store implementation. Kafka is a publish-subscribe data model, where a producer publishes data to a topic and a consumer can subscribe to a topic to eventually receive every message that was published to the given topic. The PIMAP-Store component is divided into two interfaces, PIMAP-Store.store() and PIMAP-Store.retrieve(), which fits naturally into Kafka as PIMAP-Store-Kafka.store() corresponds to a producer and PIMAP-Store-Kafka.retrieve(topic) corresponds to a consumer. Kafka has the added benefit that data can be distributed across multiple sites (multiple brokers). As PIMAP is written in Python we leverage the confluent-kafka API [110]. A Kafka broker must be setup independently of PIMAP, but this is relatively easy for a developer to setup or a paid cloud-based service can be used instead.

PIMAP-Store-Kafka uses the *sample_type* or *metric_type* as the Kafka topic and the PIMAP-sample/metric as the Kafka value. We create a consumer per topic requested. Kafka consumers have two parameters that greatly affect the throughput, the number of messages and timeout. When a consumer requests a topic from Kafka it will return after the given number of messages is reached or a given timeout is reached. To make PIMAP-Store-Kafka.retrieve() adaptive we fix the timeout to $100ms$ and decrease the number of messages parameter if a timeout occurs, otherwise we increase the number of messages parameter. We demonstrate how this configuration can dynamically adapt as system conditions change in Section 3.5.1.

To profile PIMAP-Store-Kafka.store() we initialize PIMAP-Store-Kafka to output system-samples and store as many PIMAP-samples as possible in a single process. We monitor the generated system-samples and average the reported throughput over the length of profiling. To profile the PIMAP-Store.retrieve() interface we retrieve the samples that were sent previously in the profile of PIMAP-Store.store(), monitor the system-samples generated and average the throughput reported over the length of profiling. The results of profiling can be seen in Table 3.1.

3.4.3 PIMAP-Analyze-Objective-Mobility

Objective Mobility [76] has been proposed as a metric to quantify patient mobility based on pressure readings collected by a custom wearable pressure bandage. Using the four by four grid of pressure sensors embedded into the bandage Objective Mobility reports an approximate angle of the patient on a bed and based on the angle variations in time calculates the amount of movements a patient makes.

PIMAP-Analyze-Objective-Mobility is written in Python and takes advantage of numpy's [95] libraries as well as Python's map utility to perform parallel calculations as often as possible.

To enable PIMAP-Analyze-Objective-Mobility to dynamically adaptive to current workload and network conditions we added an aggregation buffer that adjusts based on a timeout. If a timeout occurs we increase the aggregation buffer until we reach a maximum buffer length based on a maximum processing delay tolerance. If aggregation increases past the maximum processing delay tolerance we cut the aggregation buffer in half. Section 3.5.1 reports on PIMAP-Analyze's ability to dynamically adapt to the underlying system dynamics.

To profile the analyze component we send as many pressure bandage PIMAP-samples as possible in a single process. We monitor the system samples generated and average the throughput over the length of profiling The results of profiling can be seen in Table 3.1.

3.4.4 PIMAP-Visualize-Plt-Graph

For our initial visualize component we leverage the matplotlib [78] Python library, a common library used to display data. We focus on graphing data over time. Often when we visualize data in this way we can observe phenomena that would otherwise be unnoticed when looking at a singular value. An example of this is the difference between a time-lapsed photograph and video of the night sky. In the time-lapsed photograph the stars appear as streaks in the sky, whereas a video depicts the stars moving across the sky (of course the reality is the stars are actually not moving from the vantage point of the Earth, but that is besides the point).

To make PIMAP-Visualize-Plt-Graph adaptive we use an aggregation buffer similar to PIMAP-Analyze-Objective-Mobility, a limit on the amount of data that can be displayed, and an update period, which determines how often data is displayed. If the time to process data is above a threshold we decrease the size of the aggregation buffer otherwise we increase the aggregation buffer. If the time it takes to visualize data is greater than the update period we downsample the data to be displayed. This will decrease the resolution, but the only alternative is to increase the update period. We demonstrate this configuration is adaptive in Section 3.5.1.

To profile the visualize component we visualize as many PIMAP-samples as possible in a single process. We monitor the system-samples generated and average the throughput over the length of profiling. The results of profiling can be seen in Table 3.1.

Table 3.2: Description of throughput experimental setup

Application	Location run
Low throughput pressure bandage data, 1 sample/s via UDP	iMac 2010
Medium throughput pressure bandage data, 100 sample/s via UDP	iMac 2010
High throughput pressure bandage data, 2,000 sample/s via UDP	iMac 2010
PIMAP-Sense-UDP	iMac 2010
PIMAP-Store-Kafka	iMac 2010
Kafka	iMac 2010
PIMAP-Analyze-Objective-Mobility	iMac 2010
PIMAP-Visualize-Plt-Graph, update period 1s	iMac 2010
Network	Local network

3.5 PIMAP Evaluation

To evaluate PIMAP's features and performance, we ran a variety of experiments including: (1) we evaluate PIMAP's performance in low-, medium-, and high throughput scenarios to demonstrate that PIMAP has low end-to-end latency from the time data is sampled to the time data is visualized regardless of throughput; (2) we demonstrate PIMAP's ability to integrate new sensors, in particular a custom skin health sensor; (3) we connect PIMAP to a sensor network simulation platform (COOJA [97]); and (4) demonstrate how PIMAP can be used to analyze and visualize data in real-time by playing back data obtained from a wearable pressure bandage.

3.5.1 Low, Medium, And High Throughput Scenario With Low End To End Latency

To demonstrate PIMAP's ability to adapt to different throughput scenarios we evaluate the end to end latency, the time the data is sampled to the time it is visualized, when running PIMAP. In addition to the end to end latency we also monitor the way PIMAP adapts to low, medium, and high throughput scenarios, which is the aggregation limit for PIMAP-Analyze-Objective-Mobility, the number of messages for PIMAP-Store-Kafka retrieve, and the aggregation limit for PIMAP-Visualize-Plt-Graph.

We configure PIMAP, including Kafka, locally on one computer. See Table 3.2 for the configuration. Based on our profile seen in Table 3.1 the limiting throughput is PIMAP-Sense, which is a little over 3,500 PIMAP-samples/s. Based on this limit we experiment with a low throughput scenario of 1 sample/s, a medium throughput scenario of 100 PIMAP samples/s, and a high throughput scenario of 1,000 PIMAP samples/s. We ran each experiment for ten minutes and found on our system the average end to end latency for the low throughput scenario to be 20.0ms, for the medium throughput scenario to be 203ms, and for the high throughput scenario to be 411ms. From this we can see that the latency does not scale linearly with sample rate and is closer to logarithmic and even at a high sample rate of 1,000 samples/s we stay below 500ms.

To demonstrate the adaptiveness of PIMAP we ran an additional experiment where we adjust the throughput during the experiment. We use the same setup as in the previous experiment and for two and a half minutes we send at a low throughput of 1 sample/s, the next two and a half minutes we send at a medium throughput of 100 samples/s, the next two and a half minutes we send at a high throughput of 1,000 samples/s, and for the final two and a half minutes we send at a low throughput of 1 samples/s. The results can be seen in Figure 3.5 and it is clear to see the divisions in sample rate as each parameter adjusts with the sample rate.

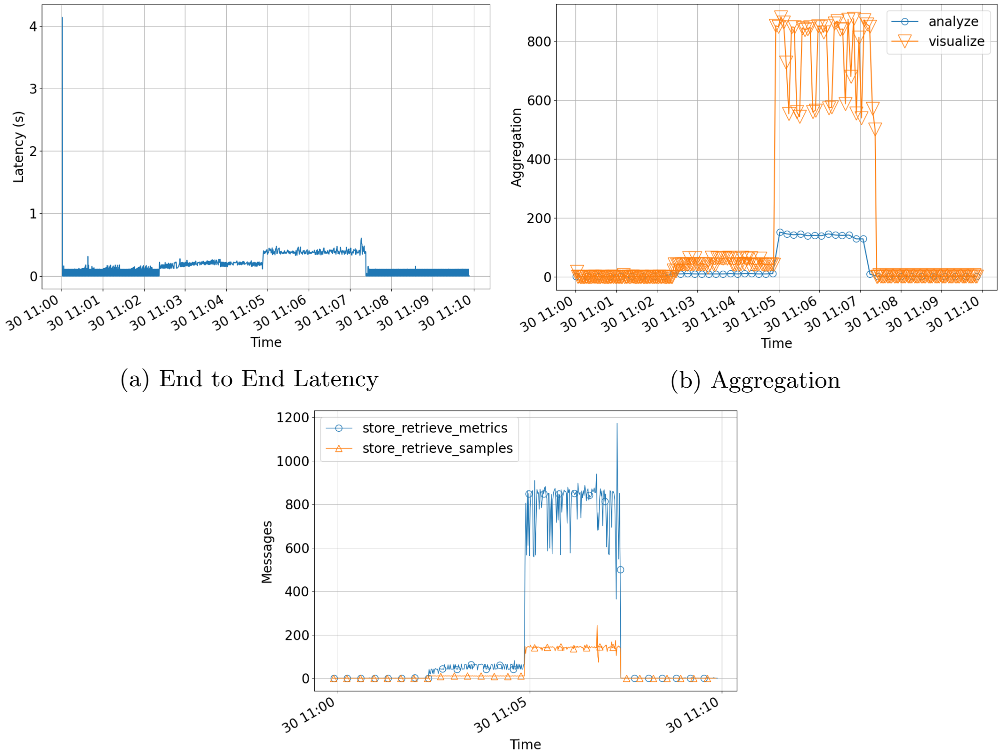
3.5.2 Support For A Variety Of Sensor Types

An important consideration when designing PIMAP was to allow for various types of sensors. If we only support one type of sensor the system does not provide value to the majority of the community and as discussed in Section 3.1 the literature tends to provide such one off systems, which makes it difficult to reuse existing implementations. To support a large variety of types of sensors we provide very loose guidelines on what type of data can be sent in a PIMAP-sample. So far in our development we have not encountered data that we cannot encode in a PIMAP-sample or PIMAP-metric.

To reiterate the sensor information of a PIMAP-sample is stored in the *sample* field. In our implementation this field is a string representation of a dictionary. For example pressure bandage data with no pressure applied, used in Objective Mobility analysis [76], can be converted to a string and will look as follows:

”sample: {’pressure_bandage’:’[[0,0,0,0],[0,0,0,0], [0,0,0,0],[0,0,0,0]]’, ’pressure_bandage_units’:’mmHg’}”

To unpack the sample one can use the built-in `ast` (Abstract Syntax Trees) Python library that can convert a syntactical grammar into its corresponding type. Another way data can be



(a) End to End Latency

(b) Aggregation

(c) Retrieved Messages

Figure 3.5: Adaptive Parameters In PIMAP

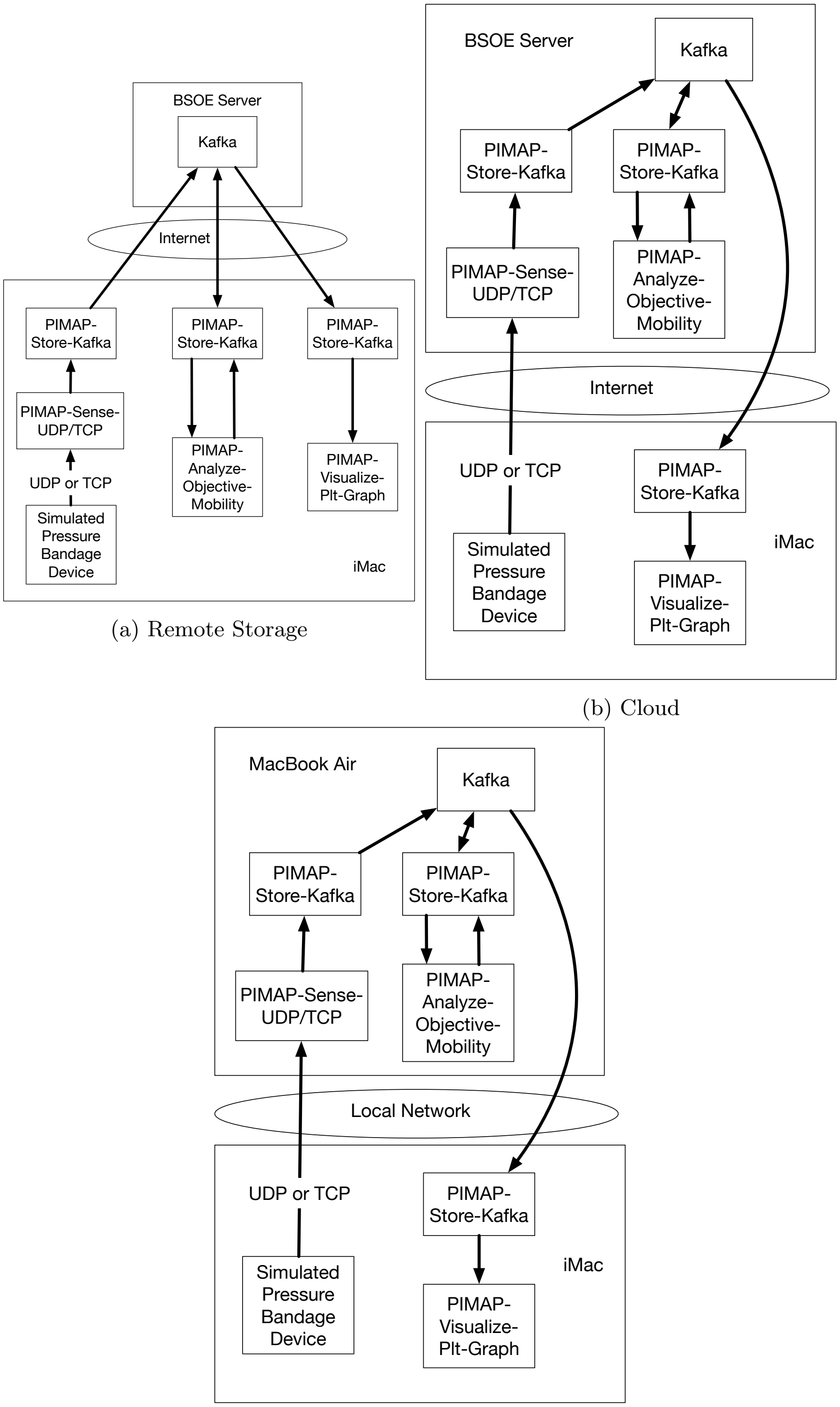
passed and converted is using the builtin `pickle` Python library, by pickling data when inserting as a *sample* and unpickling when reading the PIMAP-sample and analyzing.

To demonstrate PIMAP's ability to incorporate new sensors we setup an environment in which we gathered, analyzed, and visualized the impedance spectroscopy data from a single Sentinel bandage [132], which we obtained through our collaboration with UCSF. All tests were run on a single Linux laptop and on a local network. The Sentinel bandage generating impedance spectroscopy data was plugged into the laptop via USB and a custom PIMAP-Sense component was used to read the serial data and convert it to a PIMAP-sample. The PIMAP-samples were analyzed using a custom PIMAP-Analyze component that converted the data into a PIMAP-metric that can be displayed using a heat map. Finally we created a custom PIMAP-Visualize component that can display the heat map data.

3.5.3 Real-Time System Monitoring With And Without Reliability

PIMAP's design has built-in system monitoring features. We introduced profiling to test each component in isolation and we also introduce real-time monitoring through system-samples, which are PIMAP-samples that report the throughput and latency (as well as component specific parameters) of each component. To demonstrate how system-samples can be used to assess the characteristics of a PIMAP configuration we run PIMAP in three configurations, seen visually in Figure 3.6. We setup each configuration using an example scenario that has one pressure bandage device that sends fabricated PIMAP-sample pressure bandage data to PIMAP, which then analyzes the data using Objective Mobility analysis and the subsequent PIMAP-metrics created are visualized. The pressure bandage device sends at three data rates (1 sample/s, 100 samples/s, and 1,000 samples/s) at specified points in time to stress the PIMAP configuration. In addition we run each configuration in a UDP mode and a TCP mode to demonstrate the tradeoffs.

We name one configuration Remote Storage (Figure 3.6a), which has all PIMAP workflows running on one computer, but the storage technology, in this case Kafka, is running remotely. This configuration was named Remote Storage to approximate a PIMAP system that does the primary workload on one computer, but the data is stored remotely so that others can access it as well. We name another configuration Cloud (Figure 3.6b), which has the retrieve and visualize workflow running on one computer and the sense and store as well as the retrieve, analyze, and



(a) Remote Storage

(b) Cloud

(c) Edge

Figure 3.6: Configurations

Table 3.3: Objective Mobility Profile

Location Run	Analyze OM In w/ Movements Per Minute	Analyze OM Out w/ Movements Per Minute	Analyze OM In w/o Movements Per Minute	Analyze OM Out w/o Movements Per Minute
iMac 2010	45.25	101.9	3,993	7,985
UCSC BSOE Server	47.43	113.4	5,846	11,690
MacBook Air 2011	29.04	69.53	4,008	8,005
	PIMAP-samples/metrics a second			

store workflows running on a remote server. This configuration was named Cloud to approximate a PIMAP system that does the primary workload in the cloud. Our final configuration is named Edge (Figure 3.6c), which has the retrieve and visualize workflow running on one computer and the sense and store as well as the retrieve, analyze, and store workflows running on a separate computer on the same WiFi network. This configuration was named Edge to approximate a PIMAP system that uses edge or fog computing, where resources are placed closer than the cloud-based equivalent. The Edge and Cloud configurations are very similar except for where in the network each is run, i.e. Cloud uses resources over the Internet and Edge uses resources over the local WiFi network.

We first assessed each component in isolation using our profiling method, this gives us an upper bound on the throughput which we used to assess whether the PIMAP setup will be achievable at a given data rate. The results can be seen in Tables 3.4, 3.5, and 3.6. In order to demonstrate PIMAP with a data rate of 1,000 samples/s we disabled the Movements Per Minute calculation inside of the Objective Mobility analysis. This can be seen in Table 3.3, where the maximum incoming throughput with the Movements Per Minute calculation never exceeds 47.43, whereas without this calculation we can achieve close to 4,000 samples/s of incoming throughput.

If we examine the profiling results in Tables 3.4, 3.5, and 3.6 the limiting throughput to send UDP data is the Sense component, but when sending TCP data the limiting component is the Analyze component. The UDP upper bound is roughly 2,000-4,000 samples/s. The TCP upper bound is roughly 4,000-6,000 samples/s. As these numbers are upper bounds we chose a safe maximum data rate of 1,000 sample/s.

Table 3.4: Remote Storage Profile

Location Run	Sense UDP	Sense TCP	Analyze OM In	Analyze OM Out	Store Kafka (BSOE Server)	Retrieve Kafka (BSOE Server)	Visualize Plt Graph
iMac 2010	3,346	12,710	3,993	7,985	18,980	19,320	20,390
	PIMAP-samples/metrics a second						

Table 3.5: Cloud Profile

Location Run	Sense UDP	Sense TCP	Analyze OM In	Analyze OM Out	Store Kafka (BSOE Server)	Retrieve Kafka (BSOE Server)	Visualize Plt Graph
iMac 2010	N/A	N/A	N/A	N/A	18,980	19,320	20,390
UCSC BSOE Server	4,255	8,761	5,846	11,690	24,350	12,750	N/A
	PIMAP-samples/metrics a second						

Table 3.6: Edge Profile

Location Run	Sense UDP	Sense TCP	Analyze OM In	Analyze OM Out	Store Kafka (Air)	Retrieve Kafka (Air)	Visualize Plt Graph
iMac 2010	N/A	N/A	N/A	N/A	17,200	30,770	20,390
MacBook Air	2,214	9,166	4,008	8,005	9,288	51,080	N/A
	PIMAP-samples/metrics a second						

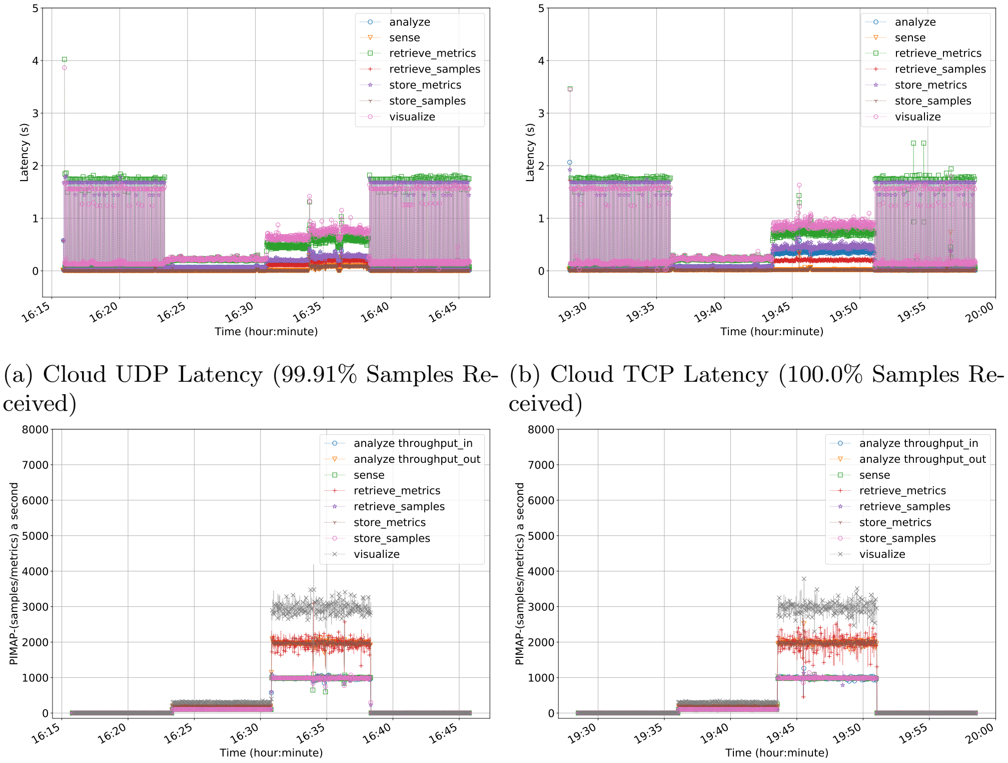
The User Datagram Protocol (UDP) [103] and the Transmission Control Protocol (TCP) [104] are two of the core Internet protocols. In brief, one of the core differences is that UDP is connectionless and therefore has less overhead, but at a cost of data loss. On the other hand TCP has reliability, but at a cost of higher overhead to establish a connection and detect data loss. To further examine these differences as a proof of concept we use the system monitoring capabilities of PIMAP to compare the three configurations: Remote Storage, Cloud, and Edge when using UDP versus TCP. We report here examples of different behaviour we discovered that could be used in future work to be detected and used internally to tune the system. We could also potentially use machine learning to learn how to achieve a low latency by training on the monitoring data.

Each graph reported is a thirty minute run of PIMAP in the given configuration as pictured in 3.6 and the data rate progresses from 1 sample/s for 7.5 minutes, 100 samples/s for 7.5 minutes, 1,000 samples/s, and 1 sample/s for the remaining 7.5 minutes. We will present various runs and explain how the phenomenon that occurred while monitoring can be used in future work.

The following are examples of somewhat stable behaviours we discovered as can be seen in Figures 3.7 and 3.8. We say the behaviour is stable in the given runs as the reported latencies and throughputs are more or less flat across the length of the experiment except at the intervals where the data changed.

In both the Cloud and Edge stable configurations when sending using UDP the latencies and throughputs are very similar with the primary difference being that in the Cloud configuration during the data rate of 1,000 samples/s there is a slight increase in the latency. The samples received percentages 99.91% for Cloud and 99.65% for Edge are also very similar. The main takeaway we found when using UDP in a stable configuration is that the end to end latency changes based on the data rate. This is as expected as we are visualizing two different metrics, one metric that has a one-to-one sample to metric ratio and another metric that has a five-to-one sample to metric ratio, which is why we see when the data rate increases from 1 sample/s to 100 samples/s the end-to-end latency decreases as it takes less time to calculate the five-to-one sample to metrics. But, when the sample rate increases from 100 samples/s to 1,000 samples/s as seen in Figure 3.9a every component's latency increases except for sensing and storing samples, which in turn increases the end-to-end latency.

Similarly in both the Cloud and Edge stable configurations when sending using TCP the



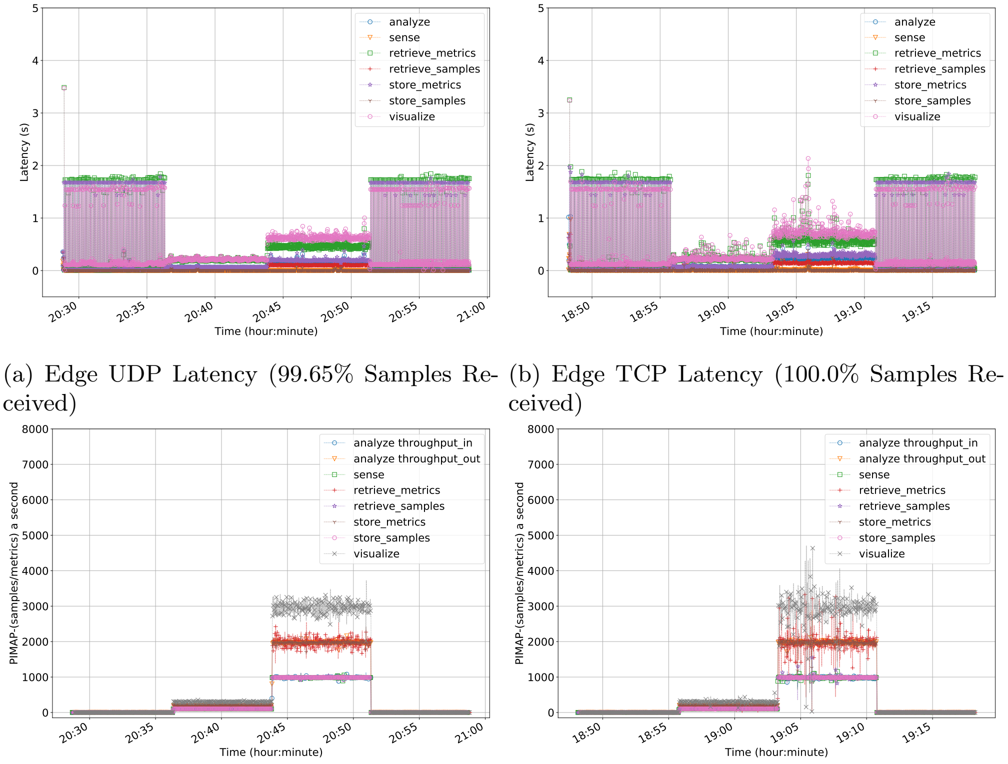
(a) Cloud UDP Latency (99.91% Samples Received)

(b) Cloud TCP Latency (100.0% Samples Received)

(c) Cloud UDP Throughput (99.91% Samples Received)

(d) Cloud TCP Throughput (100.0% Samples Received)

Figure 3.7: Cloud Stable Behaviour



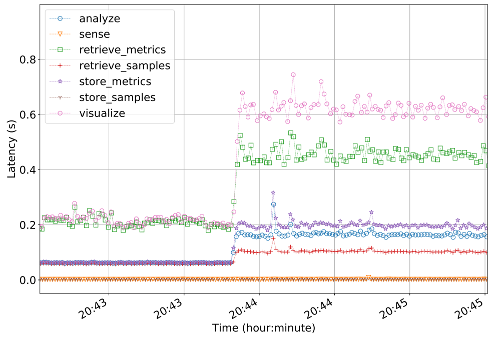
(a) Edge UDP Latency (99.65% Samples Received)

(b) Edge TCP Latency (100.0% Samples Received)

(c) Edge UDP Throughput (99.65% Samples Received)

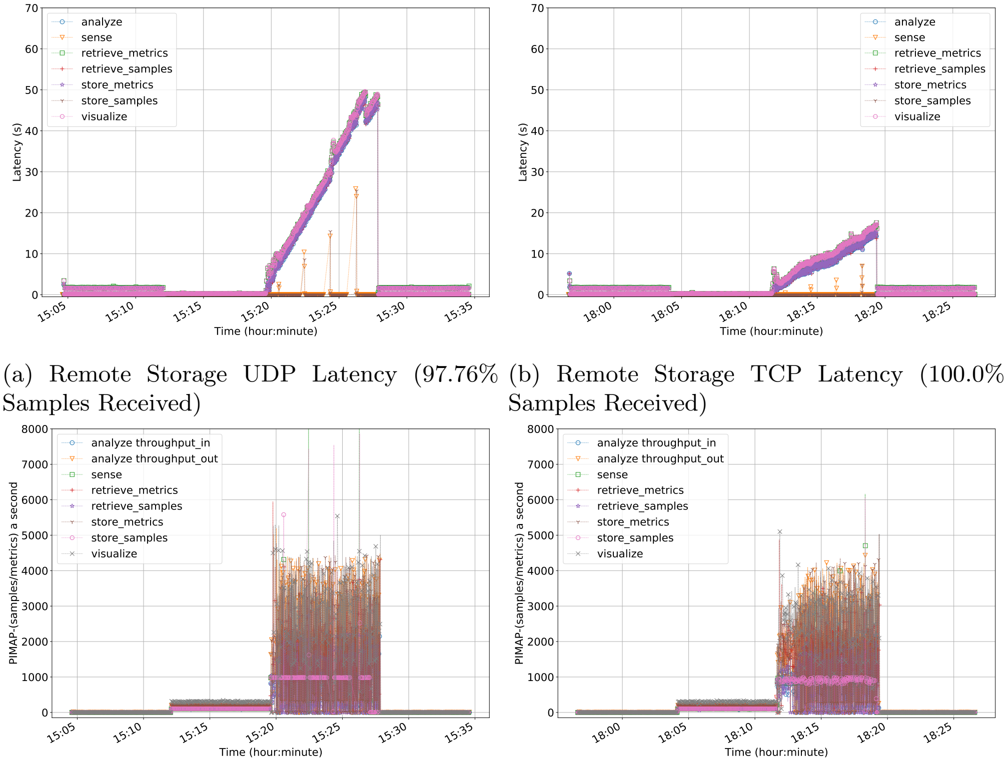
(d) Edge TCP Throughput (100.0% Samples Received)

Figure 3.8: Edge Stable Behaviour



(a) Edge UDP Latency Transition 100 Samples/s to 1,000 Samples/s

Figure 3.9: Stable Behaviour Transitions



(a) Remote Storage UDP Latency (97.76% Samples Received)

(b) Remote Storage TCP Latency (100.0% Samples Received)

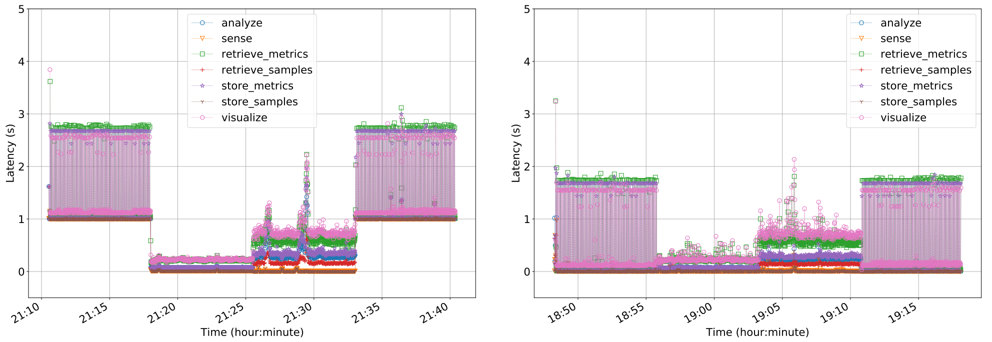
(c) Remote Storage UDP Throughput (97.76% Samples Received)

(d) Remote Storage TCP Throughput (100.0% Samples Received)

Figure 3.10: Remote Storage Unstable Behaviour

latencies and throughputs are very similar except for some occasional blips. In both configurations we received 100.0% of samples sent and we see the same phenomenon that the end-to-end latency decreases from a data rate of 1 sample/s to 100 samples/s and then the end-to-end-latency increases from a data rate of 100 samples/s to 1,000 samples/s and this is from the same phenomenon as in the UDP case because of the five-to-one sample to metric ratio and the increase in latency with the increased sample rate respectively.

During the intial experimental runs we noticed that the TCP configurations all had a 1s delay at the PIMAP-Sense component that was not present in the UDP configurations. Upon further investigation we found that it actually is a one sample period delay, e.g. if data is arriving at 1 sample/s there is a 1s delay and if data is arriving at 2 samples/s there is a 0.5s delay. We further investigated this phenomenon as this is not a feature of TCP and discovered that it was actually a bug in our implementation of PIMAP-Sense-TCP. We are not advocating that the self-monitoring features of PIMAP be used as a debugging tool, but these features did allow



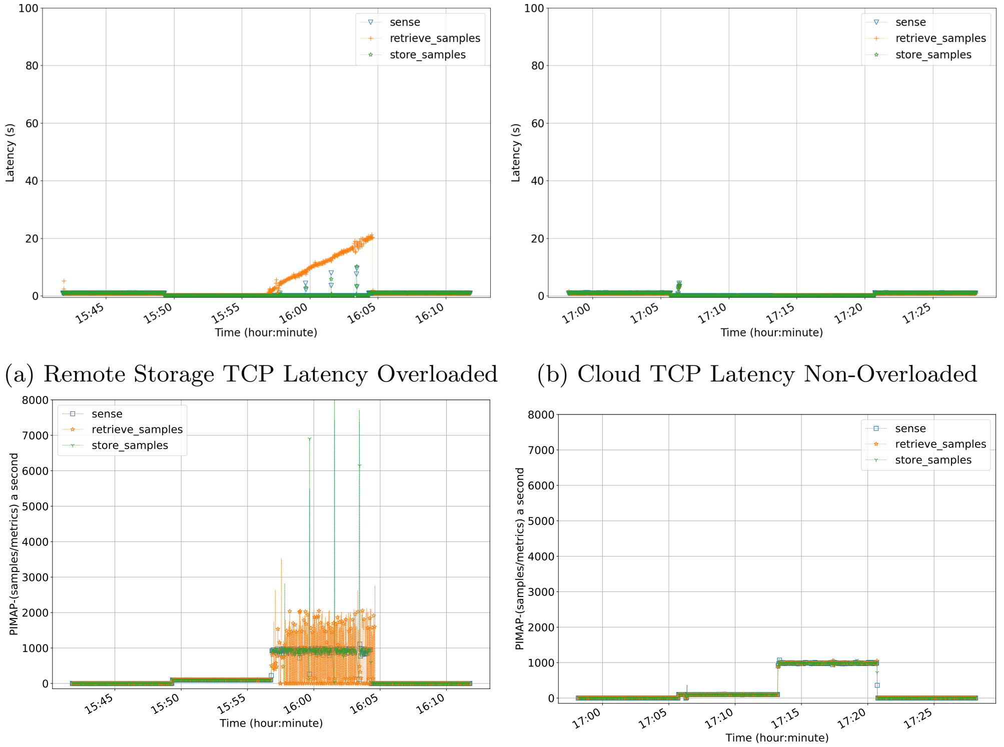
(a) Latency With PIMAP-Sense-TCP Bug (b) Latency Without PIMAP-Sense-TCP Bug

Figure 3.11: Sample Period Delay TCP Bug

us to catch a somewhat sinister bug that otherwise would probably have gone undetected as the overall application was still functional and at higher sample rates was almost unobservable becuase of low delay. In Figure 3.11 we show two experimental runs, one of which still has the sample period delay bug and the other run where the sample period delay bug is fixed.

We also discovered unstable behaviours and in fact in every run of the Remote Storage configuration when the data rate was 1,000 samples/s (15 minutes - 22.5 minutes) the latency steadily increases until the data rate goes back to 1 sample/s as seen in Figure 3.10. We are still investigating the exact cause of this behaviour, but we have narrowed it down to the retrieval of samples when analyzing the data. The retrieval appears to get overloaded and cannot catch up to the incoming data. We can see this in Figure 3.12, where we are only displaying the latency and throughput of the sense, store samples, and retrieve samples, which are the first three phases of PIMAP. In the non-overloaded configuration we see that the throughputs stay at 1,000 samples/s as expected, but in the overloaded case retrieve samples bounces between 0-2,000 samples/s, and the latency at this point runs away. The main difference between the Remote Storage configuration and the Cloud or Edge configuration is that all PIMAP components are being run on the same computer and in this case the computer is not able to maintain stability at a data rate of 1,000 samples/s.

In the Cloud configuration we found that the load on the server can have an impact on latency and throughput, which is as expected. We deployed an additional PIMAP-Sense-TCP and Store instance on the UCSC BSOE server we were using to approximate our cloud server and found when using UDP the behaviour to be less stable, but still within a normal range, but



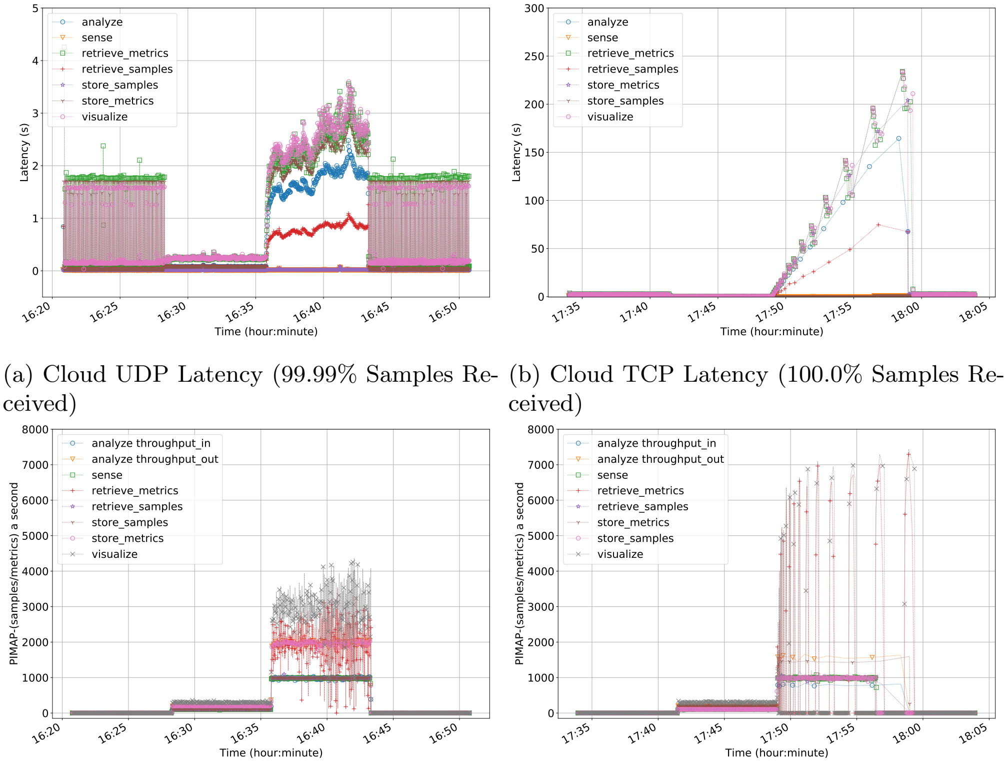
(a) Remote Storage TCP Latency Overloaded

(b) Cloud TCP Latency Non-Overloaded

(c) Remote Storage TCP Throughput Overloaded

(d) Cloud TCP Throughput Non-Overloaded

Figure 3.12: Remote Storage Overloaded Comparison



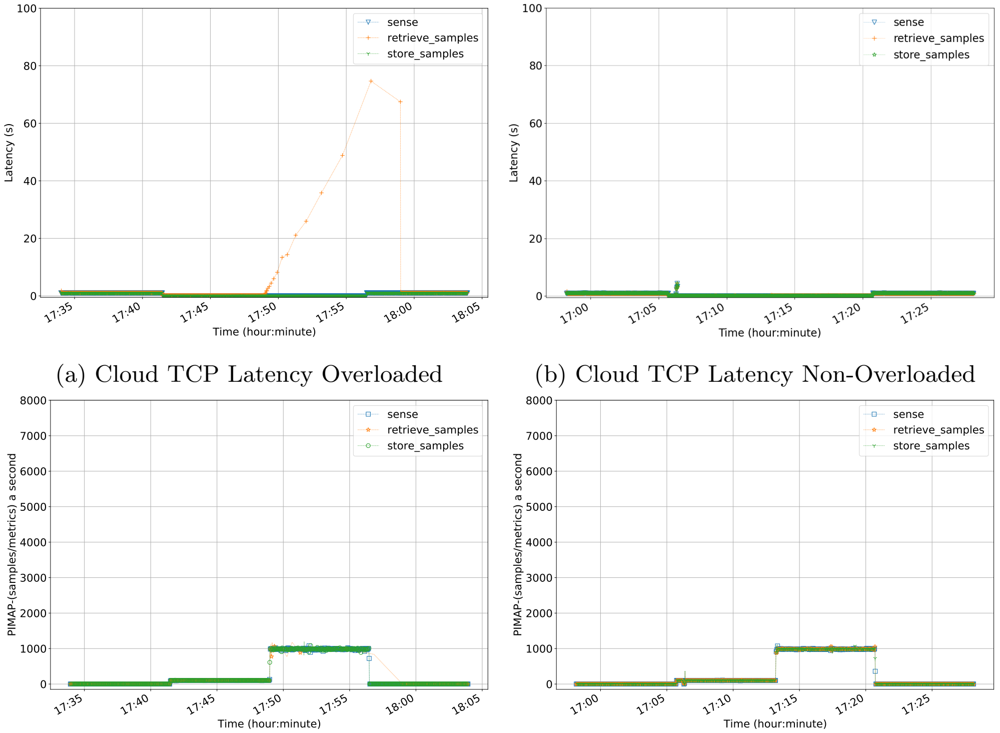
(a) Cloud UDP Latency (99.99% Samples Received)

(b) Cloud TCP Latency (100.0% Samples Received)

(c) Cloud UDP Throughput (99.99% Samples Received)

(d) Cloud TCP Throughput (100.0% Samples Received)

Figure 3.13: Cloud Unstable Behaviour



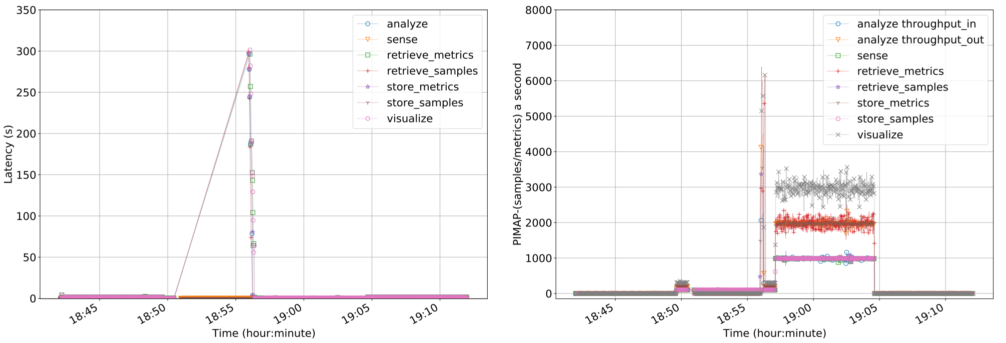
(a) Cloud TCP Latency Overloaded

(b) Cloud TCP Latency Non-Overloaded

(c) Cloud TCP Throughput Overloaded

(d) Cloud TCP Throughput Non-Overloaded

Figure 3.14: Cloud Overloaded Comparison



(a) Edge UDP Latency (98.99% Samples Received)

(b) Edge UDP Throughput (98.99% Samples Received)

Figure 3.15: Edge Kafka Restart

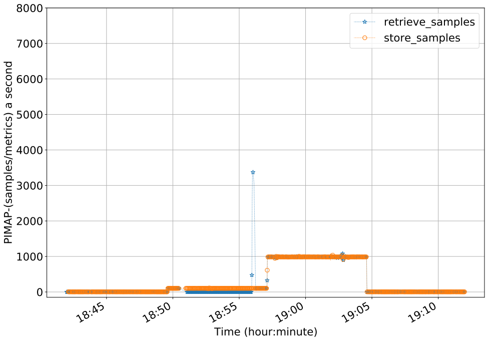


Figure 3.16: Edge UDP Stored/Retrieved Throughput During Kafka Restart

when using TCP the latency increases greatly when the data rate switches to 1,000 samples/s as can be seen in Figure 3.13. The UCSC BSOE server appears to get overloaded, but as opposed to the previous case where the retrieve samples throughput bounces between 0-2,000 samples/s in this case the retrieve samples throughput is stable at 1,000 samples/s, but even so the retrieve samples latency increases as seen in Figure 3.14, where we show the latency and throughput of sense, store samples, and retrieve samples. It is also interesting to note that even though we added additional TCP servers to the cloud server it is not the Sense TCP component that gets overloaded as we can see from monitoring that sense's latency and throughput are stable. An unmonitored patient monitoring system that are typically used would not be able to differentiate.

We also monitored a case where the Kafka instance crashed and was restarted when using the Edge configuration with UDP. The PIMAP application did recover, but we see the latency and throughput greatly increase as it recovers as seen in Figure 3.15. To detect this phenomenon we can use a combination of the stored and retrieved throughput as seen in Figure 3.16. If we are storing at a given throughput (the storage throughput reflects that data was added to the Kafka client's storage queue, which is why we still see throughput even though the Kafka instance is down), but receiving zero throughput we can assume that the Kafka instance is down and needs to be restarted.

Through these various 30 minute runs of PIMAP we observed stable behavior as well as anomalous behavior that we can use to make PIMAP a self-tuning patient monitoring system that was enabled by PIMAP's ability to self-monitor.

3.5.4 Real-Time Analysis Of Simulated Networks

A well known problem in sensor network deployment is testing whether the deployment can reliably send and receive data. To deploy even as few as ten physical devices and test whether data can be collected is time consuming and difficult.

To address this problem several sensor network simulators were developed. A well regarded simulator is COOJA [97], which simulates ContikiOS nodes. ContikiOS [25] is an operating system designed for low-power low-resource devices to connect to the Internet.

Because this is a well known problem we see the value in connecting PIMAP to COOJA in order to simulate various topologies and sensor device scenarios, so that future researchers can leverage COOJA and use simulated networks with PIMAP.

From our own experimentation we were not able to force COOJA to run in real-time. The best possible is enforcing a 100% speed limit on the simulation, but this still allows for the simulation to run slower than real-time. Because of this we only analyze the performance once data has entered PIMAP as the time given by the simulated sensor node's clock is not stable.

As a proof of concept in COOJA we setup a network with one border router and ten sensor devices that are all one hop from the border router. All sensor devices are using CSMA at the MAC layer, the standard contention based MAC protocol and RPL at the routing layer. The border router is the sink of the topology meaning all nodes will send packets to the border router if they do not have a route to the destined address.

COOJA is running inside a Docker container and PIMAP is running on the host machine. There is a tool part of the ContikiOS distribution called tunslip that connects to the border router and forwards packets onto the host network.

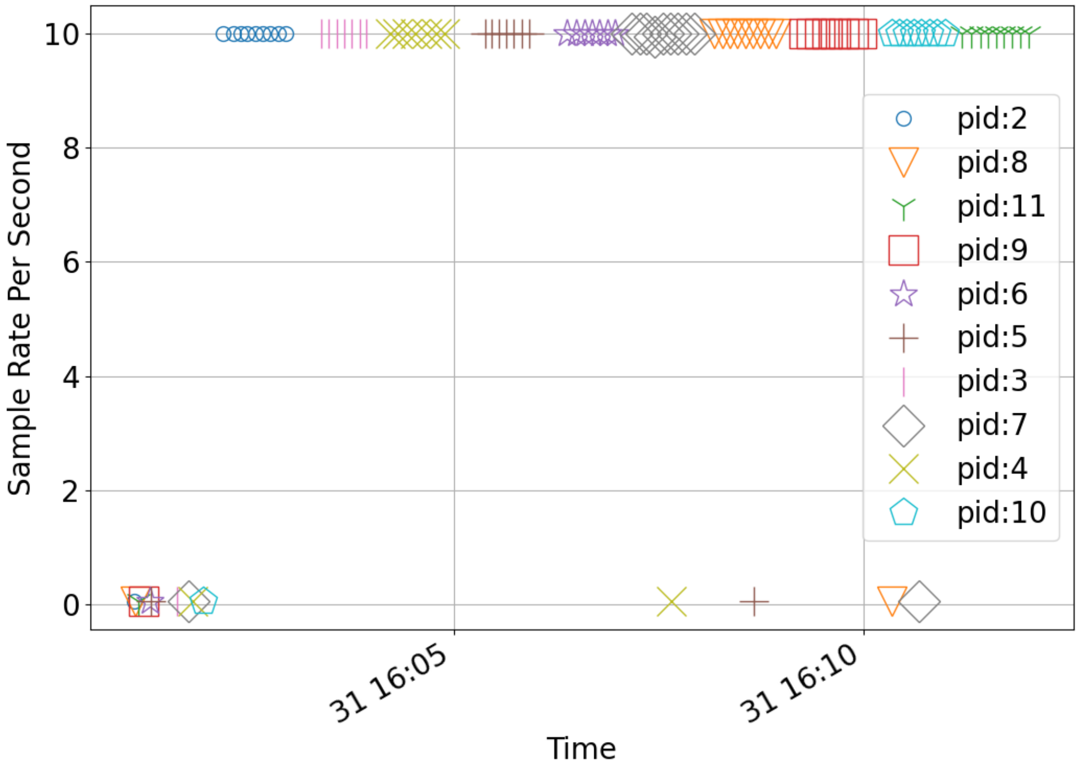


Figure 3.17: Round Robin Scheduler Using PIMAP With A Simulated COOJA Network

We perform an application layer round robin schedule to demonstrate the ability to use PIMAP to interact with COOJA and control the sample rate of the nodes. Initially all nodes sampling rates are set at 0.05 samples/s (one sample sent every 20s). We first let the simulation run for two minutes. After these two minutes we send a PIMAP-command to the lowest *patient_id* to change its sampling rate to 10 samples/s and all other nodes to set their sampling rate to 0.05 samples/s. We then wait a minute and repeat using the next lowest *patient_id*. This is an implementation of an application layer round robin scheduler.

The results can be seen in Figure 3.17. We use PIMAP-Analyze-Sample-Rate, which is an inferred sample rate analysis calculated by waiting until we have at least five PIMAP-samples, using the *timestamps* of the PIMAP-samples, which in this case is the COOJA time (time starts at zero when the node starts), calculating the gradient of the timestamps and averaging the reciprocal of the gradient.

3.6 Privacy And Security

Privacy and security are obviously very important issues to consider when designing any software system and especially when it comes to transferring and/or storing health information. Sensor information by itself is a bit of a gray area as the information by itself is not self-identifiable to a patient, but in the future it may be.

The current implementation of PIMAP as of this writing has not handled any identifiable information to a patient, meaning we as the developers and researchers do not have access to any information that is identifiable to a patient, but we expect in the future we will have to handle private information that must be stored and transferred securely and therefore intend to support all measures. Privacy and Security in regards to PIMAP will be further discussed in Chapter 5.

3.7 Limitations Of PIMAP

PIMAP may not be the best-fit patient monitoring system for all applications. PIMAP assumes that an application can be broken into the four components: sense, store, analyze, and visualize. PIMAP also assumes that streams of data are going through the system. For example PIMAP may waste resources if an application generates large amounts of data at sparse intervals in time, although PIMAP could still run in this configuration. In addition PIMAP is implemented in Python and assumes that a researcher has enough programming experience to create a new component based on an existing example. For example if a medical researcher wants to implement a new PIMAP-Analyze component they would need to look at previous PIMAP-Analyze components or an example PIMAP-Analyze component and know how to make the appropriate changes to incorporate the intended analysis. In fact when incorporating a novel sensor into PIMAP the PIMAP-Analyze component is often going to be custom as novel sensor-based analytics are heavily dependent on the novel sensor. Finally, because PIMAP abstracts different technologies (such as Kafka) and can be run in distributed configurations it does take someone with a good understanding of networking to appropriately setup a distributed PIMAP configuration. But, in the future this is a service that could potentially be commercialized for those who do not have experience to setup a distributed PIMAP configuration on on their own, but still want to use PIMAP.

Most of the experimentation and data used to test and improve PIMAP are on the order of hours of data. But, eventually PIMAP could be used in applications with years of data. There are different requirements when working with data at this scale and the current development of PIMAP has not had to deal with this order of data. This is not a limitation of PIMAP, but instead a future development of PIMAP, primarily around data navigation. With years of data even at a relatively low sample rate of one sample a second one cannot simply traverse through all the data without large delays in time, which would make PIMAP difficult to use. Instead PIMAP would need data traversal to jump through the data to find/analyze data one is looking for.

3.8 Future Work And Conclusion

We will continue to develop the PIMAP system to integrate new sensing, storage, analytics, and visualization technologies and compare and contrast when it is beneficial to use a specific technology. We are also working towards using PIMAP in a clinical trial at UCSF to monitor pressure injuries and study how care changes when visualizations are presented to healthcare workers. Our code will be released open source on Github at github.com/pimap, where additional documentation as well as numerous examples will be presented. In addition we are continuing to develop user interface tools to make PIMAP deployment both easy and secure.

In this work we present PIMAP a system framework to *Sense*, *Store*, *Analyze*, and *Visualize* patient data. PIMAP is designed so that it is not dependent on any specific technology, but instead can integrate new technologies. PIMAP is motivated by the lack of a patient monitoring framework for medical researchers to test novel sensors and novel algorithms and for clinicians to view novel sensor data and metrics in real-time. We demonstrated PIMAP's potential with real patient data to simulate a real-time scenario and present risk stratification and mobility metrics that can be presented to clinicians. We demonstrated PIMAP has low latency in both low throughput and high throughput scenarios, PIMAP is able to accommodate new sensor types, and PIMAP can integrate with a sensor network simulator to test a configuration of sensors before it is deployed. PIMAP is released as open source and we will continue to study and improve the system.

Chapter 4

Pressure Injury Risk Assessment Using A Wearable Pressure Sensor And Wearable Skin Impedance Sensor

In this chapter we discuss two USCF collaborations, one of which we generate objective analytics from a novel wearable pressure-sensing device that can be calculated and displayed to healthcare staff in real-time and the second of which we discuss integrating a novel wearable skin impdeance device. The proposed systems can be used to determine pressure injury risk of a patient in real-time. We will compare and contrast our work with current patient monitoring approaches to reducing pressure injuries in Section 4.1. We will discuss the design of the wearable pressure-sensing device and the experiment design in Section 4.2. In Section 4.3 we will discuss how we analyzed the wearable pressure sensor data to construct a best-fit plane. In Section 4.4 we discuss how we used the best-fit plane to assess the mobility of a patient. In Section 4.5 we discuss how we used the best-fit plane to assess the posture of the patient. In Section 4.6 we discuss how we applied the analysis in a real-time system. In Section 4.8 we discuss an additional collaboration with UCSF integrating a skin impedance bandage into PIMAP. We discuss future work and conclude this chatper in Section 4.9.

4.1 Related Work

Recent technologies are currently being developed to prevent pressure injuries. Some of these technologies could be used in conjunction with our sensing and others offer different trade offs. The are two current commercialized sensing strategies: Continuous Bedside Pressure Mapping (CBPM) and Inertial Measurement Unit (IMU) monitoring. Two other very promising strategies are limb-identification algorithms that can track the amount of pressure per limb as the patient rotates and skin integrity monitoring.

CBPM uses a pressure sensing mattress overlay to monitor the real-time interface pressure of a patient on a mattress. In this way healthcare staff can use the real-time visual to assess the repositioning of the patient. Two controlled trials have been performed using CBPM and a lower incidence of pressure injuries was reported [126] [7], but a Randomized Controlled Trial showed no reduction in pressure injuries using CBPM [43]. This system may help with repositioning, but the current software does not assess the risk of the patient meaning healthcare staff would still spend the same amount of time per patient making it not sustainable. These system can also be expensive because of the number of sensors required to make the pressure sensing mattress overlays.

IMU monitoring uses accelerometers, gyroscopes, and sometimes magnetometers to track the movement and orientation of a patient. IMU monitoring was clinically tested in a Randomized Controlled Trial and found that turning compliance, the reliability that healthcare staff turn a patient at a designated periodic time, increased [100]. This sensing strategy is inexpensive and could possibly be used to stratify patients based on risk using the level of mobility of a patient. The advantage of using a wearable pressure sensor is the same mobility can also be tracked and in addition pressure accumulation can also be tracked as it is established that it is pressure over time that correlates to pressure injuries [116].

A software-based approach using the same pressure sensing mattress overlay as in CBPM can be used to identify the limbs of a patient making it possible to track the amount of pressure per body part over time [32] [96] [68] [106] [108]. This solution offers the highest granularity of risk as each individual limb can be assessed, but the pressure overlay is expensive and how this data would be used in a clinical setting has not yet been evaluated. A wearable pressure sensor offers a subset of the abilities as pressure limb-tracking as individual areas of the body can be tracked where a wearable is placed at a much lower cost.

Skin integrity monitoring aims to determine the health of the skin, which can be used to detect the formation of a pressure injury or track the healing progress of a pressure injury. A hand-held scanner was developed and tested [82] as well as two bandages [132] [31]. This type of monitoring offers the benefit of assessing skin health, but it is limited to the area being monitored and currently does not offer a way to stratify patients based on the risk of forming a pressure injury. A wearable pressure sensor can offer stratification of patients and in addition monitor the pressure at a location.

There are also many other types of monitoring that can be used assess pressure injury such as temperature and humidity, blood flow, biomarker, electrocardiography, camera, ultrasound, and others. But, these works currently are too preliminary to compare and contrast with a wearable pressure sensor.

4.2 Wearable Pressure-Sensing Device Design And Data Collection

The wearable pressure-sensing device was developed by Dr. Lee's group at UCSF as a way to continuously measure interface pressure from a patient to objectively assess the risk a patient has of forming a pressure injury. Interface pressure was measured using a wearable pressure sensing array placed between a Mepilex Border Sacrum (Mölnlycke Health Care, Gothenburg, Sweden) adhesive wound dressing and Tegaderm (3M, Maplewood, USA) transparent film dressing. The sensing array consisted of circular (1cm diameter) flexible piezoresistive pressure sensors (Micro Deformable Piezoresistive "Uneo" sensors; Uneo Inc., New Taipei City, Taiwan) placed in a 4x4 array with 1 cm spacing between each cell. These 16 cells were connected using 8 traces (4 vertical, 4 horizontal), which were routed through a 30 cm flexible cable to a printed circuit board. The PCB (Printed Circuit Board) consisted of a voltage divider circuit with fixed 10k ohm resistors in series with the variable pressure sensor resistors. The change in pressure sensors' resistance was measured using a microcontroller and Bluetooth transmitter/receiver chip (BLE 112 module; Silicon Labs, Austin, USA), which scanned and transmitted (frequency = 1 Hertz)

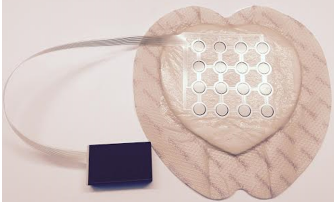
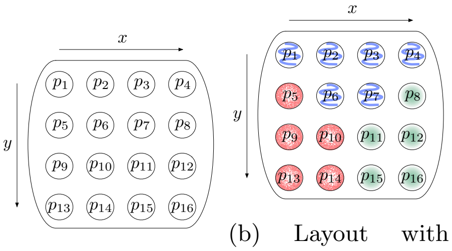


Figure 4.1: Wearable Pressure Sensor


x
y
p_1 p_2 p_3 p_4
p_5 p_6 p_7 p_8
p_9 p_{10} p_{11} p_{12}
p_{13} p_{14} p_{15} p_{16}
x
y
p_1 p_2 p_3 p_4
p_5 p_6 p_7 p_8
p_9 p_{10} p_{11} p_{12}
p_{13} p_{14} p_{15} p_{16}

(a) Layout

(b) Layout with color-coded coordinates

Figure 4.2: Pressure Sensor Layout

the measurements to an iPad mini 2 (Apple Inc., Cupertino, USA) running an application written for this study. Figure 4.1 shows the actual wearable pressure sensor used in the study.

A patient is consented and enrolled in the study. The patient's skin is observed for lesions, baseline documentation is performed, and the pressure sensitive wound dressing is placed on the sacrum of the patient following standard Mepilex application procedures. The end of the flexible flat cable is plugged into the electronics box and a coin cell battery is placed into the electronics box. The study application is selected on the iPad and the patient's study number is entered into the application. Once the patient's number is entered into the application, the iPad connects to the BLE112 and commences data collection. Data is collected until the dressing is changed (during which data is not collected, but after which data collection resumes), the patient leaves the ICU (Intensive Care Unit), or the patient is disenrolled from the study. There were a total of five patients enrolled in the study.

4.3 Fitting A Pressure Plane

At every reading of the sensor we obtain one pressure reading from each pressure location for a total of sixteen pressure readings, where the relative position of each pressure location is known as depicted in Figure 4.2. Our goal is to interpret this data to infer the mobility and posture of the patient. We have found a good way to do this is to form a best-fit linear plane to the sensors pressure readings and then use the planar characteristics to infer movement and posture.

For the purpose of discussion we will define a *cell* as a triple with the three values x, y, p, where x is the horizontal location, y is the vertical location, and p is the pressure reading at that x, y location.

We assign an x, y location at each cell and normalize the three values x, y, p. x and y are normalized on assignment such that the highest value of x and y is 1. The pressure values are normalized such that the maximum possible pressure value is 1. When we use the variables x, y, p in this paper we are using the normalized values. As an example the four corner cells have values of:

$$cell_1 = (0, 0, p_1), cell_4 = (1, 0, p_4),$$
$$cell_{13} = (0, 1, p_{13}), cell_{16} = (1, 1, p_{16})$$

Given any three points we can form a plane, so our next task is to determine how to combine the cells to from three cells that represent the data. The way we chose to do this is to average the x, y, p values in the following manner:

$$threecell_1 = average(cell_1, cell_2, cell_3, cell_4, cell_6, cell_7)$$
$$threecell_2 = average(cell_5, cell_9, cell_{10}, cell_{13}, cell_{14})$$
$$threecell_3 = average(cell_8, cell_{11}, cell_{12}, cell_{15}, cell_{16})$$

The cells chosen for each *threecell* is based on their spatial locality. The averaging of the cells helps to form a best-fit plane, but in addition it helps to eliminate some of the noise captured by the pressure sensors. Figure 4.2 depicts a pictorial presentation of the grouping.

From *threecell* we can form a plane by finding the normal to the plane by performing the cross product of two vectors in the plane. Using the normal and a point on the plane we can then solve for the final value in the planar equation. The following is the general planar equation and how the normal relates to the general planar equation.

$$ax + by + cz = d$$
$$n = (a, b, c)$$

A calculation to find the complete planar equation using *threecell* is as follows:

$$\begin{aligned}
\vec{v}_{12} &= threecell_2 - threecell_1 \\
\vec{v}_{13} &= threecell_3 - threecell_1 \\
n &= \vec{v}_{12} \times \vec{v}_{13} \\
d &= threecell_1 n^T
\end{aligned}$$

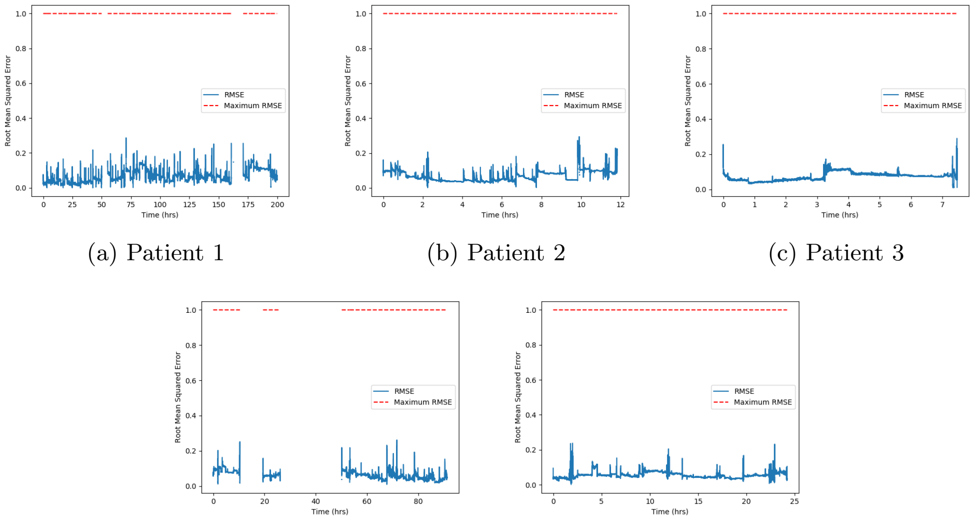
From the plane we are interested in two characteristics, the x_{slope} and y_{slope}. Although we could use the x and y slope directly for our mobility and posture analysis in the later sections we instead use a new metric related to the x and y slope we call the x_{angle} and y_{angle}, which are related to the slope as follows:

$$\begin{aligned}
x_{slope} &= -a/c \\
y_{slope} &= -b/c \\
x_{angle} &= \frac{360}{2\pi} \arctan(x_{slope}) \qquad (4.1)\\
y_{angle} &= \frac{360}{2\pi} \arctan(y_{slope}) \qquad (4.2)
\end{aligned}$$

The x_{angle} and y_{angle} give us a more intuitive way to think about the rotation of the patient, although it is important to keep in mind that we do not relate the degree of these metrics to an actual degree of rotation the patient is experiencing, but instead use it as a relative metric.

We verify our best-fit plane by calculating the Root Mean Squared Error. Our results can be seen in Figure 4.3. Each experiment lasted for a different amount of time and therefore the graph time scales are not equivalent. Also a gap in data means the sensor was disconnected for that period of time.

We see our best-fit plane is close to the actual values, but we also do not expect the Root



(a) Patient 1

(b) Patient 2

(c) Patient 3

(d) Patient 4

(e) Patient 5

Figure 4.3: Root Mean Squared Error Of Patient Data

Mean Squared Error to be extremely small as we are approximating the sensor information as a linear plane, which is not always accurate as current pressure values may be better modelled as a quadratic, exponential, logarithmic, etc. plane and in the process of forming the best-fit linear plane we are also eliminating noise, which we do eliminate when calculating the Root Mean Squared Error.

4.4 Mobility Analysis

We are able to provide an objective mobility metric based on the sensor readings. We will describe why mobility is an important metric to monitor, the relation to the Braden Scale [13], and our methods for calculating the metric.

We use the term mobility as defined in the Braden Scale [13], the patient's ability to change body position. The mobility score can have one of four values. A one, "completely immobile", indicates the patient cannot change body position without assistance. A two, "very limited", indicates the patient can make slight changes that are not frequent or significant. A three, "slightly limited", indicates the patient can make frequent small movements. A four, "no limitations", indicates the patient can make significant changes frequently and independently. We do not use the same scores in our results, but we reproduce the mobility scale as a reference and

$$x_{gradient}(i) = \begin{cases} x_{angle}(i+1) - x_{angle}(i) & \text{if } i = 0 \\ x_{angle}(i) - x_{angle}(i-1) & \text{if } i = \text{last value} \\ (x_{angle}(i+1) - x_{angle}(i-1))/2 & \text{otherwise} \end{cases} \tag{4.3}$$

$$y_{gradient}(i) = \begin{cases} y_{angle}(i+1) - y_{angle}(i) & \text{if } i = 0 \\ y_{angle}(i) - y_{angle}(i-1) & \text{if } i = \text{last value} \\ (y_{angle}(i+1) - y_{angle}(i-1))/2 & \text{otherwise} \end{cases} \tag{4.4}$$

for future comparison.

Garcia-Fernandez et al conducted a study to determine the top risk dimensions that cause pressure injuries from an expert panel. The expert panel determined that mobility is in the top five risk dimensions that lead to pressure injuries [34]. In addition Alderden et all surveyed the literature and also identified mobility as one of the top five risk factors for pressure injuries [4].

We developed two metrics to assess mobility: movements per minute $Movements/min$ and movement strength $Movement\ Strength$. Both metrics are based on our definition of a movement. A movement is calculated by setting a threshold on both the x and y angle gradients. We calculate the gradient at location i, where i is the ith sample. In our case the data is sampled at every second and therefore i corresponds to the number of seconds. The gradient is calculated based on the x and y angles as seen in Equations 4.3 and 4.4:

The x and y gradients are then combined into a singular $xy_{gradient}$ metric that does not differentiate between positive and negative angle and the threshold of movement was defined as $xy_{gradient}(i) > 2$ from visual inspection of the data.

$$xy_{gradient}(i) = max(|x_{gradient}(i)|, |y_{gradient}(i)|) \tag{4.5}$$

In Figure 4.4 we show graphs of mobility based on test data and actual patient data. Below the legend of each graph we show the calculated metrics $Movements/min$ and $Movement\ Strength$. The metric $Movements/min$ is the number of entries in $xy_{gradient}$ that have a value larger than 2 divided by the number of minutes that have elapsed. $Movement\ Strength$ is the average of the $xy_{gradient}$ values that are greater than 2.

$$movement = \begin{cases} True & \text{if } xy_{gradient} > 2 \\ \\ False & \text{otherwise} \end{cases} \tag{4.6}$$

The data in Figure 4.4 shows the mobility of five patients. The length of each experiment varied, but for comparison we display the first 24 hours of each experiment. For certain segments of time no data is displayed, e.g. patient 2, patient 3, patient 4. This is either because the experiment was shorter than 24 hours or the pressure-sensing device was disconnected for a certain amount of time.

On each mobility graph we also mark with a red asterisk where the patient was repositioned by healthcare staff to give the reader a sense of which movements were made by the patient and which were assisted. It is easy to see visually how a higher $Movements/min$ corresponds to more frequent movements by the patient. Although we use the same length of time to display the graphs for comparison the objective metrics $Movements/min$ and $Movement\ Strength$ are calculated based on the entire length of the experiment and only when the device is connected.

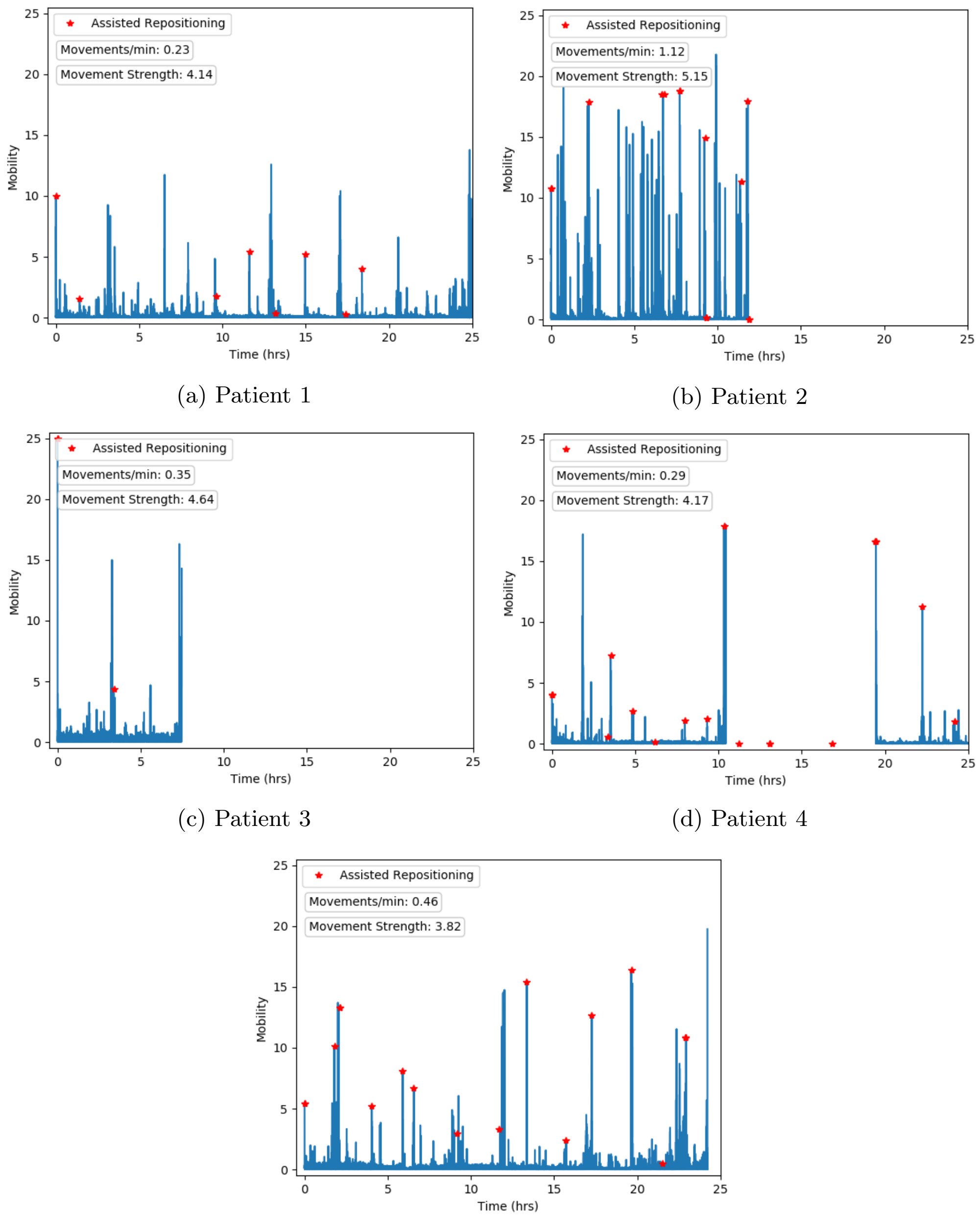
4.5 Posture Analysis

We attempt to track the posture of the patient using the x_{angle} and y_{angle} from Equations 4.1 and 4.2. We use the data from a short experiment of a healthy volunteer that includes reliable labels of posture. We first filter out movement based on our definition from Equation 4.6. We then plot the x_{angle} and y_{angle} against the labels.

The x_{angle} and y_{angle} will have a value of 0 when there is an equal of amount of pressure across the sensor device. This means in theory regardless of posture the x_{angle} and y_{angle} can be 0. From our data we find often that there is a gradient across the sensor and we attempt to use this to infer the posture of the patient.

The intuitive expectation based on the orientation of the sensor we would expect that when the x_{angle} is negative the patient is on their left side and when the x_{angle} is positive the patient is on their right side. When the patient is supine we would expect an x_{angle} close to 0. Likewise the y_{angle} should correspond to the amount of elevation the patient's head is above the patient's back.

In Figure 4.5 we present our results using a healthy volunteer experiment. We do not have enough data to evaluate the simple intuitive analysis based on the sign of the x_{angle}. With more data if this approach does not work a Machine Learning approach would be a natural analysis choice.



(a) Patient 1

(b) Patient 2

(c) Patient 3

(d) Patient 4

(e) Patient 5

Figure 4.4: Mobility Metrics And Graphs Of Patient Data

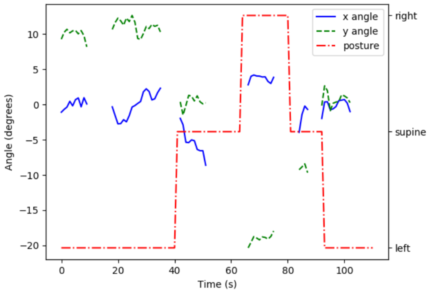


Figure 4.5: Posture Compared To X And Y Angle

4.6 Real-Time Risk Stratification Using Objective Mobility

In the previous sections we present our analysis based on collected data. Because we are using collected data we can perform analysis, such as finding the max pressure of an experiment, that is not possible in real-time. In this section we will outline the types of analysis we used and how we can adjust our technique to apply the same analysis in real-time by integrating Objective Mobility analysis into PIMAP as presented in Chapter 3.

The prominent technique we use in this paper is creating a best-fit plane. From the best-fit plane we calculate the x_{angle} and y_{angle}, which we use for posture analysis, and the $xy_{gradient}$, which we use for mobility analysis.

Calculating the best-fit plane relies on normalization, i.e. giving equal weight between the position of the sensors and recorded pressures. In our case we examine pressure, which when calibrated gives a fixed range of values. Pressure is defined as a force divided by an area. The sensor we use to collect data is a fixed area and force is a mass times an acceleration. The majority of the time acceleration for this application is gravity, which is fixed. This boils down to the pressure we measure is proportional to the mass of the patient. In a real-time system we can normalize the pressure based on a minimum pressure (a mass of 0) and a maximum pressure (close to maximum mass of a patient).

We do not need the actual maximum mass of the patient. An approximation will work fine because having a patient that exceeds the maximum mass will cause anomalous results, but the patient is anomalous and therefore the system should react accordingly.

Table 4.1: Objective Mobility Real-Time Risk Stratification

A		B		C		D		E		Average	
PID	Mvmnts Per Minute	PID	Mvmnts Per Minute	PID	Mvmnts Per Minute	PID	Mvmnts Per Minute	PID	Mvmnts Per Minute	PID	Mvmnts Per Minute
4	0.05	1	0.00	5	0.03	4	0.00	1	0.00	4	0.10
5	0.09	3	0.13	2	0.25	3	0.05	4	0.01	1	0.12
1	0.16	4	1.68	4	0.28	1	0.07	5	0.25	5	0.19
3	0.19	2	2.41	1	1.03	5	0.42	3	0.40	2	0.27
2	1.22	5	3.02	3	1.46	2	1.06	2	1.57	3	0.32

Table 4.2: Description of real-time Objective Mobility experimental setup

Application	Location run
Patient data (x5) sent at 1 sample/s via UDP	iMac 2010
PIMAP-Sense-UDP	iMac 2010
PIMAP-Store-Kafka	iMac 2010
Kafka	Remote Server
PIMAP-Analyze-Objective-Mobility	iMac 2010
PIMAP-Visualize-Plt-Graph, update period 1s	iMac 2010
Network	Internet

After normalization we calculate the best-fit plane on a per sample basis. There will be a maximum sampling rate based on the time to calculate the best-fit plane, but this maximum sampling rate should be much below the typical rates of one sample a second unless running on a bizarre system, in which case there are bigger problems than the maximum sampling rate.

The x_{angle} and y_{angle} can be calculated directly from the best-fit plane and add little to the calculation overhead. The $xy_{gradient}$ relies on time and therefore needs in the worst case three samples (if i is the current location of the gradient calculation we need $i-1$ and $i+1$ to calculate the gradient at i) to calculate the previous gradient value, meaning there is a delay of one sample. At typical rates of one sample a second this is a one second delay, which is not significant for this application.

We use the pressure bandage data collected during its clinical trials to demonstrate how PIMAP can be used to objectively assess pressure injury risk. We fed the pressure data to PIMAP as if it were being collected in real time. PIMAP then calculated patient Objective Mobility over time and displayed the results in real-time.

While in the original trial five patients were monitored at non-overlapping times, we instead simulated the patients as if they were being monitored simultaneously and having their risk assessed in real-time. The original pressure bandage patient data has gaps when the bandage became disconnected (this is often intentional if the patient needs to be moved to a different location). In order to present data that is closest to reality we ran the experiment for 7 hours as this was the amount of time that all patients had consistent pressure bandage data.

We present the results in Figure 4.6. Figures 4.6a, 4.6b, 4.6c, 4.6d, and 4.6e display the strength of assessed movements over the length of the experiment. Figure 4.6g displays The Movements Per Minute metric for all patients over the length of the experiment. The Movements Per Minute metric is a real-time risk assessment of which patients are moving the least, regardless if the movement was clinic-assisted. The y-axis is inverted such that the patient at the top of the graph is the most at risk of forming a pressure injury as they are making the least amount of movements. As can be seen the risk changes over time and there is no one patient that is always most at risk, which is how the status quo Braden Scale assesses patients. The patient data used had an enrollment criteria that all patients must score a 1 (the lowest score) for activity, mobility, and friction/sheer on the Braden Scale. Even though all patients had similar Braden Scale risk scores our metric is able to further distinguish patients in real-time based on their movement.

In Figure 4.6g we label five moments in time, labelled A-E. In Table 4.1 we rank the patients at each respective moment in time based on the Movements Per Minute metric. We demonstrate that a real-time risk assessment can highlight at any moment in time which patient is most at risk. This is invaluable to clinicians in a busy clinical setting as the healthcare staff can focus their efforts on the patients that are most at risk and not waste time blindly rotating a patient periodically that is moving on their own. In addition in Table 4.1 we highlight the average Movements Per Minute over the length of the experiment by patient. It is clear that this this fixed value does not provide the insight that the risk a patient has of forming a pressure injury changes over time. The real-time assessment that we present does not need clinician interference and can be used in addition to any tools or standard of care.

Figure 4.6f presents the end to end latency from when the data was sent in this experimental setup to when the data in Figures 4.6a, 4.6b, 4.6c, 4.6d, and 4.6e were displayed. These metrics are set with a five sample delay per patient at one sample/s. The average latency of this experiment to display the movement data across all patients is $3.05s$, which includes the time it takes to process the data and the time it takes to send the data over the network. See Table 4.2 for the configuration and Sections 3.2, 3.3, 3.4 to fully understand the configuration.

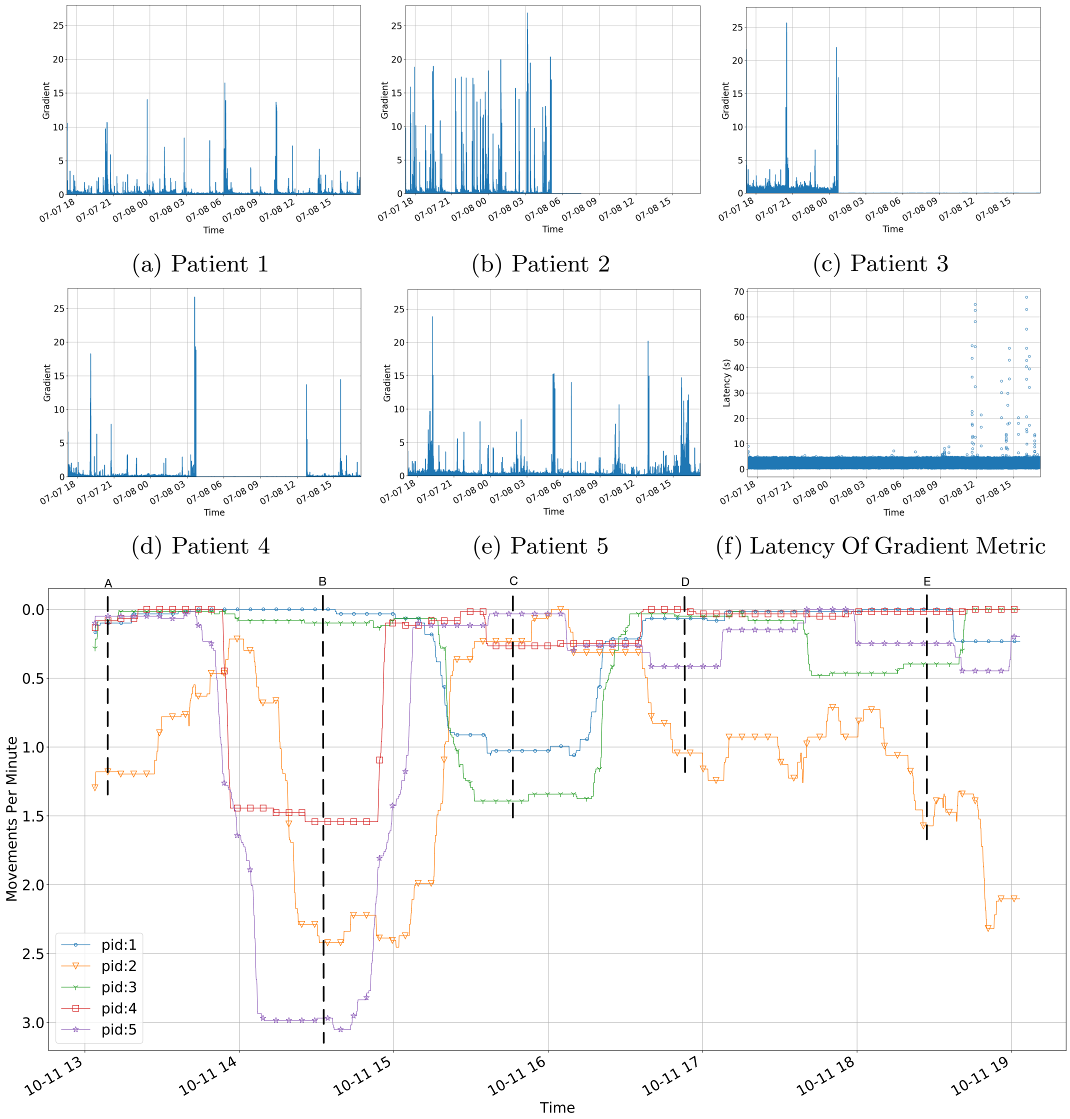
In an ideal scenario with zero network and processing latency we would anticipate a $2s$ latency from the five sample delay that it takes to calculate and visualize the data. For purposes of explanation let us assume that PIMAP-Analyze-Objective-Mobility, which is described in Section 3.4.3, receives sample 1 at time $0s$, sample 2 at time $1s$, sample 3 at time $2s$, sample 4 at time $3s$, and sample 5 at time $4s$, at which point the movement metric is created stored, retrieved, and visualized. Retaining the ideal scenario criteria that there is zero network and processing delay sample 1 would have latency of $4s$, sample 2 would have a latency of $3s$, sample 3 would have a latency of $2s$, sample 4 would have a latency of $1s$, and sample 5 would have a latency of $0s$. When we average these results the resulting latency is $2s$. From this experiment we see on average a $1.05s$ network and processing latency.

We also see in Figure 4.6f a clear increase in latency approximately an hour into the experiment as Movements Per Minute metrics begin to be generated. The metric Movements Per Minute is calculated after 3,600 movement metrics per patient are calculated and is a sliding window, so after a 3,600 gradient metric delay (approximately a one hour delay) every new movement metric a new Movement Per Minute metric is calculated.

4.7 Pressure Bandage Version 2.0

The UCSF group that invented the pressure bandage, SmartDerm, have iterated the design and created a version 2.0 of the bandage, which now has three 4x4 grids of pressure sensors that would be placed horizontally across the sacrum (one on the sacrum and one on either side of the sacrum towards each side of the patient). In addition the version 2.0 of the bandage also includes an accelerometer. The bandage sends data via Bluetooth.

The SmartDerm group is preparing to use this version 2.0 of the bandage in a new clinical trial. The group has staff engineers that they employ to help them setup the system integration of the bandage and how this data will be collected. We proposed to this group to use PIMAP to sense Bluetooth data and then use a modified version of the original Objective Mobility algorithm with the Kafka storage and Matplotlib visualization. The SmartDerm group had already started work on an Android app that gathers data from the pressure bandage v2.0 via



(a) Patient 1

(b) Patient 2

(c) Patient 3

(d) Patient 4

(e) Patient 5

(f) Latency Of Gradient Metric

(g) Movements Per Minute

Figure 4.6: Objective Mobility Real-Time Risk Stratification

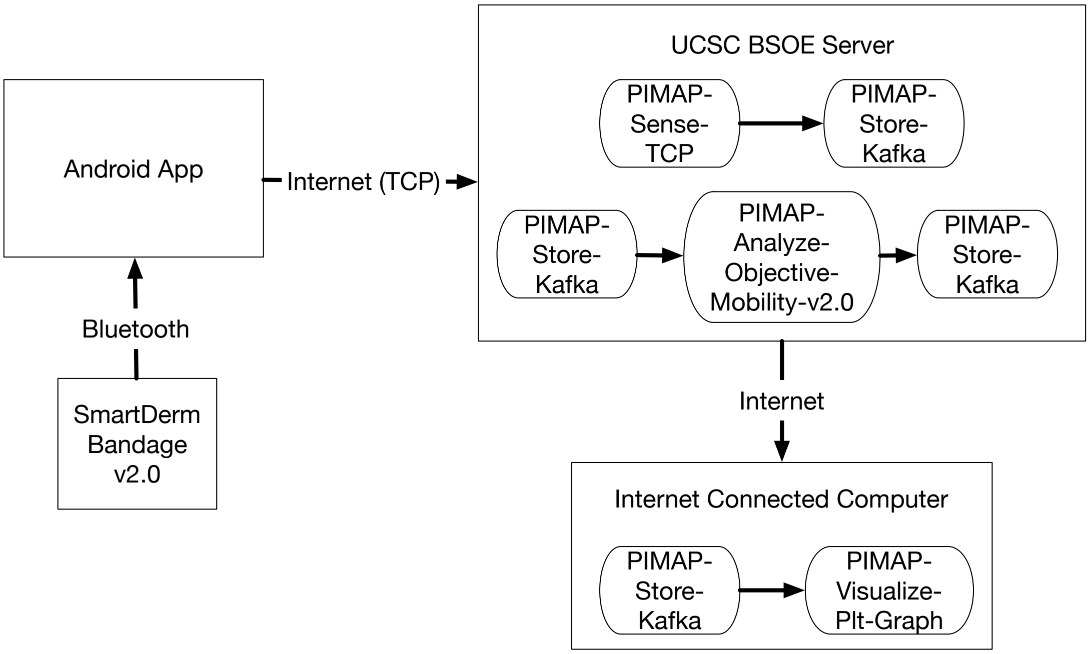


Figure 4.7: PIMAP Cloud Setup For Pressure Bandage v2.0

Bluetooth, so instead of using PIMAP from device to visualization we instead setup PIMAP as a cloud configuration, where data is sent via TCP to the PIMAP cloud and from there is stored, analyzed, and visualized. See Figure 4.7 for a visual diagram of how PIMAP is setup as a cloud configuration. This work is ongoing.

4.8 Addressing Pressure Injury Risk Using A Wearable Impedance Spectroscopy Sensor

Through our collaboration with UCSF we were introduced to Dr. David Young a a clinician and researcher who developed a skin impedance bandage, the Sentinel Bandage, that has the potential to assess whether a patient is forming a pressure injury where the bandage is placed. The core principle behind the bandage is impedance spectroscopy, which is a technique that measures the impedance across two terminals at various frequencies. Impedance is a combination of resistance, capacitance, and inductance, which change based on frequency at different rates. Skin impedance is a well researched area [130] and in brief is based upon the idea that skin can be modelled as a combination of resistances and capacitances. The Sentinel Bandage was previously validated in animal trials that it can detect skin deterioration and potentially pressure injuries [132]. The trials involved machinery that were not practical to use as a patient monitoring system as it could not easily be used by clinicians. In addition the viability of using skin impedance as a proxy for pressure injury formation is further validated by commercial devices such as Bruin Electronics that developed a handheld pressure injury formation detector based on the impedance spectroscopy [82], the same principles as the Sentinel Bandage, although the main difference is that a handheld device requires a healthcare worker to manually take a measurement whereas a bandage can potentially monitor a high risk area of a patient, such as the sacrum, without a healthcare worker.

We formed a collaboration with Dr. Young and the Sentinel Bandage group to integrate the bandage into PIMAP to allow for autonomous data collection from the bandage as well as potential analysis that could be presented to clinicians. Although the concept of the bandage was proven [132] the actual Sentinel Bandage was suspected by Dr. Young to be behaving inaccurately, but a system to validate whether the bandage was working correctly was not established. We would like to acknowledge Eric Vin as a contributor to this Chapter, primarily for his work creating the PIMAP Sense Sentinel component, which will be discussed later, as well as his work gathering and analyzing data to validate the Sentinel Bandage.

4.8.1 PIMAP Prototype

In order to both test PIMAP and the Sentinel Bandage in the wild we decided to create a PIMAP Prototype device that would be able to gather data from the Sentinel Bandage and store it into PIMAP. The main criteria for the PIMAP Prototype was that it be extremely simple to use and therefore we created it to be plug-and-play and just by plugging it into power data would automatically be collected and stored.

The current state of the Sentinel Bandage upon our initial collaboration was a bandage overlaid with 28 conduction points in a grid, 14 transmission points and 14 reception points. An impedance at a given frequency can be sampled between every transmission point and reception point. The bandage itself has a connector. Prior to our involvement a device (Sentinel Device) was developed by a hired engineer that attached to the Sentinel Bandage connector and to a computer via USB or Bluetooth (although we were never able to verify the Bluetooth connection). A Windows application was also developed by this same engineer that could take samples of data from the bandage and save them to a CSV file. The Windows application did not have visualization capabilities.

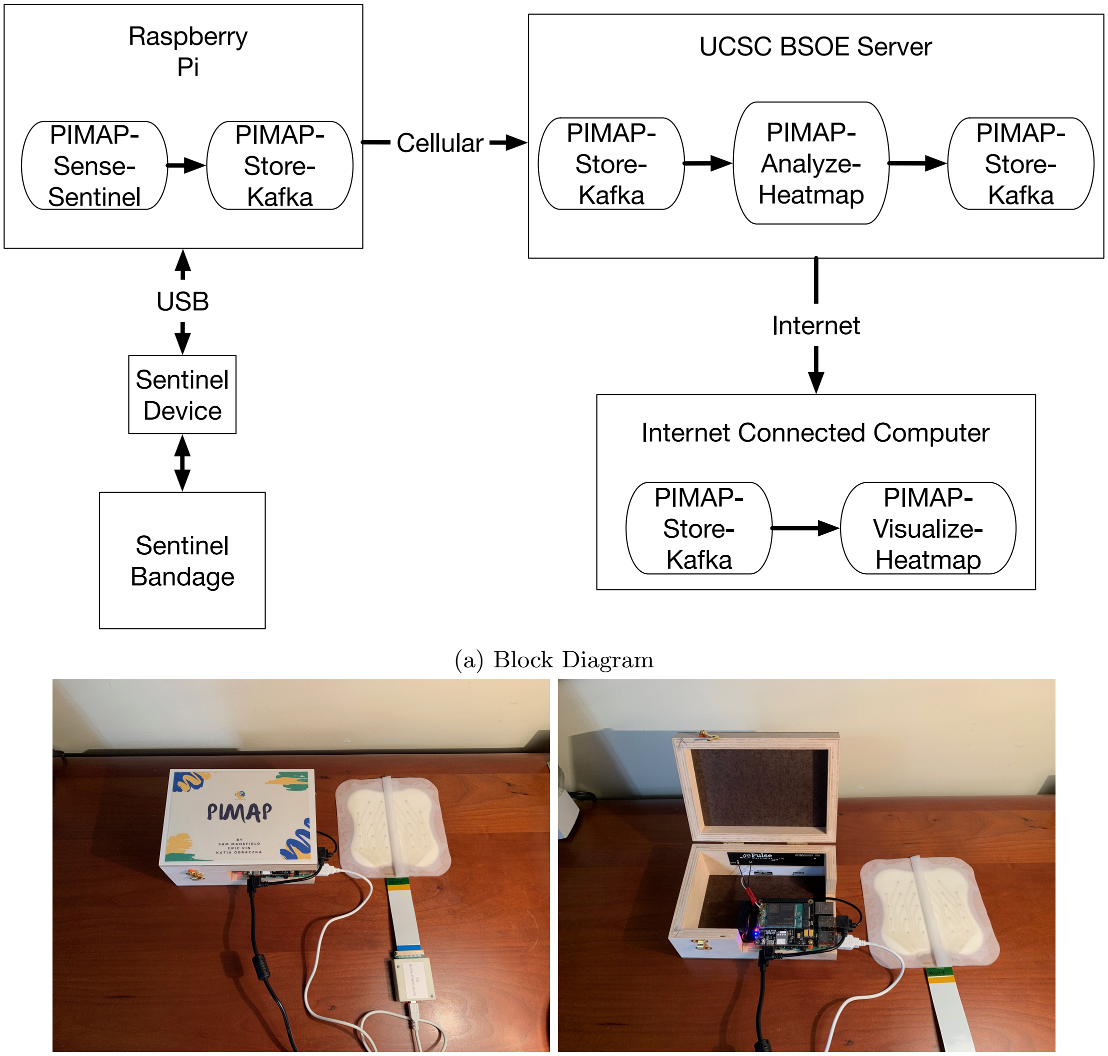
We were able to reverse engineer that the Sentinel Device was sending USB Serial data. Eric Vin took the lead on further reverse engineering what commands could be sent to the Sentinel Device. It was discovered that the device would respond with uncalibrated impedance data and by gathering data with a known resistance across a transmission and reception conductive point we can calibrate the data. Eric Vin further developed a technique to calibrate based on multiple resistance values that was more accurate than just a single resistance value.

Based on our reverse engineering we developed a custom PIMAP-Sense component, which we entitled PIMAP-Sense-Sentinel, which in essence connected to the Sentinel Device using USB Serial and at each call to sense would send a command to the Sentinel Device to receive uncalibrated data. The data was then calibrated and returned.

For this application we also developed a PIMAP-Visualize-Heat-Map component to visualize in real-time the impedance data coming from the Sentinel Bandage, but in order to do so the data must be converted into a format that the PIMAP-Visualize component can recognize. To perform this conversion we created a PIMAP-Analyze-Heat-Map component. We use the impedance values of neighboring transmission and reception points to generate the heat map.

In total the PIMAP Sentinel applications is as follows: a PIMAP-sample is generated from the PIMAP-Sense-Sentinel component the data is stored using the pre-existing PIMAP-Store-Kafka component, the PIMAP-samples are retrieved from PIMAP-Store-Kafka analyzed by PIMAP-Analyze-Heat-Map, converted to PIMAP-metrics, and stored. The PIMAP-metrics are then retrieved and visualized by PIMAP-Visualize-Heat-Map. This can be seen visually in Figure 4.8a.

The actual PIMAP Prototype was created using a Raspberry Pi 4 and Sixfab's Raspberry Pi 4G/LTE Cellular Modem Kit to send cellular data via T-Mobile. The Raspberry Pi runs a sense and store workflow, where the sense component is PIMAP-Sense-Sentinel and the store component is PIMAP-Store-Kafka. We use systemd to enable this workflow to start on startup and restart when needed, such as when the Sentinel Device is not plugged in. In addition we created an ssh reverse tunnel from the Raspberry Pi to the UCSC BSOE server so that we can ssh in and change/update configurations when necessary. We created a custom enclosure for the PIMAP Prototype and it can be seen in Figures 4.8b and 4.8c.



(a) Block Diagram

(b) Photo Closed

(c) Photo Open

Figure 4.8: PIMAP Sentinel Prototype

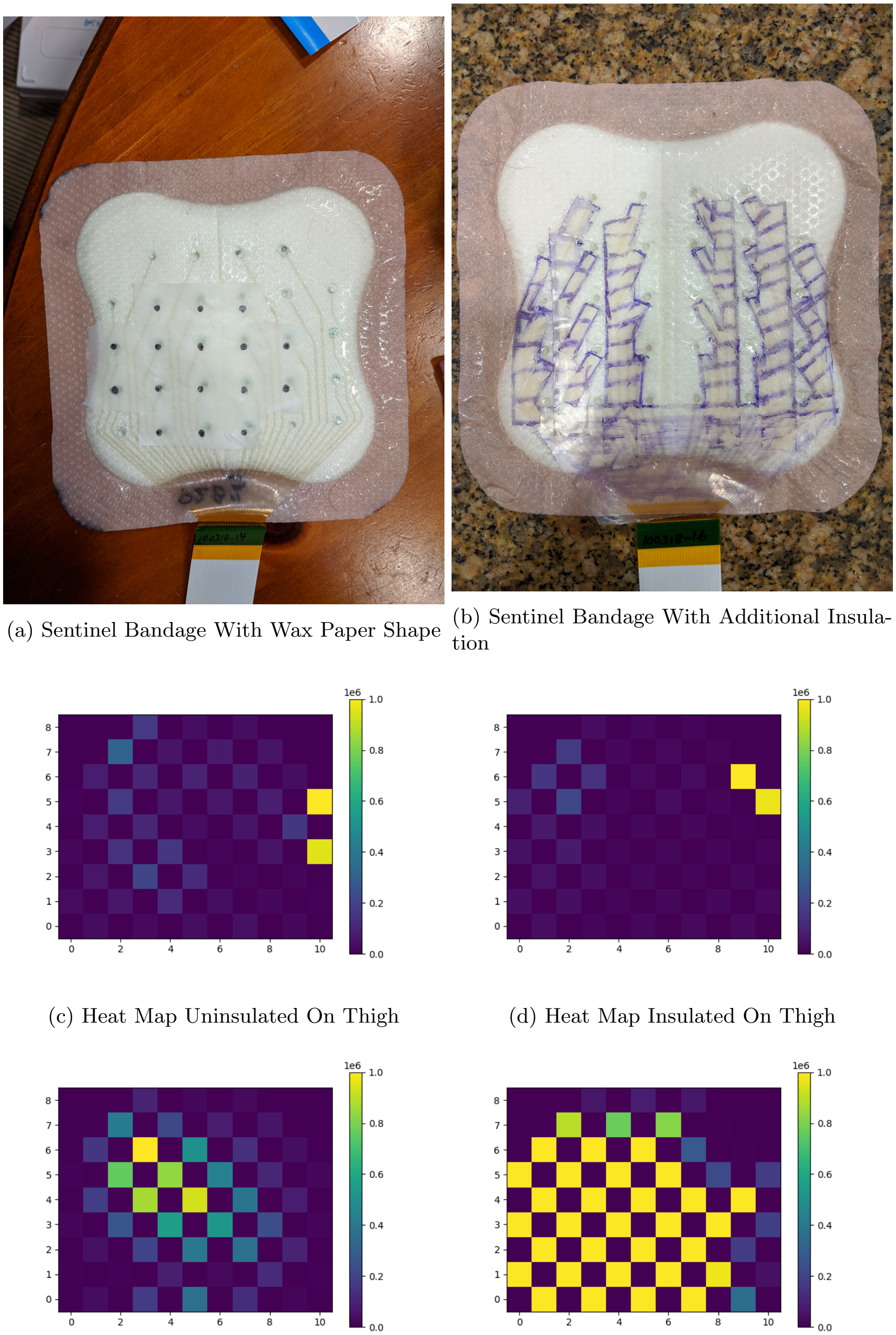
4.8.2 Validation Of Data

We entered this collaboration when the Sentinel Bandage was unvalidated, meaning it was operational in the sense that data could be collected, but the data collection took a lot of manual steps and could not be visualized. Dr. Young suspected that the Sentinel Bandage was not working properly as every time data was collected on a healthy volunteer the data seemed like noise. We used PIMAP based on our PIMAP Prototype design presented in the previous section to sense and visualize data in a test experiment to see if we could identify whether the Sentinel Bandage and Sentinel Box were operating correctly. These tests were conducted by Eric Vin.

Through observational experiments Eric discovered that there was a shorting issue with the bandage where touching a supposedly non-conductive lead would behave as if the conductive terminal were touched. A photo of the bandage is displayed in Figure 4.9a with a wax shape placed on the bandage that was used for experimentation. In this photo one can see the leads that connect with each conductive silver terminal that all come together at the connector of the bandage. These leads are supposedly non-conductive, but if touched the data sensed looks as if each lead touched was touched on the conductive silver terminal. The exact phenomenon that causes this is still not known, but we were able to verify that this phenomenon does occur and were able to offer a temporary fix that can be used upon the next iteration of bandage production.

There were four experiments conducted using a local PIMAP configuration where data stays on the local computer and does not travel over the Internet as it simplifies the design of the experiments. Each experiment was conducted for ten minutes at three frequencies: 15,000, 25,000, and 50,000. In Figure 4.9 we display a heat map at 15,000 Hz for each experiment that is representative of each experiment as a whole. Each heat map is the impedance at the given location in the same orientation as the Sentinel Bandage as depicted in Figures 4.9a and 4.9b.

During these experiments Eric found a connection error into the Sentinel Device where two of the connections would give erroneous data, which can be seen on the right hand side of Figures 4.9c and 4.9d. We only had one Sentinel Device to work with, but even with this connection error were still able to show that further insulation of the Sentinel Bandage produces stable data.



(a) Sentinel Bandage With Wax Paper Shape

(b) Sentinel Bandage With Additional Insulation

(c) Heat Map Uninsulated On Thigh

(d) Heat Map Insulated On Thigh

(e) Heat Map Uninsulated On Thigh With Wax Paper Shape

(f) Heat Map Insulated On Thigh With Wax Paper Shape

Figure 4.9: Validation Of Sentinel Bandage Experimental Data

To verify the shorting issue with the unmodified Sentinel Bandage a wax paper shape that can be seen in Figure 4.9a was placed on the Sentinel Bandage before being placed on the thigh. Wax paper is non-conductive and therefore the wax paper shape should clearly be seen in the visualizations as being high impedance (yellow in our heat maps) at these locations. In Figure 4.9e a heat map of the unmodified bandage with the wax paper shape can be seen and it is clear that the shape is only semi-visible and in fact the conductiveness of the bandage actually fades over the ten minutes of the experiment making the wax paper shape less and less visible. In Figure 4.9f the insulated bandage as depicted in Figure 4.9b was used with the wax paper shape and the shape is clearly visible as high impedance and does not fade over the length of the experiment. The Sentinel Bandage was insulated by placing non-conductive silicon tape over the exposed leads leading to the terminals the outline of the silicon tape is highlighted with pen for visibility in Figure 4.9b.

Through these experiments using PIMAP to sense, store, analyze, and visualize data it is clear that insulation fixes the shorting issue even though the exact cause is not known. We both validated the Sentinel Bandage in its use on human skin in addition to validating that PIMAP can be used to quality control an unverified device primarily through visualizing the data in real-time.

The Sentinel collaboration is ongoing and we intend to continue the development of PIMAP through the use of the PIMAP prototype used by Dr. Young to conduct experiments. The custom configuration we created can incorporate additional Sentinel Bandages and could potentially be used in a clinical trial. In addition PIMAP can be used to validate new bandages by visualizing the data as a heat map. There are still many avenues of study using the Sentinel Bandage and PIMAP is an enabling system to sense, store, analyze, and visualize this data in real-time. With this new ability enabled by PIMAP new analysis techniques that are currently not implemented can easily be incorporated without modification of the system as a whole and instead can be deployed on the UCSC BSOE servers and with the addition of new sensor data from many Sentinel Bandages there is a potential for risk stratification based on the sensed and analyzed data.

4.9 Future Work And Conclusion

With more well-labeled data we can most likely provide an accurate prediction of posture. If using only the x_{angle} as the predictor is not enough it would be a very natural fit to use Machine Learning techniques to predict the current posture by stochastically predicting the current posture based on the x_{angle}, y_{angle}, and maybe the current average pressure.

This work is a proof of concept of how to use a wearable pressure-sensing device to objectively measure mobility. We also demonstrated how a real-time system that can be used in the clinic can present this data directly to healthcare staff. We entitled this system PIMAP (Pressure Injury Monitoring And Prevention). In addition to being a real-time system to monitor pressure injuries we also view this system as a research tool to study pressure injury formation as there is currently no system dedicated to this task.

In this work we presented an objective way to assess the pressure injury risk of a patient in real-time using a novel wearable pressure-sensing device. We discussed our methods on how to analyze the data to assess the mobility of the patient. We also discussed analysis on how to track the posture of the patient, but admittedly more well-labelled data is needed before we can confirm this method.

Our initial analysis methods were performed on collected data, but we also discuss how and why we applied these same techniques to a real-time system and in future work we aim to continue investigation on providing accurate posture measurements. This work is ongoing and with the design of the new pressure bandage 2.0 and the associated PIMAP cloud configuration was designed with the goal to use PIMAP in a clinical trial to present real-time objective metrics directly to clinicians.

Chapter 5

Privacy And Security

In this chapter we discuss privacy and security considerations of a patient monitoring system. In addition we describe how and where in PIMAP's design privacy and security can be implemented. We use the term privacy in this chapter to mean handling information that can be identified to an individual regardless of the data's perceived importance, i.e. we do not distinguish between location data versus pressure data as both must be private even though at first glance location data may seem "more" private than pressure data. We use the term security in this chapter to mean data security, i.e. storing/transmitting data in a way that even if accessed by an unauthorized user it will be extremely difficult to interpret. This chapter is theoretical as of this writing as we have not implemented any of the strategies mentioned, the primary reason is throughout this process we have never had in our possession any data that contained identifying information and therefore we never had the need to secure this data. But, PIMAP has the potential to handle a large repository of sensor data that may need to be both private and secure.

Medical sensor data in general falls into a gray area in regards to privacy. If the sensor data is not labelled with patient identifiers, such as name, date of birth, etc. it is not clear whether such sensor data can be identifiable to a patient. For example pressure data is currently not identifiable to a patient by itself, but it is possible in the future that pressure data may contain a "pressure fingerprint" that can be identifiable to a patient or maybe the combination of pressure data and temperature data associated with the same patient may become a fingerprint.

HIPAA [47] in the U.S. provides regulations on both privacy and security of medical related information. We are by no means experts on HIPAA and can only provide our interpretation of the regulations. Based on the HIPAA guidelines [47], PIMAP would be required to only allow authorized entities to access data and PIMAP would be required to store data so that it would be very difficult to interpret if somehow this data was accessed by an unauthorized entity. There are many other regulations in HIPAA that we will not cover, for example if data is breached how to notify the patients that are effected, but this is beyond the scope of this discussion. There are several other government regulations that would also potentially affect how PIMAP handles data such as if PIMAP's use needs to be approved by the FDA it needs to follow the FDA's guidance on electronic records, FDA Part 11 [98]. In the case PIMAP needs to be used commercially, as we are in California, the California Consumer Privacy Act (CCPA) [15] specifies rights of consumers to the access, deletion, and sharing of personal information by businesses.

There are some general techniques that can be used to address privacy and security. For example to keep data private a common technique is to anonymize the data, e.g. "John Doe" becomes patient 1. This can be achieved using a one-way hash function so that patient 1 can be used to categorize data, but patient 1 cannot be identified back to "John Doe". To keep data secure a common technique is using encryption either with a symmetric key, i.e. one key, that if obtained can unencrypt the data or a public/private key pair so that if data is encrypted using the public key, only those with the private key can unencrypt the data. The advantage of public/private keys over symmetric keys is that a private key is only needed to unencrypt data whereas a symmetric key is needed to both encrypt and unencrypt, therefore there are more places where the symmetric key can be intercepted.

The remainder of this chapter is broken into the following sections. In Section 5.1 we will discuss the related work around privacy and security in regards to patient monitoring and IoT. In Section 5.2 we will examine three distributed configurations of PIMAP that we defined in Chapter 3 and describe where we would implement privacy and security measures. In Section 5.3 we will conclude.

5.1 Privacy And Security Techniques For Patient Monitoring

We highlight three research areas around patient monitoring with regards to security and privacy. One approach is based on end-to-end encryption from medical device to cloud thereby securing the medical sensor data. Another approach uses attribute-based authentication to secure data in such a way that only those with the correct attributes can decrypt the data, e.g. only your doctor can access your data. A third approach is using blockchain [84] to have a distributed authority of records that can also restrict access to authorized users.

One end-to-end encryption patient monitoring approach, consisting of devices, gateways, and the cloud relies on the Datagram Transport Layer Security (DTLS) protocol [115], which offloads the computationally expensive security protocols to the gateways, which act as a bridge between device and cloud so that resource-restricted medical devices can still encrypt medical data that would otherwise be too computationally expensive [83]. This approach secures the medical data upon production so that it cannot be accessed or interpreted easily by others and is only unencrypted at a secure location, i.e. data is never unencrypted when sent over a network or when stored. The drawbacks are that it requires a central trusted authority to manage the security keys and in this particular work there are no additional steps to restrict access to medical data, i.e. if you have access to the data you can view all the data.

Attribute-based authentication [120] [38] [10] for patient monitoring was proposed as a way to both secure and restrict access to data based on attributes, e.g. only the healthcare professionals monitoring patient x can view patient x's data [71]. Attribute-based authentication uses attributes, e.g. doctor x or patient y, to encrypt the data, so that the data itself is secured and can only be viewed by those with the same attributes. The drawback of this approach is that if attributes are added or removed the affected data must be re-encrypted, which is computationally expensive. This method also requires a centralized authority.

A third approach to secure data in patient monitoring is by using blockchain to have a distributed authority of medical sensor data [136]. A blockchain can be computationally expensive when miners provide Proof of Work, so the authors propose a miner selection method to reduce this cost. In addition the authors also propose a way to restrict access to data. The drawback of this method is it requires a high degree of coordination on top of the patient monitoring

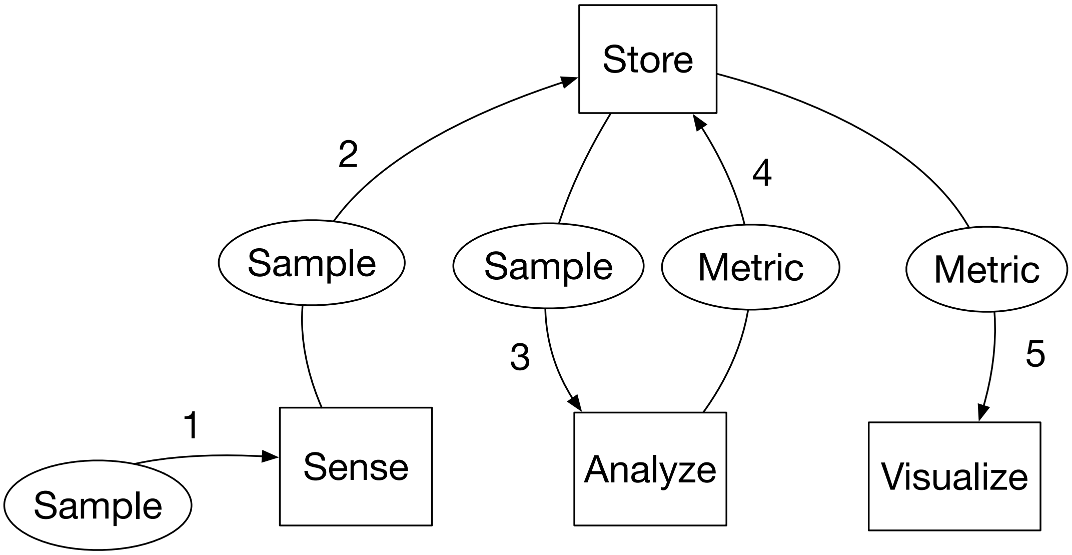


Figure 5.1: General Data Flow For Patient Monitoring

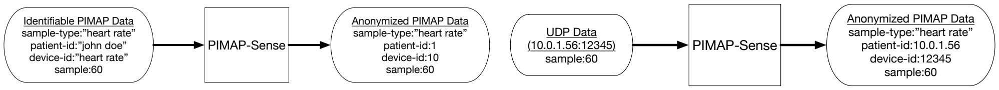
application itself.

There are many other approaches to patient monitoring security. We highlighted three works that give a representative view of the literature, but we expect the privacy and security landscape to change over time and we expect the newest developments to also make it into patient monitoring.

5.2 Addressing Privacy And Security In Three Distributed Configurations Of PIMAP

The general data flow of any patient monitoring application as discussed in Section 3.2 is shown in Figure 5.1. Recall that PIMAP has four components: Sense, Store, Analyze, and Visualize. There is a also a typical ordering where raw data is sensed and stored, the raw data is then retrieved, analyzed, and the metric created from the raw data created by the Analyze component is stored, the metric is then retrieved and visualized. Also each PIMAP data-type we defined in Section 3.2.1 have a patient-id and device-id that identify which patient and which device on the respective patient that the raw data or metric is associated with.

To assure data is private, i.e. that data cannot be identified to a patient, privacy could be



(a) Anonymizing PIMAP Data

(b) Anonymizing Non-PIMAP-Data

Figure 5.2: Anonymizing Patient-id And Device-id

directly addressed at the Sense component, i.e. as patient data enters PIMAP. But, this is not a trivial task. If PIMAP data is sent from the medical device that identifies the patient we can start with keeping the patient-id and device-id private, i.e. anonymizing the patient-id and device-id, for example by using a hash function to create a one-way mapping from patient-id or device-id to anonymized patient-id or device-id we can assure privacy, but each PIMAP-Sense component must then keep some sort of record of the mapping to either output to a human or into another private system, see Figure 5.2a for a graphical representation. If non-PIMAP data is sent we have already put in place a pseudo-anonymous method in place, e.g. the PIMAP-Sense-UDP component upon receiving non-PIMAP data uses the medical device's host as the patient-id and the port as the device-id, in this way data entering PIMAP is somewhat private, see Figure 5.2b for a graphical representation.

It is very difficult to anonymize the sensor data and metric data as changing the data will make it very difficult to analyze. As stated previously sensor data is often in a gray area when it comes to privacy as it is not obvious that pressure data for example could be used to identify a patient, but there is a reality that with enough different types of sensor data it is probably possible to create a sensor "fingerprint" of a patient. We leave this problem to a future date as it will take experimentation and validation to truly examine the problem of anonymizing sensor data while still being able to analyze the data.

Securing data, storing and transmitting data in a way that is difficult to understand if accessed by an unauthorized user is straightforward, but time-consuming to implement. To maintain security in PIMAP the transfer of data to and from each component must be secured. In addition fine-grained control of access to data takes additional steps such as using attribute-based encryption [38] or other access control methods. PIMAP has natural access control based on sample-type, metric-type, patient-id, and device-id because the data is already divided by these identifiers. For example the storage component is configured such that you request data by sample/metric-type and patient-id and therefore we can limit access based on these criteria

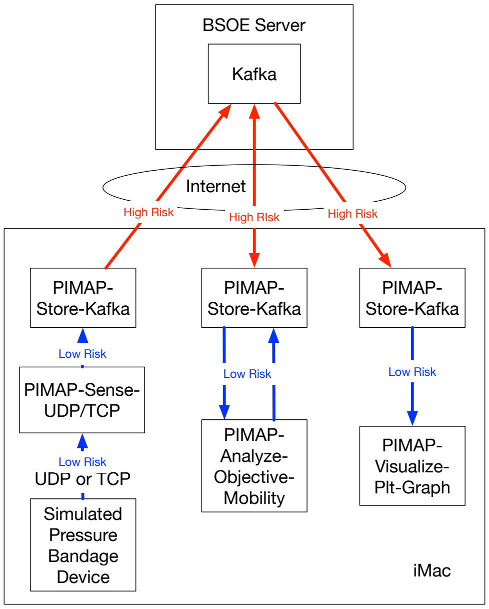


Figure 5.3: Remote Storage Configuration Security Risk

as well and in addition we could limit by device-id for fine-grained control. This would be application level access control as an application can be limited to only store or retrieve a subset of all data.

We will discuss security in more depth using examples of three PIMAP configurations: Remote Storage, Cloud, and Edge. For each configuration we classify links between components as low, medium, and high risk in regards to security and discuss different ways to secure the data and have access control. Keep in mind that we do not label any links as no risk, so just because some links are labelled low risk, that is just relative to the medium and high risk links. Low risk still has risk associated with that link.

5.2.1 Remote Storage Configuration

In Figure 5.3 we graphically depict the Remote Storage Configuration with each link between components and Kafka labelled as low, medium, and high risk. There are a total of eight links, three of which we designated high risk and five of which we designated low risk. The high risk links are designated as such because they go over the Internet. The low risk links are designated as such because they are local to a personal computer.

The Remote Storage Configuration was configured so that all PIMAP components are run on one computer, but the storage of data is remote so that the data can be accessible to other researchers. The high risk links in this configuration all have to do with storing/retrieving data to/from Kafka. To handle the data across these high risk links we can leverage Kafka's security tools to encrypt the data, but we will not be able to leverage these tools when using a different storage technology. Alternatively we can integrate into PIMAP's storage component an encryption scheme, either as a symmetric key or a public/private key managed by the PIMAP application designer. For example when data is stored by PIMAP-Store-Kafka the data would be encrypted using a symmetric key or a public key so that only those with this symmetric key or the private key can unencrypt the data, such as the PIMAP-Store-Kafka component that is retrieving the PIMAP-samples for analysis. The keys would be setup when configuring the PIMAP application. The low risk links can also be handled with symmetric keys, public/private keys, or not use any encryption if the risk is low enough, but this depends on the situation. One approach might be to use a simple symmetric key for the low risk links and public/private keys for the high risk links.

Access to the data in this configuration can be managed in a straight forward manner by limiting access based on the sample-types, metric-types, patient-id device-id pairs. For example in this application the sample-type may be "pressure_bandage" and the metric-type may be "objective_mobility" for patient-ids: 1, 2, 3, 4, and 5 with one device-id per patient. Based on this we can limit access to this data, which would be through Kafka, by encrypting data using this application's public key so that only those with the private key of this application can access this data. The encrypted data would still be accessible unless we also encrypt the sample/metric-type identifiers.

5.2.2 Cloud Configuration

In Figure 5.4 we graphically depict the Cloud Configuration with each link between components and Kafka labelled as low, medium, and high risk. There are a total of eight links, two of which we designated high risk, five of which we designated medium risk, and one of which we designated low risk. We designated the links with high risk as they go over the Internet, we designated the links with medium risk as they are running on a remote server that others may have access to, and we designated the links as low risk as they are on a personal computer.

The Cloud Configuration was created to approximate a cloud-based solution where most

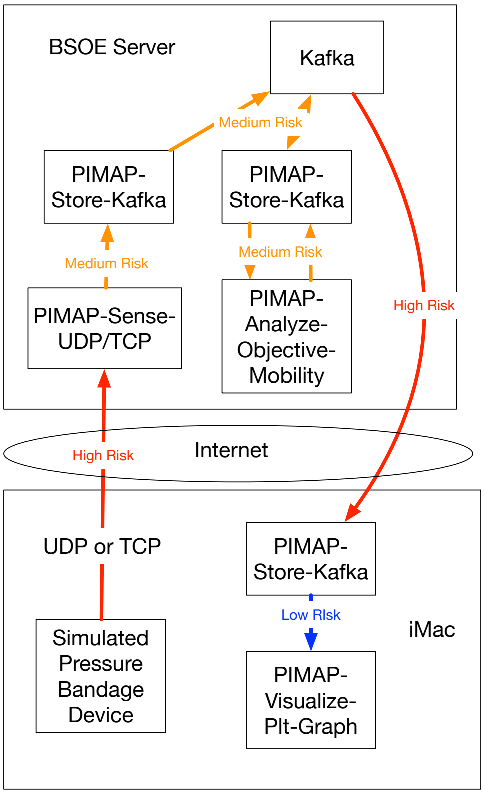


Figure 5.4: Cloud Configuration Security Risk

PIMAP components are run in the cloud where there are more resources. When we look at the security implications we can see that when compared to the Remote Storage Configuration in general there is higher risk when using the Cloud configuration. The Cloud configuration is also difficult to secure because the PIMAP application components are running on two separate computers and therefore the key whether symmetric or public/private must be stored on both computers. Unfortunately in this configuration the analysis is performed on the remote server and therefore the sensor data must be unencrypted to be useful and therefore a key must be stored somewhere on this remote server, which by nature is less secure. Also unique to this configuration is that the simulated pressure bandage device is sending data over the Internet and must also encrypt the data to be considered secure and as the device is not really part of the PIMAP system it may need a separate key associated with the PIMAP-Sense component. Transport Layer Security (TLS) [24] can be run on top of TCP or if running UDP we can use Datagram Transport Layer Security (DTLS) [115].

Access to the data in this configuration can be manged similarly to the Remote Configuration scheme by having a symmetric or public/private key for this application so that only those with this key can understand and use the data. The encrypted data would still be accessible to all. We can see that this Cloud Configuration is not the most secure configuration as we will need to store a key on the remote server, which may be accessible to others. There may be additional security schemes that accommodate this scenario in a secure way, but it is beyond the scope of this discussion.

5.2.3 Edge Configuration

In Figure 5.5 we graphically depict the Edge Configuration with each link between components and Kafka labelled as low, medium, and high risk. There are a total of eight links, we designated two of these links as medium risk and the remaining six links as low risk. We classified the links that go over the local network as medium risk and the links run on the two separate personal computers as low risk.

The Edge Configuration based on our classification is the lowest risk as it does not contain any high risk links and contains the most low risk links. This is inherent to the configuration as no data is transmitted over the Internet, the drawback of this is that data is not accessible to those outside of the local network. We can apply the same techniques as discussed for

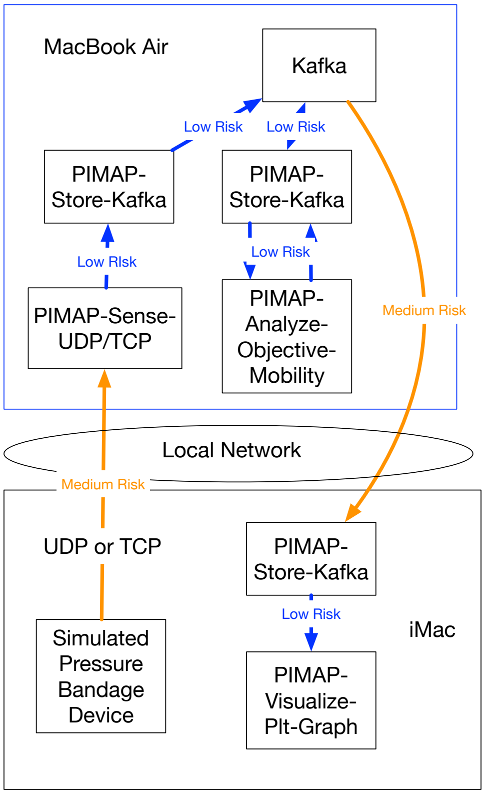


Figure 5.5: Edge Configuration Security Risk

the previous configurations using a symmetric or public/private key that is accessible to this PIMAP application. This configuration may be slightly less secure than the Remote Storage Configuration as the security keys must be stored in two different locations, although these location are both more secure than a remote server. Similar to the Cloud Configuration the simulated pressure bandage device is sending data over a network, although in this case it is a local network, so one can use TLS or DTLS depending on the transport layer in use.

Access to data can also be restricted based on sample/metric-type and patient-id device-id pairs. Because data is encrypted when transmitted and stored only those with the necessary security keys will be able to unencrypt the data, although the encrypted data will still be accessible.

5.3 Conclusion

Privacy and security are very important topics especially in regards to distributed systems such as PIMAP. In addition because PIMAP deals with medical data this makes privacy and security even more important. PIMAP to date has not incorporated the techniques mentioned in this chapter, but has the potential to do so. We discussed related works that focussed on security mechanisms using a centralized server, attribute-based encryption, and blockchain in relation to a patient monitoring system. We can leverage these techniques for the PIMAP system and in addition because of the structure of PIMAP into components and each data-type contains classifiers such as sample/metric-type, patient-ids, and device-ids there is a natural separation of data.

We discussed three configurations of PIMAP and assessed the risk for each link in the configuration, which revealed that the lowest risk configuration is the Edge Configuration as it does not transmit over the Internet and does not rely on a remote server, the next lowest risk is the Remote Storage Configuration, and the highest risk configuration is the Cloud Configuration as data is transmitted over the Internet and the components of PIMAP are run on a remote server, so a security key must be stored on the remote server.

We discussed privacy in regards to patient-ids and device-ids, which PIMAP can anonymize although this does require that another entity has the mapping to deanonymize the data as needed. The sensor data itself is in a gray area as it is not clear at this time whether data such as temperature and pressure will be able to identify a patient and it is an extremely difficult task to anonymize the sensor data as the raw data is needed for analysis.

We discussed how data can be secured in the three configurations of PIMAP and how this security leads to fine-grained access control as only the PIMAP application would be able to unencrypt the data for the given sample/metric-type and patient-id device-id pairs. For future work we aim to make PIMAP secure so that many PIMAP applications can be run simultaneously using the same storage technology and also so that PIMAP can be adopted by those who need security in place to use PIMAP.

Chapter 6

Future Work And Conclusion

There are many future directions for PIMAP. In the past we explored modeling the healthcare clinic as a Cyber-Physical System (CPS), see Appendix A, and this still has potential to explore how PIMAP can affect the clinic and in addition can help tune algorithms based on the results found in the CPS model. The limitation of the CPS model is that it relies on a closed loop system, which in our case means we are modelling how a clinician will react to the data presented, which may not be accurate. In addition this work was conducted before PIMAP was fully standardized and only represents the general aspects of the system as opposed to actually using PIMAP in the CPS model.

Some related works focus on the use of a Smartphone as a bridge between a medical sensor and the Internet. PIMAP is built upon Python and both iOS and Android have Python interpreters, but as we have currently not explored this we do not know how useful these interpreters are and whether they will be able to handle the current implementations of PIMAP. But, even if the current implementations will not run it seems likely that alternative PIMAP implementations could be created that rely only on the libraries supported by the iOS and Android interpreters, or a different version for each. This work would make PIMAP truly cross-platform as then it could be run on OSX, Windows, Linux, iOS, and Android as they all have Python interpreters.

One of the big potential projects for PIMAP down the line when there are many sensing, storing, analytic, and visual technologies implemented PIMAP can assess based on a configuration, e.g. cloud, edge, local, or custom, which technology is best suited for the application. For example for the storage component we currently implement Kafka, but two other well known time-series data-stores are influxDB [54] and Prometheus [109]. PIMAP can be used to compare and contrast based on different configurations, which of these data stores is appropriate for a given configuration. The same can be applied to visualization technologies.

The most promising way for PIMAP to develop is for researchers to start using and improving upon it. This not an easy task and so far we have had the most difficulty in establishing collaborations. We hope that by releasing our Github repository with documentation and examples we will provide a resource for any researcher to use PIMAP without having to directly consult with us. This may be a lofty goal, but I think it is the only way to truly proliferate PIMAP as making ourselves a singular bottleneck greatly reduces the availability of PIMAP's use.

Our collaboration with UCSF's SmartDerm team is ongoing and with their new design of a pressure bandage 2.0 that integrates with a cloud-based PIMAP configuration could potentially be used in a clinical trial to assess how real-time objective metrics can influence care.

Our collaboration with UCSF's Sentinel team is ongoing. We validated that the Sentinel Bandage operates as expected with additional insulation. The Sentinel team is planning on using the PIMAP prototype to test the insulated bandage and in addition plans to create a new bandage that has built-in insulation when manufactured. PIMAP also has potential to be used in a clinical trial in conjunction with the Sentinel Bandage to study how real-time visualization of skin health can influence care.

We recently started a collaboration with UC Davis researchers and clinicians Holly Kirkland-Walsh and Sarina Fazio to create and visualize a metric to assess the risk of a pressure injury based on the blood pressure of the patient. The idea behind the work is based on Holly Kirkland-Walsh's research based on a retrospective study that for every mmHg decrease in diastolic blood pressure, the odds of a deep tissue injury increased by 7.5% [59]. In future work we aim to use PIMAP to automatically calculate this metric and present it in real-time to clinicians with the goal of stratifying patients based on the risk of pressure injury formation.

PIMAP may not be the best-fit patient monitoring system for all applications. PIMAP assumes that an application can be broken into the four components: sense, store, analyze, and visualize. PIMAP also assumes that streams of data are going through the system. For example PIMAP may waste resources if an application generates large amounts of data at sparse intervals in time, although PIMAP could still run in this configuration. In addition PIMAP is implemented in Python and assumes that a researcher has enough programming experience to create a new component based on an existing example. For example if a medical researcher wants to implement a new PIMAP-Analyze component they would need to look at previous PIMAP-Analyze components or an example PIMAP-Analyze component and know how to make the appropriate changes to incorporate the intended analysis. In fact when incorporating a novel sensor into PIMAP the PIMAP-Analyze component is often going to be custom as novel sensor-based analytics are heavily dependent on the novel sensor. Finally, because PIMAP abstracts different technologies (such as Kafka) and can be run in distributed configurations it does take someone with a good understanding of networking to appropriately setup a distributed PIMAP configuration. But, in the future this is a service that could potentially be commercialized for those who do not have experience to setup a distributed PIMAP configuration on on their own, but still want to use PIMAP.

Most of the experimentation and data used to test and improve PIMAP are on the order of hours of data. But, eventually PIMAP could be used in applications with years of data. There are different requirements when working with data at this scale and the current development of PIMAP has not had to deal with this order of data. This is not a limitation of PIMAP, but instead a future development of PIMAP, primarily around data navigation. With years of data even at a relatively low sample rate of one sample a second one cannot simply traverse through all the data without large delays in time, which would make PIMAP difficult to use. Instead PIMAP would need data traversal to jump through the data to find/analyze data one is looking for.

We presented PIMAP a patient monitoring system framework based on the need to monitor and assess the risk of medical conditions such as pressure injuries and the lack of reusable and modifiable software in this area. Pressure injuries are still an unsolved problem in healthcare as current methods to assess risk and prevent pressure injuries require a large amount of clinician time, which is often not available in large facilities.

Through an extensive literature survey we identified patient monitoring as a promising approach to assess the risk a patient has of forming a pressure injury, but found all patient monitoring approaches in the literature either were commercial and therefore were expensive and locked into what the commercial entity offered and therefore could not be modified for other applications or one-off systems that had little documentation and were often not used again.

PIMAP was designed with this patient monitoring need in mind and focuses on a system framework that can easily incorporate new sensing, analysis, storage, and visualization while leveraging existing PIMAP components that do not need to be reinvented. In addition PIMAP is designed to be deployed in a distributed manner so that PIMAP can handle both small studies and large multi-site studies without changes to the underlying framework.

We used PIMAP to study new objective metrics based on novel sensing devices such as a pressure bandage and an impedance spectroscopy bandage and existing FDA approved clinical devices and observed how these metrics could be used in real-time to assess the risk a patient has of forming a pressure injury.

PIMAP is released with a license allowing all noncommercial and research institutions to use and is described in detail so that others can create similar systems if there is a need. PIMAP can be easily modified and extended for new applications while leveraging existing implementations such as PIMAP-Store-Kafka, PIMAP-Visualize-Plt-Graph, and PIMAP-Sense-UDP.

We intend PIMAP to be used in the future by medical researchers to accelerate medical research by lowering the barrier of testing new medical devices and algorithms in real-time, so that these new methods can be validated and used in the EMR.

Appendix A

Modeling The Healthcare Clinic As A CPS

The exact external forces that cause decubitus ulcers, a.k.a. bedsores, pressure sores, pressure ulcers, or pressure injuries is an open-ended question. The related work indicates that it is some combination of pressure, friction, shear force, temperature, humidity, and restriction of blood flow [34]. But, bed sores are also patient dependent [34] and are more prevalent in patients with chronic health problems [74].

Bed sore detection and prevention is split into two categories: sitting-acquired and laying-acquired. This project will focus on laying-acquired bed sores, as they were the original problem presented by UCSF, but some of the concepts may be applicable to sitting- acquired ulcers as well

Previous studies of bed sore detection are mainly clinical studies relating patients to prevalence of bed sores [74] or patient facilities to prevalence of bed sores [16]. bed sore prevention has varied from placing foam bandages over high risk areas [121], placing a layer of soft foam on top of beds [112], dynamic mattresses [7], moisturizing the patient [112], and rotating the patient manually [112].

Technology-based solutions targeting bed sore detection have been the focus of a number of patents. For example, one method describes a mattress that notifies a nurse when a patient is in one position for too long [127], reducing the need for a nurse to constantly check on the patient (e.g., every two hours), if the patient has moved on their own. A more extensive patent involves a video camera that monitors the patient and keeps a log [55]. Both of these approaches are interesting, but have not been studied to validate whether their method is effective.

The goal of this project is to create a system that can both monitor bed sores and also study what the exact cause of a bed sore is. In this report I will be presenting a model of the PIMAP to help study how the parameters interact and how to further develop the system.

A.1 Model

The entire model can be broken down into five submodels: the human, an adc, the network sending to the server, the server, and the network sending back to the human. I use the following state variables to keep track of the system.

Human On Bed

variables: $posture_h, \tau_h$

input: $change_{ndn}$

output: $posture_h$

The variable $posture_h$ keeps track of what posture the human is lying on the bed. For simplicity $posture_h$ only has two states: $posture_h = 1$ if the human is lying on their left side and $posture_h = 2$ if the human is lying on their right side. The variable τ_h keeps track of the elapsed time the human is in a posture. τ_h counts up to $T_{mobility}$ minutes and the human then changes postures.

The human responds to input *change*, which is controlled by the server and signals what side the human should be lying on. For example if $change = 1$ the human will change to be in state $posture_h = 1$. For the mobility experiment this feature is turned off.

ADC Sampling

variables: $posture_{adc}, \tau_{adc}$

input: $posture_h$

output: $posture_{adc}$

The variable τ_{adc} keeps track of the time elapsed between samples. The variable counts up to T_{nmax}, the maximum network delay and samples $posture_h$ and stores it into $posture_{adc}$.

Up Network

variables: $posture_{nup}, \tau_{nup}, seq_{nup}$

input: $posture_{adc}$

output: $posture_{nup}, seq_{nup}$

The variable τ_{nup} keeps track of the time elapsed between packets sent to the server. For this project I am assuming the entire state can be sent in one packet. τ_{nup} is reset at a random time selected at reset between T_{nmin} and T_{nmax}. The network samples $posture_{adc}$ and stores it into $posture_{nup}$ as well as incrementing seq_{nup}.

Server

variables: $posture_s, \tau_s, i, p_l, p_r, change_s, risk$

input: $posture_{nup}$

output: $change_s, risk$

The variable τ_s keeps track of the elapsed time and resets every T_s. At every reset the server stores the state information it receives from the network, $posture_{nup}$ into the variable $posture_s$. The variable i keeps track of the index to store the pressure data, which is stored into a pressure map. The variables p_l and p_r keep track of the current amount of pressure based on the current window size. When the current amount of pressure exceeds P_{warn}, $change_s$ indicates what state the human should switch to.

Down Network

variables: $\tau_{ndn}, change_{ndn}$

The down network sends the information from the server to the human. The variable τ_{ndn} keeps track of the network delay. The variable $change_{ndn}$ stores the $change_s$ variable from the server.

Each submodel is discussed in depth below and the entire model can be seen in Figure A.1.

A.1.1 Global Variables

$T_{mobility}$, the amount of minutes until the human model changes positions.

T_{nmin}, the minimum network delay.

T_{nmax}, the maximum network delay.

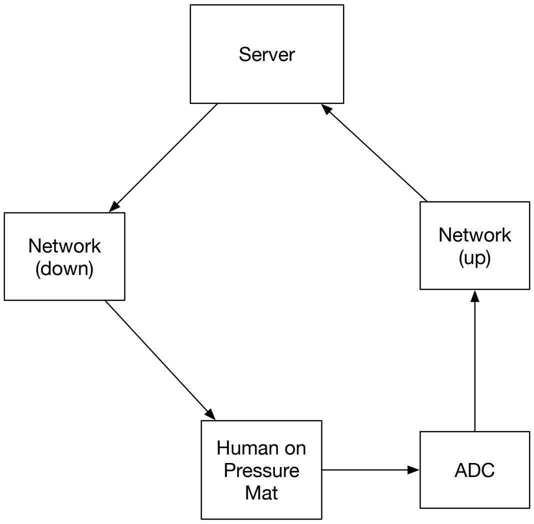


Figure A.1: A Block Diagram Of The Entire CPS Model

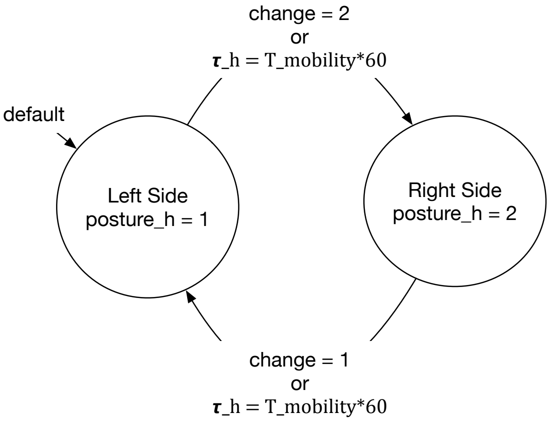


Figure A.2: The FSM Used For The Simple Human Model

A.1.2 Human

There are no human sleeping models that I am aware of. So I use a very simple model. The human can either be in two states, left side or right side, which correspond to the side the human is sleeping on. The human moves based on a set $T_{mobility}$.

Another features is that the human will be moved by an external force *change*, where $change = x$ will move the human to state $posture_h = x$. If $change = -1$ this signals that the human should ignore the input *change*.

This simple human model results in the FSM seen in Figure A.2.

```
pimg_array =
  {left_pimg, right_pimg}

left_state = 1
% Return left pressure image
pimg_array{left_state}

right_state = 2
% Return right pressure image
pimg_array{right_state}
```

Figure A.3: Example Of How The State And Pressure Images Are Related

Implementation

The left and right side state are stored as 2-D arrays and can be thought of as gray-scale images, where each pixel represents pressure instead of color. The left and right images are stored into a cell array where the index corresponds to the state. The left state is defined to be 1 and the right state is defined to be 2. An example relating the human state and the pressure images can be seen in Figure A.3.

The model can be represented using the following differential and difference equations.

$$u = (change_{ndn}) \in \{\{-1, 1, 2\}\}$$

$$x = (posture_h, \tau_h) \in \{\{1, 2\} \times [0, \infty)\}$$

$$\dot{x} = \begin{bmatrix} \dot{posture}_h \\ \dot{\tau}_h \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \text{Flow Condition: } \tau_h <= T_{mobility}$$

$$x^+ = \begin{bmatrix} posture_h^+ \\ \tau_h^+ \end{bmatrix} = \begin{bmatrix} 3 - posture_h \\ 0 \end{bmatrix} \qquad \text{Jump Condition: } \tau_h > T_{mobility}$$

$$x^+ = \begin{bmatrix} posture_h^+ \\ \tau_h^+ \end{bmatrix} = \begin{bmatrix} change_{ndn} \\ 0 \end{bmatrix} \qquad \text{Jump Condition: } posture_h \neq change_{ndn} \wedge change_{ndn} \neq -1$$

A.1.3 ADC

The state of the human on the pressure mat is sampled by an ADC. The pressure mat will be battery powered and therefore it is necessary to conserve power. One of the largest power-saving strategies is to decrease the sample rate. Therefore it is necessary to use an ADC and sample only as needed.

The sample rate of the ADC and the network delay to reach the server in the next stage are related. If the sample rate is faster than the network delay the network will not be able to deliver the information fast enough and old data in the buffer will either be overwritten or incoming data will be dropped. In the simulations the network does not actually have a buffer, but I still implemented the design with this idea in mind. In order to avoid this problem I sampled the ADC the worst case network delay. The ADC samples the pressure data, $posture_h$ every T_{nmax} seconds.

Sampling this infrequently is not ideal. Future work will be to identify bottlenecks in the network and minimize them as much as possible, such as sampling many times and then sending a packet as opposed to sending a packet every time.

Implementation

The model can be represented using the following differential and difference equations.

$$u = (posture_h) \in \{\{1, 2\}\}$$

$$x = (posture_{adc}, \tau_{adc}) \in \{\{1, 2\} \times [0, \infty)\}$$

$$\dot{x} = \begin{bmatrix} posture\dot{}_{adc} \\ \dot{\tau_{adc}} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \qquad \text{Flow Condition: } \tau_{adc} <= T_{nmax}$$

$$x^+ = \begin{bmatrix} posture^+_{adc} \\ \tau^+_{adc} \end{bmatrix} = \begin{bmatrix} posture_h \\ 0 \end{bmatrix} \qquad \text{Jump Condition: } \tau_{adc} > T_{nmax}$$

A.1.4 Up Network

In order to get the pressure data to the server it must be sent through a network. The network delay is modeled using the constants T_{nmin} the minimum network delay and T_{nmax} the maximum network delay. The network delivers the pressure mat data, $posture_{adc}$ at a random interval between $[T_{nmin}, T_{nmax}]$. I am also assuming the pressure mat data will be delivered in one packet, or another way to look at it is that the network delay is for the entire transmission of the pressure data. The variable seq_{nup} indicates the current packet being sent.

Implementation

The model can be represented using the following differential and difference equations.

```
newNetDelay() {
  return Tnmin +
    rand()*(Tnmax - Tnmin)
}
```

$$u = (posture_{adc}) \in \{\{1, 2\}\}$$

$$x = (posture_{nup}, \tau_{nup}, seqout_{nup}) \in \{\{1, 2\} \times (-\infty, T_{nmax}] \times \mathbb{Z}\}$$

$$\dot{x} = \begin{bmatrix} \dot{posture}_{nup} \\ \dot{\tau}_{nup} \\ \dot{seqout}_{nup} \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \qquad \text{Flow Condition: } \tau_{nup} >= 0$$

$$x^{+} = \begin{bmatrix} posture_{nup}^{+} \\ \tau_{nup}^{+} \\ seqout_{nup}^{+} \end{bmatrix} = \begin{bmatrix} posture_{adc} \\ \text{newNetDelay()} \\ seqout_{nup} + 1 \end{bmatrix} \qquad \text{Jump Condition: } \tau_{nup} < 0$$

A.1.5 Server

The server is the brain behind the PIMAP system and keeps track of the pressure per body part over time. For simplicity I am only keeping track of two body parts for this project, left side and right side. Based on the state of the human I store the pressure of each body part each second. For instance if the patient is lying on their right side the server will store a 1 meaning pressure is applied on that side and record 0 for all other body parts that do not have pressure applied.

The pressure for each body part is stored in an array, where the index i keeps track of where each pressure data should be stored in the array. The array keeps track of pressure over time for three hours worth of data. An average moving window is also calculated each second for two hours worth of pressure data.

I define two thresholds, P_{warn}, which signals that the patient should rotate, and P_{max}, which signals that a bed sore has formed. The goal for the server is to control the patient so that a bed sore never forms or P_{max} is never reached.

Because of the simplicity of the model it is easy to calculate the two thresholds. I will set $P_{warn} = 7200$, meaning pressure was applied for two hours, the standard amount of time a nurse will rotate a patient, and $P_{max} = 9000$, meaning pressure was applied for 30 min longer than the patient should be rotated. In simulation I will tune values so that using this simple model a bed sore never forms. These values were only used for the closer to real world simulation. The simple simulations use much lower values.

Implementation

The model can be represented using the following differential and difference equations.

```
inc_i(i) {
  i++
  if i > window_size
    i = 1
  return i
}

sum_pressure(s) {
  reutrn sum(window_size(pmap{s}))
```

```
        }

        sum_pressure() {
          for m in pmap
            sum_p = sum(window_size(pmap{m}))
            if sum_p > Pwarn
              return m

          return 0
        }

        store_pressure(mprev, m, i) {
          if mprev != m
            log("position changed")
            log(mprev + "->" + m)
            log("after")
            log("avg(pmap{m} + "pressure")

          pmap{m}(i) = 1
        }

```

$$u = (posture_{nup}, seqout_{nup}) \in \{\{1,2\} \times \mathbb{Z}\}$$

$$x = (posture_s, \tau_s, seqin_s, seqout_s, change_s, mobility_s) \in$$

$$\{\{1,2\} \times (-\infty, T_{nmax}] \times \mathbb{Z} \times \mathbb{Z} \times \{-1,1,2\} \times [0,\infty)\}$$

$$\dot{x} = \begin{bmatrix} \dot{posture}_s \\ \dot{\tau}_s \\ \dot{seqin}_s \\ \dot{seqout}_s \\ \dot{change}_s \\ \dot{mobility}_s \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{Flow Condition: } \tau_s >= 0$$

$$x^+ = \begin{bmatrix} posture_s^+ \\ \tau_s^+ \\ seqin_s^+ \\ seqout_s^+ \\ change_s^+ \\ mobility_s^+ \end{bmatrix} = \begin{bmatrix} posture_{nup} \\ \text{newNetDelay()} \\ seqin_s \\ seqout_s + 1 \\ change_s \\ mobility_s \end{bmatrix} \qquad \text{Jump Condition: } \tau_s < 0 \wedge seqout_{nup} == seqin_s + 1$$

$$x^+ = \begin{bmatrix} posture_s^+ \\ \tau_s^+ \\ seqin_s^+ \\ seqout_s^+ \\ change_s^+ \\ mobility_s^+ \end{bmatrix} = \begin{bmatrix} posture_{nup} \\ \text{newNetDelay} \\ seqin_s \\ seqout_s + 1 \\ change_s \\ mobility_s \end{bmatrix} \qquad \text{Jump Condition: } seqout_{nup} == seqin_s + 1$$

A.1.6 Network Down

To get the signal from the server to the human it is sent through a network. The network sends the signal $change_s$ from the server to the human in the range of $[T_{nmin}, T_{nmax}]$.

Implementation

The model can be represented using the following differential and difference equations.

```
        new_net_delay() {
          return Tnmin +
            rand()*(Tnmax - Tnmin)
        }

```

$$u = (change_s, seqout_s) \in \{\{-1, 1, 2\} \times \mathbb{Z}\}$$

$$x = (change_{ndn}, seqin_{ndn}) \in \{\{1, 2\} \times \mathbb{Z}\}$$

$$\dot{x} = \begin{bmatrix} \dot{change}_{ndn} \\ \dot{seqin}_{ndn} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \text{Flow Condition: No Flows}$$

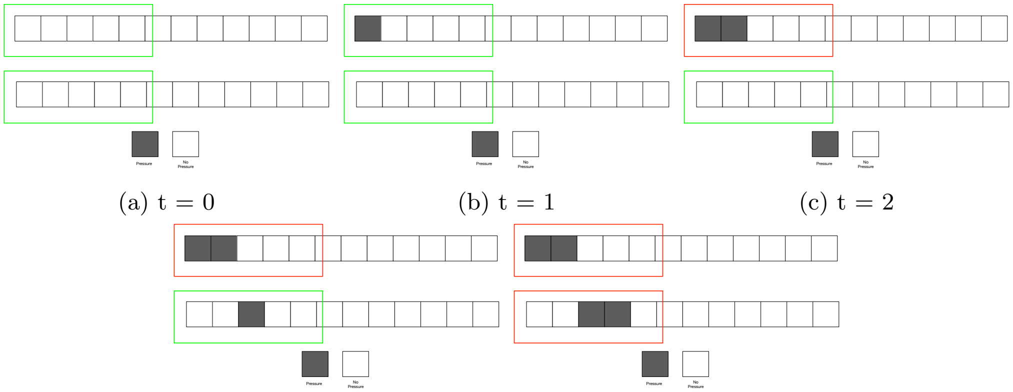
$$x^+ = \begin{bmatrix} change^+_{ndn} \\ seqin^+_{ndn} \end{bmatrix} = \begin{bmatrix} change_s \\ seqin_{ndn} + 1 \end{bmatrix} \qquad \text{Jump Condition: } seqout_s == seqin_{ndn} + 1$$

A.2 Simulation

All simulation was created in Matlab using the Hybrid Equation Toolbox created by Ricardo Sanfelice. I ran five different simulations. Four were simpler simulations that were used to gain insight into the model. The fifth simulation uses parameters that are closer to the real implementation.

A.2.1 Simple Simulations

One thing I discovered when testing the model is that the window size used to calculate pressure and the P_{warn} threshold are closely related. If you use a window size that is smaller than P_{warn} given that pressure will always increase at a rate of one you will never reach P_{warn}, the server will never notify the patient to rotate, and the patient will form a bed sore.. If you use a window size that is larger than P_{warn} it ends up taking too long to switch states and a bed sore can form.



(a) t = 0 (b) t = 1 (c) t = 2

(d) t = 3 (e) t = 4

Figure A.4: Simple Window Size Example

The intuition behind this is that when the window size is larger than P_{warn} the threshold is reached before the entire window is filled. Now as time is increasing the amount of pressure does not decrease because even though we are moving through time the pressure that was above the threshold is still in the window. Meanwhile the server has signalled for the patient to switch sides and what can happen is that the patient can reach the threshold in both states because the window is too big. A visual representation of this can be seen in Figure A.4.

I ran four simulations where I varied P_{warn} while keeping all other parameters constant. For a list of the parameters used in these simple simulations see A.1. I set the values to unrealistic values in order to gain insight into how the model functioned. Then based on this insight I adjusted the parameters for the next section where I run a simulation that is closer to reality.

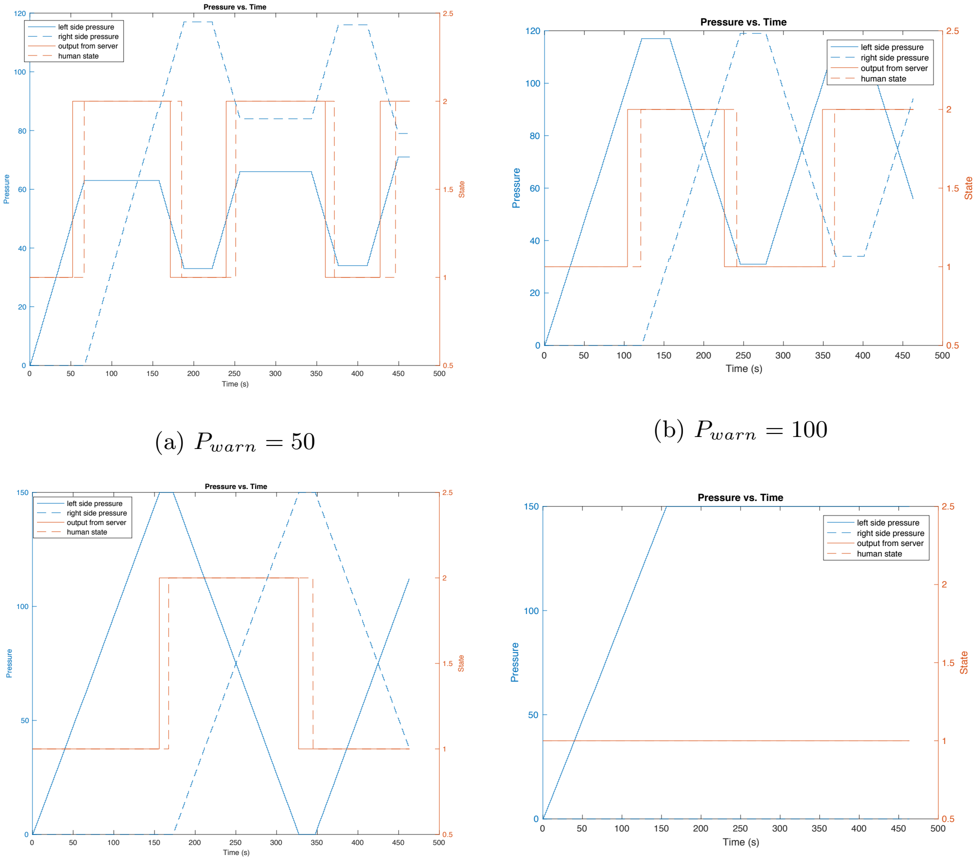
The simulations can be seen in Figure A.5. We can see the close to optimum window size is setting the size equal to P_{warn}.

A.2.2 Closer To Real World Simulation

I also ran a simulation, which is closer to real world parameters. The table of parameters is in Figure A.6. The notable changes is that I changed $P_{warn} = 7200$, or 2 hours as this is the required time that a patient must be rotated in a hospital, and accordingly I changed the window size to 7200. I also increased the network delay min and max to a more realistic delay, although more experimentation needs to be done with the network delay. See Figure A.7 for the simulation. Although a bed sore does not form on the imaginary patient there is an odd

Table A.1: Simulation Parameters For The Simple Simulations

Paramters	Values
Pressure History	10800
Pwarn	50, 100, 150, 200
Window Size	150
Tmin	1
Tmax	3
Ts	1
x0_h	[1]
x0_adc	[1; 0]
x0_nup	[1; Tnmax]
x0_s	[1; 0; 0; 0; 0; 1]
x0_ndn	[1; Tnmax]



(a) $P_{warn} = 50$

(b) $P_{warn} = 100$

(c) $P_{warn} = 150$

(d) $P_{warn} = 200$

Figure A.5: Comparison Of Different Window Sizes Used In The Simple Simulation

Paramters	Values
Pressure History	10800
Pwarn	7200
Window Size	7200
Tmin	3
Tmax	15
Ts	1
x0_h	[1]
x0_adc	[1; 0]
x0_nup	[1; Tnmax]
x0_s	[1; 0; 0; 0; 0; 1]
x0_ndn	[1; Tnmax]

Figure A.6: Simulation Parameters For The Closer To Real World Simulation

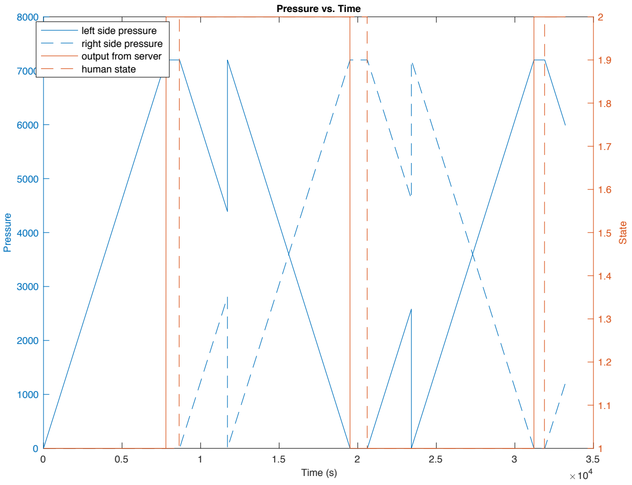


Figure A.7: A Closer To Real World Simulation

state transition at time 12000 and again at 24000. Because the simulation takes a considerable amount of time to run it was difficult to debug this particular problem since it only occurs during long running simulations.

A.3 Future Work

There were multiple parts to this project that I would like to improve. Currently I am just passing the state, either 1 or 2, to each model, but ideally I would like to pass the actual pressure data. This way I can test actual compression algorithms and also use computer vision techniques to distinguish body parts and how much pressure is applied to each part.

The network model could also be improved. Right now I am just using a pure delay. I don't want to implement an actual protocol, which be the most realistic, but treating the network as a simple delay is not ideal. It would be interesting to maybe use a packet model. I am also using a pure random delay for the network. This could be improved by using a delay based on packets in a buffer or a randomness that reflects more of delays in actual networks.

In the actual network I would use a low power radio and a bridge, but I treated the network that sends the packet from the pressure mat to the server as one network. I would like to break the network down into something that closer resembles the actual implementation.

A big future work project would be to create an actual human sleeping model. I don't have a lot of experience with human modelling, but an inbetween step would be to create more states that are based of of real pressure images of humans sleeping. This is relatively easy to do with image processing. It is more a matter of how to string the states together.

Another big future work would be to integrate some form of bed model, such as an air mattress with individual pockets of air, which could be controllable from the server. This way instead of needing a nurse to rotate a patient the bed itself would re-distribute the pressure.

A.4 Conclusion

This work will be extremely beneficial to creating a model for PIMAP. By creating this model I can study how the system performs and tweak it without having to modify actual hardware. There are many simplifications that are currently being used, but there is also a lot of room for improvement to create a close to realism model.

I was able to study the behavior of the model and realized how related the pressure thresholds are to the pressure window and in fact the close to optimum is solutions is to set the window equal to the warning threshold. In addition I was able to run a close to real world simulation that avoided bed sore formation in an imaginary patient.

Bibliography

[1] Karoon Agrawal and Neha Chauhan. "Pressure ulcers: Back to the basics". In: *Indian Journal of Plastic Surgery: Official Publication of the Association of Plastic Surgeons of India* 45.2 (2012), p. 244.

[2] Alisa Ahmetović et al. "Safety and feasibility of intermittent electrical stimulation for the prevention of deep tissue injury". In: *Advances in Wound Care* 4.3 (2015), pp. 192–201.

[3] H. Akbari and MA Younessi Heravi. "Designing and Constructing Blood Flow Monitoring System to Predict Pressure Ulcers on Heel". In: *Journal of Biomedical Physics & Engineering* 4.2 (2014), p. 61.

[4] Jenny Alderden et al. "Risk factors for pressure injuries among critical care patients: A systematic review". In: *International Journal of Nursing Studies* 71 (June 1, 2017), pp. 97–114. ISSN: 0020-7489. DOI: 10.1016/j.ijnurstu.2017.03.012.

[5] Dan L. Bader and Peter R. Worsley. "Technologies to monitor the health of loaded skin tissues". In: *BioMedical Engineering OnLine* 17 (Apr. 12, 2018), p. 40. ISSN: 1475-925X. DOI: 10.1186/s12938-018-0470-z.

[6] Valentina Baljak et al. "A scalable realtime analytics pipeline and storage architecture for physiological monitoring big data". In: *Smart Health.* CHASE 2018 Special Issue 9-10 (Dec. 1, 2018), pp. 275–286. ISSN: 2352-6483. DOI: 10.1016/j.smhl.2018.07.013.

[7] Robert Behrendt et al. "Continuous Bedside Pressure Mapping and Rates of Hospital-Associated Pressure Ulcers in a Medical Intensive Care Unit". In: *American Journal of Critical Care* 23.2 (Mar. 1, 2014), pp. 127–133. ISSN: 1062-3264, 1937-710X. DOI: 10.4037/ajcc2014192.

[8] Stephanie L Bennett, Rafik Goubran, and Frank Knoefel. "Long term monitoring of a pressure ulcer risk patient using thermal images". In: *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE.* IEEE, 2017, pp. 1461–1464.

[9] Dan R Berlowitz and David M Brienza. "Are all pressure ulcers the result of deep tissue injury? A review of the literature." In: *Ostomy/Wound Management* 53.10 (2007), pp. 34–38.

[10] John Bethencourt, Amit Sahai, and Brent Waters. "Ciphertext-Policy Attribute-Based Encryption". In: *2007 IEEE Symposium on Security and Privacy (SP '07).* 2007 IEEE Symposium on Security and Privacy (SP '07). Berkeley, CA: IEEE, May 2007, pp. 321–334. ISBN: 978-0-7695-2848-9. DOI: 10.1109/SP.2007.11.

[11] Joyce M Black et al. "Medical device related pressure ulcers in hospitalized patients". In: *International Wound Journal* 7.5 (2010), pp. 358–365.

[12] Joyce M Black et al. "Pressure ulcers: avoidable or unavoidable? Results of the national pressure ulcer advisory panel consensus conference". In: *Ostomy-Wound Management* 57.2 (2011), p. 24.

[13] Barbara J Braden and Nancy Bergstrom. "Clinical utility of the Braden scale for predicting pressure sore risk". In: *Advances in Skin & Wound Care* 2.3 (1989), pp. 44–51.

[14] Barbara Braden and Nancy Bergstrom. "A Conceptual Schema for the Study of the Etiology of Pressure Sores". In: *Rehabilitation Nursing* 12.1 (1987), pp. 8–16.

[15] *California Consumer Privacy Act of 2018.*

[16] M. B. A. Catherine VanGilder et al. "Results of the 2008–2009 International Pressure Ulcer Prevalence$^{\text{TM}}$ Survey and a 3-year, acute care, unit-specific analysis". In: *Ostomy Wound Management* 55.11 (2009), pp. 39–45.

[17] Emanuele Cereda et al. "A nutritional formula enriched with arginine, zinc, and antioxidants for the healing of pressure ulcers: a randomized trial". In: *Annals of internal medicine* 162.3 (2015), pp. 167–174.

[18] M. C. Chang et al. "In-bed patient motion and pose analysis using depth videos for pressure ulcer prevention". In: *2017 IEEE International Conference on Image Processing (ICIP).* 2017 IEEE International Conference on Image Processing (ICIP). Sept. 2017, pp. 4118–4122. DOI: `10.1109/ICIP.2017.8297057`.

[19] P. Chanyagorn and W. Chanyagorn. "Wireless activity reminder system for pressure ulcer prevention in IPD patients". In: *2017 International Electrical Engineering Congress (iEECON).* 2017 International Electrical Engineering Congress (iEECON). Mar. 2017, pp. 1–4. DOI: `10.1109/IEECON.2017.8075810`.

[20] Fiona M Coyer, Nancy A Stotts, and Virginia Schmied Blackman. "A prospective window into medical device-related pressure ulcers in intensive care". In: *International Wound Journal* 11.6 (2014), pp. 656–664.

[21] Aidan Curtis et al. "HealthSense: Software-defined Mobile-based Clinical Trials". In: *The 25th Annual International Conference on Mobile Computing and Networking - MobiCom '19.* The 25th Annual International Conference. Los Cabos, Mexico: ACM Press, 2019, pp. 1–15. ISBN: 978-1-4503-6169-9. DOI: `10.1145/3300061.3345433`.

[22] Tom Defloor et al. "Pressure ulcer prevention and repositioning". In: *Science and Practice of Pressure Ulcer Management.* Springer, 2006, pp. 67–73. ISBN: 1-85233-839-3.

[23] David Diaz et al. "Pressure injury prediction using diffusely scattered light". In: *Journal of Biomedical Optics* 22.2 (Feb. 2017), p. 025003. ISSN: 1083-3668, 1560-2281. DOI: `10.1117/1.JBO.22.2.025003`.

[24] T Dierks and E Rescorla. *RFC5246: The Transport Layer Security (TLS) Protocol Version 1.2.* Aug. 2008.

[25] Adam Dunkels, Björn Grönvall, and Thiemo Voigt. "Contiki-a lightweight and flexible operating system for tiny networked sensors". In: *Local Computer Networks, 2004. 29th Annual IEEE International Conference on.* IEEE, 2004, pp. 455–462.

[26] Laura E. Edsberg et al. "Unavoidable pressure injury: state of the science and consensus outcomes". In: *Journal of Wound Ostomy & Continence Nursing* 41.4 (2014), pp. 313–334.

[27] *Eliminating serious, preventable, and costly medical errors—never events.* Centers for Medicare & Medicaid Services, 2006.

[28] William J. Ennis et al. "Advanced Technologies to Improve Wound Healing: Electrical Stimulation, Vibration Therapy, and Ultrasound—What Is the Evidence?" In: *Plastic and Reconstructive Surgery* 138.3 (Sept. 2016), 94S. ISSN: 0032-1052. DOI: `10.1097/PRS.0000000000002680`.

[29] Lorenz Esch et al. “MNE Scan: Software for real-time processing of electrophysiological data”. In: *Journal of Neuroscience Methods* 303 (June 1, 2018), pp. 55–67. ISSN: 1872-678X. DOI: 10.1016/j.jneumeth.2018.03.020.

[30] Farve Daneshvar Fard, Sahar Moghimi, and Reza Lotfi. “Pressure ulcer risk assessment by monitoring interface pressure and temperature”. In: Electrical Engineering (ICEE), 2013 21st Iranian Conference on. IEEE, May 2013, pp. 1–5. ISBN: 978-1-4673-5634-3. DOI: 10.1109/IranianCEE.2013.6599875.

[31] Muhammad Fahad Farooqui and Atif Shamim. “Low Cost Inkjet Printed Smart Bandage for Wireless Monitoring of Chronic Wounds”. In: *Scientific Reports* 6 (June 29, 2016), p. 28949. ISSN: 2045-2322. DOI: 10.1038/srep28949.

[32] Masoud Farshbaf et al. “Detecting high-risk regions for pressure ulcer risk assessment”. In: *Bioinformatics and Biomedicine (BIBM), 2013 IEEE International Conference on.* IEEE, 2013, pp. 255–260.

[33] Wei Gao et al. “Fully integrated wearable sensor arrays for multiplexed *in situ* perspiration analysis”. In: *Nature* 529.7587 (Jan. 2016), pp. 509–514. ISSN: 1476-4687. DOI: 10.1038/nature16521.

[34] Francisco Pedro García-Fernández et al. “A New Theoretical Model for the Development of Pressure Ulcers and Other Dependence-Related Lesions”. In: *Journal of Nursing Scholarship* 46.1 (2014), pp. 28–38.

[35] Amit Gefen. “Reswick and Rogers pressure-time curve for pressure ulcer risk. Part 1”. In: *Nursing Standard* 23.45 (2009), pp. 64–74.

[36] Tahereh Najafi Ghezeljeh et al. “The Comparison of the Effect of the Head of Bed Elevation to 30 and 45 Degreess on the Incidence of Ventilator Associated Pneumonia and the Risk for Pressure Ulcers: A Controlled Randomized Clinical Trial”. In: *Iranian Red Crescent Medical Journal* 19.7 (2017).

[37] Gorecki Claudia et al. “Impact of Pressure Ulcers on Quality of Life in Older Patients: A Systematic Review”. In: *Journal of the American Geriatrics Society* 57.7 (June 30, 2009), pp. 1175–1183. ISSN: 0002-8614. DOI: 10.1111/j.1532-5415.2009.02307.x.

[38] Vipul Goyal et al. “Attribute-based encryption for fine-grained access control of encrypted data”. In: *Proceedings of the 13th ACM conference on Computer and communications security - CCS ’06.* the 13th ACM conference. Alexandria, Virginia, USA: ACM Press, 2006, pp. 89–98. ISBN: 978-1-59593-518-2. DOI: 10.1145/1180405.1180418.

[39] Nikolaus Gravenstein, Johannes H. van Oostrom PhD, and Lawrence J. Caruso. “Patient repositioning and pressure ulcer risk-Monitoring interface pressures of at-risk patients”. In: *Journal of Rehabilitation Research and Development* 50.4 (2013), p. 477.

[40] Timo Grimm et al. “Sleep position classification from a depth camera using bed aligned maps”. In: *Pattern Recognition (ICPR), 2016 23rd International Conference on.* IEEE, 2016, pp. 319–324.

[41] KE Groth. “Clinical observations and experimental studies of the pathogenesis of decubitus ulcers”. In: *Acta Chir Scand* 87.76 (1942), pp. 1–209.

[42] Lena Gunningberg and Cheryl Carli. “Reduced pressure for fewer pressure ulcers: can real-time feedback of interface pressure optimise repositioning in bed?” In: *International Wound Journal* 13.5 (2016), pp. 774–779.

[43] Lena Gunningberg et al. “Pressure mapping to prevent pressure ulcers in a hospital setting: A pragmatic randomised controlled trial”. In: *International Journal of Nursing Studies* 72 (2017), pp. 53–59.

[44] Seungyong Han et al. "Battery-free, wireless sensors for full-body pressure and temperature mapping". In: *Science Translational Medicine* 10.435 (Apr. 4, 2018), eaan4950. ISSN: 1946-6234, 1946-6242. DOI: 10.1126/scitranslmed.aan4950.

[45] Dieter Hayn et al. "An eHealth system for pressure ulcer risk assessment based on accelerometer and pressure data". In: *Journal of Sensors* 2015 (2015).

[46] Takuya Higashino et al. "Combination of thermographic and ultrasonographic assessments for early detection of deep tissue injury". In: *International Wound Journal* 11.5 (2014), pp. 509–516. ISSN: 1742-481X. DOI: 10.1111/j.1742-481X.2012.01117.x.

[47] *HIPAA Administrative Simplification.* Regulation Text 45 CFR Parts 160, 162, and 164. U.S. Department of Health and Human Services Office for Civil Rights, Mar. 26, 2013.

[48] Hopkins Alison et al. "Patient stories of living with a pressure ulcer". In: *Journal of Advanced Nursing* 56.4 (Sept. 12, 2006), pp. 345–353. ISSN: 0309-2402. DOI: 10.1111/j.1365-2648.2006.04007.x.

[49] Enamul Hoque, Robert F Dickerson, and John A Stankovic. "Monitoring body positions and movements during sleep using wisps". In: *Wireless Health 2010.* ACM, 2010, pp. 44–53.

[50] C. C. Hsia et al. "Analysis and comparison of sleeping posture classification methods using pressure sensitive bed system". In: *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE.* IEEE, 2009, pp. 6131–6134.

[51] Chi-Chun Hsia et al. "Bayesian classification for bed posture detection based on kurtosis and skewness estimation". In: *HealthCom 2008 - 10th International Conference on e-health Networking, Applications and Services.* HealthCom 2008 - 10th International Conference on e-health Networking, Applications and Services. July 2008, pp. 165–168. DOI: 10.1109/HEALTH.2008.4600129.

[52] Weimin Huang et al. "Multimodal sleeping posture classification". In: *Pattern Recognition (ICPR), 2010 20th International Conference on.* IEEE, 2010, pp. 4336–4339.

[53] Tafazzul Husain. "An experimental study of some pressure effects on tissues, with reference to the bed-sore problem". In: *The Journal of Pathology* 66.2 (1953), pp. 347–358.

[54] *InfluxDB.* URL: influxdata.com.

[55] Steve Gail Johnson et al. *System and method for using a video monitoring system to prevent and manage decubitus ulcers in patients.* Google Patents, Mar. 2014.

[56] *Kafka.* URL: kafka.apache.org.

[57] Lisa Kawasaki et al. "The mechanisms and evidence of efficacy of electrical stimulation for healing of pressure ulcer: A systematic review". In: *Wound Repair and Regeneration* 22.2 (2014), pp. 161–173. ISSN: 1524-475X. DOI: 10.1111/wrr.12134.

[58] Ahsan H Khandoker, Marimuthu Palaniswami, and Chandan K Karmakar. "Support vector machines for automated recognition of obstructive sleep apnea syndrome from ECG recordings". In: *IEEE Transactions on Information Technology in Biomedicine* 13.1 (2009), pp. 37–48.

[59] Holly Kirkland-Kyhn, Oleg Teleten, and Machelle Wilson. "A Retrospective, Descriptive, Comparative Study to Identify Patient Variables That Contribute to the Development of Deep Tissue Injury Among Patients in Intensive Care Units." In: *Ostomy/wound Management* 63.2 (2017), pp. 42–47.

[60] M. Kosiak. "Etiology of decubitus ulcers". In: *Archives of Physical Medicine and Rehabilitation* 42 (Jan. 1961), pp. 19–29. ISSN: 0003-9993.

[61] Michael Kosiak. “Etiology and pathology of ischemic ulcers.” In: *Archives of Physical Medicine and Rehabilitation* 40.2 (1959), pp. 62–69.

[62] Richard M. Kwasnicki et al. “A lightweight sensing platform for monitoring sleep quality and posture: a simulated validation study”. In: *European Journal of Medical Research* 23 (May 30, 2018), p. 28. ISSN: 2047-783X. DOI: 10.1186/s40001-018-0326-9.

[63] Gero Langer and Astrid Fink. “Nutritional interventions for preventing and treating pressure ulcers”. In: *The Cochrane Library*. John Wiley & Sons, Ltd, June 12, 2014.

[64] Hong Ji Lee et al. “Estimation of body postures on bed using unconstrained ECG measurements”. In: *IEEE Journal of Biomedical and Health Informatics* 17.6 (2013), pp. 985–993.

[65] Jeffrey M. Levine. “100 Years of Bedsores: How Much Have We Learned?” In: *Advances in Skin & Wound Care* 31.3 (Mar. 2018), pp. 139–141. ISSN: 1527-7941. DOI: 10.1097/01.ASW.0000530066.59878.2b.

[66] Y. Y. Li et al. “Sleep posture classification with multi-stream CNN using vertical distance map”. In: *2018 International Workshop on Advanced Image Technology (IWAIT)*. 2018 International Workshop on Advanced Image Technology (IWAIT). Jan. 2018, pp. 1–4. DOI: 10.1109/IWAIT.2018.8369761.

[67] Eran Linder-Ganz et al. “Assessment of mechanical conditions in sub-dermal tissues during sitting: a combined experimental-MRI and finite element approach”. In: *Journal of Biomechanics* 40.7 (2007), pp. 1443–1454.

[68] Jason J. Liu et al. “Bodypart localization for pressure ulcer prevention”. In: *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*. IEEE, 2014, pp. 766–769.

[69] Jason J. Liu et al. “Sleep posture analysis using a dense pressure sensitive bedsheet”. In: *Pervasive and Mobile Computing*. Selected Papers from the Eleventh Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2013) 10 (Feb. 1, 2014), pp. 34–50. ISSN: 1574-1192. DOI: 10.1016/j.pmcj.2013.10.008.

[70] Wei-Min Liu et al. “Multimodal and Multispectral Imaging for Chronic Pressure Ulcer Assessment”. In: *Proceedings of the 2nd International Conference on Biomedical Signal and Image Processing*. ICBIP 2017. New York, NY, USA: ACM, 2017, pp. 47–52. ISBN: 978-1-4503-5268-0. DOI: 10.1145/3133793.3133802.

[71] Ahmed Lounis et al. “Healing on the cloud: Secure cloud architecture for medical wireless sensor networks”. In: *Future Generation Computer Systems* 55 (2016), pp. 266–277. DOI: 10.1016/j.future.2015.01.009.

[72] Mandy T. Lowery. “A pressure sore risk calculator for intensive care patients:‘the Sunderland experience’”. In: *Intensive and Critical Care Nursing* 11.6 (1995), pp. 344–353.

[73] Valentina S. Lucas et al. “Utility of High-Frequency Ultrasound: Moving Beyond the Surface to Detect Changes in Skin Integrity”. In: *Plastic Surgical Nursing : Official Journal of the American Society of Plastic and Reconstructive Surgical Nurses* 34.1 (2014), pp. 34–38. ISSN: 0741-5206. DOI: 10.1097/PSN.0000000000000031.

[74] Courtney H. Lyder et al. “Hospital-Acquired pressure ulcers: Results from the national medicare patient safety monitoring system study”. In: *Journal of the American Geriatrics Society* 60.9 (2012), pp. 1603–1608.

[75] S. Mansfield, K. Obraczka, and S. Roy. “Pressure Injury Prevention: A Survey”. In: *IEEE Reviews In Biomedical Engineering* (2019), pp. 352–368. ISSN: 1941-1189. DOI: 10.1109/RBME.2019.2927200.

[76] Sam Mansfield et al. "Objective Pressure Injury Risk Assessment Using A Wearable Pressure Sensor". In: *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). San Diego, CA, USA: IEEE, Nov. 2019, pp. 1561–1568. ISBN: 978-1-72811-867-3. DOI: 10.1109/BIBM47256.2019.8982939.

[77] F. G. Marchione, L. M. Q. Araújo, and L. V. Araújo. "Approaches that use software to support the prevention of pressure ulcer: A systematic review". In: *International Journal of Medical Informatics* 84.10 (Oct. 1, 2015), pp. 725–736. ISSN: 1386-5056. DOI: 10.1016/j.ijmedinf.2015.05.013.

[78] *Matplotlib*. URL: matplotlib.org.

[79] Elizabeth McInnes et al. "Support surfaces for pressure ulcer prevention". In: *Cochrane Database of Systematic Reviews* 9 (2015), pp. 1–119.

[80] John McNeill et al. "Wearable wireless sensor patch for continuous monitoring of skin temperature, pressure, and relative humidity". In: *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–4.

[81] Akitsugu MISAKI et al. "Body Pressure Sensing Mattress for Bedsore Prevention". In: *SEI Technical Review* 78 (2014), p. 95.

[82] Zena Moore et al. "Subepidermal moisture (SEM) and bioimpedance: a literature review of a novel method for early detection of pressure-induced tissue damage (pressure ulcers)". In: *International Wound Journal* 14.2 (2016), pp. 331–337. ISSN: 1742-481X. DOI: 10.1111/iwj.12604.

[83] Sanaz Rahimi Moosavi et al. "End-to-end security scheme for mobility enabled healthcare Internet of Things". In: *Future Generation Computer Systems* 64 (Nov. 2016), pp. 108–124. ISSN: 0167739X. DOI: 10.1016/j.future.2016.02.020.

[84] Satoshi Nakamoto. *Bitcoin: A peer-to-peer electronic cash system.* Manubot, 2019.

[85] *National Pressure Ulcer Advisory Panel (NPUAP) announces a change in terminology from pressure ulcer to pressure injury and updates the stages of pressure injury — The National Pressure Ulcer Advisory Panel - NPUAP*. URL: http://www.npuap.org/national-pressure-ulcer-advisory-panel-npuap-announces-a-change-in-terminology-from-pressure-ulcer-to-pressure-injury-and-updates-the-stages-of-pressure-injury/ (visited on 04/18/2018).

[86] Sebastian Neubert et al. "Multi-Sensor-Fusion Approach for a Data-Science-Oriented Preventive Health Management System: Concept and Development of a Decentralized Data Collection Approach for Heterogeneous Data Sources". In: *International Journal of Telemedicine and Applications* 2019 (Oct. 8, 2019), pp. 1–18. ISSN: 1687-6415, 1687-6423. DOI: 10.1155/2019/9864246.

[87] Stacy Newbern. "Why your facility needs a full-time certified wound care nurse:" in: *Nursing* 48.2 (Feb. 2018), pp. 66–68. ISSN: 0360-4039. DOI: 10.1097/01.NURSE.0000529914.98433.76.

[88] J. C. L. Neyens et al. "Arginine-enriched oral nutritional supplementation in the treatment of pressure ulcers: A literature review". In: *Wound Medicine* 16 (Mar. 1, 2017), pp. 46–51. ISSN: 2213-9095. DOI: 10.1016/j.wndm.2016.07.002.

[89] Van Nguyen, Abdul Q Javaid, and Mary A Weitnauer. "Detection of motion and posture change using an IR-UWB radar". In: *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*. IEEE, 2016, pp. 3650–3653.

[90] Hareem Nisar et al. “An electrical stimulation based therapeutic wearable for pressure ulcer prevention”. In: *Biomedical Engineering and Sciences (IECBES), 2016 IEEE EMBS Conference on*. IEEE, 2016, pp. 411–414.

[91] Doreen Norton, Rhoda McLaren, and Arthur Norman Exton-Smith. *An investigation of geriatric nursing problems in hospital*. Churchill Livingstone Edinburgh, 1962.

[92] NPUAP. *National Pressure Ulcer Advisory Panel Support Surface Standards Initiative*. Jan. 29, 2007.

[93] National Quality Forum (NQF). *Serious Reportable Events in Healthcare—2011 Update: A Consensus Report*. NQF, 2011.

[94] Lerit Nuksawn, Ekawit Nantajeewarawat, and Surapa Thiemjarus. “Real-time sensor- and camera-based logging of sleep postures”. In: *Computer Science and Engineering Conference (ICSEC), 2015 International*. IEEE, 2015, pp. 1–6.

[95] *NumPy*. URL: `numpy.org`.

[96] Sarah Ostadabbas et al. “In-bed posture classification and limb identification”. In: *Biomedical Circuits and Systems Conference (BioCAS), 2014 IEEE*. IEEE, 2014, pp. 133–136.

[97] Fredrik Osterlind et al. “Cross-level sensor network simulation with cooja”. In: *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. IEEE, 2006, pp. 641–648.

[98] *Part 11, Electronic Records; Electronic Signatures - Scope and Application*. Sept. 2003.

[99] Matthew J. Peterson et al. “Effects of turning on skin-bed interface pressures in healthy adults”. In: *Journal of Advanced Nursing* 66.7 (2010), pp. 1556–1564.

[100] David Pickham et al. “Effect of a wearable patient sensor on care delivery for preventing pressure injuries in acutely ill adults: A pragmatic randomized clinical trial (LS-HAPI study)”. In: *International Journal of Nursing Studies* 80 (2018), pp. 12–19.

[101] Anna Polak et al. “A Randomized, Controlled Clinical Study to Assess the Effect of Anodal and Cathodal Electrical Stimulation on Periwound Skin Blood Flow and Pressure Ulcer Size Reduction in Persons with Neurological Injuries.” In: *Ostomy/Wound Management* 64.2 (2018), pp. 10–29.

[102] Matthew Q. Pompeo. “Pressure map technology for pressure ulcer patients: can we handle the truth?” In: *Wounds: A Compendium of Clinical Research and Practice* 25.2 (2013), pp. 34–40.

[103] Jon Postel. *RFC0768: User Datagram Protocol*. RFC Editor, 1980.

[104] Jon Postel. *RFC0793: Transmission control protocol*. RFC Editor, 1981.

[105] Mary Ellen Posthauer et al. “The Role of Nutrition for Pressure Ulcer Management: National Pressure Ulcer Advisory Panel, European Pressure Ulcer Advisory Panel, and Pan Pacific Pressure Injury Alliance White Paper”. In: *Advances in Skin & Wound Care* 28.4 (Apr. 2015), p. 175. ISSN: 1527-7941. DOI: `10.1097/01.ASW.0000461911.31139.62`.

[106] M. Baran Pouyan, Mehrdad Nourani, and Matthew Pompeo. “Clustering-based limb identification for pressure ulcer risk assessment”. In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 4230–4233.

[107] M. B. Pouyan et al. “Continuous eight-posture classification for bed-bound patients”. In: *2013 6th International Conference on Biomedical Engineering and Informatics*. 2013 6th International Conference on Biomedical Engineering and Informatics. Dec. 2013, pp. 121–126. DOI: `10.1109/BMEI.2013.6746919`.

[108] Maziyar Baran Pouyan et al. "Automatic limb identification and sleeping parameters assessment for pressure ulcer prevention". In: *Computers in Biology and Medicine* 75 (2016), pp. 98–108.

[109] *Prometheus*. URL: prometheus.io.

[110] *Python Client for Apache Kafka*. URL: https://github.com/confluentinc/confluent-kafka-python.

[111] Uvais Qidwai et al. "Intelligent integrated instrumentation platform for monitoring long-term bedridden patients". In: *Biomedical Engineering and Sciences (IECBES), 2016 IEEE EMBS Conference on*. IEEE, 2016, pp. 561–564.

[112] Madhuri Reddy, Sudeep S. Gill, and Paula A. Rochon. "Preventing pressure ulcers: a systematic review". In: *Jama* 296.8 (2006), pp. 974–984.

[113] Gregory B Rehm et al. "Development of a research-oriented system for collecting mechanical ventilator waveform data". In: *Journal of the American Medical Informatics Association* 25.3 (Mar. 1, 2018), pp. 295–299. ISSN: 1067-5027, 1527-974X. DOI: 10.1093/jamia/ocx116.

[114] B. S. Renganathan et al. "A novel system to tackle hospital acquired pressure ulcers". In: *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). Aug. 2016, pp. 4780–4783. DOI: 10.1109/EMBC.2016.7591796.

[115] E. Rescorla and N. Modadugu. *RFC6347: Datagram Transport Layer Security Version 1.2*. Jan. 2012.

[116] J. B. Reswick and J. E. Rogers. "Experience at Rancho Los Amigos Hospital with devices and techniques to prevent pressure sores". In: *Bed sore biomechanics*. Springer, 1976, pp. 301–310.

[117] Anna Rodriguez et al. "Medical Device Connectivity Challenges Outline the Technical Requirements and Standards For Promoting Big Data Research and Personalized Medicine in Neurocritical Care". In: *Military Medicine* 183 (suppl_1 Mar. 1, 2018), pp. 99–104. ISSN: 0026-4075. DOI: 10.1093/milmed/usx146.

[118] C. Allison Russo, Claudia Steiner, and William Spector. "Hospitalizations related to pressure ulcers among adults 18 years and older, 2006: statistical brief# 64". In: (2006).

[119] Seied Hadi Saghaleini et al. "Pressure Ulcer and Nutrition". In: *Indian Journal of Critical Care Medicine : Peer-reviewed, Official Publication of Indian Society of Critical Care Medicine* 22.4 (Apr. 2018), pp. 283–289. ISSN: 0972-5229. DOI: 10.4103/ijccm.IJCCM_277_17.

[120] Amit Sahai and Brent Waters. "Fuzzy Identity-Based Encryption". In: *Advances in Cryptology – EUROCRYPT 2005*. Ed. by Ronald Cramer. Red. by David Hutchison et al. Vol. 3494. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 457–473. ISBN: 978-3-540-25910-7 978-3-540-32055-5.

[121] Nick Santamaria et al. "A randomised controlled trial of the effectiveness of soft silicone multi-layered foam dressings in the prevention of sacral and heel pressure ulcers in trauma and critically ill patients: the border trial". In: *International Wound Journal* 12.3 (2015), pp. 302–308.

[122] Ronald G. Scott and Kristen M. Thurman. "Visual feedback of continuous bedside pressure mapping to optimize effective patient repositioning". In: *Advances in Wound Care* 3.5 (2014), pp. 376–382.

[123] Raffaele Serra et al. “Low serum albumin level as an independent risk factor for the onset of pressure ulcers in intensive care unit patients”. In: *International Wound Journal* 11.5 (2014), pp. 550–553.

[124] Syed Aziz Shah et al. “Posture Recognition to Prevent Bedsores for Multiple Patients Using Leaking Coaxial Cable”. In: *IEEE Access* 4 (2016), pp. 8065–8072.

[125] David C. Shieh et al. “Dramatic Reduction in Hospital-Acquired Pressure Injuries Using a Pink Paper Reminder System:” in: *Advances in Skin & Wound Care* 31.3 (Mar. 2018), pp. 118–122. ISSN: 1527-7941. DOI: 10.1097/01.ASW.0000527966.72494.61.

[126] Aamir Siddiqui et al. “A continuous bedside pressure mapping system for prevention of pressure ulcer development in the medical ICU: a retrospective analysis”. In: *Wounds* 25.12 (2013), pp. 333–339.

[127] Toby E. Smith and Craig L. Cooper. *Method and apparatus for mitigating the risk of pressure sores.* Google Patents, May 2008.

[128] Timothy M Snavely. “A brief economic analysis of the looming nursing shortage in the United States”. In: *Nursing Economics* 34.2 (2016), p. 98.

[129] Leandro R. Solis et al. “Effects of intermittent electrical stimulation on superficial pressure, tissue oxygenation, and discomfort levels for the prevention of deep tissue injury”. In: *Annals of Biomedical Engineering* 39.2 (2011), pp. 649–663.

[130] Jocelyn Evelyn Songer. “Tissue ischemia monitoring using impedance spectroscopy: Clinical evaluation”. In: (2001).

[131] Christoph Stach, Frank Steimle, and Ana Cristina Franco da Silva. “TIROL: the extensible interconnectivity layer for mHealth applications”. In: *International Conference on Information and Software Technologies.* Springer, 2017, pp. 190–202.

[132] Sarah L Swisher et al. “Impedance sensing device enables early detection of pressure ulcers in vivo”. In: *Nature Communications* 6 (2015), ncomms7575.

[133] Farid Touati et al. “Feasibility and performance evaluation of a 6LoWPAN-enabled platform for ubiquitous healthcare monitoring”. In: *Wireless Communications and Mobile Computing* 16.10 (2016), pp. 1271–1281. ISSN: 1530-8677. DOI: 10.1002/wcm.2601.

[134] Jacquelynn P. Tran et al. “Prevention of Pressure Ulcers in the Acute Care Setting: New Innovations and Technologies”. In: *Plastic and Reconstructive Surgery* 138 (Sept. 2016), 232S–240S. ISSN: 0032-1052. DOI: 10.1097/PRS.0000000000002644.

[135] Ahmad Tubaishat et al. “Pressure Injuries Among Hospitalized Patients With Cancer: Prevalence and Use of Preventive Interventions”. In: *Journal of Wound Ostomy & Continence Nursing* 45.3 (2018), pp. 227–232.

[136] Md. Ashraf Uddin et al. “Continuous Patient Monitoring With a Patient Centric Agent: A Block Architecture”. In: *IEEE Access* 6 (2018), pp. 32700–32726. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2846779.

[137] National Pressure Ulcer Advisory Panel (US) and Emily Haesler. *Prevention and treatment of pressure ulcers: quick reference guide.* Cambridge Media, 2014.

[138] Catherine VanGilder et al. “The International Pressure Ulcer Prevalence$^{\mathrm{TM}}$ Survey: 2006-2015”. In: *Journal of Wound, Ostomy and Continence Nursing* 44.1 (2017), pp. 20–28.

[139] Judy Waterlow. “Pressure sores: a risk assessment card.” In: *Nursing Times* 81.48 (1985), pp. 49–55.

[140] X. Xu et al. “Body-Earth Mover’s Distance: A Matching-Based Approach for Sleep Posture Recognition”. In: *IEEE Transactions on Biomedical Circuits and Systems* 10.5 (Oct. 2016), pp. 1023–1035. ISSN: 1932-4545. DOI: 10.1109/TBCAS.2016.2543686.

[141] Heenam Yoon et al. “Estimation of sleep posture using a patch-type accelerometer based device”. In: *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*. IEEE, 2015, pp. 4942–4945.

[142] Rasoul Yousefi et al. “A smart bed platform for monitoring & Ulcer prevention”. In: *Biomedical Engineering and Informatics (BMEI), 2011 4th International Conference on*. Vol. 3. IEEE, 2011, pp. 1362–1366.

[143] Saldy Yusuf et al. “Microclimate and development of pressure ulcers and superficial skin changes”. In: *International Wound Journal* 12.1 (Feb. 1, 2015), pp. 40–46. ISSN: 1742-481X. DOI: 10.1111/iwj.12048.