ABSTRACT: In this paper, we present a system for monitoring the sleep and sleep posture of individuals residing in residential care facilities. In this system, a pressure sensing mat was developed to be placed on the mattress of the patient's bed. A data acquisition circuit collects data from the sensor mat and transmits it to the cloud for processing. An Android application allows caregivers to monitor the sleeping posture of the patient in real-time and an alert is produced if the patient is detected to be sleeping too close to the edge of the bed. Finally, experiments were undertaken to demonstrate the feasibility of our solution and we discuss future additions to our system.

1. Introduction

In New Zealand, the proportion of the total population aged 65 and over is increasing, which is a common trend in countries all around the world. In 2016, 15% of New Zealand's population was aged 65+, with the percentage estimated to reach 20% by 2032 [1]. The elderly tend to face higher rates of physical and cognitive decline that can cause aging-related diseases such as dementia and poor mobility. This may prevent them from living independently and they may require monitoring and/or assisted living in residential care facilities.

The elderly tend to suffer from poor sleep quality and this has effects on their physical health, cognitive function and overall quality of life [2]. These effects are more devastating on the elderly due to their physical status and can cause further sleep-related health issues or injuries. Studies have found that sleep issues were more prevalent within the residential care population [3]. Health issues stemming from poor sleep quality include weakened immune system and cardiovascular disease, and improper sleep posture can exacerbate sleep apnea. This leads in increased healthcare costs and can be a significant source of stress for the patient, as well as their family. Detecting and monitoring for these symptoms can be challenging and may also require use of extra staff resources. In 2014/2015, 60% of the $NZD983 million budget allocated to district health boards in New Zealand was spent towards aged residential care [4]. These costs are expected to rise in the future.

Falls are a major risk to the elderly and can lead to hospitalisation from injury, or even death in extreme cases. Falls can be attributed to fatigue or weak balance. Lack of sleep can cause fatigue, and poor sleep posture can affect the spine, which is mainly responsible for supporting the movement of a person [5]. Accident and compensation claims for falls in New Zealand have risen steadily in the last five years. In the 2017/2018 financial year, total costs of injury claims due to falls in the 65+ age bracket were NZD$191 million [6].

Another issue that is prevalent in the elderly population are pressure injuries (PI). These can be caused by prolonged sleep in a single posture without moving. The effects of PI can be constant pain, loss of mobility, depression, and even death. In 2016, a report into pressure injuries was commissioned by the New Zealand Ministry of Health that found that the prevalence of PI could be reduced if caregivers were more vigilant and took proactive action to prevent PI [7]. The risk of PI can be reduced if staff actively monitor and regularly change the patient's sleeping posture.

The challenges, rising costs of care and effects of sleep related issues on the elderly in New Zealand motivates the need for a system that can help medical practitioners and caregivers in residential care monitor their patients more efficiently. An autonomous system that monitors the sleep posture and records sleeping patterns is presented. An Android smartphone application provides caregivers and healthcare professionals with sleep related data, so they can implement preventative measures to reduce and manage the risks of poor sleep as necessary. The overall aim is to reduce the use of resources and burden on caregivers, while also improving the patient's sense of security and quality of life.

2. Background Information

2.1. Internet of Things (IoT)

The Internet of Things is a concept that is rapidly gaining popularity all over the world. It refers to the network of everyday items and devices that are embedded with electronics, software, sensors, actuators and internet connectivity that allows the collection and exchange of data. This allows the integration of the physical environment into digital systems and creates opportunities for automation, which produces economical and efficiency improvements. In this project, we follow an IoT approach in the development of our system. The sensor mat is sampled by our data acquisition circuit and information is sent to the database on the cloud via WiFi.

2.2. Cloud Computing

Use of cloud computing is another practice that is gaining popularity all over the world. Cloud computing is the use of shared pools of computing resources and services through the internet. It allows companies to store, manage and process data without the need to spend additional resources and costs on IT infrastructure. Cloud computing is scalable and flexible. We utilise cloud computing within our project as it reduces the need for server infrastructure. Cloud computing and IoT are concepts that accompany each other due their common use of the internet. In our project, we utilise an Elastic Compute 2 instance provided by Amazon Web Services (AWS). EC2 is a cloud virtual machine service that is flexible and allows you to run programs and scripts as you would with any desktop and OS.

2.3. Machine Learning

Machine learning is becoming popular around the world. It is a subset of artificial intelligence and enables computers to have the ability to self-learn without being programmed. When exposed to new data, these programs learn, grow, adapt and develop autonomously. Within our project, we use convolutional neural networks (CNN), which are a subset of machine learning, to classify sleeping postures. CNN's consist of a network of nodes, like the network of neurons in a brain. Each node processes a part of the input data. As a result, CNN's have been applied in computationally intensive image recognition tasks, with good performance. The CNN can consist of many layers, the more layers there are, the more features that can be recognised. An example of a 2D CNN is shown in Figure 1.

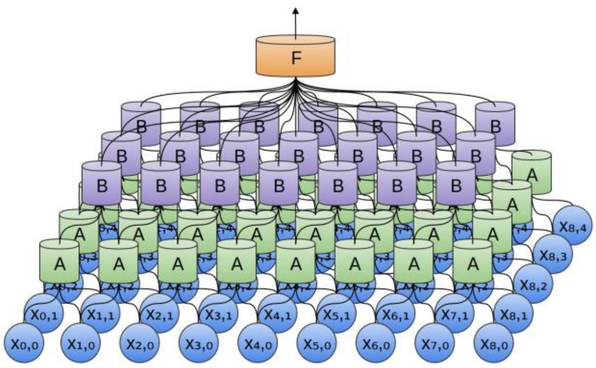

Figure 1: 2D Convolutional Neural Network [8]

2.4. TensorFlow

TensorFlow [9] is an open source machine learning framework developed by Google. Specifically, we use a CNN model called Inception-v3, which is the most recent model to be released. The last layer has been left untrained to cater for user specific applications. The Inception-v3 model was trained with a set of sleep posture images and we used it to classify new incoming data into a posture category.

2.5. Force Sensitive Resistor (FSR)

Force sensitive resistors [10] are simple tactile sensors that are used in applications where changes and differences in pressure need to be detected. They could me made using conductive polymers, elastomers or semiconducting polymers. They are also termed piezoresistive sensors, as the resistance is pressure dependent. In our research, we implement an array of FSR's to capture the pressure distribution of a person's sleeping posture.

3. Related Work

Currently, there are various solutions in that have been researched and implemented for the monitoring of sleep and/or sleep posture. These were impractical due to their high cost, low accuracy and unsuitability for use in residential care facility. Our research was focused on an implementation that resolved these issues. It should be easy for caregivers to monitor patients at the same time, without the need for extra exertion and tasks to be undertaken in their daily routines.

The current 'gold standard' in terms for diagnosing sleep disorders and issues is the use of polysomnography (PSG) [11]. This method involves the measurement of multiple physiological parameters, such as brain activity, blood oxygen level, heart rate, breathing, and leg and eye movements. This requires a lot of sensors and equipment to be physically attached to the patient's body. Although this method can yield extremely accurate results and insight into a person's sleep, this method is obtrusive and disruptive, so it may not reflect a normal period of sleep. This could be a source of inaccuracy or misdiagnosis. This approach is suitable for medical-supervised evaluations but was infeasible in terms of daily use.

WatchPAT [12] was a wrist-worn medical device with a finger clip, which allowed similar information to be obtained without the using PSG. This meant that the patient could be monitored within the comfort of their own home and usual sleeping environment. However, the device was required to be tightly worn, which would be uncomfortable to the elderly. Commercially available wrist worn sleep trackers could also be used. These are popular and widely available. Such examples were the Fitbit Charge2 [13] and the Jawbone UP3 [14]. As they are for the consumer market, only two or three parameters could be measured. Movement and heart rate were the most common metrics that could be monitored. Sleep could be tracked based on these two parameters alone, but research has shown that the accuracy of sleep monitoring by these devices were questionable [15]. In addition, wrist worn devices are battery powered and often require Bluetooth connectivity (in the case of consumer devices) to gather information. This would mean extra tasks for the caregivers to complete within their daily routines. Lastly, wrist worn systems cannot detect sleep posture.

Within our context, we considered monitoring methods that required physical contact with the patient, such as wrist worn devices, to be intrusive. There were also many alternative solutions that are based on non-intrusive monitoring methods. Smartphone applications such as ‘Sleep as Android’ [16] and ‘Runtastic Sleep Better’ [17] could monitor sleep using sensors built into the phone. The smartphone is required to be placed on the bed for it to work, which means one phone is required for each user. This would not be very economical when there are a lot of patients. Lastly, these require user intervention to start and stop the application which, again, adds more tasks for the caregivers.

As an alternative to smartphone-based systems, Emfit [18] developed a small sensing mat that was placed between the mattress and bed sheets. This system autonomously monitors sleep and transmits information through WiFi, which is ideal. A web application is also provided for users to view information about their sleep patterns. The main drawback was that it costs USD$299. In addition, the system was oriented to a single user. For our project, we envision a similar system, but at a cost that is accessible and has support for multiple patients.

4. System Architecture

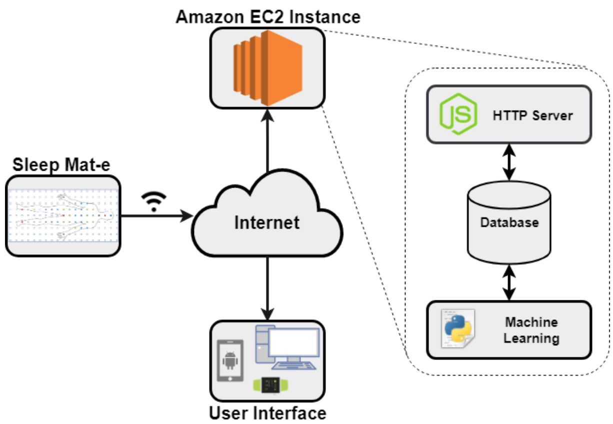

Figure 2: Sleep Mat-e system architecture

Our system architecture is shown in Figure 2. The goal of the project was to produce a sleep monitoring system catered for elderly in residential care facilities and their caregivers. Pressure data would be captured and after processing, useful results would be displayed to caregivers on an Android application. Firstly, the Sleep Mat-e hardware was developed. A thin plastic mat embedded with 171 custom-built Velostat FSR’s and data acquisition circuit with WiFi connectivity was to be located at each bed. This system would capture pressure distribution data and transmit it wirelessly through the internet to the cloud.

Within the cloud, a HTTP server was used to process incoming data and store it in a database. A script executing TensorFlow machine learning algorithm would run periodically and categorise new incoming data into one of six posture categories. This is further explained in the sections to follow.

An Android application for caregivers was set up to allow remote monitoring of patients. The live status of a patient’s sleep posture could be viewed through the application. The daily overall time in bed and split of times spent in each posture were also displayed. Finally, a fall warning is produced to alert caregivers whenever the patient sleeps too close to the edge of the bed.

5. Methodology

Many project phases were undertaken before a complete system could be produced. First, a way to capture the posture of a sleeping patient was researched. As our target demographic were elderly patients in residential care facilities, the sensor technology needed to be comfortable and non-invasive. A thin mat embedded with an array of FSR’s was developed. We found that a common bed sizes in care facilities were close to the dimensions of a king single bed. The mat was made large enough to completely cover a king single mattress. The resolution of the sensor array required and the number of sleeping postures we could identify were interdependent.

The sensor mat was constructed with an array of 171 FSR’s over a 1m x 2m mat as shown in Figure 3. The sensor array started 5cm from the edge of the mat, and the sensors were placed 10cm apart from each other, giving a total of 9 rows and 19 columns. The sensors were spaced 10cm apart to allow more sensors to be added if necessary. Testing was done with this configuration to view the pressure distribution of several different postures. However, due to the low resolution of the sensor array, we could only identify four main sleeping postures: face up, face down, left lateral and right lateral, as seen in Figure 4. We wanted to double the number of sensors rows and columns, but time constraints and material availability meant we had to continue with this configuration for the rest of the project.

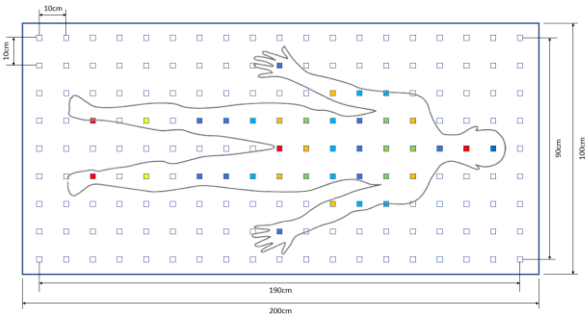

Figure 3: Sensing mat topology and dimensions

Figure 4: Sleep postures identified

To make our solution capable of autonomous monitoring, a data acquisition circuit and cloud processing system was implemented. This allowed the sensor values to be periodically captured and then transmitted through WiFi to the cloud without the need for caregivers to be physically present. A web server and database was set up to allow data to be received and stored. Machine learning was used to classify data into a specific posture category.

Lastly, a mobile application was created to provide a user interface for caregivers to monitor patients remotely. Data was accessed from the web server. Current and historical sleep data could be viewed, and the healthcare needs of the patient could be reviewed as necessary. The functionality of the whole system was tested by just using the mat and the live posture monitoring feature on the mobile application. Results in Section 7 were gathered this way.

6. Implementation

6.1. Connectivity

Choosing of the communication method between our data acquisition circuit and cloud server was an initial step in the project, as it laid the foundations for later development. Four wireless communication options were analysed: WiFi, Bluetooth, Zigbee and GSM. The key metrics that were desirable in our implementation were: long range, high data transfer rate, low complexity, reduced amount of equipment to use and maintain, and low cost.

Use of Bluetooth as the communication method would require an intermediate device to act as a receiver, such as a smartphone, to receive the data before further transmitting it to the cloud via the internet. This was not very suitable as there would need to be a smartphone nearby, with an application running and actively waiting for data at all times. This would make the application slow and power consuming. An alternative method was to use GSM, which is a long-range communication method and does not require any additional routers to be maintained. The drawbacks of GSM were that costs were associated with sending data each time. Furthermore, GSM modules and related electronics would need to be implemented into the data acquisition circuit which increases the complexity and cost. Zigbee is similar to Bluetooth in that it requires intermediate routers for data ‘hopping’ but has a longer transmission range. However, the protocol is proprietary and requires Zigbee certified transmitters and receivers to be implemented in a network. The last option considered was WiFi, which has a high data transfer rate, but is power intensive. This was chosen as our wireless communication method due to the wide accessibility and existing infrastructure in many buildings. The effects of high-power consumption were negated as we were not aiming to operating our electronics using batteries.

6.2. Hardware

The development of a sensor mat was a crucial initial step in the research project. Dimensions of standard bed sizes in New Zealand were researched and it was found that a king single size mattress was the middle-ground in terms of size for a single person. The dimensions of the sensor mat to be developed were to correspond with a king single mattress (92 cm x 203 cm). The sensing mat was envisioned to be placed on top of the mattress, but underneath the bed sheets. Therefore, the sensor mat was required to be thin and flexible, to be as unobtrusive to the patient as possible. Various materials and sensors were researched to find the most suitable option.

As we were interested in determining sleeping posture, sensors capable of capturing the pressure distribution and contact area between the body and the mattress were required. Options considered for evaluation were: premade mattress sensor arrays, limit switches, FSR's and piezoresistive materials. They were judged according to their physical and technical specifications, in addition to costs. The options were analysed against the ideal sensor criteria:

1. Flat sensor shape
2. Able to detect a large pressure range
3. Low cost

Flat sensors were required as the sensing mat was to be as unobtrusive as possible, as mentioned earlier, to gather accurate information of the patient in their most natural state during sleep. The sensors needed to be reactive to a wide range of different pressure values to return useful data and increase the accuracy of the posture detection process. This property was also ideal so that individuals of varying body weights and sizes could use that mat and still yield optimal results. Finally, the sensor mat should aim to be cost effective to ensure that it is readily accessible to many users.

Specialised pressure mapping sensors integrated large arrays of FSR's into thin profile mats. These could provide extremely accurate and high-resolution images as they had very high sensor densities, with many available solutions containing thousands of pressure sensors. An example of the sensing capabilities of these systems is shown in Figure 5. The pressure ranges were also found to be very large. Although they were very suitable for our project, the costs of these systems were very high, upwards of thousands of $NZD, which meant it was unsuitable for widespread use. In addition, proprietary hardware such as data acquisition units and bundled visualisation software meant that integration into our project would be difficult.

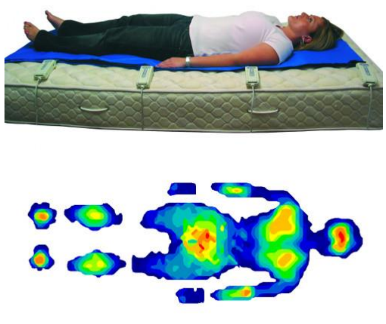

Figure 5: Tekscan BPMS System [19]

Limiting switches were researched as an alternative to pressure mapping sensors. The advantages of using switches as the sensors were that they would be easier to interface, as no voltage dividers are required, which is the case with FSR's. The compromise would be that pressure sensing data would be lost, as the switches only return binary data. When deployed in an array, only a binary image of the pressure distribution can be obtained. In the context of our application, trying to differentiate between a face-up or a face-down posture would be difficult, as these postures yield almost identical binary images. An example of a binary image is shown in Figure 6. Furthermore, many commercially available options required large activation forces, as they were designed for applications where detection of a person sitting or stepping on an area were required.

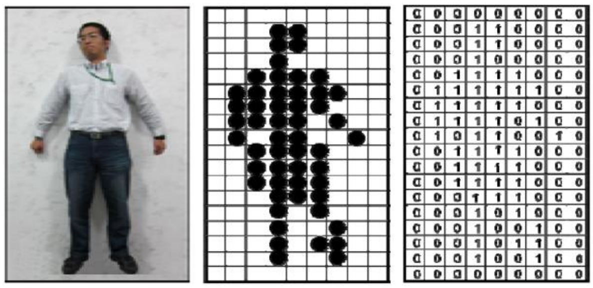

Figure 6: Binary image of sensing mattress [20]

The third option considered were FSR's. These were the same type of sensor technology used in pressure mapping systems. However, by using FSR's we could produce our own mat system using however many sensors that were required, in the configuration we wanted. FSR's are made of piezoresistive material and so they are flat and flexible in nature, which was perfect for our application. Furthermore, they can detect large forces, which would make the system applicable to a lot of different users. Due to the design of the sensors, two connection wires were needed for each sensor. With these sensors in an array, the sensor mat would become very messy due to the amount of wires required. Furthermore, the connections would have to be affixed, to ensure they do not break or become loose while the patient is sleeping on the mat. Use of these sensors would contribute to the complexity of the construction process. Commercially available options showed that, in low quantities, the costs would be reasonably low, but when large numbers were needed for an array, it became uneconomical.

The last consideration was the use of piezoresistive material to make our own sensors. Velostat [21] was found to be readily available and at a low cost. It is a thin plastic containing carbon black, which makes it conductive. It is also pressure sensitive. These properties are like those found in FSR's, making it suitable for use within our system. As Velostat is supplied in sheets, it was very versatile, and we could cut pieces into our desired shape and size. Conductive material would be needed to contact the Velostat in order to use it as an FSR.

Type	Name	Dimensions	Sensors	Max Pressure (kgf/cm^2)
Pressure Sensing Mat	Tekscan BPMS HMER3	173cm x 88cm	5304	6.6
	SPI Tactilus Bodyfitter	185cm x 76cm	1728	14.1
Limiting Switch	Tapeswitch Sensing Cell	2.5cm x 1.9cm	Each	-
FSR	Leanstar Tech DF9-40	0.9cm diameter	Each	77.1
	Tekscan A301	1cm diameter	Each	35.3
Piezoresistive Material	Velostat	-	-	3.7

Table 1: Summary of sensor options

Table 1 summarises the sensor options with key metrics displayed. We decided to go with Velostat as it was flexible and thin, which would help to provide an unobtrusive and seamless experience to the user. It is also cost effective. A typical resistance vs force graph for Velostat is shown in Figure 7. Choosing Velostat also meant that more research was needed to find the optimal way to create and embed an array of FSR's into a thin mat. A three-layer approach was found to be the most practical method for constructing a pressure mapping mat. Heavy duty polythene film was chosen as the material for the top and bottom layers, and copper tape was used for the connection traces for the columns and rows. The idea of this layering approach is shown in Figure 8. The copper tape on the top layer was laid out perpendicular to those found on the bottom and at each intersection point, a 2cm x 2cm piece of Velostat material separated the copper tape. Through this approach, a total of 171 FSR's were embedded into the mat while maintaining a thin profile and without the need for wires. Female jumper wires were soldered onto the end of each strip of copper tape, so that connections could be made to the data acquisition circuit.

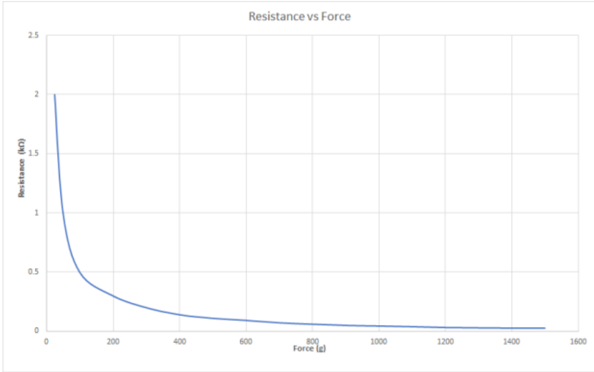

Figure 7: Velostat performance [22]

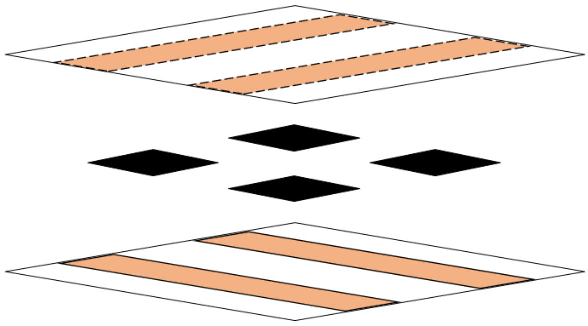

Figure 8: Sensing mat structure

Next, the data acquisition circuit was developed, using a systematic approach. The purpose of the data acquisition circuit was to capture a frame of data (the values of all the FSR's on the mat at an instance) and send it wirelessly to the cloud database. To obtain one frame, a common matrix scanning strategy was used. The concept is to supply voltage to one line at a time, for example, a column, and then the analogue to digital (ADC) values for all the intersecting lines will be read, which would be the rows in this case. After this is done, the supply voltage would be cut off for that column and then the next adjacent column would be activated, until all columns have been cycled through.

An initial design consideration that was required was deciding whether the rows or columns would correspond to the voltage supplies or ADC lines. If the 19 columns were connected to the voltage supplies, then only 9 ADC channels would be required, but the frame scanning time would be longer due to the number of voltage lines to cycle through. On the other hand, if 9 rows were connected to the voltage supplies, then 19 ADC channels would be required, but with shorter frame capture time. The main issue was the number of ADC channels, as it was difficult to source a microcontroller with at least 19 ADC channels. Microcontrollers with many ADC channels were also found to be quite costly. Ultimately, we decided to connect the columns to the voltage supplies and rows to the ADC channels.

Shift registers, analog multiplexers (muxes) and direct I/O connections were considered for the supply of voltage to the 19 columns. A microcontroller would control the sequential activation of each line. Direct I/O connections were considered as no additional hardware was required. However, microcontrollers with lots of I/O pins were generally expensive to source and had large amounts of resources, such as memory and processing power, which would be unused in our application. Multiplexers can activate specific lines and could cater for all columns with just a single chip. However, multiplexers with 16 or more outputs per IC were found to be very expensive. The more economically feasible options were found to be the two, four or eight output ICs. Since 19 outputs were required, we could use three eight-output muxes. This would require at least nine control pins from a microcontroller. The last option were to deploy shift registers in a daisy-chain configuration, which essentially creates a single large shift register while using the same common control signals for each chip. At most, only five control signal lines were required for any number of shift registers in a daisy-chain configuration. This was an attractive reason that lead us to use shift registers for our circuit. We could use shift registers as we were cycling through each column in a sequential fashion. Subsequently, the nine rows were connected to nine voltage divider circuits. The layout of our connections to and from the sensing mat is shown in Figure 9.

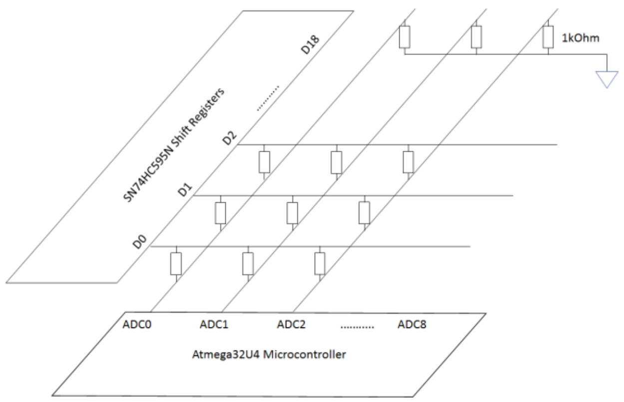

Figure 9: Layout of connections

A common eight output shift register, the Texas Instruments SN74HC595N [23] was chosen for our application. Out of five possible control pins, four were connected to the microcontroller and one was tied to ground, which was the output enable pin. This further reduced the I/O pin requirements for our microcontroller. In our initial breadboard testing of our circuit, we received unexpected ADC readings. Through further research, we found that the shift register outputs could be in one of three states: high, low, or high impedance. Since we had tied output enable to ground, our shift register configuration could only produce high or low outputs. While one output is pulling high and the other pins are pulling low, we found that current was being sunk into pins that were pulling low. To rectify this issue, a 1N4148 diode [24] was placed at each shift register output to prevent current from flowing back into the pins. The use of a diode induced a forward voltage drop of 1V in the worst case, reducing the shift register output from 5V to 4V. This in turn reduced the effective range of our FSR's. The maximum current for each shift register output was rated at 35mA [23]. In the worst case, with a 1kOhm fixed resistor, where all the FSR's in a column were pressed at maximum pressure, the current draw was calculated to be 35.1mA. The worst-case current draw could be further reduced by having a larger fixed resistor value in the voltage divider, which would again reduce the range of our ADC readings. However, a safe assumption to make was that maximum pressure would not be exerted on a whole column at any time.

For the wireless transmission of data to our cloud database, we used a WiFi module. The ESP8266 WiFi module [25] was chosen as it is readily available and low cost. Furthermore, it is the most used WiFi module in IoT applications and because of this popularity, there is an abundance of documentation and support available online. This module communicates serially through UART to the microcontroller and for our application, only one microcontroller pin is needed for sending requests. The default firmware of the WiFi module was called AT. AT commands are sent from the microcontroller serially to instruct the WiFi module to perform actions such as reset, open connection, access point connection and send data to a server. During use of our system, this was found to be very slow and inconsistent, which could be partly attributed to the size of data being transmitted at once (171 sensor values). Data was received on the cloud end around every 10 seconds and sometimes longer when packets were dropped. An alternative, more efficient firmware called NodeMCU was found and tested with the ESP8266. However, this firmware required commands to be in a programming language called Lua. The use of Lua caused issues with our data formatting and did not work as we expected, so the default firmware was kept.

During prototyping, we used the Arduino platform, including the IDE and development boards. The Arduino platform abstracts away the low-level aspects of microcontroller programming, such as bit and register manipulation. Furthermore, there were plenty of libraries, support and documentation for a wide range of applications, along with a user-friendly user interface. When testing on a small scale, we used the Arduino Uno, but that was limited to eight ADC channels, so the sensing area of the mat was narrowed by one whole row (10cm). To rectify this, an Arduino Leonardo development board was obtained, which consisted of 12 ADC channels. By using this board, we could confirm the functionality of our whole circuit design. After verification of our breadboard design, we decided to make a PCB for our circuit. This would reduce the footprint of our circuit and provide secure and reliable connections between the sensing mat and our electronics.

A microcontroller was required as the last step for our PCB and schematic design. We decided to keep with the Microchip AVR family of microcontrollers due to experience and its use on the Arduino platform. The main criteria for our microcontroller choice was that it had to consist of at a minimum: nine ADC channels, four I/O pins for shift register control, a UART interface for the WiFi module, and an SPI interface for serial programming. Enough memory was also needed for our complex program and storage of run-time variables, such as the 171 sensor values. We chose the AVR ATmega32U4 [26] as it fit the criteria perfectly. It has 12 ADC channels, 26 I/O pins, 32KB program flash, 2.5KB SRAM and serial programming capability. Our completed PCB is shown in Figure 10.

Finally, a 9V DC barrel jack adapter was used to power the board. These were widely available and are compatible with Arduino boards. Switching regulators were used to step down the 9V input into separate 5V and 3.3V voltage rails. Switching regulators offer an advantage over linear regulators due to their higher DC-DC conversion efficiency. The Recom switching regulators we used were up to 95% efficient [27]. The 5V rail was used to power the microcontroller and shift registers, and the 3.3V rail was used to power the WiFi module. A power toggle switch was added to save power while not in use.

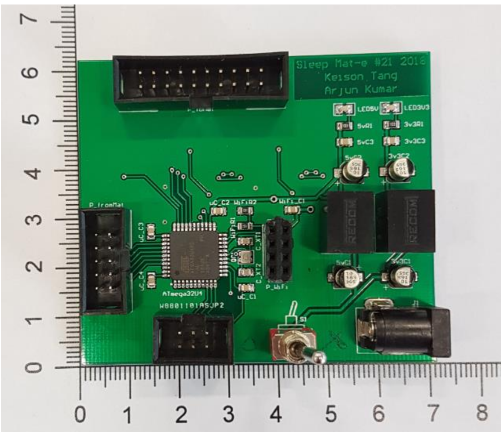

Figure 10: Completed PCB

6.3. Software

AWS was used as our cloud service provider. An EC2 instance was used to run our HTTP server, machine learning scripts and database. We used a cloud virtual machine as opposed to a serverless architecture as it was more flexible and easier to keep track of constant changes during development. Unused or previous versions of code and applications could also be kept in case future reference was needed. Microsoft Azure and Google Cloud offered similar capabilities, but we decided to use AWS based on the greater support and documentation available online. The user interface also made it easy and intuitive to deploy and monitor our instances.

For the development of our mobile application, we considered using native and hybrid development approaches. Xamarin hybrid development was considered due to previous experience and familiarity with the .NET framework. Implementation using Xamarin would mean Android, iOS and Windows platforms could be targeted with just a single codebase. This could mean any smartphone could be used to run our application and users would not be limited by their choice of device. However, the nature of hybrid development means that the user interface and functionality may vary between different platforms. We decided to go with native development targeting the Android platform as it is open source, resulting in greater flexibility and easier access of device features, such as in-built sensors. The Android Studio IDE was used as it is the official IDE for creating Android applications. It is always kept up-to date and has a lot of support and documentation available. Android Studio also offers an easy drag-and-drop interface for layout, meaning user interface and layout elements can be quickly altered without having to code. Native development targeting the Android platform was taken forward with the use of Android Studio, as we aimed to provide an intuitive user interface.

Our Android application was designed to provide caregivers the ability to remotely monitor patients and allow alerts to be sent whenever the patient was close to the edge of the bed. Therefore, the application had to be user friendly and display important information in a clear manner. Screenshots of the Sleep Mat-e Android application are shown in Figure 11. When the user launches the application, they will be greeted with the login page shown in Figure 11a. Figure 11b shows the dashboard that shows up after the user logs in. There they can go to the live posture monitor as shown in Figure 11c. The posture image updates as the patient on the mat changes posture, but after a 10 second delay, as discussed in the Hardware sub-section. A patient's sleep posture can also be viewed through the application, as shown in Figures 11d-e. Caregivers can select the date through a calendar popup to view past sleep posture information on certain days. Finally, a fall detection warning (Figure 11f) pops up and the phone vibrates to alert the caregiver whenever the patient is detected to be sleeping on the edges of the mattress. The caregiver can then check on the patient and take preventative measures as necessary.

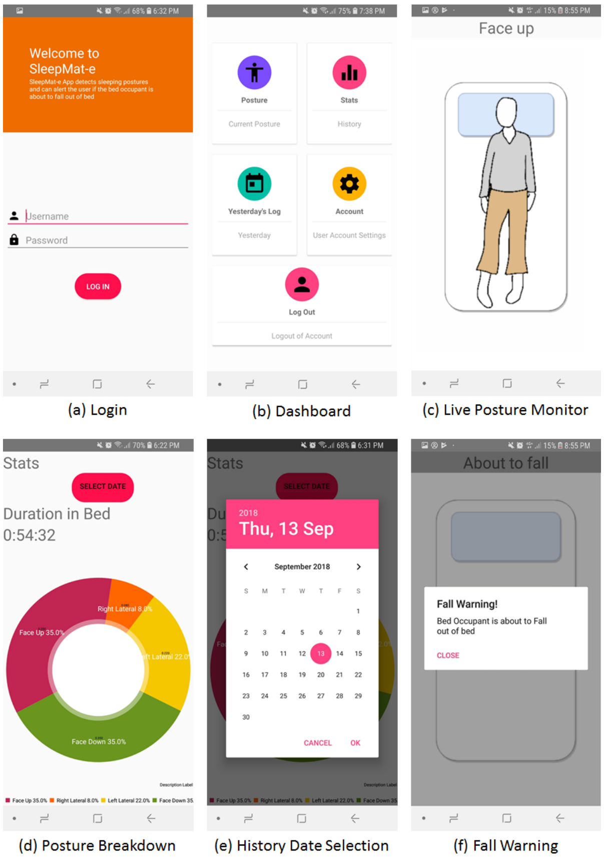

Figure 11: Sleep Mat-e Android application

CherryPy and NodeJS were considered as options for our HTTP web server. The purpose of the web server is to allow requests to be made to POST or GET information from the database. Our web server processes requests and accesses the database to write and read entries. NodeJS was chosen over CherryPy as it is asynchronous and non-I/O blocking, meaning multiple requests could be made at the same time. Furthermore, CherryPy was found to be more complex to create applications with as opposed to NodeJS, which was much more simplistic and easier to use. As NodeJS is a JavaScript framework, the same language could be used for front-end and back-end web programming. NodeJS is becoming more popular, which means there is an increasing user support network along with better quality documentation. Many libraries were available, which made prototyping and development quicker and more efficient.

Three different databases were considered for our application. First was MongoDB, which is a free and open source database aimed at document storage. The data is stored as JSON and is flexible as there is no fixed schema. However, as the data is stored in JSON format, the keys will be repeated every time entries are inserted. The structure can also be changed, which could result in inconsistent data. MySQL was tested as it offers relational databases, which means separate tables can be linked together based on keys. However, it was not as flexible as MongoDB, as the schema had to be predefined, and could not be altered once the database was created. Lastly, SQLite3 was considered. It is useful for local applications but does not allow remote connections to access the database. MySQL was used in our implementation due to the advantages it offered over MongoDB and SQLite3. XAMPP was a GUI used to start the MySQL server and provided a visual interface to the database.

TensorFlow and K nearest neighbours (KNN) machine learning algorithms were analysed and studied for their suitability in our application. The KNN algorithm makes predictions based on the location of similar occurrences of data within the training dataset. It is a simple machine learning algorithm that works by finding similar patterns between values. However, this method requires the whole dataset to be available to the algorithm. In our application, where there are 171 sensor values the algorithm would be inefficient as it must trawl through all entries in the dataset. TensorFlow utilises a convolutional neural network (CNN) for image classification. The Inception V3 model that is used by TensorFlow has been trained over millions of images, but the last layer of the network has been left untrained. We could supply our own dataset to complete the last layer of training. After training, a graph file is created, which contains information regarding nodes and weighting. This is the advantage of TensorFlow, as the training dataset is not needed after the graph file is produced. The training process can take significantly more time than KNN algorithm, but once training is complete, a classification could be quicker than using KNN. In addition, Python libraries numpy and matplotlib could be used to create a heatmap of the data frame and save it as images, as shown in Figure 12. These factors motivated us to use TensorFlow over KNN, in addition to fact that our data was large and complex.

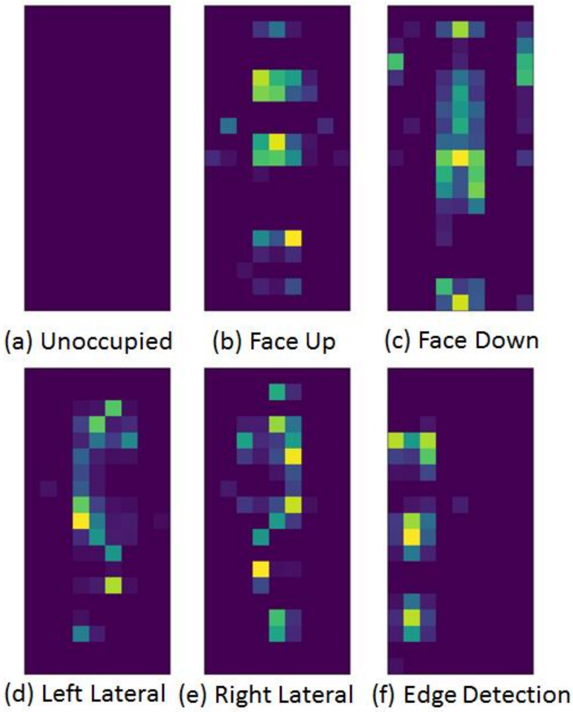

Figure 12: Pressure distribution heatmap

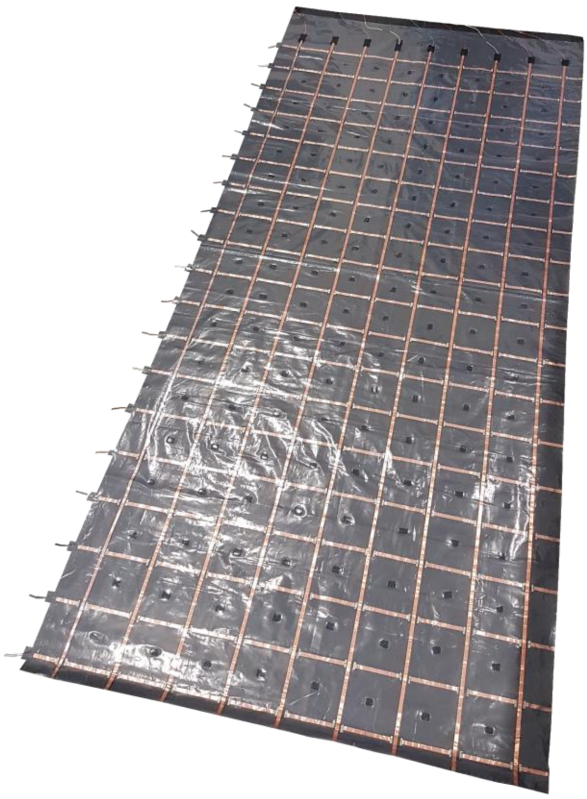

Figure 13: Sleep Mat-e prototype

7. Results

In this research project, a complete prototype system has been implemented, as shown in Figure 13. Hardware elements have been connected to the cloud and an Android application has been created to visualise the data for caregivers. A script running TensorFlow executes periodically within the cloud, to classify incoming data into sleep postures. Real testing has been used to obtain the following results. Each posture was tested 40 times. The classification was successful if the correct posture was predicted and any other posture classification was deemed incorrect. The results of our testing are shown in Table 2. The overall accuracy is the average of all six posture categories.

Posture Category	Accuracy
Unoccupied	100%
Face Up	92.5%
Face Down	90.0%
Left Lateral	85.0%
Right Lateral	80%
Edge	95%
Overall	90.4%

Table 2: Accuracies of posture categories

8. Future Work

More features could be implemented to improve the overall usability of the system. Currently, sleep time is measured based on whether the bed is occupied or not, so an algorithm could be developed to detect light and deep sleep based on the rate of change of posture. The resolution of the pressure sensing mat could be increased to allow many more postures to be detected. This would provide caregivers with much more precise data about sleeping habits. A website could also be produced to allow visualisation and monitoring of multiple patients at once. When the system becomes popular, further enhancements to the software would be required, such as the implementation of security features and focus on scalable software development such as load balancing.

9. Conclusions

This paper presents a feasible and low-cost solution for sleep and sleep posture monitoring of elderly in residential care facilities. The four major sleep postures were identified and could be tracked through our system. A thin and unobtrusive sensing mat, along with accompanying data acquisition hardware has been implemented. An Android application catered for caregivers has been developed to combat and prevent the side effects of poor sleep within the elderly population. The solution has been tested and its overall accuracy of detection has been found to be 90.4%.

10. Acknowledgements

The researchers would like to thank Dr Muhammad Nadeem for his supervision and academic support throughout the duration of the project. They would also like to thank Dr Duleepa Thrimawithana and Wai Leung Yeung for their assistance in hardware aspects of the project.

11. References

[1] Population projections overview, *Stats NZ*, 8 March 2017. [Online]. Available: http://archive.stats.govt.nz/browse_for_stats/population/estimates_and_projections/projections-overview/nat-pop-proj.aspx. [Accessed: 28 April 2018].

[2] Verhaert, Vincent, Bart Haex, Tom De Wilde, Daniel Berckmans, Marie Vandekerckhove, Johan Verbraecken, and Jos Vander Sloten. "Unobtrusive assessment of motor patterns during sleep based on mattress indentation measurements." *IEEE Transactions on Information Technology in Biomedicine* 15, no. 5 (2011): pp. 787-794.

[3] D. Seitz, N. Purandare, D. Conn, "Prevalence of psychiatric disorders among older adults in long-term care homes: A systematic review," *International Psychogeriatrics*, vol. 22, no. 7, p.1025-1039, June 2010. [Abstract]. Available: PubMed, https://www.ncbi.nlm.nih.gov/pubmed/20522279. [Accessed: 28 Apr. 2018].

[4] Ministry of Health, "DHB spending on services for older people," 13 July 2016. [Online]. Available: https://www.health.govt.nz/nz-health-statistics/health-statistics-and-data-sets/older-peoples-health-data-and-stats/dhb-spending-services-older-people. [Accessed 24 Aug. 2018]

[5] "Falls in older people: causes and prevention", Best Practice Journal, vol. 26, pp. 6-13, 2010.

[6] ACC, "Statistics on our claims," 29 April 2017. [Online]. Available: https://www.acc.co.nz/about-us/statistics/#injury-stats-nav. [Accessed 24 Aug. 2018].

[7] R. Catto, "The case for investment in: A quality improvement programme to reduce pressure injuries in New Zealand," KPMG Advisory, New Zealand. Report. 19 Nov. 2015.

[8] Conv Nets: A Modular Perspective, 8 July 2014. [Online]. Available: https://colah.github.io/posts/2014-07-Conv-Nets-Modular/img/Conv2-9x5-Conv2Conv2.png. [Accessed: 8 October 2018].

[9] TensorFlow, Image Recognition, 28 September 2018. [Online]. Available: https://www.tensorflow.org/tutorials/images/image_recognition. [Accessed: 8 October 2018]

[10] N. Ida, Sensors, Actuators, and their Interfaces: A Multidisciplinary Introduction. Edison, NJ: SciTech Publishing, 2014.

[11] Mayo Clinic, Polysomnography (sleep study). [Online]. Available: https://www.mayoclinic.org/tests-procedures/polysomnography/about/pac-20394877. [Accessed: 8 October 2018].

[12] Itamar-medical, WatchPAT. [Online]. Available: https://www.itamar-medical.com/watchpat-home-sleep-testing-made-simple/. [Accessed: 8 October 2018].

[13] Fitbit Inc., "Fitbit Charge 2," 2018. [Online]. Available: https://www.fitbit.com/nz/charge2. [Accessed 28 April 2018].

[14] Jawbone, "Up3," 2018. [Online]. Available: https://jawbone.com/up/trackers/up3. [Accessed 28 April 2018].

[15] Jeon, L., and Joseph Finkelstein. "Consumer sleep tracking devices: a critical review." In *Proc. Stud. Health Technol. Inform.*, vol. 210, pp. 458-460. 2015.

[16] Google Play Store, "Sleep as Android: Sleep cycle tracker, smart alarm," 2018. [Online]. Available: https://play.google.com/store/apps/details?id=com.urbandroid.sleep. [Accessed 28 April 2018].

[17] Google Play Store, "Runtastic Sleep Better: Sleep Cycle & Smart Alarm," 2018. [Online]. Available: https://play.google.com/store/apps/details?id=com.runtastic.android.sleepbetter.lite. [Accessed 28 April 2018].

[18] Emfit Limited, "Emfit QS – heart rate variability based contact-free recovery analysis," 2018. [Online]. Available: https://shop-us.emfit.com/products/emfit-qs. [Accessed 28 Apr. 2018]

[19] Tekscan, Body Pressure Distribution Mattress, 2018. [Online]. Availalable: https://www.tekscan.com/sites/default/files/body-pressure-distribution-mattress.jpg. [Accessed: 8 October 2018].

[20] Hsu, Hsiu-Chen, and Rong-Chin Lo. "A new mattress development based on pressure sensors for body-contouring uniform support." *arXiv preprint arXiv:1308.2196* (2013).

[21] Adafruit, Pressure-Sensitive Conductive Sheet, 2018. [Online]. Available: https://www.adafruit.com/product/1361. [Accessed: 7 October 2018].

[22] Marco Reps, Pressure Sensor Matrix Mat Project, 31 December 2017. [Online]. Available:

https://reps.cc/?p=50. [Accessed 6 October 2018].
[23] Texas Instruments, "SNx4HC595 8-Bit Shift Registers With 3-State Output Registers," SN74HC595 datasheet, Nov. 1982 [Revised Sept. 2015].
[24] Vishay Semiconductors, "Small Signal Fast Switching Diodes," 1N4148 datasheet, July 2017.
[25] Sparkfun Electronics, "ESP8266 Module (WRL-13678)," ESP8266 datasheet, 2018.
[26] Atmel, "8-bit Microcontroller with 16/32K bytes of ISP Flash and USB Controller," ATmega32U4 datasheet, July 2008 [Revised Apr. 2016].
[27] RECOM, "Switching Regulator," R-78E-0.5 datasheet [Revised Mar. 2018].