

Article

An Edge Computing and Ambient Data Capture System for Clinical and Home Environments

Pradyumna Byappanahalli Suresha ^{1,5} , Chaitra Hegde ^{2,5} , Zifan Jiang ^{3,5}  and Gari D. Clifford ^{4,5*} 

¹ School of Electrical and Computer Engineering, Georgia Institute of Technology; pradyumna@gatech.edu

² School of Electrical and Computer Engineering, Georgia Institute of Technology; chegde@gatech.edu

³ Department of Biomedical Engineering, Georgia Institute of Technology and Emory University; zifanjiang@gatech.edu

⁴ Department of Biomedical Engineering, Georgia Institute of Technology and Emory University; gari@gatech.edu

⁵ Department of Biomedical Informatics, Emory University

* Correspondence: gari@gatech.edu

Abstract: The non-contact patient monitoring paradigm moves patient care into their homes and enables long-term patient studies. The challenge, however, is to make the system non-intrusive, privacy-preserving, and low-cost. To this end, we describe an open-source edge computing and ambient data capture system, developed using low-cost and readily available hardware. We describe five applications of our ambient data capture system. Namely: (a) Estimating occupancy and human activity phenotyping; (b) Medical equipment alarm classification; (c) Geolocation of humans in a built environment; (d) Ambient light logging; and (e) Ambient temperature and humidity logging. We obtained an accuracy of 94% for estimating occupancy from video. We stress-tested the alarm note classification in the absence and presence of speech and obtained micro averaged *F1* scores of 0.98 and 0.93, respectively. The geolocation tracking provided a room-level accuracy of 98.7%. The root mean square error in the temperature sensor validation task was 0.3°C and for the humidity sensor, it was 1% Relative Humidity. The low-cost edge computing system presented here demonstrated the ability to capture and analyze a wide range of activities in a privacy-preserving manner in clinical and home environments and is able to provide key insights into the healthcare practices and patient behaviors.

Keywords: Raspberry Pi; Edge Computing; Ambient Health Monitoring; Privacy-preserving; Bluetooth; Geolocation Tracking; Patient Alarm; Illuminance;



Citation: Suresha, PB.; Hegde, C.; Jiang, Z.; Clifford, GD. An Edge Computing and Ambient Data Capture System. *Preprints* 2022, 1, 0. <https://doi.org/>

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the years, sensor technologies have played a critical role in patient monitoring in clinical and home environments. Despite this, much of the captured data is poorly integrated for research and retrospective analysis. Moreover, certain key events occurring in these settings remain undocumented with any level of spatiotemporal precision.

In clinical environments such as operating rooms (ORs) and intensive care units (ICUs), key events during patient monitoring include: (1) Patient movements while lying in bed and in mobility within the room [1,2]; (2) Bedside monitor alarm triggers and noise pollution [2–6]; (3) Presence, absence and movement of clinical personnel in the patient's vicinity [7–9]; and (4) Variations in the ambient light, temperature, and humidity [2,6,10]. In home environments, key events that are generally untracked but are beneficial for patient monitoring include: (1) Patient bodily movement during sleep [11,12]; (2) Patient movement around their residence [13]; (3) Doorbell triggers, smoke-detector triggers, microwave beeps, and phone rings [14]; and (4) Changes in the ambient light, temperature, and humidity [15]. Recently, non-contact sensors or nearables [15–20] such as microphones, video cameras, light-intensity sensors, temperature and humidity sensors, are becoming more popular for hassle-free patient monitoring. They not only collect valuable patient behavior data but also pick up key information about the patient's ambient environment while not interfering with the patient's day-to-day activities.

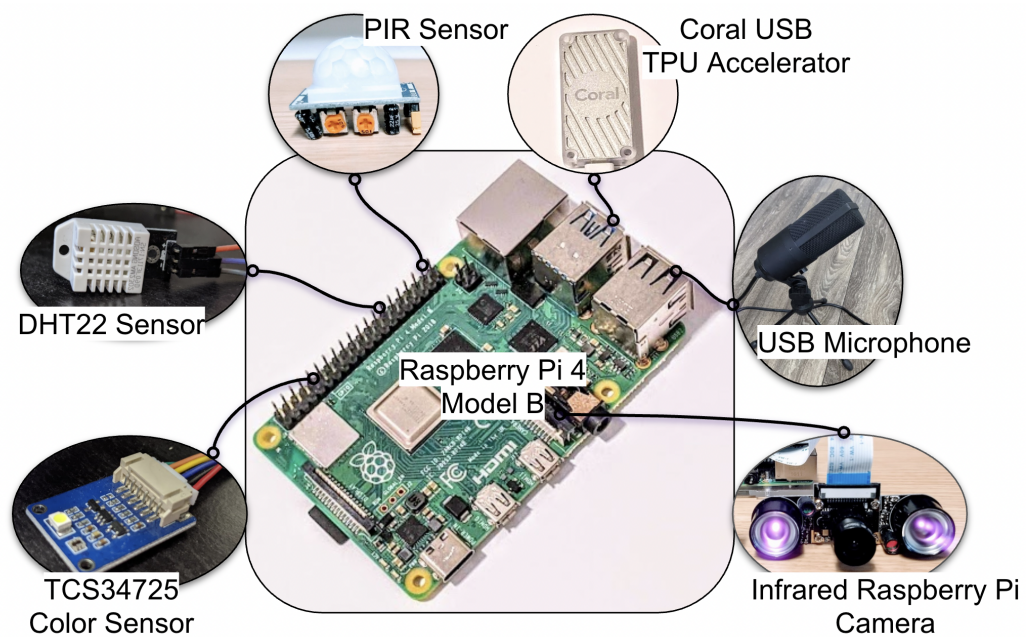


Figure 1. Edge computing and ambient data capture system. (PIR: Passive Infrared Sensor; USB: Universal Serial Bus; TPU: Tensor Processing Unit)

The objective of this study was to develop a non-contact data capture and archival system to capture patient behavior and ambient environment information. While obtaining patient behavior and ambient information is crucial in understanding the effects of healthcare practices on patient health, maintaining patient privacy is as important if not more. For this, we utilized the edge computing paradigm. In *edge computing*, algorithms are decentralized and moved closer to the point of data capture to reduce latency and bandwidth requirements. This paradigm can be defined as computing outside the cloud, happening at the edge of the network, specifically in applications where real-time data processing is required. In our work, we utilized a Raspberry Pi (RPi) as a hub for data collection and edge computing. We extracted patient privacy-preserving features from the captured data on the RPi before discarding the raw underlying signal and transferred the computed features to a Health Insurance Portability and Accountability Act (HIPAA) compliant storage.

Non-contact monitoring of patients is becoming more prevalent, especially in elderly patients [21–23] and neurodevelopmental populations (such as Autism Spectrum Disorder) [24] as these systems cause no burden on the patients in terms of wearing and operating the device (in contrast to a wearable such as the smart watch). Further, non-contact monitoring allows for monitoring the patient's global movements in contrast to wearables, thus providing additional information about patient behaviors. The advent of the COVID-19 pandemic further increased the need for such systems [19] as they allow passive patient monitoring from a distance. However, there are multiple challenges in building such a system. First challenge lies in integrating different sensors to capture multiple data modalities under a single clock. Second, this system should asynchronously transfer the data to a HIPAA-compliant database. Lastly, the system should maintain patient privacy while capturing the various data modalities. We overcame these challenges by developing a novel software system that ran on an RPi. Using this system we integrated the following five sensors: (1) Passive Infrared (PIR) sensor (2) RPi-Infrared (IR) camera (3) Universal Serial Bus (USB) Microphone (4) TCS34725 color sensor and (5) DHT22 temperature-humidity sensor. Further, we utilized the onboard Bluetooth receiver to geolocate humans using Bluetooth beacons. The main novelty of our work lies in achieving privacy-preserved patient monitoring and data fusion. It was a specific design consideration that all hardware

could be easily purchased at a low cost. This effectively helped scale the system and enabled us to capture data in large clinical environments. Further, the system was intended to be deployed rapidly without the need for expert fabrication of hardware.

2. Related Work

2.1. Wearables and Body Area Networks

Since the turn of the 21st century, wearables (on-body sensors) have been at the forefront of non-traditional health monitoring systems [25–27]. These sensors collect high-resolution physiological signal data such as the electrocardiogram (ECG) and galvanic skin response. Further, multiple wearables can be used in a network for remote patient monitoring. Body area networks (BANs) are one such system where multiple wearables continuously monitor human physiology and track the patient's health status [28]. BANs utilize wireless technologies including the ultra-wideband [29], Bluetooth [30], and Zigbee [30,31] for this purpose. Although BANs capture high-resolution information regarding human physiology, they suffer from the following drawbacks:

- BAN sensors perform localized measurements. For example, a wrist-worn accelerometer measures the acceleration of hand/wrist (local) and does not reliably measure overall body movement (global). Although a network of accelerometers will alleviate this problem, it comes at the cost of causing inconvenience to the human subject as they have to wear multiple sensors on their body over long periods.
- Data from BAN sensors are often corrupted by missing data due to motion artifacts and low compliance by the subjects. Human bodily movement causes motion artifacts in the physiological signals (say ECG) being captured by the wearable and thus leads to data degradation. Further, the wearer (a human) has to comply with a data acquisition schedule and follow the instructions diligently to generate *good quality* data.

2.2. Non-contact Health Monitoring Systems

Non-contact health monitoring systems, on the other hand, capture global signals (e.g., overall body movements via video camera [16]) and are less dependent on patient compliance for data capture. A popular way of performing non-contact monitoring of patients is to use the Doppler radar technology [32,33]. The Doppler radar is a specialized radar system that can measure target displacement remotely by using the Doppler effect. It has been used for gait-assessment of older adults [34], capturing human respiration signal [35] and human vital sign measurement [36]. While it does an excellent job of detecting body movements and measuring the vital signs of a patient, it does not capture auditory cues and other ambient environmental signatures. Thus, it does not suit our needs.

In our work, we monitor both patient movements and the ambient environment they inhabit utilizing more traditional sensors such as infrared cameras, microphones, and ambient environment sensors. To ensure patient privacy, we utilized the edge computing paradigm to extract patient privacy-preserving features and discarded raw video and audio recordings. Numerous works have proposed the usage of some of our system's individual components for patient monitoring. Extensive research has been performed to study the effect of noise pollution on patient and staff health, the performance of staff, and patient safety in clinical environments [37–41]. However, very few works describe methods to capture privacy-preserving ambient sound in clinical and home environments. In particular, Guerlain et al. [42] presented a methodology for archiving multi-channel audio and video recordings of OR during surgeries to facilitate prospective studies of operative performance. In contrast, our system avoided capturing raw audio signals. We developed a method to compute useful audio features from the captured raw audio on the RPi, archive the feature vectors, and discard the raw audio signals.

To geolocate clinical personnel at fine-resolution, Azevedo-Coste et al. [43] proposed using multiple cameras installed in ORs along with a wireless network of inertial sensors. On the other hand, recently, there has been a surge in the development of radio frequency (RF)-based non-contact human movement detectors and geolocators [44–46]. However,

to keep the system low-cost, we avoided using these components in our edge computing system. Instead, our method utilized PIR sensors and movement and pose signals from RPi-IR cameras to capture patient movements and avoided retaining the raw video signals. Further, we proposed performing geolocation in different environments (clinical and home) using Bluetooth technology.

The effect of ambient temperature, humidity, and light intensity on the human circadian rhythm and sleep is a widely studied phenomenon. Aschoff et al. [47] provide a comprehensive explanation about these effects. As there is growing evidence [48–53] about these effects we included methods to asynchronously capture and archive these signals.

3. Materials and Methods

3.1. System Architecture

The system architecture described here is a low-cost, high-compliance design. At its core is the Python script that interfaces with the sensors via the Raspberry Pi. A picture illustrating the hardware components of the system is shown in Figure 1. The bill-of-materials, along with the total cost for the hardware components and the system dimensions, are provided in our open-sourced Github repository [54]. We now describe the system's individual hardware components along with the associated software.

3.1.1. Raspberry Pi

The RPi is a \$35 computer that is about the size of a deck of cards. It functioned as the central hub in our data collection pipeline. In our work, we used the RPi 4 model B (Figure 1), which was released in June 2019 and was the most recent model during the software development stage of the project. The Debian-based operating system ('Raspbian-Buster') that is optimized for the RPi was installed for developers and users to interact with the hardware. Among the onboard peripherals on the RPi were two USB 2.0 ports and two USB 3.0 ports, a 40-pin General Purpose Input-Output (GPIO) header, and a USB-C port to supply power to the RPi. The RPi was powered using a 5V 3A power adapter.

3.1.2. PIR Sensor Based Human Movement Detection

We used a PIR sensor (Figure 1) for coarse human movement detection. The PIR sensor consists of a pair of IR sensitive slots housed in a hermetically sealed metal casing. A Fresnel lens acts as the outermost cover, which increases the range and sensitivity of the sensor. When the sensor is idle, both the IR slots receive the same amount of IR radiation. Whereas, when an IR emanating object moves past the field of view of the first IR slot, this slot detects an increased IR radiation, and thus, a differential signal C between the two slots is generated. A differential signal C' , which is completely out of phase with respect to C , is generated when the object moves past the other IR-sensitive slot. These differentials are then processed to form the output signal. Our system was designed to capture data at a sampling frequency of 1Hz. The data itself was a binary spike train taking the value 1 when a movement was detected and 0 otherwise. In [12] we presented the method for capturing movement data using a PIR sensor during sleep from a single person. Further, we used the captured movement data to build a binary classifier for obstructive sleep apnea classification and obtained a classification accuracy equal to 91%. In the current work, we extended this method and generalized PIR sensor-based human movement detection to function in different clinical and home environments. Based on the positioning and orientation of the sensor, one could capture different information. For instance, the timestamps when the patient's knee was operated on could be obtained by placing the sensor to monitor a patient's knee during their knee replacement surgery.

3.1.3. IR Camera-Based Human Movement Detection

It is possible to use a video feed [16] from an RPi-IR camera in place of the PIR sensor to perform human movement detection. This method allowed us to capture the human movement signal with more than two quantization levels and obtain a finer signal than

the binary signal captured using the PIR sensor. Besides capturing the occurrence of movements, the RPi-IR camera-based analysis allowed us to compute the intensity and direction of these movements. We used the No Infrared (NoIR) version [55] of the RPi-IR camera (Figure 1). In contrast to the regular RPi-camera, the NoIR RPi-IR camera did not employ infrared filters and gave us the ability to see in the dark with infrared lighting.

We now describe the method to extract human movement signal from the RPi-IR camera feed and describe its utility. Without loss of generality let us assume we capture the video at 1 Hz (i.e. 1 frame per second). Let the video frame at time t (in seconds) with a pixel resolution equal to $M \times N$ be denoted as F_t , and the previous video frame be F_{t-1} . The frame difference between the two frames at time t (D_t) is defined as $D_t = F_t - F_{t-1}$. The difference-frame D_t has the same pixel resolution as F_t and F_{t-1} , i.e. $M \times N$. For a given video V , the corresponding difference frame-stack is given by the set $D = \{D_t\}_{t \in [2, T]}$ where T is the total number of frames in V and D_t is the difference frame at t seconds. We extract four different signals from D namely: (1) Global Difference Sum (GDS); (2) Global δ -Pixel Count (GDPC); (3) Local Difference Sum (LDS); and (4) Local δ -Pixel Count (LDPC). The global signals (GDS and GDPC) for a given difference frame D_t are computed as follows:

$$GDS[t] = \sum_{i \in [1, M]} \sum_{j \in [1, N]} |D_t[i, j]| \quad (1)$$

$$GDPC[t] = \#(\{d \in D_t \mid |d| > \delta\}) \quad (2)$$

where d denotes an individual pixel in the difference frame D_t , $|\cdot|$ denotes the absolute value and $\#(\cdot)$ denotes the set cardinality. $GDS[t]$ is the sum total of the absolute values of the pixels in the difference frame at time t seconds and $GDPC[t]$ is the total number of pixels in the difference frame at time t seconds that have an absolute value greater than δ . If $\delta = 0$, $GDPC[t]$ denotes the total number of non-zero pixels in the difference frame D_t . The local signals (LDS and LDPC) for a given difference frame D_t are computed likewise to the global signals but are calculated on smaller blocks in the difference frame. For this, we divide a difference frame of pixel resolution $M \times N$ into K parts of equal size along the M -axis and L parts of equal size along the N -axis. Each of the K parts contains $m = \frac{M}{K}$ columns and each of the L parts contains $n = \frac{N}{L}$ rows. This division along the rows and columns of a difference frame D_t creates $nBlocks = K * L$ local-blocks of size $m \times n$. Let the s^{th} local-block be denoted as $DL_{t,s}$. Then the local signals (LDS and LDPC) at time t for the s^{th} local-block are given as:

$$LDS[t, s] = \sum_{i \in [1, m]} \sum_{j \in [1, n]} |DL_{t,s}[i, j]| \quad (3)$$

$$LDPC[t, s] = \#(\{d \in DL_{t,s} \mid |d| > \delta\}) \quad (4)$$

where d denotes an individual pixel in the difference frame local-block $DL_{t,s}$, $|\cdot|$ denotes the absolute value and $\#(\cdot)$ denotes the set cardinality. The $LDS[t, s]$ is the sum total of the absolute values of the pixels in the s^{th} local-block of the difference frame at time t seconds and the $LDPC[t, s]$ is the total number of pixels in the s^{th} local-block of the difference frame at time t seconds that have an absolute value greater than δ . Similar to $GDPC$, if $\delta = 0$, $LDPC[t, s]$ denotes the total number of non-zero pixels in the s^{th} local-block of the difference frame D_t . The different parameters that needed to be set were δ , K , and L . The default values we set in our work were $\delta = 0$, $K = 5$, $L = 4$ and the videos we experimented on had a pixel resolution equal to 320×240 . Thus, in our work, we had $m = 64$, $n = 64$, and $nBlocks = 20$.

Together, the host of time-series data described in Appendix A gave us information about the temporal and spatial variations in the video V . We needed certain assumptions so that these signals gave us information about human movement. The assumptions were as follows (1) We had one person in the entire video. (2) The background was static, and the only moving object in the video was the person or an object attached to the person. (3) The video was not corrupted or affected by noise.

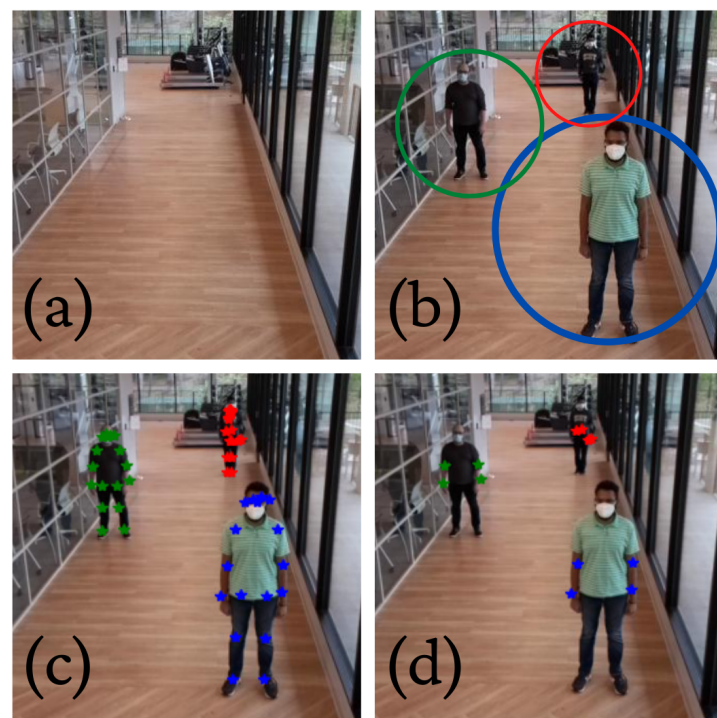


Figure 2. PoseNet [56] for human pose detection. (a) We show the scene without any humans. (b) The scene with three humans standing. (c) The identified keypoints using PoseNet have been overlaid on the individual humans. (d) The keypoints corresponding to the elbows and wrists have been retained. This is useful for analyzing hand movements.

While the PIR sensor provided binary movement signals (movement occurred vs. no movement), the movement signals from the IR cameras were finer and had a higher spatial resolution. These two solutions were useful in different scenarios. For instance, while the PIR sensor could be used to detect the presence or absence of a human in a room, the movement signal from the IR camera could be used to perform a privacy-preserving analysis of a patient's sleep.

3.1.4. Human Pose and Activity Phenotyping

Technologies such as PoseNet [56,57], OpenPose [58] or DeepLabCut [59] could be used to obtain canonical representations of the human form, and when tracked over time, one could obtain information concerning pose and types of activities. In our implementation, we used a *Coral USB Tensor Processing Unit (TPU) Accelerator* (Figure 1) and Tensorflow Lite to render these abstractions in real time.

This data representation is known as *keypoint representation* and comprises the x-y coordinate positions of various interest points on the body, including the knees, elbows, and eyes. It has the advantage of preserving the privacy of individuals while also reducing the dimensionality of the data. One can use this technology to understand behavior of the neurodiverse populations, both at a group level and at an individual level, to capture social interaction metrics and predict certain behaviors of interest ahead of time.

In its implementation, we used a Google Coral Accelerator USB device in addition to the RPi and the RPi camera module V2 (8 megapixel). The RPi 4 was most suitable for this work compared to the older versions of the RPi due to the presence of USB-A 3.0 ports. These ports ensured fast communication between the Coral device and RPi. The absence of USB-A 3.0 ports in previous versions of the RPi significantly increased the communication time between the RPi and Coral, thus increasing the total run time of the algorithm implemented on it. We implemented the Tensorflow Lite model of the PoseNet

algorithm on the Google Coral Accelerator that was connected to the USB-A 3.0 port of the RPi 4. The camera attached to the RPi collected videos, which were converted to the keypoint representations in real-time by PoseNet. These keypoint representations were stored on a secure server for further processing, and the raw video was discarded, thus achieving patient privacy. An example is given in Figure 2.

3.1.5. Privacy Preserving Audio Data Capture

The proposed audio data capturing system consisted of a USB microphone connected to the RPi. In this work, we tested the *Fifine Conference USB Microphone* (Figure 1). In order to record the audio signal, we used the *python-sounddevice* package available on GitHub [60,61] under the MIT License. Specifically, we modified the script *rec_unlimited.py* [60] to be able to continuously record audio data and perform audio feature computation at regular intervals. To extract features from these audio snippets, we used a 30 millisecond Hanning window with a 50% overlap. Feature computation was done using the *librosa* package available on Github[62] under the Internet Systems Consortium (ISC) license. We utilized the spectral representation method *stft* in the Core IO and Digital Signal Processing toolbox and the spectral features method *mfcc* in the feature extraction toolbox in order to compute short-time Fourier transform (STFT) and Mel-frequency cepstral coefficients (MFCC), respectively. Further, we used the filter bank construction method *mel* in the Filters toolbox to create a filter bank with 10 frequency bins. For each 30 millisecond window, we then computed the signal energy in different frequency bins by performing the following matrix multiplication:

$$E = MS \quad (5)$$

where S was the STFT coefficient vector for the current window, M was the mel filter mask matrix with each row corresponding to a different mask, and E was the signal energy in different frequency bins corresponding to the mel filter masks. Note that the default shape of mel filter masks was a triangle with the mask values summing to one. We further included scripts for computing sample entropy of the windowed audio snippets using the *mse.c* script, which was available on Physionet [63] under the GNU general public license. Further, we developed scripts to archive the computed audio features to a secure cloud storage and discard the underlying audio snippets. The above implementations were developed using Python 3.7.3, C, and Bash scripting. These were representative edge computing methods that could extract different features from audio signals. Other feature extraction algorithms which could be run on the constrained environment of an RPi could be easily incorporated as additional methods. The discarding of raw audio data ensured patient privacy and speaker identification was not possible. Furthermore, we did not record the speaker's pitch information in the audio snippets or deploy methods to determine if a given window contained voiced speech.

3.1.6. Human Location Tracking via Bluetooth

The Bluetooth scanning system utilized the onboard Bluetooth receiver of an RPi. In this work, we tested the *Smart Beacon SB18-3* by *kontakt.io* with an RPi 4 model B. We leveraged the *scanner* package by *bluepy - a Bluetooth LE interface for Python* [64] for this purpose. The code implementation was done in Python 3.7.3. The software recorded the received signal strength indicator (RSSI) value from all the beacons transmitting the Bluetooth signal in the vicinity. We used the media access control (MAC) addresses of the Bluetooth beacons to identify them. A Python script would poll for RSSI values from all the beacons in the vicinity at regular intervals. The received RSSI values, the unique MAC address, and the recording timestamp were dumped into a file.

3.1.7. Ambient Light Intensity Assessment

In this work, we used the Waveshare TCS34725 Color Sensor (Figure 1) in conjunction with an RPi to capture ambient light intensity. Among other signals, the color sensor captured the following signals which were of interest to us: (1) Red, Green, and Blue

values in the RGB888 format (8-bit representation for each of the three color-channels); (2) Illuminosity in Lumen per square foot (LUX); and (3) Color Temperature in Kelvin. The RGB values and the color temperature gave us information about the ambient light color, and the illuminosity values gave us information about the ambient light intensity.

3.1.8. Temperature and Humidity Detection

For temperature and humidity detection we used the DHT22 Temperature-Humidity sensor module (Figure 1) in conjunction with an RPi. The DHT22 sensor comprised a thermistor and a capacitive humidity sensor that measured the surrounding air to provide calibrated temperature and humidity values. Further, the sensor module came with a digital board that housed three pins, namely VCC, GND, and OUT. The sensor had an operating voltage of 3.3/5V (DC), and the OUT could be read from a GPIO pin on the RPi. The temperature range was -40 to 80 °C, and the humidity range was 0 – 100% Relative Humidity (RH).

3.1.9. Thermal Camera-Based Temperature Measurement

In [19], we showed that it is possible to perform febrile state detection using the combination of an RPi camera and the FLIR Lepton 3.5 Radiometry Long-Wave Infrared Camera with its associated Input-Output module. This system has been included in the codebase without further experimental validation.

3.2. Data Fusion

The following data modalities were captured at a sampling frequency of 1 Hz: (1) PIR sensor-based human movement; (2) IR camera-based human movement; (3) Audio data; (4) Bluetooth RSSI signal; (5) Ambient light intensity; and (6) Temperature and humidity. A single main script facilitated the capture of all the above data modalities and the individual time stamps corresponding to each sample. The human pose signal was recorded in an ad-hoc manner when the algorithm detected a human body. Nevertheless, the corresponding timestamps were recorded using a single clock onboard the RPi to ensure all data modalities were recorded synchronously. Further, the recorded data was easy to access via a simple directory structure consisting of separate folders for each data modality. The data collection was performed in parallel by each RPi and transmitted in real-time to a HIPAA compliant central server, which aggregated the data to perform patient state analysis.

3.3. Applications

The applications of our edge computing and ambient data capture system range from monitoring patient sleep in sleep labs to tracking neurodegenerative patients at their homes. In our work, we describe five experiments to demonstrate our system's utility:

1. Estimating occupancy and human activity phenotyping: *This utility enables us to perform patient sleep monitoring, human location tracking and activity phenotyping.*
2. Medical equipment alarm classification using audio: *This utility facilitates patient monitor alarm monitoring in ORs or ICU rooms, where there are many system not centrally integrated.*
3. Geolocation of humans in a built environment: *We can track humans in a built environment and model social distancing for quantifying epidemic disease exposure [65].*
4. Ambient light logging: *This system can be used to study the effect of ambient light on human circadian rhythm.*
5. Ambient temperature and humidity logging: *We can perform long-term monitoring of the effects of ambient environmental conditions on patient behavior.*

In each of these experiments, we either processed the data captured by our system or tested the utility of our onboard feature extraction methods to understand the associated environments better.

Table 2. Class labels including the Musical Notes in International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 60601 – 1 – 8 Alarm and their count [*E* stands for the *Empty* class and *T* stands for the *Transition* class]

Musical Note	Fundamental Frequency (Hz)	Count
C_4	261.63	207
D_4	293.66	43
E_4	329.63	32
F_4	349.23	46
F_4^\sharp	369.99	27
G_4	392.00	66
A_4	440.00	45
B_4	493.88	17
C_5	523.25	81
<i>E</i>	-	969
<i>T</i>	-	432

3.3.1. Estimating Occupancy and Human Activity Phenotyping

Table 1. Estimating occupancy and activity phenotyping

Start Time (second)	End Time (second)	Duration (seconds)	Action
0	30	30	One person standing
30	60	30	Two people standing
60	90	30	Three people standing
90	120	30	Three people exercising
120	150	30	Two people exercising
150	180	30	One person exercising

In this experiment, we developed and tested algorithms to (1) Track the number of people in a given video; and (2) Differentiate people standing still from people performing a hand exercise (activity). For this, we utilized PoseNet to record keypoint locations of stick figures of humans in the video. We recorded a three-minute structured video in employing three subjects (S1, S2 & S3). The data collection was performed as illustrated in Table 1.

The video frames and the corresponding keypoints were retrospectively processed to compute the number of people in each frame. For every video frame, the corresponding keypoints file contained the x-y coordinates of the stick figures. For each stick figure (human being), a separate stick figure x-y coordinate array was stored. Thus, by calculating the number of non-zero arrays in each keypoints file, we counted the number of people present in the corresponding video frame.

For human activity phenotyping, we considered the case of contrasting a standing human from an exercising human. For this, we considered the following four keypoints amongst the 17 keypoints recorded by PoseNet: (1) Left Elbow; (2) Right Elbow; (3) Left Wrist; and (4) Right wrist. In our retrospective analysis, we computed the distance (in pixels) between frames for all four keypoint locations. A distance-vector was formed for each of the four keypoint locations for all three subjects separately when they were standing and exercising. Further, we computed the interquartile range (IQR) of these distance vectors and plotted these values for all four locations for each subject both while standing and exercising. The IQR is a suitable method for suppressing outliers and capturing the spread of the data. The statistical variance, on the other hand, is affected by outliers. Hence, we chose to compute IQR over the variance to measure the spread of the data.

3.3.2. Medical Equipment Alarm Classification Using Audio

In this experiment, we tested the utility of the audio feature extraction methods (energy in mel frequency bins and MFCC) proposed by us for clinical audio classification. For this, we utilized an external clinical audio database, extracted the proposed features, and performed multi-class classification.

Dataset. We analyzed the International Organization for Standardization / International Electrotechnical Commission (ISO/IEC) 60601 – 1 – 8 type medical equipment alarm sounds [66]. The alarm sounds comprised eight categories: general, oxygen, ventilation, cardiovascular, temperature, drug delivery, artificial perfusion, and power failure. Further, each category had two alarm sounds, namely, medium priority alarm and high priority alarm. The medium priority alarm sounds were about one second long and consisted of three musical notes that were played once, whereas the high priority alarm sounds were about 4.5 seconds long. They consisted of five musical notes that were played twice. All the alarm audio recordings were single channel, sampled at 22050 Hz, and recorded in the *Waveform* audio file format. Nine musical notes were used to construct these 16 different alarm audio recordings. Table 2 lists these nine musical notes with their fundamental frequencies. The works [66,67] provide more information on the individual alarm sound recordings.

Feature Extraction. We used the audio data capture software described in Section 3.1.5 and computed 20 MFCC features and 10 filter bank energy features on 30 millisecond snippets of the 16 alarm sound recordings. Further, we computed STFT coefficients for the audio clips and annotated each snippet to belong to one of the following 11 classes: $\{Empty, C_4, D_4, E_4, F_4, F_4^\#, G_4, A_4, B_4, C_5, Transition\}$. The *Empty* class was assigned when all the STFT coefficients of a snippet were equal to zero. If a particular audio snippet was partially made up of a specific note with the rest of the samples equal to zeros, such windows were annotated as the *Transition* class. We annotated the musical notes by comparing the fundamental frequency in STFT with the values shown in Table 2. Moreover, we used the note sheets provided in [66,67] to confirm our annotations. We had a total of 1965 data points. Table 2 further provides the breakdown of the number of data points in each class.

Classification. Using the 30 features described, we performed an 11 class classification using five-fold cross-validation and an XGBoost [68] classifier. All codes were written in Python 3.6.3 and XGBoost was implemented using the package provided in [69]. The following hyperparameters were used without any tuning: $n_estimators = 150$, $objective = 'multi:softmax'$, $num_class = 11$, $max_depth = 6$. All other hyperparameters were set to their default values. As illustrated in Table 2, the dataset contained class imbalance. We thus used both the macro averaged $\mathcal{F}1$ ($\mathcal{F}1 - macro$) score and the micro averaged $\mathcal{F}1$ ($\mathcal{F}1 - micro$) score as the measures for assessing classification performance. The $\mathcal{F}1 - macro$ score gives equal importance to each class irrespective of the number of samples in each class thus providing a balanced assessment of the multi-class classification performance when the dataset is imbalanced. The $\mathcal{F}1 - micro$ score on the other hand aggregates samples from all classes before computing the $\mathcal{F}1$ score. Please refer [70] for the individual expressions for computing the two $\mathcal{F}1$ scores.

Speech Mixing. Next, we measured the performance of the note classification algorithm in the presence of speech. For this, in addition to the ISO/IEC 60601 – 1 – 8 dataset, we used a speech record consisting of five speakers [four male and one female] from the Oxford Lip Reading Sentences 2 dataset [71]. First, we resampled the speech record to match the sampling frequency of the alarm audio recordings (22050Hz) and extracted the first channel of this resampled speech record, denoted by \mathcal{S} . Next, for each of the 16 alarm audio records $\mathcal{A}_i, i \in [1, 16]$, we uniformly randomly pick an audio snippet from the speech record \mathcal{S}_i

which was of the same length as \mathcal{A}_i . We then generated 10 audio records per alarm audio recording as follows:

$$\mathcal{M}_i[\alpha] = \alpha * \mathcal{S}_i + (1 - \alpha) * \mathcal{A}_i \quad (6)$$

where $\mathcal{M}_i[\alpha]$ was the mixed audio recording for a given α and $\alpha \in \{0, 0.1, 0.2, \dots, 0.8, 0.9\}$ was the mixing parameter that combined speech recording with alarm audio recordings. Note that $*$ and $+$ denoted scalar multiplication and sample-wise addition, respectively. When α was equal to 0, we had no speech component, and thus, $\mathcal{M}_i[0]$ was equal to the original alarm audio recording \mathcal{A}_i . As α increased from 0 to 0.9, the speech component in $\mathcal{M}_i[\alpha]$ increased linearly, and the alarm audio component decreases linearly.

We obtained a total of 160 different audio recordings (10 mixed audio recordings per clean alarm audio recording). We re-computed 20 MFCC features and 10 filter bank energy features for these 160 audio recordings using 30 millisecond Hanning windows and a 50% overlap and obtained a total of 19650 feature vectors. The ground truth labels for the feature vector at different α values were the same as those for $\alpha = 0$. Utilizing these 19650 feature vectors and corresponding labels, we performed an 11 class classification of musical notes using five-fold cross-validation and XGBoost [68] classifier. The hyperparameters were the same as it was when there was no speech mixing.

3.3.3. Geolocation of Humans in a Built Environment

Table 3. The human tracking experiment

Start Time (second)	End Time (second)	Duration (seconds)	Action
0	146	146	Stay in room 1
146	159	13	Move from room 1 to room 2
159	268	109	Stay in room 2
268	286	18	Move from room 2 to room 3
286	470	184	Stay in room 3
470	480	10	Move from room 3 to room 1
480	600	120	Stay in room 1

In this experiment, we processed the RSSI signal received by the RPi to perform room-level location detection of humans using a Bluetooth beacon. We set up nine RPis in a built environment where each RPi was loaded with the software to capture RSSI values as received from a specific Bluetooth beacon. The built environment consisted of three rooms, and three RPis were present in each room, approximately equidistant from the center of the room. A *Kontakt.io* Bluetooth beacon with a unique MAC address was used in the experiment. A human subject carried the Bluetooth beacon and moved around the space as illustrated in Table 3.

The processing of the collected RSSI values to perform geolocation of humans was done on a central server. This was because we collected data from multiple RPis to perform geolocation. Once the data was transferred to the cloud from each RPi, we downloaded the data onto a single computer and performed geolocation. The RSSI signals captured by each RPi were non-uniformly sampled. Hence, these signals were converted to a uniformly sampled signal with a sampling frequency equal to 1 Hz by filling missing data using the following equation.

$$\begin{aligned} RSSI_n[currentTime] &= RSSI_n[previous] \\ &\times \max([\beta \times (currentTime - previous), 1]) \end{aligned} \quad (7)$$

where, $n \in [1, 9]$ was the index variable to recognize RPis, $RSSI_n$ was the vector of RSSI values captured by n^{th} RPi, $currentTime$ was the time (in seconds) at which we did not have a reading of the RSSI value, $previous$ was the closest predecessor time point (in seconds) to $currentTime$ at which we had a reading of the RSSI value, β was the decay parameter that

controlled the rate at which the RSSI value decayed when RSSI values were missing and max was a function to compute the maximum value in the input vector. The time difference ($currentTime - previous$) was expressed in seconds.

Further, any RSSI value less than -200 dBm was clamped to -200 dBm to have all RSSI values in a fixed range. We set β to 0.2, which corresponded to maintaining the previous value for 5 seconds before the RSSI values were decayed when the RSSI values were missing. Further, we computed the average RSSI signal for each room by computing the mean value of the RSSI signals captured by the three RPi in each room. We used the softmax function to obtain a probability vector that gave the probability of the human subject with the Bluetooth beacon to be present in each of the three rooms at any given point in time. The averaging of RSSI values from multiple RPi receivers and the further usage of the softmax function significantly suppressed the effect of noisy RSSI samples.

3.3.4. Ambient Light Logging

Table 4. The ambient light tracking experiment

Day	Time (HH:MM)	Action
Clear Day (D_{clear})	07:32	Sunrise
	18:10	Sunset
Night-1 (N_1)	18:55	Lights-ON
	23:45	Lights-OFF
	01:00	Data upload start
	02:00	Restart data collection
Cloudy Day (D_{cloudy})	07:31	Sunrise
	16:58	Lights-ON
	18:11	Sunset
Night-2 (N_2)	23:42	Lights-OFF

To perform ambient data logging, we set up an RPi with the Waveshare TCS34725 color sensor in a built environment. The RPi was loaded with the associated software needed to capture the ambient light intensity values. Table 4 provides a timeline of events that occurred during ambient light data capture. The duration between 1 AM and 2 AM was reserved for data upload, and no data capture was performed during this period. The color sensor was set up in a place that received natural sunlight during the day and received light from light sources in the room during the night. The lights in the room were turned ON when the natural sunlight was not adequate for a normal human lifestyle. The lights in the room remained ON until “sleep time” of the residents in the built environment when the lights were turned OFF. We recorded the ambient light data for two consecutive days. In parallel, we tracked the weather conditions of the data collection site and recorded the minute-to-minute local cloud cover information. With this setup, we studied the effect of cloud cover, sunrise and sunset times, artificial lights in the room, and buildings around the data collection site on the ambient light data captured by the color sensor. We divided the entire time period into 4 sections: D_{clear} , N_1 , D_{cloudy} , N_2 . Here, D_{clear} represents the day-period (sunrise to sunset) on the first day when the skies were clear, N_1 represents the first night (sunset to sunrise), D_{cloudy} represents the day-period (sunrise to sunset) on the second day when the skies were extremely cloudy (average cloud cover $> 80\%$), and N_2 represents the second night (sunset to sunrise). Since the cloud cover information during the night had little or no effect on the ambient light intensity in the room, in our analysis, we only used the cloud cover data tracked during D_{clear} and D_{cloudy} .

3.3.5. Ambient Temperature and Humidity Logging

We validated the DHT22 temperature and humidity sensor against a commercially available sensor in this experiment. We set up an RPi with the DHT22 sensor in a built

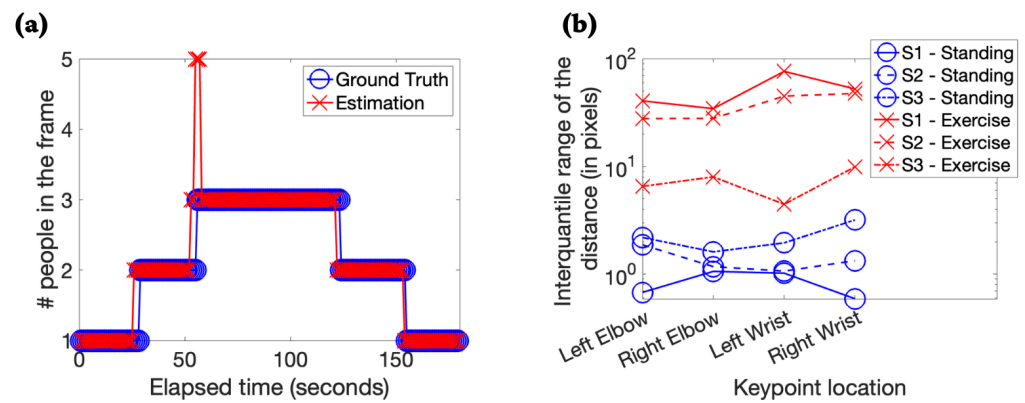


Figure 3. Results for occupancy estimation and activity phenotyping. (a) The comparison between the ground truth and the estimation by the algorithm for the number of people in each video frame is shown. (b) We show the interquartile range (IQR) of the distance (in pixels) between the keypoints from consecutive frames (sampling frequency = 1Hz) for three subjects standing and performing exercise.

environment. The RPi was loaded with the necessary software to continuously capture temperature and humidity values and the associated UTC timestamps. We collected the temperature (T_{RPi}) and humidity (H_{RPi}) values with this setup at a sampling frequency of 1Hz for three consecutive days (about 72 hours). Further, we set up the *ORIA mini Bluetooth Temperature-Humidity sensor* (a commercial sensor) beside our RPi setup and simultaneously performed temperature (T_{cs}) and humidity (H_{cs}) measurements using the commercial sensor. The commercial sensor allowed the export of the recorded T_{cs} and H_{cs} measurements in the form of comma-separated value files via an Android application. The commercial sensor output contained measurement values at a sampling rate of 0.001667Hz (one sample per 10 minutes). Hence, we retrospectively processed the T_{RPi} and H_{RPi} measurements captured by the DHT22 sensor to match the number of samples and the measurement timestamps corresponding to the commercial sensor via the following procedure. For every timestamp (ts_{cs}) at which we had the temperature and humidity values from the commercial sensor, we constructed a 10-minute window which spanned from $ts_{cs} - (10 \text{ minutes})$ to ts_{cs} . We collated all T_{RPi} and H_{RPi} measurements in this time window and computed the mean value of these measurements to obtain $T_{RPi-\mu}[ts_{cs}]$ and $H_{RPi-\mu}[ts_{cs}]$. We then compared the $T_{RPi-\mu}$ with T_{cs} and $H_{RPi-\mu}$ with H_{cs} by plotting the signals one over the other. Further, we performed correlation analysis and fit separate linear models for the temperature and humidity measurements. Finally, we created separate Bland-Altman plots for the temperature and humidity measurements.

4. Results

4.1. Estimating Occupancy and Human Activity Phenotyping

Figure 3A shows the comparison of the algorithm's estimation for human occupancy in each video frame with respect to the ground truth values. We obtained an accuracy of 94% for the occupancy estimation experiment. For contrasting humans performing hand exercises from humans standing still, we visualized the IQR values of the frame-to-frame distances (in pixels) for four keypoints corresponding to the human hand. The IQR values

Table 5. Note Classification in Medical Equipment Alarm

Setting	$\mathcal{F1} - \text{micro}$	$\mathcal{F1} - \text{macro}$
Without Speech	0.98	0.97
With Speech Mixing	0.93	0.91

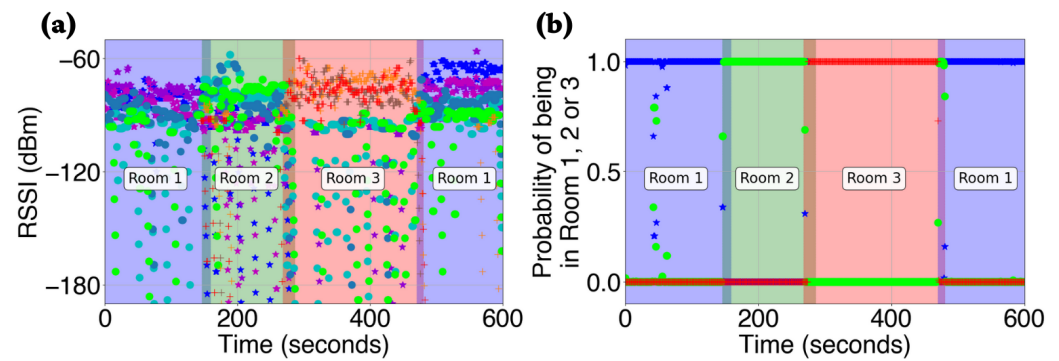


Figure 4. Human tracking via Bluetooth. The translucent background colors indicate the ground truth values, and the corresponding room number has been labeled. (a) The received signal strength indicator (RSSI) values as captured by each of the nine Raspberry Pis during the experiment are shown. It is measured in decibels with reference to one milliwatt (dBm). (b) The corresponding probability values of the subject being in Rooms 1, 2, or 3 during the experiment. The blue color depicts Room 1, the green color depicts Room 2, and the red color depicts Room 3.

were always larger for exercising than standing still for all four keypoints and all three subjects. When a subject was standing still, the keypoints barely moved and thus would typically have low values (< 5 pixels). The individual IQR values for all three subjects at the four keypoint locations are illustrated in Figure 3B.

4.2. Medical Equipment Alarm Classification Using Audio

Table 5 shows the results for the 11-class medical equipment alarm note classification. We computed the micro averaged ($\mathcal{F1} - micro$) and the macro-averaged ($\mathcal{F1} - macro$) $\mathcal{F1}$ scores for the two experiments. When there was no speech content in the alarm audio recordings, we obtained an $\mathcal{F1} - micro$ equal to 0.98 and an $\mathcal{F1} - macro$ equal to 0.97. Retraining with speech resulted in a drop of 5.1% and 6.2% in $\mathcal{F1} - micro$ and $\mathcal{F1} - macro$ scores, respectively.

4.3. Geolocation of Humans in a Built Environment

Figure 4A shows interpolated RSSI values captured by each of the nine RPIs that were placed in the built environment, and Figure 4B illustrates the corresponding probability of the subject being in rooms 1, 2, or 3. The ground truth of the subject's presence is shown using translucent colors in the background. Specifically, translucent blue denoted being present in room 1, translucent green denoted being present in room 2, and translucent red denoted being present in room 3. Further, the transition from one room to another was illustrated by overlapping colors corresponding to the two rooms. It is evident from Figure 4 that our system did an excellent job of identifying the room in which the person was present. Specifically, for 592 out of 600 seconds, the human tracking system correctly identified the subject's presence in one of the three rooms, which corresponded to an accuracy of 98.67%.

4.4. Ambient Light Logging

Figure 5 depicts the variation of ambient light intensity over two days. The minimum, median, and maximum illuminance values during the D_{clear} period were equal to 0.56 LUX, 71.08 LUX, and 186.92 LUX, respectively, whereas the minimum, median, and maximum illuminance values during the D_{cloudy} period were equal to 0.26 LUX, 26.93 LUX, and 117.41 LUX, respectively. Thus, the median difference in illuminance between the clear and cloudy days was equal to 44.15 LUX. The minimum, median and maximum illuminance between the lights-ON and lights-OFF times was equal to 15.08 LUX, 15.30 LUX, 16.66 LUX on the first night (N_1) and equal to 9.39 LUX, 10.15 LUX, 15.02 LUX on the second night (N_2).

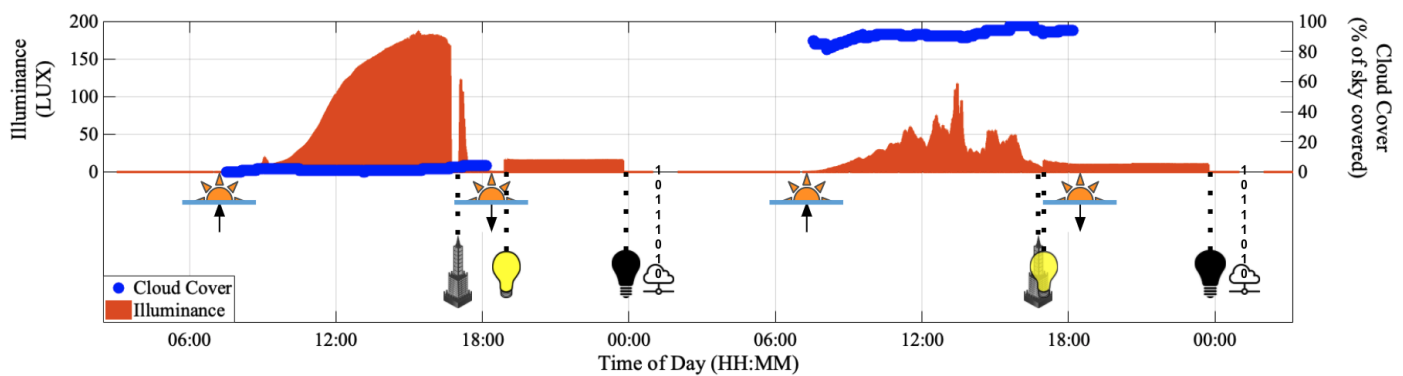


Figure 5. Observing the effects of cloud cover, sunrise, sunset, lights-ON, lights-OFF and buildings around the data collection site on the ambient light in a built environment for two continuous days (48 hours). The solid-orange bars (■) depict the amount of ambient light sensed by the color sensor in Lumen per Square Foot (LUX). The solid-blue circles (●) depict the local cloud cover in percentage of local sky covered by clouds. The sunrise and sunset times are indicated with the rising and setting sun symbols, respectively, using the upward and downward arrows. The yellow and black bulbs specify the lights-ON and lights-OFF times, respectively. The skyscraper symbol indicates the time when the Sun goes behind a skyscraper and causes a shadow onto the location where ambient light was being tracked. Data upload is depicted by binary values and a cloud node.

The illuminance was consistently equal to zero between the lights-OFF and sunrise times. Further, we observed a dip in illuminance when the Sun hid behind a skyscraper and cast a shadow on the data collection site during the day.

4.5. Ambient Temperature and Humidity Logging

Figure 6 illustrates the comparison between the processed temperature and humidity values from the DHT22 sensor with the outputs from a commercial sensor. The temperature values from the two sensors closely followed each other with a root mean squared error (RMSE) between the two measurements equal to 0.28°C and a coefficient of determination (r^2) equal to 0.97. Over 97% of the samples lay within the limits of agreement $[-1.96\text{SD}, +1.96\text{SD}]$ in the Bland-Altman plot. Further, the mean difference was equal to -0.4°C . The humidity values from the two sensors closely followed each other with an RMSE between the two measurements equal to $1.00\%\text{RH}$ and an r^2 equal to 0.90. Over 95% of the samples lay within the limits of agreement $[-1.96\text{SD}, +1.96\text{SD}]$ in the Bland-Altman plot. Further, the mean difference was equal to $-1.2\%\text{RH}$.

5. Discussion

The work described in this article aims to extend the types of data found in traditional clinical monitoring environments and provide a simple system to capture data in the built environment, outside of clinical settings. Many commercial (clinical and consumer) systems are either designed to keep data in a proprietary ‘walled-garden’ to reduce competition or are not designed for the high throughput needed to transmit/record the data. The RPi-based edge computing system described in this work allows direct data import via USB and upload to the cloud asynchronously to overcome these issues.

We have included methods to capture audio data, physical movement, and location of subjects. As we have demonstrated, audio data allows capturing of all alarms in the clinical space. While some monitors transmit some of these events or signals over the network, it is often costly or impossible to gain access to such data, and data integration and synchronization are highly problematic. Moreover, such systems do not provide a holistic picture of the environment. For example, the volume of the alarm, together with the background noise, contributes to noise pollution and has been shown to affect caregivers and patients alike [4,5,72]. Beyond alarms in the clinical environment, it is possible to

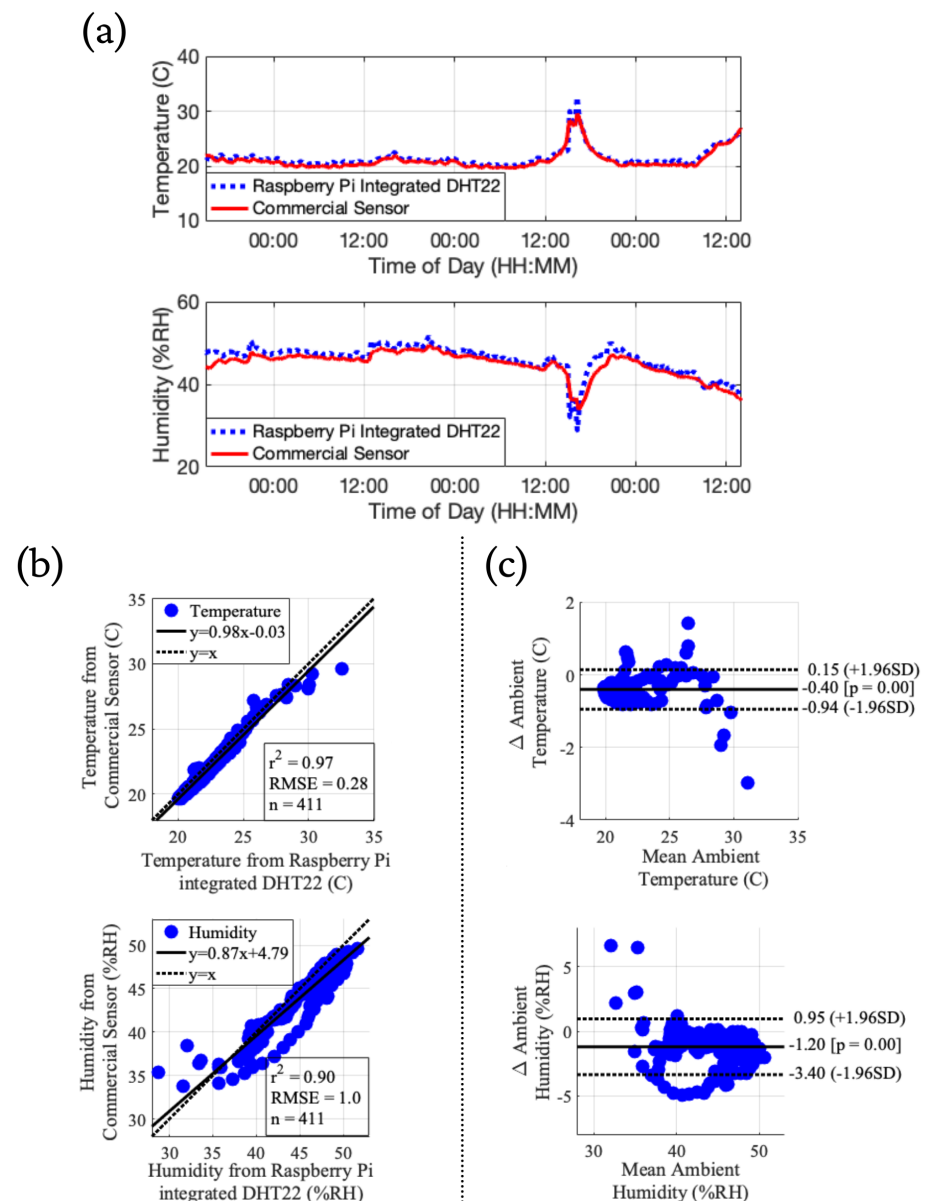


Figure 6. Comparison of temperature and humidity values captured by the Raspberry Pi (RPi) integrated DHT22 sensor and a commercial sensor. The top plot corresponds to temperature in each subplot, and the bottom plot corresponds to humidity values. (a) The dashed blue lines indicate the processed temperature and humidity values captured by the RPi integrated DHT22 sensor. The solid red lines indicate the corresponding values captured by the commercial sensor. (b) The correlation plots between values captured from the DHT22 sensor and the commercial sensor. The solid-blue circles (●) indicate individual temperature and humidity tuples. The linear fits on the data and their deviations from the 45° line are depicted in the plot. (c) The Bland-Altman plot between the two measurements. The solid black lines indicate the mean difference between the two measurements. The dashed black lines indicate the +1.96 and the -1.96 standard deviation (SD) lines for the difference between the two measurements.

capture whether a patient is being mechanically ventilated (and at what frequency), groans and expression of pain, and other non-verbal utterances. It is possible to add speaker and voice recognition to the code base, to identify who is speaking and about what, providing insight into clinical (and non-clinical) discussions that may provide additional diagnostic power. For instance, by differentiating patients from family members, it is possible to assess both the level of clinical team support and frequency of bedside visits and the social support that a patient may have (inferred by the number and duration of visitations by friends and family). Tracking the time clinicians spend with patients and the level of expertise available could help identify gaps in care. In addition, by tracking Bluetooth transmitter strength of body-worn devices (e.g., badges or phones), it is possible to infer motion, an individual's identity (through a look-up table), and even the exact location of an individual if more than one Bluetooth receiver base is used. Real-time and accurate tracking of humans using Bluetooth beacons needs a receiver (RPi) sensor network. Further, we can have a central server where all the RPi's communicate and update the collected RSSI values. We can then have algorithms operating on this database in real-time to perform the geolocation of humans. We have implemented this system in a clinical environment at Emory Healthcare, Atlanta, USA, to monitor the real-time location of humans.

By capturing motion via video, we can probe even deeper into assessing the patient and their environment. For instance, we can estimate the quantity of sleep, sedation, and agitation that a patient experiences, all of which have been linked to recovery [73]. At the same time, if the motion is associated with clinical care, then the intensity of activity can indicate when treatments, observations, or specific activities (such as replacing drips) took place. While we know that the ratio of nursing staff to patient impacts outcomes [7], there are no studies that examine the time at the bedside and the actions taken at the bedside in terms of their impact on the outcome. However, it is known that time spent at the bedside is linked to improved patient satisfaction [74].

Finally, the data modalities we capture provide us a unique opportunity to perform multimodal analysis of the patient state. For instance, consider the case of human sleep monitoring in a home environment. All we need to do is to place the Bluetooth beacon in the patient's clothes before they sleep. The motion signals captured during the patient's sleep give us the timestamps when the patient moved in the bed. Based on the intensity of the motion signal, we can delineate minor movements (rolling over) from major movements (sitting up in bed). The simultaneous recordings of audio-features and illuminance, which can act both as sleep inhibitors (flushing toilet or turning ON lights) and wake-event markers, give us valuable information about the patient's sleeping patterns. Further, the RSSI signal analysis will provide us with all the times that the patient exits the room during the night. Finally, via a long-term monitoring protocol, we can recommend ideal sleeping conditions to the patient by monitoring the temperature and humidity in the room. Thus, the system presented here provides a low-cost method for performing deep analysis, both at home and in a clinical setting. The system itself has been deployed for patient monitoring and data collection in two separate healthcare facilities located in the United States of America: (1) A New York state department of health funded center for excellence facility that offers residential, medical, clinical and special education programs to the residents (25 units); and (2) A Mild Cognitive Impairment rehabilitation program facility at Emory Healthcare in Atlanta (40 units). For a 12 hour (7 AM to 7 PM) recording of all data modalities (except PoseNet), a total of 832 MB of data was recorded.

6. Conclusion

This article presents an overview of a generalized open-source system for edge computing in clinical and home environments. It provides real-time data elements and analysis that are not generally present in electronic medical records yet are associated with clinical performance, diagnosis, and outcomes. In particular, we focused on the acoustic environment (such as speech, alarms, and environmental noise), the motion of clinicians and patients, and location (absolute or relative to others). We also added optional camera-

based analysis and environmental sensors (temperature, humidity, and light). Further, we included methods for privacy-preserving feature extraction to provide a generally acceptable system that is unlikely to violate hospital policies and other privacy regulations, which may reduce the anxiety of administrators and clinicians concerning the level of monitoring. Encryption and data transfer protocols were not included as these are specific to each institution. The implementation on a state-of-the-art extensible edge computing system at a relatively low cost provides a high degree of flexibility in the design. The bill-of-materials and open-source code to replicate the work described here have been made publicly available under an open-source license [54].

Author Contributions: Conceptualization, G.D.C.; methodology, P.B.S., C.H., Z.J. and G.D.C.; software, P.B.S.; validation, P.B.S., C.H. and G.D.C.; formal analysis, P.B.S.; writing—original draft preparation, P.B.S., C.H., Z.J. and G.D.C.; writing—review and editing, P.B.S. and G.D.C.; supervision, G.D.C.; project administration, G.D.C.; funding acquisition, G.D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by an Amazon Web Services Machine Learning Research Award, the Cox Foundation and the National Science Foundation, grant # 1822378 ‘Leveraging Heterogeneous Data Across International Borders in a Privacy Preserving Manner for Clinical Deep Learning’. Gari D. Clifford is also funded by the National Center for Advancing Translational Sciences of the National Institutes of Health under Award Number UL1TR002378 and National Institute of Environmental Health Sciences under grant # 2P30ES019776-05. The content is solely the responsibility of the authors and does not necessarily represent the official views of the authors’ sponsors and employers.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable

Data Availability Statement: Not applicable

Acknowledgments: We would like to thank the Cognitive Empowerment program which is supported by Emory’s Brain Health Center, Georgia Institute of Technology, and a generous investment from Cox Enterprises, Inc. and the James M. Cox Foundation. Further, we would like to thank Amazon Web Services, Inc., National Center for Advancing Translational Sciences and National Institute of Environmental Health Sciences for their support. We would like to thank Matt Doiron, Technology Specialist in the Mild Cognitive Impairment Executive Park (MCIEP) program and Venkata Siva Krishna Madala, Graduate Assistant, Georgia Institute of Technology for agreeing to be photographed which has been used in Figure 2 in this paper. Further, we would like to thank Jacob Zelko, Georgia Tech Research Institute for his contribution in developing the core scripts for detecting Bluetooth beacons and recording ambient temperature and humidity values using a Raspberry Pi.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

OR	Operating Room
ICU	Intensive Care Unit
PIR	Passive Infrared
USB	Universal Serial Bus
TPU	Tensor Processing Unit
RPi	Raspberry Pi
HIPAA	Three letter acronym
IR	Infrared
ECG	Electrocardiogram
BAN	Body Area Network
RF	Radio Frequency
GPIO	General Purpose Input Output
NoIR	No Infrared
GDS	Global Difference Sum
GDPC	Global δ -Pixel Count
LDS	Local Difference Sum
LDPC	Local δ -Pixel Count
ISC	Internet Systems Consortium
STFT	Short Time Fourier Transform
MFCC	Mel Frequency Cepstral Coefficient
RSSI	Received Signal Strength Indicator
MAC	Media Access Control
LUX	Lumen per Square Foot
DC	Direct Current
RH	Relative Humidity
IQR	Interquartile Range
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
dBm	Decibels with reference to one milliwatt
RMSE	Root Mean Square Error
SD	Standard Deviation
MCIEP	Mild Cognitive Impairment Executive Park

References

1. Ma, A.J.; Rawat, N.; Reiter, A.; Shrock, C.; Zhan, A.; Stone, A.; Rabiee, A.; Griffin, S.; Needham, D.M.; Saria, S. Measuring patient mobility in the ICU using a novel noninvasive sensor. *Critical Care Medicine* **2017**, *45*, 630.

2. Davoudi, A.; Malhotra, K.R.; Shickel, B.; Siegel, S.; Williams, S.; Ruppert, M.; Bihorac, E.; Ozrazgat-Baslanti, T.; Tighe, P.J.; Bihorac, A.; et al. Intelligent ICU for autonomous patient monitoring using pervasive sensing and deep learning. *Scientific Reports* **2019**, *9*, 1–13.

3. Schmid, F.; Goepfert, M.S.; Reuter, D.A. Patient monitoring alarms in the ICU and in the operating room. *Annual Update in Intensive Care and Emergency Medicine* **2013**, pp. 359–371.

4. Ryherd, E.; Ackerman, J.; Zimring, C.; Okcu, S.; Persson Wayne, K. Noise pollution in Hospitals: Impacts on staff. *Journal of Clinical Outcomes Management* **2012**, *19*, 1–10.

5. Choiniere, D.B. The effects of hospital noise. *Nursing Administration Quarterly* **2010**, *34*, 327–333. doi:10.1097/naq.0b013e3181f563db.

6. Jaiswal, S.J.; Garcia, S.; Owens, R.L. Sound and light levels are similarly disruptive in ICU and non-ICU wards. *Journal of Hospital Medicine* **2017**, *12*, 798–804.

7. Wise, J. Higher nurse to patient ratio is linked to reduced risk of inpatient death. *BMJ* **2016**, 352.

8. Frisby, J.; Smith, V.; Traub, S.; Patel, V.L. Contextual computing: A Bluetooth based approach for tracking healthcare providers in the emergency room. *Journal of Biomedical Informatics* **2017**, *65*, 97–104.

9. Vankipuram, A.; Patel, V.L. Automated Location Tracking in Clinical Environments: A Review of Systems and Impact on Workflow Analysis. In *Cognitive Informatics*; Springer, 2019; pp. 235–253.

10. White, R.D. Recommended standards for the newborn ICU. *Journal of Perinatology* **2007**, *27*, S4–S19.

11. Wilde-Frenz, J.; Schulz, H. Rate and distribution of body movements during sleep in humans. *Perceptual and Motor Skills* **1983**, *56*, 275–283.

12. Suresha, P.B.; Cakmak, A.S.; Da Poian, G.; Shah, A.J.; Vaccarino, V.; Bliwise, D.; Clifford, G.D. Obstructive Sleep Apnea Classification in a Mixed-Disorder Elderly Male Population Using a Low-Cost Off-Body Movement Sensor. In Proceedings of the Proceedings of 2019 IEEE-EMBS International Conference on Biomedical & Health Informatics. IEEE, 2019, pp. 1–4.

13. Liang, P.C.; Krause, P. Real-time indoor patient movement pattern telemonitoring with one-meter precision. In Proceedings of the 4th International Conference on Wireless Mobile Communication and Healthcare-Transforming Healthcare Through Innovations in Mobile and Wireless Technologies. IEEE, 2014, pp. 141–144.
14. Baker, C.R.; Armijo, K.; Belka, S.; Benhabib, M.; Bhargava, V.; Burkhart, N.; Der Minassians, A.; Dervisoglu, G.; Gutnik, L.; Haick, M.B.; et al. Wireless sensor networks for home health care. In Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops. IEEE, 2007, Vol. 2, pp. 832–837.
15. Kim, S.H.; Jeong, J.M.; Hwang, M.T.; Kang, C.S. Development of an IoT-based atmospheric environment monitoring system. In Proceedings of the International Conference on Information and Communication Technology Convergence. IEEE, 2017, pp. 861–863.
16. Geder, E.; Clifford, G.D. Fusion of image and signal processing for the detection of obstructive sleep apnea. In Proceedings of the Proceedings of 2012 IEEE-EMBS International Conference on Biomedical and Health Informatics. IEEE, 2012, pp. 890–893.
17. Wang, C.W.; Hunter, A.; Gravill, N.; Matusiewicz, S. Unconstrained video monitoring of breathing behavior and application to diagnosis of sleep apnea. *IEEE Transactions on Biomedical Engineering* **2013**, *61*, 396–404.
18. Deng, F.; Dong, J.; Wang, X.; Fang, Y.; Liu, Y.; Yu, Z.; Liu, J.; Chen, F. Design and implementation of a noncontact sleep monitoring system using infrared cameras and motion sensor. *IEEE Transactions on Instrumentation and Measurement* **2018**, *67*, 1555–1563.
19. Hegde, C.; Jiang, Z.; Suresha, P.B.; Zelko, J.; Seyedi, S.; Smith, M.A.; Wright, D.W.; Kamaleswaran, R.; Reyna, M.A.; Clifford, G.D. Autotriage - an open source edge computing raspberry pi-based clinical screening system. *medRxiv* **2020**.
20. Rienzo, M.D.; Mukkamala, R. Wearable and Nearable Biosensors and Systems for Healthcare. *Sensors (Basel)* **2021**. doi:10.3390/s21041291.
21. Romdhane, R.; Mulin, E.; Derreumeaux, A.; Zouba, N.; Piano, J.; Lee, L.; Leroi, I.; Mallea, P.; David, R.; Thonnat, M.; et al. Automatic video monitoring system for assessment of Alzheimer's disease symptoms. *The Journal of Nutrition, Health & Aging* **2012**, *16*, 213–218.
22. Suzuki, S.; Matsui, T.; Kagawa, M.; Asao, T.; Kotani, K. An approach to a non-contact vital sign monitoring using dual-frequency microwave radars for elderly care. *Journal of Biomedical Science and Engineering* **2013**, *6*. doi:10.4236/jbise.2013.67086.
23. Matsui, T.; Yoshida, Y.; Kagawa, M.; Kubota, M.; Kurita, A. Development of a practicable non-contact bedside autonomic activation monitoring system using microwave radars and its clinical application in elderly people. *Journal of Clinical Monitoring and Computing* **2013**, *27*, 351–356.
24. Tsumura, N.; Kanai, C.; Masui, K.; Mitsunashi, R.; Nagasawa, T.; et al. Automatic classification of adult males with and without autism spectrum disorder by non-contact measurement of autonomic nervous system activation. *Frontiers in Psychiatry* **2021**, *12*, 573.
25. Piwek, L.; Ellis, D.A.; Andrews, S.; Joinson, A. The Rise of Consumer Health Wearables: Promises and Barriers. *PLOS Medicine* **2016**, *13*, e1001953. doi:10.1371/journal.pmed.1001953.
26. Dunn, J.; Runge, R.; Snyder, M. Wearables and the medical revolution. *Personalized Medicine* **2018**, *15*, 429–448. doi:10.2217/pme-2018-0044.
27. Yetisen, A.K.; Martinez-Hurtado, J.L.; Ünal, B.; Khademhosseini, A.; Butt, H. Wearables in Medicine. *Advanced Materials* **2018**, *30*, 1706910. doi:10.1002/adma.201706910.
28. Chen, M.; Gonzalez, S.; Vasilakos, A.; Cao, H.; Leung, V.C.M. Body Area Networks: A Survey. *Mobile Networks and Applications* **2011**, *16*, 171–193. doi:10.1007/s11036-010-0260-8.
29. Catherwood, P.A.; Bukhari, S.S.; Watt, G.; Whittow, W.G.; McLaughlin, J. Body-centric wireless hospital patient monitoring networks using body-contoured flexible antennas. *IET Microwaves, Antennas & Propagation* **2018**, *12*, 203–210. doi:10.1049/iet-map.2017.0604.
30. Movassaghi, S.; Abolhasan, M.; Lipman, J.; Smith, D.; Jamalipour, A. Wireless Body Area Networks: A Survey. *IEEE Communications Surveys Tutorials* **2014**, *16*, 1658–1686. doi:10.1109/SURV.2013.121313.00064.
31. Monton, E.; Hernandez, J.; Blasco, J.; Herve, T.; Micallef, J.; Grech, I.; Brincat, A.; Traver, V. Body area network for wireless patient monitoring. *IET Communications* **2008**, *2*, 215. doi:10.1049/iet-com:20070046.
32. Lin, F.; Zhuang, Y.; Song, C.; Wang, A.; Li, Y.; Gu, C.; Li, C.; Xu, W. SleepSense: A Noncontact and Cost-Effective Sleep Monitoring System. *IEEE Transactions on Biomedical Circuits and Systems* **2017**, *11*, 189–202. doi:10.1109/TBCAS.2016.2541680.
33. Liang, Q.; Xu, L.; Bao, N.; Qi, L.; Shi, J.; Yang, Y.; Yao, Y. Research on Non-Contact Monitoring System for Human Physiological Signal and Body Movement. *Biosensors* **2019**, *9*, 58. doi:10.3390/bios9020058.
34. Wang, F.; Skubic, M.; Rantz, M.; Cuddihy, P.E. Quantitative Gait Measurement With Pulse-Doppler Radar for Passive In-Home Gait Assessment. *IEEE Transactions on Biomedical Engineering* **2014**, *61*, 2434–2443. doi:10.1109/TBME.2014.2319333.
35. Lee, Y.S.; Pathirana, P.N.; Caelli, T.; Evans, R. Doppler radar in respiratory monitoring: Detection and analysis. IEEE, 2013, pp. 224–228. doi:10.1109/ICCAIS.2013.6720558.
36. Gu, C.; Li, C.; Lin, J.; Long, J.; Huangfu, J.; Ran, L. Instrument-Based Noncontact Doppler Radar Vital Sign Detection System Using Heterodyne Digital Quadrature Demodulation Architecture. *IEEE Transactions on Instrumentation and Measurement* **2010**, *59*, 1580–1588. doi:10.1109/TIM.2009.2028208.
37. Drzymalski, D.; Ceruzzi, J.; Camann, W. Noise in the obstetric operating room. *International Journal of Obstetric Anesthesia* **2017**, *29*, 87–88.

38. Keller, S.; Tschan, F.; Semmer, N.K.; Holzer, E.; Candinas, D.; Brink, M.; Beldi, G. Noise in the operating room distracts members of the surgical team. An observational study. *World Journal of Surgery* **2018**, *42*, 3880–3887.
39. Murthy, V.; Malhotra, S.; Bala, I.; Raghunathan, M. Detrimental effects of noise on anaesthetists. *Canadian Journal of Anaesthesia* **1995**, *42*, 608.
40. Shapiro, R.A.; Berland, T. Noise in the operating room. *New England Journal of Medicine* **1972**, *287*, 1236–1238.
41. Way, T.J.; Long, A.; Weihing, J.; Ritchie, R.; Jones, R.; Bush, M.; Shinn, J.B. Effect of noise on auditory processing in the operating room. *Journal of the American College of Surgeons* **2013**, *216*, 933–938.
42. Guerlain, S.; Adams, R.B.; Turrentine, F.B.; Shin, T.; Guo, H.; Collins, S.R.; Calland, J.F. Assessing team performance in the operating room: Development and use of a “black-box” recorder and other tools for the intraoperative environment. *Journal of the American College of Surgeons* **2005**, *200*, 29–37.
43. Azevedo-Coste, C.; Pissard-Gibollet, R.; Toupet, G.; Fleury, É.; Lucet, J.C.; Birgand, G. Tracking Clinical Staff Behaviors in an Operating Room. *Sensors* **2019**, *19*, 2287.
44. Adib, F.; Kabelac, Z.; Katabi, D. Multi-person localization via RF body reflections. In Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), 2015, pp. 279–292.
45. Kianoush, S.; Savazzi, S.; Vicentini, F.; Rampa, V.; Giussani, M. Device-free RF human body fall detection and localization in industrial workplaces. *IEEE Internet of Things Journal* **2016**, *4*, 351–362.
46. Tian, Y.; Lee, G.H.; He, H.; Hsu, C.Y.; Katabi, D. RF-based fall monitoring using convolutional neural networks. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **2018**, *2*, 1–24.
47. Aschoff, J. Circadian rhythms: Influences of internal and external factors on the period measured in constant conditions. *Zeitschrift für Tierpsychologie* **1979**, *49*, 225–249.
48. Campbell, S.S.; Dawson, D. Enhancement of nighttime alertness and performance with bright ambient light. *Physiology & Behavior* **1990**, *48*, 317–320.
49. Honma, K.; Honma, S.; Kohsaka, M.; Fukuda, N. Seasonal variation in the human circadian rhythm: Dissociation between sleep and temperature rhythm. *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* **1992**, *262*, R885–R891.
50. Saper, C.B.; Lu, J.; Chou, T.C.; Gooley, J. The hypothalamic integrator for circadian rhythms. *Trends in Neurosciences* **2005**, *28*, 152–157.
51. Okamoto-Mizuno, K.; Mizuno, K. Effects of thermal environment on sleep and circadian rhythm. *Journal of Physiological Anthropology* **2012**, *31*, 1–9.
52. Verceles, A.C.; Silhan, L.; Terrin, M.; Netzer, G.; Shanholtz, C.; Scharf, S.M. Circadian rhythm disruption in severe sepsis: The effect of ambient light on urinary 6-sulfatoxymelatonin secretion. *Intensive Care Medicine* **2012**, *38*, 804–810.
53. Blume, C.; Garbaza, C.; Spitschan, M. Effects of light on human circadian rhythms, sleep and mood. *Somnologie* **2019**, *23*, 147–156.
54. Suresha, P.B.; Clifford, G.D. Sedline-Root-EEG-Toolbox. https://github.com/cliffordlab/OS_Edge_Compute_Data_Capture_RPi, 2021.
55. Pi NoIR Camera V2. <https://www.raspberrypi.org/products/pi-noir-camera-v2/>. [Online; accessed 24-Feb-2021].
56. Papandreou, G.; Zhu, T.; Chen, L.C.; Gidaris, S.; Tompson, J.; Murphy, K. Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model. In Proceedings of the Proceedings of the European Conference on Computer Vision, 2018, pp. 269–286.
57. Papandreou, G.; Zhu, T.; Kanazawa, N.; Toshev, A.; Tompson, J.; Bregler, C.; Murphy, K. Towards accurate multi-person pose estimation in the wild. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4903–4911.
58. Cao, Z.; Hidalgo, G.; Simon, T.; Wei, S.E.; Sheikh, Y. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2019**, *43*, 172–186.
59. Mathis, A.; Mamidanna, P.; Cury, K.M.; Abe, T.; Murthy, V.N.; Mathis, M.W.; Bethge, M. DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience* **2018**, *21*, 1281–1289.
60. Python-sounddevice. <https://python-sounddevice.readthedocs.io/>. [Online; accessed 24-Feb-2021].
61. Github: Python-sounddevice. <https://github.com/spatialaudio/python-sounddevice/>. [Online; accessed 24-Feb-2021].
62. Librosa. <https://librosa.org/doc/latest/index.html>. [Online; accessed 24-Feb-2021].
63. Multiscale Entropy. <https://www.physionet.org/physiotools/mse/mse.c>. [Online; accessed 24-Feb-2021].
64. Bluepy. <https://ianharvey.github.io/bluepy-doc/>. [Online; accessed 24-Feb-2021].
65. Hegde, C.; Sameni, R.; Rad, A.B.; Clifford, G.D. Modeling Social Distancing and Quantifying Epidemic Disease Exposure in a Built Environment. *IEEE Journal of Selected Topics in Signal Processing* **2022**, pp. 1–1. doi:10.1109/JSTSP.2022.3145622.
66. Block Jr, F.E.; Rouse, J.D.; Hakala, M.; Thompson, C.L. A proposed new set of alarm sounds which satisfy standards and rationale to encode source information. *Journal of clinical monitoring and computing* **2000**, *16*, 541.
67. ISO/IEC 60601-1-8, Patterson and other alarms in medical equipment. <http://www.anaesthesia.med.usyd.edu.au/resources/alarms/>. [Online; accessed 02-Jun-2020].
68. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
69. XGBoost. <https://xgboost.readthedocs.io/en/latest/index.html>. [Online; accessed 24-Feb-2021].

-
70. Takahashi, K.; Yamamoto, K.; Kuchiba, A.; Koyama, T. Confidence interval for micro-averaged F1 and macro-averaged F1 scores. *Applied Intelligence* **2021**, pp. 1–12.
 71. Afouras, T.; Chung, J.S.; Zisserman, A. The Conversation: Deep Audio-Visual Speech Enhancement. In Proceedings of the Interspeech, 2018.
 72. Wallace, C.; Robins, J.; Alvord, L.; Walker, J. The effect of earplugs on sleep measures during exposure to simulated intensive care unit noise. *American Journal of Critical Care* **1999**, *8*, 210–219.
 73. Weinhouse, G.L.; Watson, P.L. Sedation and sleep disturbances in the ICU. *Critical Care Clinics* **2009**, *25*, 539–549.
 74. Mitchell, A.; Gudeczauskas, K.; Therrien, A.; Zauher, A. Bedside reporting is a key to communication. *Journal of Healthcare Communications* **2018**, *3*, 13.