# Non-Invasive Motion Detection and Classification in NICU Patients using Ballistographic Signals from a Pressure Sensitive Mat

by

Samreen Aziz

A thesis submitted in conformity with the requirements

for the degree of:

Masters in Applied Sciences

in

Biomedical Engineering

with Specialization in Data Science

Department of Systems and Computer Engineering

Carleton University

# Abstract

Patient movements can cause motion artifacts on physiological signals and can result in false alarms in a continuous patient care environment. This thesis explores the use of centre of pressure (COP) signals from a pressure sensitive mat, placed below neonates in the neonatal intensive care unit to (a) detect patient movement in real-time, and (b) classify the source of movement as upper or lower body. The COP is the sum of all vectors of pressure acting on the PSM; it is the point where the total force due to pressure is equal on both halves of the PSM. To achieve (a) the sum distance travelled by the COP is tracked over time using a sliding window with data from seven patients. Windows exhibiting large deviations in the COP are indicative of patient motion. Window boundary suppression led to improved movement detection with precision scores of 0.84 and recall of 0.71 at a window length of 10 seconds in a real-time sliding window approach. To address problem (b) – which has not previously been attempted in non-invasive monitoring research – six features were derived from the COP, and feature selection was done with out-of-bag error feature importance ranking with a random forest to remove anomalous features, then the feature set was further reduced using sequential forward selection for use in SVMs trained with leave-one-group-out cross-validation. Balancing training and test data with SMOTE or majority under sampling improved classifier performance by approximately two-fold. It was found that using a sample imputation approach of adding ~13 minutes of hand-annotated new subject data to the training set makes the classifier most useful to a new patient, producing accuracy scores of ~87.29%, precision of 0.90, and recall of 0.84. These models and insights may be used in a real-time motion detection algorithm and create a foundation for future work in limb movement classification and detection.

# Acknowledgments

First, I would like to thank the thesis committee – Dr. James Green, Dr. Marzieh Amini, and Dr. Hala Assal from Carleton University and Dr. WonSook Lee from the University of Ottawa – for taking the time to read this thesis and participate in my defense. Without you there would be no finish line for me to strive towards, so I thank you for your time, evaluation, and consideration of the work I have done throughout my degree.

I would also like to thank my husband, Arman, for doing everything in his power to preserve in me a tough mindset over the final few months of my thesis, spending countless hours reviewing my dense work despite being a social science major, staying in Toronto with me so I would have the support of my family. Thank you for bringing your life over to Canada from the United States to not only spend a lifetime with me, but also to be a supportive part of my journey as I completed this degree. Thank you for being so kind when I got lazy, patient when I got frustrated, and excited when I achieved milestones. Thank you for loving me unconditionally, and for holding me when I needed it most. I promise to support you and be patient with your growth as you've been with mine, now and forever.

I thank my Nanima, Dr. Afsari Begum, a celebrated pediatrician in Hyderabad, India. Thank you for your work in paediatrics in India and for being a constant pillar of support and prayer in my life.

I thank my parents for their faith in me as I embarked into a completely different field of study in a completely different city for the first time on my own. You brought Maryam and I to Canada and overcame hardships along the way for the sake of our successes– I hope that I have been able to

make you proud, truly. Baba, you have been a good listener when the going got tough and provided the encouragement and objective perspectives that I needed. Ammi, you were always there to celebrate every little success, to be my biggest support in the challenges I have overcome in life, and as a woman you have shown me what it means to be strong and to make it through in a new field. Without my parents, I would hardly know the meaning of strength and perseverance.

I thank my best friend and sister, Maryam. Thank you for always picking up your phone, talking to me over FaceTime, and making sure I didn't go through stressors alone. Your love and light are a big part of why I made it through, a big part of why I never gave up. You never stopped believing in me. My best friend, my right hand.

I thank my supervisor Dr. Green for believing in me before I even entered the program and for his continuous guidance and support throughout the writing process, for his continuous patience and understanding, and his encouragement and positive energy no matter what came my way. I am endlessly grateful to have found a supervisor like him, I couldn't have asked for better and I can't imagine a better fit. I would like to thank my team at Aviva and the department of graduate studies at Carleton for their patience with my progress. I would also like to thank my friend, colleague, and upcoming PhD, Yasmina Souley Dosso for the countless nights she spent keeping me company as I worked, celebrating successes, and being my mentor and big sister. I cherish the friendship I have made with you and thank you endlessly for your constant support.

Lastly, I would like to thank the Children's Hospital of Eastern Ontario, IBM, and Carleton University for making it possible for me to have the necessary data and funding required to complete my thesis.

# Table of Contents

# List of Tables

# List of Figures

# List of Appendices

# Chapter 1

# Introduction

This chapter introduces the motivation of this work, the problem statement, all publications arising from this research, and the organization of this thesis.

## 1.1 Motivation

The neonatal intensive care unit (NICU) is populated with some of the most fragile patients; some of whom have not completed a full gestational term and many of whom are subjects of neurological and other physiological diseases. Newborns in the NICU are kept under continuous monitoring for variations in vital signs such as respiration rate, heart rate, blood-oxygen saturation, as well as the periodic monitoring of neurological function using electroencephalographic signals. Currently, the most effective digital methods for continuous monitoring impede the patient and caregiver experiences. Multiple electrodes adhered to newborn skin can cause discomfort to the infant while in the incubator or crib. Additionally, the monitoring wires can impede both parental interactions with the patient. Lastly, patient movements can reduce the accuracy of monitoring data due to motion artifacts that can lead to false alarms, exacerbating nurse fatigue and encouraging them to overlook or doubt the validity of alarms.

False alarms often occur when vital signs fall below or above certain thresholds. Such threshold breaches may be falsely caused by motion artifacts during periods of patient movement.

To address the problem of invasive monitoring and false alarm gating, an algorithm using a non-invasive technology to detect patient motion during the monitoring period can

inform monitoring systems of motion activity before it alerts a nurse with an alarm. Additionally, an algorithm that can distinguish between upper and lower body motion could form the basis for developing a more complex algorithm that can classify patterns of movements corresponding to specific clinical interventions. Specifically, if an algorithm can detect lower body motion as the positive class from upper body motion, false alarms caused by blood pressure monitors and pulse oximeters during lower body movements can be gated.

The pressure sensor mat (PSM) has been previously shown to be effective in detecting movements in adults [1]–[6]. It is unclear whether this technology is also suitable for movement detection in neonates. Automated monitoring of movement events in the NICU can improve resource management, reduce alarm fatigue in clinicians, and improve the quality of care that neonates receive in the NICU.

## 1.2 Problem Statement

This project will address the problem of false alarms caused by movement artifacts during periods of patient monitoring in the NICU by developing a motion detection and classification algorithm that uses raw ballistographic signal data from neonates resting on non-invasive PSM. Features such the sum distance travelled by the centre of pressure (COP), average lateral COP, average longitudinal COP, lateral COP variance, longitudinal COP variance, and velocity of COP movement will be extracted from the PSM data to develop algorithms that can detect the presence of motion, and further classify motion originating from upper versus lower extremities. The primary goals of this work are to establish that PSM can be used for movement detection in neonates and to create a set of algorithms to non-invasively and unobtrusively detecting motion and motion subtypes

during periods of neonatal patient monitoring. Such a system could ultimately be used to gate false alarms caused by motion artifacts.

## 1.3 Publications Arising from this Research

Several publications have arisen from the research reported in this thesis. For each publication below, I clarify my role in the study:

- Dosso, Y. S., Aziz, S., Nizami, S., Greenwood, K., Harrold, J., & Green, J. R. (2020, July). Video-based neonatal motion detection. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 6135-6138). IEEE.
    - *Contribution:* Participated as a bed-side observer annotating movement events, collecting PSM signal data and RGB-D video data. This paper detailed the use of video for movement detection in neonates and differs from the current thesis in the sensor modality (video vs. PSM).
- Dosso, Y. S., Aziz, S., Nizami, S., Greenwood, K., Harrold, J., & Green, J. R. (2020, June). Neonatal Face Tracking for Non-Contact Continuous Patient Monitoring. In *2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* (pp. 1-6). IEEE.
    - *Contribution:* Participated as a bed-side observer annotating movement events, collecting PSM signal data and RGB-D video data. The video data used in this research was collected by myself and the team.
- Aziz, S., Dosso, Y. S., Nizami, S., Greenwood, K., Harrold, J., & Green, J. R. (2020, June). Detection of Neonatal Patient Motion Using a Pressure-Sensitive Mat.

In *2020 IEEE International Symposium on Medical Measurements and Applications (MeMeA)* (pp. 1-6). IEEE.

- *Contribution:* Established that a PSM can be used to detect neonatal patient movements and developed a proof-of-concept algorithm for motion detection using ballistographic signals. Demonstrated that *motion* and *no_motion* events can be distinguished using a threshold applied to the sum of distance travelled by the COP over a sliding window.
- Awarded the ***Best Student Paper Award*** at the 2020 IEEE Measurements and Medical Applications (MeMeA) Conference held in Bari, Italy (*virtual due to COVID-19*).

## 1.4 Thesis Structure

This thesis is divided into six chapters:

- *Chapter 1* introduces the motivation of this work, the problem statement, and all publications arising from this research.
- *Chapter 2* discusses the background knowledge required to understand the work in this research. The first section reviews the NICU and alarm fatigue, the PSM which is the technology through which primary data for this work was collected, and machine learning in medical applications. The second part reviews existing literature related to this research including: general motion detection systems in patient monitoring, applications of the PSM in motion detection, and other applications of the PSM in patient monitoring.
- *Chapter 3* reviews characteristics of the patients in this study as well as the inclusion and exclusion criteria for selection, the experimental set-up at the time of

data collection and the technologies involved, the event classification protocol, and a description of the data and data preparation methods.

- *Chapter 4* describes the methods used to develop a discrete classification and real-time detection algorithm that distinguishes between baseline *motion* and *no_motion* events using the sum of distances travelled by the COP as a feature.
- *Chapter 5* describes the methods used to develop a discrete classification and real-time detection algorithm that distinguishes between upper and lower body motion for use as a classification and refinement step following predicted motion by the baseline model.
- *Chapter 6* discusses the outcomes that have been achieved in this research, the limitations of this work at present, implications of the results, and future work.

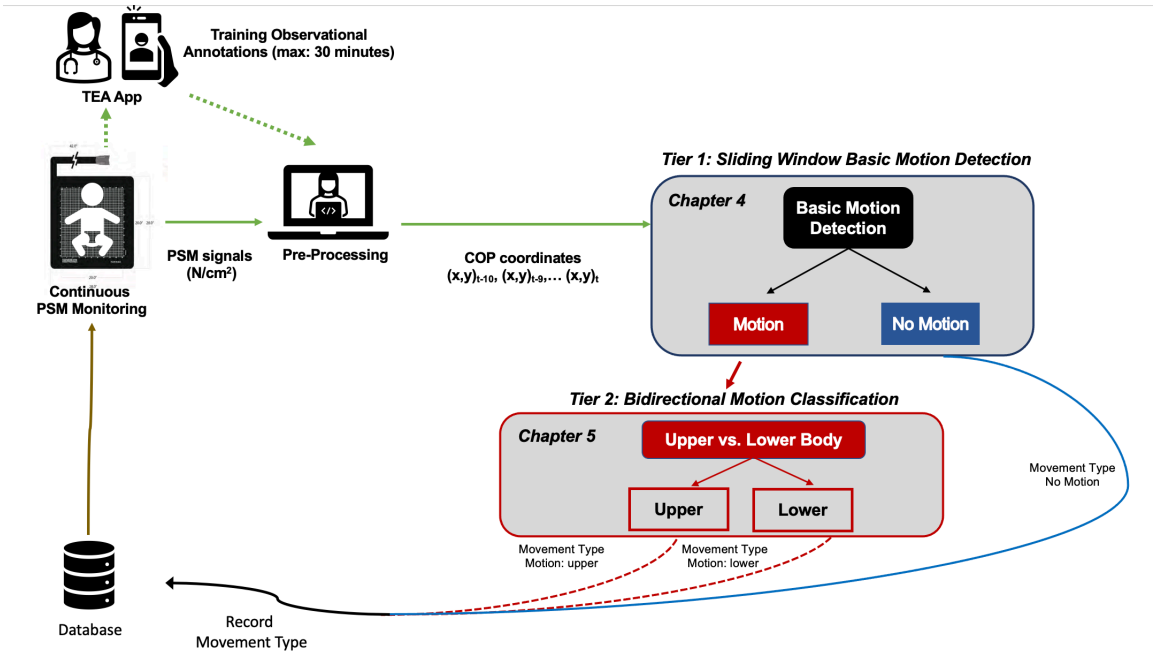



**Figure 1.** Thesis framework with data flow from PSM, detection, classification and recording steps.

# Chapter 2

# Background and Literature Review

## 2.1 Background

### 2.1.1 The NICU Environment and Alarm Fatigue

In the first few days after birth, infants require little to no stimulation; ill or premature patients need even less as they are not fully developed, their brains cannot cope with extreme reactions, and their hearts cannot cope with elevated heart rates. For these reasons, the neonatal intensive care unit is characterized as a controlled and low stimulation environment.

Infants are cared for in one of three types of beds: an incubator, crib, or a radiant/overhead warmer. Incubators are enclosed clear plastic cribs that keep infants warm and minimize their exposure to environmental germs and potential hazards, while also providing access ports for nurses to provide care as required. Overhead warmers are not enclosed in a full-coverage plastic crib but provide a radiant heat source. They are used for infants that require more frequent handling by nurses and other care staff. Cribs are used for healthier and more developed neonates, often approaching their release date.

Given the fragility of newborns and namely those that are pre-term or ill, they require continuous monitoring. The types of monitors in the NICU include the blood pressure monitor, cardiopulmonary monitor, carbon dioxide/transcutaneous oxygen monitors, and an electroencephalogram monitor as required. When monitoring levels drop below or exceed healthy specified thresholds, an alarm sounds in the NICU requiring a nurse to respond [7]. Each alarm has a different level of urgency, but every alarm necessarily

requires the primary care nurse to identify, assess, and confirm the source of the alarm – this can be mentally and physically taxing. The most common source of false alarms is the transcutaneous oxygen monitor (SpO2) followed by the cardiopulmonary monitor which alarms in the case of bradycardia. A 2002 study found that in each hour a primary care nurse typically responds to 13 – 15 alarms per hour, that 68% were false alarms, and 94% had no clinical significance [8][9].

The issue of NICU false alarms is associated with alarm fatigue which can negatively impact the quality of patient care and patient safety [10]. Fatigue is defined as a lack of energy to act; it can be acute and pass after a brief period of rest, or it can be chronic as characterized by irreversible physical and mental exhaustion, as can be caused by an excessive exposure to monitoring devices in the NICU [11].

In a survey of 48 nurses in Australia, 93% reported that fatigue caused by alarms can lead to silencing or ignoring them. 81% of nurses believed that fatigue is caused mostly by the mental exhaustion of reacting to an alarm in panic only to realize it is a false alarm; it is a consumption of mental and physical resources that could be used towards a true alarm. 59% of nurses associate the false alarms with improperly set thresholds, and more importantly, improperly trained alarm accuracy [12].

In another study, it was found that 87.5% of false alarms are caused by the movement of neonates [13]. For these reasons, a false alarm gating solution that is independent of or integrated with monitoring devices should be innovated to assure that the quality of care, patient safety, and the mental health and physical safety of nurses is managed more effectively.

### 2.1.2 Pressure Sensor Mat

The PSM is an unobtrusive and non-invasive tool used for measuring time-varying contact pressure and mapping pressure at multiple sensing sites (sensels). Whether the contact is achieved by human subjects sitting, standing, or laying down on the mat, the PSM can produce a range of pressure signals over time and space to describe movement and pressure. PSMs are available in a variety of different shapes and sizes depending on application type, and often come with accompanying software for visualization and data extraction. Various technologies can be used to create the sensels used in the PSM including optical, relative, and capacitive sensors.

The PSM's capabilities have been explored in healthcare research as an unobtrusive monitoring tool for a range of clinical areas and conditions. To name a few applications of the PSM, the PSM has also been used as a real-time data collection tool to support geriatric in-bed mobility detection for fall prevention, ulcer prevention, sleep state classification, respiration rate estimation, and the detection of detrimental sitting positions in children to avoid musculoskeletal disorders [1], [14]–[17]. Applications of the PSM in motion detection and healthcare are further discussed in *Section 2.2.2* and *Section 2.2.3*.

As mentioned above, the pressure sensor mat is a tool for measuring contact pressure but does not necessarily require direct contact with skin or other objects that will apply pressure on it. The structure of the PSM in this study includes isolation, conduction, and protective layers above and below the sensing layer. The protective layer prevents electrical signals from the device to have any contact with the external environment. The conductive layers transfer pressure signals to the signal acquisition circuit which is connected by a wire between the mat and the input port of the receiving computer. Other PSMs may use

optical sensors with no electrical signals in the PSM itself. The mat can be placed between bedding, within seat cushions, etc., and still capture variations in pressure signals.

PSMs are excellent candidates for use as pressure data acquisition devices used for non-invasive neonatal monitoring in the NICU. Their unobtrusive nature makes pressure sensor mats ideal candidates for use in clinical settings to monitor patient behaviour or vitals. Variations in pressure in the abdominal or thoracic regions can mimic patterns of respiration cycles [18]–[21]. Variations in pressure are also present when movement occurs on the mat; in research with adult participants, these variations were quantified with features such as changes in the total contact area, features derived from the COP, and average signal strength [22], [23].

Monitoring tools that employ pressure sensor technologies are available in the market based on clinical research for lay consumer use, but these technologies have not yet been adopted in clinical settings.

### 2.1.3 Machine Learning in Medical Applications

Machine learning as it pertains to medical applications combines statistics and computer science to develop algorithms that can be predictive, evaluative using detection or classification, or both. It is the process where advanced statistics-based computer algorithms process large amounts of data to learn patterns or associations within the data to ultimately come to some conclusion. When the algorithm is applied to new data, it may outperform clinical experts in terms of speed and/or accuracy. To the advantage of standardizing healthcare practices, these algorithms produce consistent responses from a single system even if trained on biased data, and are less likely to produce fatigue-related errors that could be made by clinical professionals [24].

In developing a generalizable algorithm, the algorithm undergoes a training process where it learns from a (training) subset of the full dataset. It is then tested on a separate (validation) partition to tune learned parameters. Lastly, the fully trained algorithm is tested with never-before-seen data to observe the generalizability of the algorithm. One can return to the tuning and training phase if performance is inadequate [24].

There are three strands of known machine learning: (*1) supervised, (2) unsupervised,* and *(3) reinforcement learning*:

(1) *Supervised learning* occurs when the machine learning algorithm is trained on a labelled dataset; each class of data learned by the algorithm is labelled by a human expert, and the algorithm must differentiate between the labelled classes. Supervised learning is ideal for classification, regression, or prediction problems.

(2) *Unsupervised learning* occurs when a machine learning algorithm is given unlabelled data with multiple features or variables and is required to cluster those data into groups based on similar features between groups. The clustering process is also effective in detecting anomalous data that cluster separately to the majority classes.

(3) *Reinforcement learning* most closely mimics the way human beings learn to achieve goals and favourable outcomes, beginning with trial and error to learn about associated outcomes and actions, and ultimately tuning parameters and methods towards attaining the most desirable results. The model is considered to be performing well when clinically favorable outcomes are achieved; whether ‘close to favourable’ outcome achievement is acceptable depends on the use of the model and the cost of not achieving perfect or ideal outcomes.

#### *2.1.3.1 Support Vector Machines (SVMs)*

The support vector machine (SVM) is a supervised learning algorithm that can be used for classification problems, as in the case of classifying *motion* events from *no_motion* events or classifying the direction of movement with COP variations in PSM data. SVMs try to find a hyperplane or a dividing threshold between two classes based on feature data. Sometimes the data is perfectly separable on a 2D linear plane, and that is usually the case when there are very clear distinguishing features between the two classes. Other times, the data is required to be transformed into higher and higher dimensions until a linearly separable plane can divide the two classes; this is called kernelling and is typically used for datasets where the two classes of interest seem to 'overlap' in the original feature space. The most common kernels are linear, polynomial, and radial basis function (rbf). The radial basis function is most used when the data is not perfectly separable and requires hyperplanes in higher dimensions. The limitation of SVMs is that they take a very long time to train when datasets are large.

#### *2.1.3.2 Decision Trees*

Decision trees use a supervised learning approach to create branched classification rules based on feature thresholds. Decision forests may also be used where multiple decision trees with different classification branching are combined and the dominant classification across all classifying trees is favoured as the outcome; they can also be used to identify the most important features in a training set by computing the out of bag error (OOB-error) which is used and explained in *Chapter 5*.

#### *2.1.3.3 K-Nearest Neighbours (KNNs)*

K-nearest neighbours (KNN) is a supervised machine learning algorithm that can be used to solve regression and classification problems. KNN is based on the idea of distance between a query point – that is a new instance – and existing labelled training points. Instead of Euclidean distance, Manhattan, Hamming, or Minkowski distance functions can also be used. KNN analyzes the labels of the *k* closest points to the query point, and the majority label is applied. The number of *k* neighbours is a hyperparameter; if *k* is too small it can be sensitive to noisy random points that exist within a cluster by chance and if *k* is too large it may include too much data from majority classes and lead to false classifications. Typically, *k* should be less than the square root of the number of all the samples in the dataset. A KNN classifier performs well on large datasets, but the selection of *k* may require multiple rounds of experimentation to improve results.

### 2.1.4 Evaluation of a Classification Algorithm

Evaluation metrics are used to assess the overall classification or prediction efficacy of an algorithm. The metric used to evaluate the performance of the classifier may vary based on the user requirements. For example, the clinical importance to a nurse of detecting *motion* events more often than *no_motion* events – false alarms may be gated but this risks the possibility of *no_motion* events being misclassified as a *motion* event, which poses a risk on patient safety. Would the NICU nurse prefer to compromise alertness if they are aware of the possibility of a critical event occurring with a still baby? Should the classifier be able to perform reasonably well in either situation, with a likelihood of slightly reduced performance in both? The following evaluation metrics quantify the algorithm's ability to

meet user requirements, and produce insights into the quality of algorithm performance and the quality of the data.

#### *2.1.4.1 Confusion Matrix*

The confusion matrix is a contingency table that tabulates actual class labels with predicted class labels and allows for the performance of the classifier to be evaluated based on several derivative metrics such as the true positive rate, false positive rate, precision, recall, accuracy, F1 score, and Matthew's Correlation Coefficient.

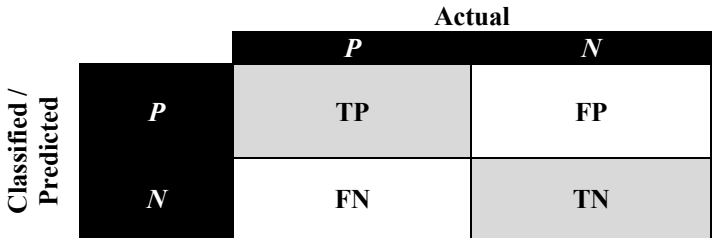



**Figure 2.** Structural overview of two-class confusion matrix.

The confusion matrix in *Figure 2* displays a two-class confusion matrix, however, the matrix can be expanded vertically and horizontally as more classes are added. The metrics that can be derived from the confusion matrix are discussed further below. A true positive (TP) is when the positive class is correctly classified, whereas a true negative is when a truly negative sample is correctly classified as negative. A false positive (FP) is misclassified negative sample as positive, while a false negative (FN) is a misclassified positive class as negative.

#### *2.1.4.2 Metrics Derived from the Confusion Matrix*

- *True Positive Rate (TPR), Recall, and Sensitivity:* the ratio of correctly classified positive classes divided by the sum of correctly and incorrectly classified positive

classes as negative class. This number ranges between 0 and 1, where an ideal ratio is closer to 1.

$$TPR = \frac{TP}{TP+FN} \quad (1)$$

- *False Positive Rate (FPR):* the ratio of incorrectly classified negative classes as positive divided by the sum of correctly and incorrectly classified negative classes. This number ranges between 0 and 1, where an ideal ratio is closer to 0.

$$FPR = \frac{FP}{TN+FP} \quad (2)$$

- *Precision:* the ratio of values predicted to belong to the positive class, that actually belong to the positive class, over the sum of the former and the total number of false positives. For example, if *motion* events are the positive class, out of all the events classified as *motion*, precision reports what proportion of those predictions are true motion events.

$$precision = \frac{TP}{TP+FP} \quad (3)$$

- *Specificity or True Negative Rate (TNR):* the ratio of correctly classified positive classes divided by the sum of correctly and incorrectly classified negative classes as positive class. This number ranges between 0 and 1, where an ideal ratio is closer to 1.

$$recall = \frac{TN}{TN+FP} \quad (4)$$

- *Accuracy:* the sum true positives and true negatives divided by the sum of all values in the confusion matrix. Accuracy can be reported as a ratio between 0 and 1 or a percentage between 0 and 100 percent. A limitation of using accuracy as an evaluation metric is that it is extremely sensitive to class imbalance. For this reason, the F1-Score

or Matthew's Correlation Coefficient (MCC) should also be used to support the evaluation of the classifier.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

- *F1-Score:* a reflection of the precision and recall combined, both of which are sensitive to class imbalance alone. This score ranges between -1 and 1.

$$F1\text{-}Score = 2 \times \left(\frac{(precision \times recall)}{precision+recall}\right) \quad (6)$$

- *Matthews Correlation Coefficient (MCC)*: also known as the *phi-coefficient* (φ), the MCC computes the correlation between the true and predicted values; the MCC does not have issues of asymmetricity or a sensitivity to class imbalance. It is best used with binary classification problems and its values range between -1 to 1, where values closer to 1 or -1 represent perfect positive or negative correlations, respectively.

$$MCC = \frac{(TP \times TN)-(FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (7)$$

#### *2.1.4.3 ROC and AUC*

The receiver operating characteristic (ROC) takes labelled data from the positive and negative classes and computes the TPR and FPR when the data from all classes are separated to various degrees by a specific decision or set of decision thresholds. The optimal threshold is the point on the classifier that pulls the ROC curve closest to the top-left of the plot, where TPR is as close as possible to 1 and FPR is as close as possible to 0. The area-under-the-curve (AUC) quantifies the performance of the classifier at all possible thresholds. The AUC ranges between 0 and 1, where a value of 0.50 means the classifier's performance is only as good as random. A sample ROC plot with an AUC of nearly 1 is shown in *Figure 3.*

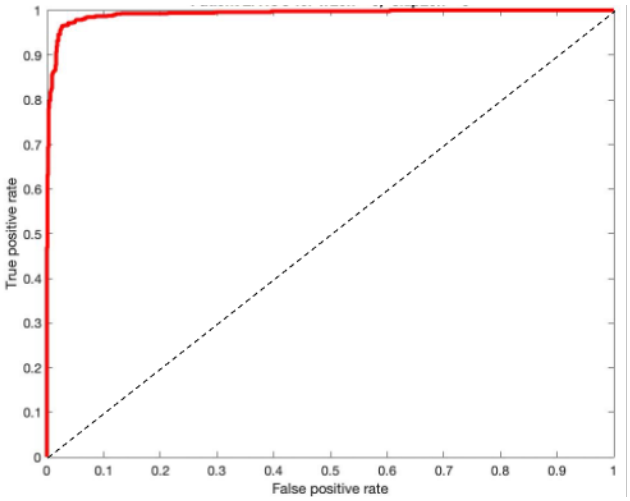



**Figure 3.** Example of an ROC curve. The AUC is the area under the red curve.

### 2.1.5 Class Imbalance Problems

Class imbalance occurs when one class has significantly more samples than another class or classes in a dataset. When class imbalance occurs, this can cause the accuracy of the classifier to appear high but only because it is correctly classifying the dominant class while incorrectly classifying the minority class. Class imbalance can produce a deceiving AUC, but when metrics like precision, recall, and MCC are calculated, the class imbalance becomes more evident. Ideally, a classifier should be trained with a balanced dataset so that it has a chance to learn all or both classes and create informed decision boundaries for classification.

Since the data in the present research come from a purely observational study, the monitoring data does not always contain an even mixture of event types. Depending on the research and the consequences of synthetic data creation, minority classes in datasets may be synthetically oversampled such that they more balanced, or at least do not have an unfair class imbalance with the majority. In this research two methods are used: *synthetic minority oversampling technique (SMOTE)* and *majority class under-sampling.*

#### *2.1.5.1 Synthetic Minority Oversampling Technique (SMOTE)*

SMOTE uses a KNN algorithm to create synthetic samples of the minority class. For each synthetic sample, SMOTE selects a random sample from the minority class, computes the *k* nearest neighbours within the minority class, then adds a new synthetic sample at a random point along the vector connecting the sample and one of its neighbours. With this method, a new set of features are created with values that are similar to other data within the class.

#### *2.1.5.2 Majority Undersampling*

Undersampling the majority class (class with the most samples of data) means taking a randomized subset of the majority class such that the sizes of the majority class match that of the minority class (class with the least samples of data) or come very close, as opposed to SMOTE which tries to increase the size of the minority class. This method creates a new dataset containing existing data and does not involve a synthesis of hypothetical data.

## 2.2 Literature Review

### 2.2.1 Motion Detection Systems in Medical Applications

Both contact and non-contact approaches exist for motion detection systems in medical applications, such as non-contact video and contact-based sensors like accelerometers.

A wide range of methods are used for motion detection using video data. Optical flow is a process that detects the presence of motion by looking at changes in perceived pixel intensity between subsequent frames. Similar to previous work in neonatal motion detection using segmentation methods using video data [25], Dosso *et al.* showed that the motion vector field is more dense in frames of *motion* events, and sparse during frames of

*no_motion* events [26]. The sum of the magnitudes in the vector field of each image during periods of *motion* and *no_motion* was calculated, and the median sum within a *motion* segment was derived to avoid outliers. When using the optical flow approach to determine thresholds denoting *motion* versus *no_motion*, 90% recall and 68% precision were achieved. They also tried using a supervised long-short-term memory model and achieved ~82% recall and ~31% precision, which suggests that optical flow is a more suitable video-based motion detection approach. Patients in low-light intensity conditions were excluded from the study, so this approach may not fare well in low lighting environments.

Previously, Kalitzin *et al.* used an image segmenting approach with optical flow to distinguish between ictal and inter-ictal motion in an adult population diagnosed with major motor seizures (MMS) [27]. Using labelled video data where ictal and non-ictal segments have been characterized and labelled, they observed changes in luminance and pixel intensity – which is the optical flow approach – to compute the motion velocity vector field across all frames to extract the rotation, translation, dilation, and rates of transformations. They also processed the data with a non-stationary wavelet technique and looked at the wavelet amplitude cross covariance and the spectral contrast features as a measure of the magnitude of an ictal event. This study was the first of its kind to provide a proof of concept that by using video data and optical flow methods, MMS and noMMS behaviour can be distinguished. These findings are extremely important as major epileptic events can lead to seizures, comas, brain damage, among other physiological complications. Interestingly, the overlap in the *motion* versus *no_motion* events was much higher with the population studied in this work, versus Dosso *et al.*'s work where the median sum of pixel intensities was used as a distinguishing or classifying feature [26].

Heinrich *et al.* employed a motion detection approach using near-infrared video data; they focused on the type of lighting, reduction of shadows, and improvement of texture [28]. They computed the local variance of intensity ratios per frame and found that areas that have a uniform texture are mistaken for shadows and stress the importance of textural clarity. They enhanced a variance algorithm to address this problem and included knowledge of shadows, spatiotemporal and motion information in their video motion detection algorithm. They created two models: a texture model (TM) that includes spatiotemporal features, and a motion compensated texture model (MCTM) that distinguishes shadow movement from subject movement. The TM uses the sum of squared difference surface (SSDS) to characterize textures; two textures within the same frame are considered unchanged if their SSDS values are less than the threshold value. While the TM can remove false positive detections where the textures of the corresponding regions remain unchanged, the MCTM eliminates false detections where the texture changes over time in an unpredictable manner. Using output from a luminance-variance based motion detector, the TM, and the MSTM in a cascaded Adaboost classifier, they were able to achieve a mean specificity of 96.5% in distinguishing between a movement and no movement event, and a mean sensitivity of 68.6% in their video motion detection model.

Optical stereo is a slightly different approach to optical flow where instead of focusing on the change in pixel intensity, this method looks at the distance that a point of interest has travelled between consecutive frames. In Henry *et al.*'s work on non-rigid motion detection for motion tracking of the head they addressed the problem of detecting head movement during MRI/CAT scans to correct for motion after the scan. Using head movement video data collected from an adult male volunteer, they computed the distances between triangulated 3D point clouds consisting of forehead landmarks between frames.

The difference in distances were computed and the median distance for the movement event was computed as represented of a rigid or non-rigid event, or a *no_motion* or *motion* event respectively. Using a PointNet deep learning classifier, rigid events were classified correctly 99.85% of the time, while non-rigid events were correctly classified 93.3% of the time.

As it pertains to image and video based motion detection, the final work discussed is 3D depth video for pressure ulcer prevention and motion analysis conducted by Chang *et al.* [29]. Many studies that explore in-bed motion detection for ulcer prevention or fall detection, also include static pose classification. However, this is outside the scope of this thesis so it will not be discussed. For their motion analysis work, Chang *et al.* use 3D depth sensor data collected from real adult patients. They first applied a foreground-background analysis on depth sequences in each frame where each depth pixel represents the distance from the sensor. Then, they created a probabilistic model to separate stationary pixels from pixels with significant changes; in other words, they created a feature to distinguish between *motion* and *no_motion* events. They applied morphological filtering for background subtraction then focused the video data on bounds of the bed to avoid environmental noise from skewing the results. Lastly, they aggregated signals over a time window to eliminate minutia of noise following a sequence of motion history images. The crux of this approach was based on gross motion detection versus the lack thereof. The process of looking at motion history images has a similar concept to optical flow. In the application of ulcer prevention, the algorithm developed in this work would be able to support the development of a tracking tool that informs nurses of when and if a patient was repositioned, and whether the patient is moving in the bed in discomfort.

Finally, this section explores motion detection using contact sensors. Hu *et al.* used radio frequency identification tags to continuously monitor patients during sleep for specific motion-related problems like restless leg syndrome and periodic limb movement disorders [30]. Based on the received signal strength indicator (RSSI), classification was done with discrete events such as arms moving, legs moving, or arms and legs moving, with data collected from participants who would simulate these movements in a bed; however, data were not collected from real patients who experience a sleep disorder. After denoising signals in each event sample, they computed the differences between expected interval of the RSSI and the actual value of the RSSI; if this difference was greater than the expected variation between RSSIs in a *no_motion* state, the event was *motion*. Using K-means clustering, they were able to detect specific movement events and then determine the underlying sleep disease.

Accelerometers are sensors whose primary use is to measure motion along reference axes. Advantages of accelerometers include that they offer opportunities for continuous monitoring, are low cost, easy to use, and easy to set-up as part of wearable monitoring technologies such as smartwatches. Accelerometers are sensitive to changes in acceleration in activities such as jumping, walking, or falling. In their work developing a motion classification algorithm to detect falls, Huang *et al.* describe motion of the human body as a sequence of body postures, where each posture is described with a set of joints that each have their own angles and positions in a 3D space [31]. They attached accelerometric sensors to the elbows, wrists, knees, and ankles to an adult male participant and obtained 69 motion clips which were 380 frames each which included footage of them walking, falling, grabbing, sleeping, sitting on a chair, and sitting on the floor. Since an event is a sequence of body postures and each posture consists of a set of measured rotation angles

and spatial positions within a set of body joints, they extracted sequences of joint angle feature vectors from the motion events by concatenating rotation angles of the relevant joints of a posture in the motion sequence. They also extracted a joint feature vector, which is a concatenation of positions of the joints in space as opposed to their angles of rotation. They used these rotation and position feature vectors in an SVM classifier and of the four kernel parameters they experimented with, the radial basis function (RBF) produced the highest classification accuracy of ~80% using either joint angle feature vectors or joint position feature vectors to classify motion types.

Lastly, accelerometers have also been used for tremor detection in patients suffering from Parkinson's disease. Zhang *et al.* had six patients wear wrist accelerometers to record movement data [32]. They extracted features from the data using four distinct approaches: a baseline set of features derived from Fisher *et al.* [33], Mel frequency cepstral coefficients (MFCCs), MFCCs on tremor activity spectra, features derived from a convolutional neural network (CNN) trained on raw signals and features derived from a CNN trained on separated tremor and activity spectra. Ultimately, they found that using CNN trained on the separated tremor and activity spectra to learn features with a multi-layer perceptron layer for classification performed better than random forest for classification, with an accuracy of 0.89 and an FPR of 0.30 at a TPR of 0.90. Relying on the CNN to extract features produced better results than using engineered features extracted directly from the accelerometer signals. Nonetheless, this research demonstrates that motion detection research is not limited to geriatric or neonatal populations, but also adults with neuromuscular disorders.

### 2.2.2 Motion Detection from PSM

The primary advantage of using a PSM is that the PSM is non-invasive and unobtrusive requiring no direct interface between a subject's skin and the sensor. Furthermore, they are less intrusive than camera systems. As it pertains to in-bed patient monitoring, video-based motion detection systems – 2D or 3D – suffer from the occlusions, changes in environmental lighting, or blankets covering a patient, [27], [28], [34]. Moreover, radiofrequency monitoring is limited by the number of sensors on the patient's blanket or on their body; in the case of sensors placed on clothes, otherwise its primary disadvantages are sensor movement and noise caused by the material and pattern of the blankets [30]. Lastly, accelerometer performance – 2D or 3D – is limited by variations in human body types, patient types, and set-up time as clinicians must find the right place to place the accelerometer to achieve the best readings of subtle movements.

Neonatal motion detection with the PSM was explored by Joshi *et al.* using a 59 cm by 29 cm electromechanical film (EMFi) sensor that was unobtrusively and non-invasively placed between the mattress and the bedding of ten infants [23]. This thin film sensor included a midsection of polypropylene layers separated by air voids. The air voids act as electrical dipoles where the film is sensitive to forces perpendicular to its surface. The EMFi sensor produces a signal by considering all sensors in the mat. The signal was used to extract a ballistographic signal (BSG), which was used to derive pressure-sensitive features, such as the signal instability index (SII). As the name suggests, signal instability measures the consistency and variability of a signal over time. They found that the SII is a strong feature to distinguish between various movements when taken from BSG signals, producing an AUC of 0.90. In this study, all ten patients weighed an average of 1500g suggesting that the signals attained from the EMFi are sensitive to very light infants and

can produce distinguishable signal data across *motion* and *no_motion*. It is interesting to note that the ten infants in this study were in a controlled private room setting during data collection, which means that the data was free from environmental noise, which is unlike a traditional NICU setting. Additionally, the bed-side observation annotations did not include upper and lower body movement annotations; conclusions cannot be made about the ability of an algorithm derived from BSG signals using SII as a primary feature to detect upper versus lower body motion in a limb specific way. Joshi *et al.'s* claims that BSG and SII are good for limb-specific motion detection requires more experimentation by collecting data that includes episodes of movement that are exclusively upper or lower body motion, and perhaps even more specifically, labelled by the laterality of each limb. Lastly, the infants in this study were turned to their left hip, right hip, supine, prone positions for the duration of the recording. Infants in the NICU are typically in supine position as they generally lack the muscle mass and energy to turn and hold themselves in other postures. Without further clarification, it is reasonable to assume that the infants may have been turned into these positions for data collection purposes. For continuous patient monitoring, it is important at its basis for the technology to be able to detect motion from the lack thereof regardless of pose. Nevertheless, Joshi and colleagues were able to successfully use a pressure sensor array that amalgamates signal data to extract features and implement an algorithm that was able to classify various types of gross motion events in a neonatal observational study, to produce an AUC of 0.90; the presence or absence of class imbalance was not reported.

Furthermore, signals communicated from multiple sensors on a PSM can also identify the directionality of movement, and static pressure points before significant movements. While the majority of research on mobility and transitions are focused on acceleration

signals obtained from wearable sensors [31], [32], [34], [35], Foubert *et al.* took advantage of these properties by using three 66 cm x 71 cm PSM with multi-sensor outputs to detect transitions between laying and sitting positions which can be useful for patient-mobility and fall prevention monitoring for the elderly in extended-care homes or geriatric inpatient units at hospitals [36]. They combined a postural recognition algorithm with a movement detection algorithm, both SVMs. The lateral (LatCP) and longitudinal (LonCP) COP values were used to compute the Euclidean distance travelled by the COP across each frame of *motion* and *no_motion*, using a lowpass averaging filter to eliminate noise and outliers. To differentiate between laying and sitting, the top-performing features were the weighted number of active sensors and weighted sum of sensor values. Signal values and counts near the head and legs were given greater weights than those near the abdomen. For the laying vs. sitting algorithm, the maximum classification accuracy of the SVM with a single feature was 94%, and greater when adding more features since five of eight tested features had at least 80% classification accuracy individually. In contrast, accelerometer-based studies have reported requiring at least 2 axes to achieve greater than 92% recognition accuracy [37]–[39].

Following Foubert *et al.'s* work, Bennett *et al.* added geometric and kinematic features derived from the COP of PSM signals to the feature-set to detect in-bed movement patterns, namely stable versus unstable sit-to-stand movements [22]. They analyzed discrete events of stable and unstable events, then classified stability and instability with time-series monitoring data from a single geriatric subject. The geometric features they extracted included: number of acute angles (NAA), the number of spatial coordinates (NSC), and polygon area, and the kinematic features were velocity and acceleration. Ultimately, the NSC and NAA were the top performing features individually in distinguishing between

state transfers, with a classification accuracy of 98% with an SVM. The kinematic features were explored later and not tested for classifier performance; however, they reported that plotting velocity and acceleration of the COP show distinguishable patterns between sit and stand motion events. These findings are applicable to this thesis and may be relevant to future research.

Ren *et al.* took an image-based approach for observing postural control behaviour in bed-exit data prior to and post-fall using the PSM and the COP [14]. They observed the magnitude and frequency of movement in the *y*-direction by *x*-values of the COP. It was found that if a patient had a fall, slower and more frequent attempts to get off the bed would occur in the days following the accident. Movement progression plots with colour-coded visualizations can help caregivers to identify that a fall has occurred and send help; this is particularly important for patients with dementia or Alzheimer's who may forget to report their accident.

Outside of motion-detection for geriatric and hospital-based applications, researchers have also studied movement during sleep to improve the technologies used in sleep studies. Movement during sleep can be informative in terms of the incidence of restless leg syndrome, it can also disrupt the collection of respiratory signals. In their PSM-based research on real-time monitoring of respiration and pulse rate during sleep, Zhu *et al.* filtered out motion artifacts from respiration signals by identifying wavelets that are four times greater than the standard deviation of previous signals and removed the following 2.5 seconds of raw pressure data [40]. Moreover, Pino *et al.* in their study on contact pressure monitoring for sleep studies using the PSM, deliberately looked for movement in respiration signals by observing large amplitude changes during two second windows in at least three sensel signals [41]. These methods could also be used to identify those time

stamps where motion is suspected, extract other pressure-based signals from that time frame and determine whether the movement was from the lower or upper body.

In their study measuring heartbeat, respiration, and movements using Smart Bed technologies, Sivanatham and Adami *et al.* described a method for classifications of movement in bed, using an array of load cell sensors, as opposed to a PSM, within the mattress of the bed [5], [21], [42]. Sivanatham focused on hand movement, leg movement, and mixed hand and leg movements; additionally, they extracted the centre of mass (COM) as opposed to the COP. The COM is the point that the body's mass is balanced around; it varies as the body moves and sways, also known as the centre of gravity, where the weight of the body acts downwards. The COP is the sum of all vectors of pressure acting on the body, in this case on the PSM; it is the point where the total force due to pressure is equal on both sides of the body. Adami *et al.* suggested that the variance of the COP's trajectory in the y-direction is a good differentiator between upper versus lower body movements. Sivanatham plotted changes in the *y-coordinate* of the COM for a patient laying on the bed across different movements. In agreement with Adami *et al.'s* findings, Sivanatham found that leg movements and hand movements could be successfully differentiated with a seven feature SVM model, but that leg movements could not always be differentiated from gross movements as the impact of the hand movement in a gross motion event contributes very little to the overall COP. Overall, in distinguishing between upper and lower body motion, Sivanatham (2016) showed 95% classification accuracy, and *Adami et al.* showed an overall 84% classification accuracy.

Motion detection applications using the PSM or an array of load sensors have been used across applications involving sleep studies, motion gating in respiration estimation, elementary stages of neonatal motion detection, fall detection, and in-bed movement

transitions. While many of these studies have been employed with young and old adults, as opposed to neonates. Each study has provided insights into baseline motion detection and upper and lower body motion detection using the PSM for a neonatal motion detection application and have suggested limitations for consideration such as the indistinguishability between mixed motion events and lower body motion.

### 2.2.3 Other Applications of PSM for Patient Monitoring

Patient monitoring research using the PSM has predominantly focused on respiration rate estimation, heart rate estimation and detection, sleep studies and apnea detection, as well as critical geriatric patient monitoring needs such as ulcer prevention, sit-to-stand detection for fall prevention, and stability.

Respiration rate estimation technologies come in contact, non-contact, and indirect contact forms; PSM technologies are considered indirect contact while non-contact technologies use cameras or infrared imaging systems detecting temperature changes in exhaled air out of patients [43]. Liu *et al.* in their study of respiratory monitoring with adult subjects used a pressure sensitive bedsheet which was 2.5 m x 1.25 m with 64 x 28 pressure sensors, and a torso localization approach. The motivation of this study stems from the understanding that there are a lack of reliable and unobtrusive respiration rate measurement systems for clinical or home use [44]. In every frame, they localize the torso before extracting a respiratory signal. Extracting the signal post-torso localization reduces the interferences caused by extremities during movement, suggesting that the signals in the head, legs, and arms change dramatically during periods of movement and if included within the frame of respiration detection, could throw off the overall signal. Overall, they found that motion caused by lower extremities had a greater impact on signal extraction;

this is likely because movement of the legs causes a shift in pressure along the y-axis. Since the signals for respiration were dependent on vertical directional coordinates, the impact of lower body movement would be more pronounced regardless of torso localization. They also visualized respiration motion; with inhalation, there is a shift in pressure upwards towards the shoulders and with exhalation, the pressure re-centers in the torso region. The geometric delay between the pressure changes was also proposed as a method for respiratory signal extraction with the PSM.

Bekele *et al.* conducted a similar study to Liu *et al.* and spatially averaged pressure signals obtained from a PSM using signals from the thorax as opposed to the entire torso. Additionally, this study was the first to use PSM data collected from neonates in the NICU to non-invasively estimate respiration rate [18]. Bekele *et al.* adopted and modified a time-frequency respiration estimation approach outlined by Zhu *et al.* who used pressure signal data obtained from sensors placed underneath a pillow to estimate respiration [40]. Zhu *et al.* looked at the dynamic response to the weight fluctuation of the head due to breathing movements, and their reference data for validation came from a nasal thermistor.

In Bekele *et al.*'s work, the signals were normalized by taking the average signal of the past 60 seconds since respiration signals can be reproduced at lower sampling rates. It should be noted that at the thorax site there are multiple sensors, so a signal selection process was also done to identify the sensor that produces the closest respiration signal to the gold standard. The movement artifacts were handled by using a high pass filter over the normalized signal, whereas Zhu *et al.* suppressed random noise with a soft-threshold method. Ultimately, using a multi-level Daubechies Wavelet Transform (DWT) Bekele *et al.* found that the Level 2 components of the DWT could be used to recompose the respiration signal, whereas Zhu *et al.* found that the fourth and fifth components work best.

These variations may be due to differences in the location of the sensors and the quality of the signals obtained from the PSMs, as well as the fact that the subjects were adults and not neonates. In line with neonatal respiration rate detection, Koch *et al.* created a tiny stretchable sensor array to be adhered to an infant's thorax, which could be considered a micro-PSM [19]. Depending on the position of the sensor array, a signal selection process selects the best sensor while the thorax presses up against the pressure sensing array to find the sensor that produces the strongest respiration signal, albeit inverted.

PSMs have also been used for estimation of heart rate and respiration rate. Sivanatham *et al.* used load sensors to record BCG signals [21]. The BCG signals include respiration signals, heart rate signals, DC power line noise, and movement artifacts. Cancelling the DC components, applying a first stage band pass, then filtering on individual load cell data, the respiration signal was removed, and the remaining signal underwent second stage bandpass filtering. Comparing this signal to the gold standard monitoring data, the heartrate signal achieved over 98% accuracy on heartbeat detection. After a smoothing and normalizing step, the respiration rate was estimated by looking at inter-peak durations in the signal. To conclude, they were able to achieve 99% respiration rate estimation accuracy when comparing to gold standard monitor data, using ballistographic signals obtained from a pressure sensor array.

PSMs have also been used in sleep monitoring and apnea detection. The detection of sleep apnea is critical in the prevention of adult diseases like obesity, arrhythmias, and coronary artery disease. Sleep apnea also has a relationship with type II diabetes, which can be caught early on if a sleep study is conducted. Apnea occurs when respiration ceases for a period of time before returning, bradypnea occurs when there is low and shallow respiration, while tachypnea occurs when there is high respiration. Townsend *et al.* studied

the use of PSMs and the effect of windowing on central apnea detection [45]. Central apnea occurs frequently in underdeveloped infants due to their premature nervous systems but the most reported cases have been in older adults where their brainstem stops sending signals to breathe [46]. Using pressure data, they found that the start and end of apneic events produce very high frequency pressure signals, illustrated by sharp peaks. Similarly, Pino *et al.* found that with respiration data extracted from BSG signals using PSM data, an apneic event was identified within a signal if the decrease in signal variance was 10% or less of its normal value [41]. Moreover, Pino *et al.* also found that detecting apnea from the respiration signal of restless sleepers with a lot of movement in their signal was more difficult and inaccurate. Returning to the work of Townsend *et al.*, they used a sliding window approach to determine a window of signal data that would most effectively identify anomalous peaks between normal signals and correctly identify apnea. In their initial windowed classification, given each apnea and no-apnea event had a label of 1 or 0 respectively, they averaged and rounded up or down to the nearest integer to determine whether the window was apneic or not. However, it is unclear 50/50 splits of mixed apneic and non-apneic seconds of signal data within even-second were labelled – it is unclear if the window would be classified as apneic, non-apneic, or neither. Ultimately, it was found that, as the window size increased, MCC also increased, but plateaued after a certain point. Similarly, TPR and FPR behaved the same way with TPR increasing and FPR decreasing before a plateau. When a moving average was added to the sliding window approach for classification, the TPR increased with a slight FPR increase. This research demonstrated a method to improve sample by sample apnea detection using a windowing and moving average approach.

PSMs have also been used to study and prevent pressure ulcers. Bennett *et al.* conducted a study to monitor the relief of pressure points using the PSM [47]. They placed three PSMs under a hospital bed mattress: under the shoulder blade region, the hips, and the feet. The sensor mats extended laterally across each side of the bed for all three mats. They looked at six global pressure points on the body to see which points would be lifted off the bed during three different types of movements. The global points were at the shoulder blades, the hips, and the feet; they learned that different movements had different fluctuations of pressure. For example, movement to the right meant decreased pressure on the left. The data was zeroed, a baseline was determined, pressure differences from the baseline were calculated and the time and place of instance during which a valley on one side and a peak on another coincided on opposite sides of the bed were identified. The algorithm was successful at detecting pressure relief at the hips and the feet, but not at the shoulder blades; this means that the identification of where and when to relieve pressure is possible with this algorithm, with modifications to the detection of pressure points at the shoulder blade.

The PSM can form part of a mechatronics approach to ulcer prevention [48]. Pressure sensors are distributed across a mattress, and if a sensor's pressure signal exceeds its set threshold, the dynamic ulcer support bed pushes redistribution of air across the mattress to recalibrate thresholds and reduce pressure

Lastly, research using the PSM for medical monitoring has also explored the assessment of standing stability. Taylor *et al.* used a 48 cm x 48 cm sensor array in the form of a tile [49]. Light *et al.* used an insole-based pressure sensor to measure similar features and argue that insole based pressure sensors are better than force plates in terms of the concurrent validity of the COP [50]. Taylor *et al.* observed the movement of the COP and computed the lean ratio, COP value, total area of sway, and the movement velocity.

The movement velocity was computed by taking the difference between consecutive COPs divided by the time between samples and then multiplied by the distance between the COPs. The velocity calculations helped to determine that the velocity of weight shifting during a sway, or a lean is faster with young people than older people.

The PSM has been shown to be suitable in multiple patient monitoring applications outside of motion detection, with COP and frequency-based features being some of the most successful variables used in various movement detection and classification models.

### 2.2.4 Conclusions

The work reviewed in this chapter makes it clear that motion detection in clinical applications is extremely important as a preventative tool for many clinical risks such as falls, apneas, seizures, and ulcers to name a few. The key gaps in many of the motion detection studies explored is that they involve direct contact or limit patient motion in some way, such as waist belt accelerometers and joint and wrist sensors. The two primary methods of non-invasive motion detection methods that have been researched with the neonatal population are accelerometry and computer vision; however, while one faces the challenge of direct contact and invasiveness, the latter faces the issue of inaccurate motion detection due to visual occlusion.

A common artifact that patient monitoring systems strive to filter out are movement artifacts; however, despite these efforts, many studies blame poor performance of their monitoring algorithms on the magnitude or duration of movement artifacts within their monitoring data. This thesis contributes to a larger monitoring system that also estimates respiration and heart rate using pressure signals and video data. In estimating respiration rate, we have also found that motion artifacts can skew the performance of the system.

Integrating a motion detection system within a patient monitoring system can gate false alarms, reduce false positives, and even be integrated into multi-modal monitoring systems that set markers to disregard future alarms based on multiple past false alarm flags.

# Chapter 3

## Experimental Set-Up & Data Collection and Preparation

This section reviews the patients in our study as well as the inclusion and exclusion criteria for selection, the experimental set-up at the time of data collection and the various technologies involved, the event classification protocol, and a description of the data and data preparation methods.

## 3.1 Patients

### 3.1.1 Children's Hospital of Eastern Ontario (CHEO) and NICU

The study was conducted in the neonatal intensive care unit (NICU) at the Children's Hospital of Eastern Ontario (CHEO) with the supervision of neonatal care teams and was approved by the research ethics boards at CHEO, the Ottawa Hospital, and Carleton University. CHEO is paediatric hospital and research institute located in Ottawa, Ontario, Canada.

The subjects in this study were admitted patients in the neonatal intensive care unit (NICU). The NICU at CHEO cares for some of the most fragile and critically ill patients in the hospital. Over 400 infants are admitted to CHEO's NICU each year and are most often premature, neurophysiologically underdeveloped, susceptible to ictal events, or biologically limited in another way. Such patients require continuous monitoring of their vitals including heart rate, respiration rate, and oxygen saturation ($SpO_2$) [51]. However, methods for patient monitoring are often intrusive involving adhesives, wires, and sensors

[52]. Adhesives can harm thin neonatal skin; wire electrodes can pose a choking hazard, but also disrupt patient movement and inhibit parental interactions with the patient [53].

As will be described in *Section 3.1.2*, the inclusion criteria for the NICU study specify the weight of the infant at the time of observation and bed type strata. When a patient was admitted to the NICU and deemed stable for study by the physician and care team, and if the patient matched the inclusion criteria, the guardian(s) was/were invited to participate in the study by the study coordinator employed by CHEO. The process of nurses allied with our study evaluating the patient for inclusion criteria, attaining guardian consent, informing our team of a new enrollment, and our team arriving on site to collect data was a process that took 24 – 48 hours.

### 3.1.2 Patient Inclusion and Exclusion Criteria

There are twenty level III beds in the NICU, and there are three bed types: crib, overhead warmer, and incubator. Infants in the NICU at the time of observation were classified as having low ($w < 1500$g), medium ($1500\text{g} \leq w < 2500$g), or high ($w \geq 2500$g) weight.

**Table 1.** Patient counts for nine strata of weight classes and bed types

| Bed Type | Weight Class | Count | | |
|---|---|---|---|---|
| | | Male | Female | Total |
| Overhead Warmer | *<1500g* | 1 | 2 | 3 |
| | *1500g ≤ w ≤ 2500g* | 4 | 1 | 5 |
| | *≥ 2500g* | 2 | 3 | 5 |
| Crib | *<1500g* | 0 | 0 | 0 |
| | *1500g ≤ w ≤ 2500g* | 2 | 3 | 5 |
| | *≥ 2500g* | 4 | 1 | 5 |
| Incubator | *<1500g* | 3 | 1 | 4 |
| | *1500g ≤ w ≤ 2500g* | 4 | 2 | 6 |
| | *≥ 2500g* | 0 | 0 | 0 |
| **Total** | | **20** | **13** | **33** |

Given we had three bed types and three weight classes of patients, we planned to collect data from 35 patients – five patients of each weight and bed type combination, except for *low* weight infants in a crib or *high* weight infants in the incubator since neither of these combinations are typically seen in the NICU. Based on the availability of patients in the NICU who met our inclusion criteria over a two-year data collection window, we collected data for 33 patients. *Table 1* presents a matrix of patients in each weight class, bed type, and sex. See *Appendix A, Table 21,* for a breakdown of individual patient-characteristics.

### 3.1.3 Selection Rationale & Summary of Study-Specific Patients

*Equation 8* shows the basic equation for pressure. Since force is the numerator and the sensing area of a pressure sensel is constant, as the force increases, the pressure increases. Force in the case of an infant laying on a mat is mass in kilograms multiplied by acceleration due to gravity. In theory, infants with a larger mass but same overall size spread themselves out over more sensels than lower mass patients as they are larger which can also contribute to larger variations in sensel pressure values with movement. However, it is important to note that the distribution of pressure is uneven; there are hot spots on the mat where the skeleton is closer to the PSM, pressure points where the skeleton is not covered by soft tissue such as the hips. There are also areas on the mat where there is no measurable contact pressure, in the case of the arch of the back.

$$pressure = \frac{force\ (N)}{area\ (cm^2)} \quad \textbf{( 8 )}$$

For the purposes of this study, it was hypothesized that clearer distinctions between *motion* and the *no_motion* would be achieved in middle and high weight patients, as their applied contact pressure would be greater, providing a more clear signal. Three patients in

the *medium* weight class, and four patients in the *high* weight class were included in the training and test datasets.

*Table 2* summarizes the subset of patients whose pressure signals were used to develop and evaluate the motion detection algorithms described in *Chapter 4* and *Chapter 5*.

**Table 2.** Summary of weights and bed-types for study-specific patients

| Patient ID | Sex | Bed Type | Weight Class (*g*) | Actual Weight *(g)* |
|---|---|---|---|---|
| 2 | M | Crib | $\geq 2500$ | 3620 |
| 10 | F | Crib | $\geq 2500$ | 3530 |
| 13 | M | Crib | $\geq 2500$ | 4435 |
| 6 | M | OVW | $\geq 2500$ | 3504 |
| 11 | F | Crib | $1500 \leq w \leq 2500$ | 2415 |
| 33 | F | Incubator | $1500 \leq w \leq 2500$ | 1825 |
| 36 | M | Incubator | $1500 \leq w \leq 2500$ | 1605 |

## 3.2 Experimental Set-Up and Data Collection

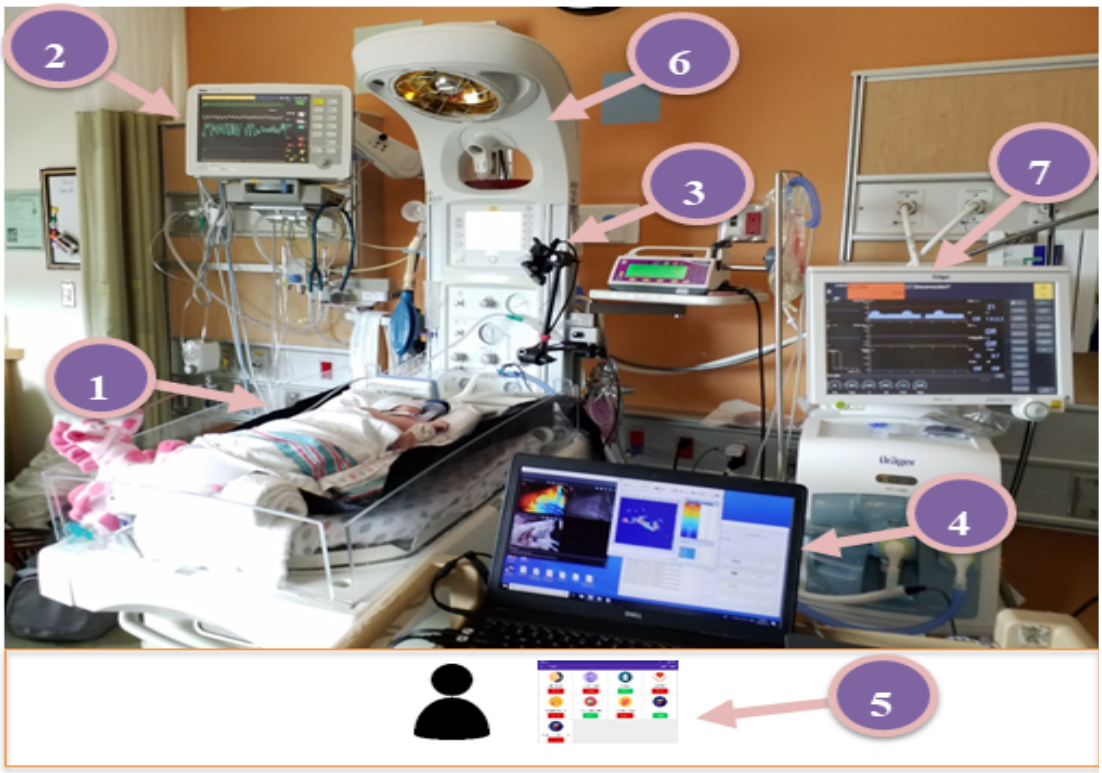



**Figure 4**. Experimental set-up: (1) LX100:100.100.05 pressure sensitive mat placed between the bedding and the infant. (2) Ventilator. (3) Overhead Intel RealSense SR300 camera to capture RGB-D video data. (4) Laptop to observe PSM and video data collection. (5) Temporal Event Annotation App (TEA) and bedside observers. (6) Overhead warmer. (7) Patient monitor with $SpO_2$ probe (Infinity Delta, Draeger Medical Systems, Inc. Telford, PA, USA). © Copyright 2018, Amente Bekele

### 3.2.1 Pressure Sensor Mat and XSensor Pro

The PSM used in this research is the LX100:100.100.05 (XSensor Technology Corp, Calgary) with a sensing area of 50.8 cm x 50.8 cm and 10,000 sensing points to capture ballistographic (BSG) signals ranging from 0.0 $N/cm^2$ to 3.41 $N/cm^2$ at 18 frames per second (fps) (Figure 5a). Given its size and range, the mat is used for low applied pressure

applications. The PSM connected to an X3 Pro Sensor Pack that fed into an X3 Pro Electronic Platform that was connected by USB to a laptop that was running the XSensor X3 Pro V8 software as shown in *Figure 4* at points 1, 4, 5, and 7; the XSensor X3 Pro software was used to record and manage PSM data.

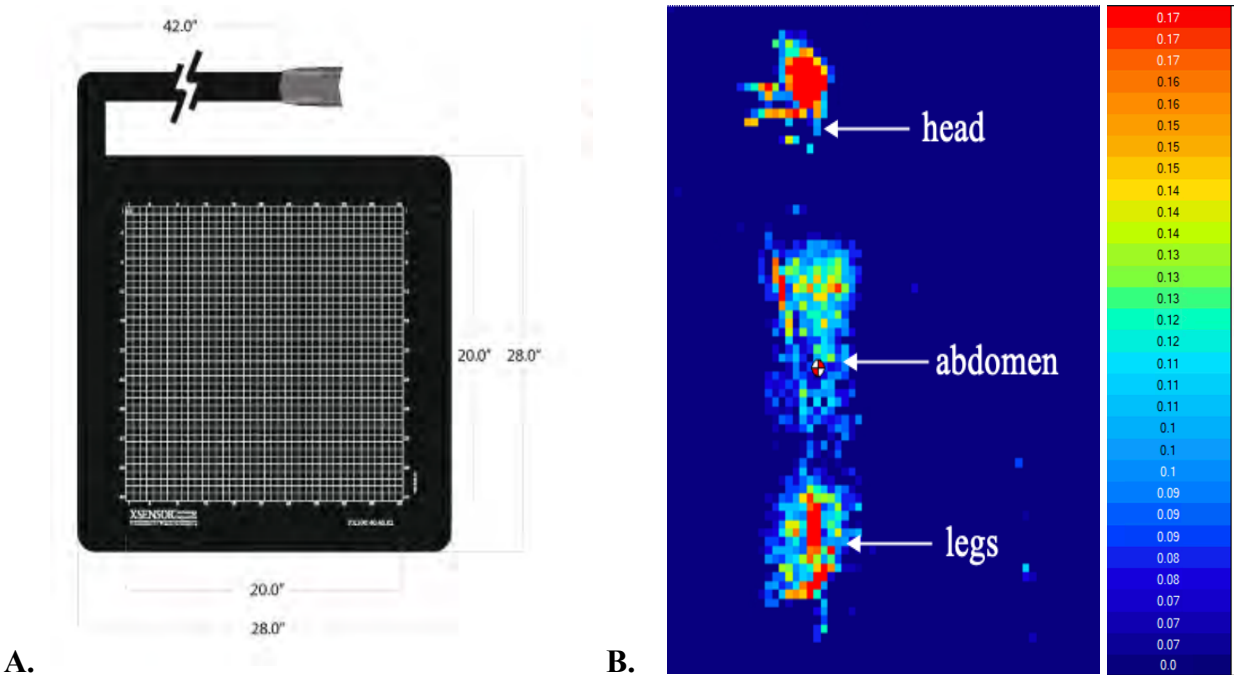



**Figure 5.** (A) LX100:100.100.05 pressure sensitive mat. (B) Single frame of pressure signal data visualized using XSensor X3 Pro V8 software for a medium weight patient in crib. Red-yellow tones depicts high pressure, while green-blue tones represent lower pressures in $N/cm^2$ as shown by heat scale on right. The red and white circle in the centre of the abdomen shows the centre of pressure (COP).

With the help of a nurse, patient and blankets were removed from the bed and the PSM was placed flat under bed sheets as shown in *Figure 4* Point 1. The sheets then covered the PSM and the patient was placed back on the bed. For some patients, the bedding was rolled on either side of the patient to keep the patient in position, and blankets were draped over

them. The PSM was then connected to the laptop to record and visualize all changes in pressure data. Between patient recordings, the PSM was sanitized and stored at CHEO.

The signal files for each patient were individually viewed with the XSensor X3 Pro V8 software to visualize heat maps on a frame-by-frame basis displaying pressure distributions over time to the millisecond (*Figure 5b*). This software was used to extract the COP at any given frame, visualize paths travelled by the COP, and view other features like average pressure contact area.

### 3.2.2 RGB-D Video

Video data was used to determine gold standard *motion* and *no_motion* annotations for this work; the annotations also noted patient monitor alarms; whether the alarms are true or false are verified by the patient's care staff. Lastly, video data was also used in refining annotation time stamps in the PSM as will be described in 3.3.

RGB-D Intel RealSense SR300 video data was collected at 15 frames-per-second (fps) for each patient for the entire data collection period. The camera was placed above the patient using a clamp attached on or near the bed, and articulated arm that would allow the camera to be positioned over an incubator, overhead warmer, or crib. For OHW beds, a custom clamp was fabricated to attach an articulated arm holding the camera to an integral side rail, as shown in *Figure 4* at points 3 and 6. For the crib, a "super clamp" was used to attach to the top rail of the crib along with the articulated arm. Lastly, for the incubator a durable and easy-to-clean silicon skirt was created to attach to the plexiglass to hold the camera in position overhead of the patient.

The duration of each patient's recording session varied between four to six hours, inclusive of periods where the patient was removed from the mat for feeding or family

time. The complete recordings include events such as gross motion, diaper changes, bedding, and swaddle readjustments.

### 3.2.3 Temporal Event Annotation (TEA) Application

In addition to the video and PSM data, two bedside annotators used an intuitive and configurable user interface called the Temporal Event Annotation (TEA) App to annotate in real-time a timeline of clinical events, event categories, and study attributes for a patient [54]. This application was built by the NICU project team at Carleton University to record movement and intervention-based events; it was used with an Android tablet at the time of observation. The annotations were saved as JSON files and served as a baseline for the process of validating timestamps for events of interest as they occurred on the PSM and in video recordings.

## 3.3 Classification Protocol for Event Annotation & Validation

Prior to importing the data into MATLAB to prepare for analysis, event annotations were reviewed, enhanced, and time-adjusted using video data using VideoLAN Client (VLC) as the media player.

It was observed during the event annotation process that the bedside annotations would often be delayed by a few seconds relative to the actual motion event. PSM and video recordings always began at different times during the experiment due to the set-up procedure, for this reason the video and PSM were not in sync from the beginning of their respective recordings. Additionally, since individual bedside annotators varied in style and accuracy, it was questioned whether each motion event was classified independently between pauses or if the motion event was continuously logged despite pauses.

As a post-processing quality assurance step, the start and end times for existing motion event annotations were reviewed for up to 2 hours of data per patient. The annotations and time stamps were adjusted by matching registration points in the PSM and video data, and the revised event annotations followed a consistent set of criteria using event inclusion and exclusion criteria described in *Table 3*.

**Table 3.** Description of event types and post-collection validation protocol

| Event Type | Description | Inclusions or Exclusions | |
|---|---|---|---|
| *Motion* | Motion (M): Any category of motion – upper, lower, gross, clinical intervention<br><br>Gross Motion (GM): No dominant limb movement; both arms, both legs, and head in sporadic motion | • **Inclusion:** Pauses lasting ≤ 2 seconds between motion events were included in the motion event<br>• **Exclusion:** Gross twitches lasting between 1-2 seconds not considered as a motion event | |
| *Upper and Lower Body Motion* | Specific limb movements – differentiate between arms, legs, and head movement | Lower Body Motion | • Legs: During observation, both legs raise or move under a blanket; annotations of specific leg movements come from 'gross' annotation, when legs move independently from gross annotation, we do not specify which leg |
| | | Upper Body Motion | • Head: Head moving up and down or side to side<br>• Arms: Left and right arms movement in any direction |
| *No_motion* | • No movement of limbs<br>• Typically, a patient is resting or sleeping. | • **Inclusive of:** Gross twitches that last between 1 – 2 seconds; these could be representative of brief sleep apnea, but this is not clinically concerning unless it lasts 15 seconds or more [55]<br>• **Inclusive of:** Hand wiggle, wrist flexation, brief head nods lasting between 1 – 3 seconds, facial grimaces | |

Throughout the recording of PSM and video data, a pen was pressed at various points on the PSM – this resulted in clustered high-pressure signals that could be identified on the

PSM. Matching the point where this occurred on the PSM and in the video allowed for an estimation of corresponding time stamps on the PSM associated with video observation data.

For each patient in this research, an Excel workbook was created with multiple tabs for each event type, with two columns each. The first column represented the start time to the nearest second and the second column represented the end time to the nearest second. Each row represented one 'episode' of that event type. These time stamps were then used in the data import and preparation steps conducted using MATLAB R2018a.

## 3.4 Description of Data & Data Preparation

For each patient, the PSM signal frames were exported into a .csv using XSensor Pro and then converted to multiple ten-minute .mat files using an in-house csv-to-matlab conversion script. Each ten-minute Mat object contained 10,800 frames of signal data and each patient had ~ 25-30 files since the total recording time for each patient averaged ~4-5 hours.

To make all signal data easier to index and load, each 10-minute .mat structure was saved as 'recording_*n*', where *n* represents the $n^{th}$ ten-minute interval of PSM data in time, inside a .mat file called 'recordings_*p*', where *p* represents the patient number.

Each 10-minute recording contained 10, 800 frames, 10, 800 COP *xy*-coordinate pairs, and 10, 800 time stamps *(hh-mm-ss-msms)*. Each frame was a Mat object with a [1 x 1] header field containing the following information: *frame number* (int), *date* (str), *time* (str), *sensor number* (str), *rows* (int), *columns* (int), *cop_x* (float), *cop_y* (float), *sensel_width* (float), *sensel_height* (float), *avg_pressure* (float), *peak_pressure* (float),

*minimum_pressure* (float), *contact_area* (float). The changing variables in each frame are the *average pressure, peak pressure, minimum pressure, cop_x, cop_y,* and *contact_area.*

### 3.4.1 Data Preparation for Motion Detection

In *Chapter 4* and *Chapter 5* the COP is used to extract features to detect movement related events from PSM data. The following steps are repeated for all 10-minute files stored in Mat objects for each patient.

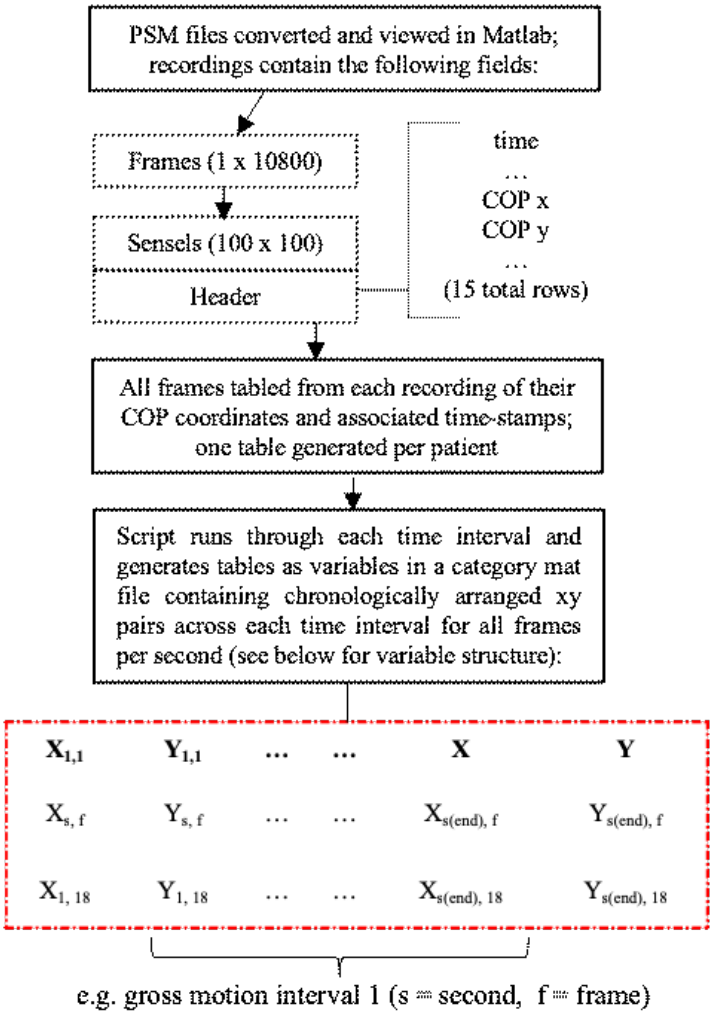



**Figure 6.** Data preparation pipeline to generate chronological sequences of centre of pressure (COP) coordinate pairs for all 18 frames per second (fps) within each event time interval.

#### *3.4.1.1 Creating COP Coordinate Tables*

Starting with the first 10-minute file and its first frame in the *recordings_[patientNumber].mat* file containing all PSM recordings for a given patient, the COP coordinates and time stamp were extracted and appended vertically to a three-column table [time_stamp x_cop y_cop]. When this process was completed for all frames in the first file, the remaining 10-minute files were looped to extract coordinates and add them to the coordinates table. Ultimately, for each patient, the COP coordinates for all frames in the complete PSM recording of the patient were appended vertically into a single three-column table. When the time stamps were added to the table, they were in 24-h format hh:mm:ss and were stripped of milliseconds. The table of coordinates was then saved as *table_patient_[patientNumber].mat.*

#### *3.4.1.2 Extracting Event Episodes from COP Coordinate Tables*

Time stamps for *motion*, *no_motion*, gross motion, upper, and lower body motion events were stored in separate sheets within each patient's Excel workbook using the annotation and classification protocol outlined in *Section 3.3.* A *time_load* function was used to extract the start and end times for each set of event annotations stored in the Excel workbook as variables and add them inside *eventType_times_[patientNumber].mat* objects which were called. The start time represented the beginning of a *motion*, *no_motion*, *etc.* event, while the end time represented the time at which the event ceased. Each pair of start and end times represented a single episode of a specific event type, so a single patient may have multiple episodes per event.

A *COP_extraction* function was then used to take individual start and end times stored in the *eventType_times* file, extract COP coordinates for that time interval from its

respective coordinate table, store it in a table with 18 rows (frames) and 2**s* columns, where 2**s* is the number of seconds in that episode multiplied by 2 since there is an *x* and *y* coordinate for all 18 frames in each second, this process is modelled in *Figure 6* above. For each episode there is a single matrix of COP coordinates where each odd column is 18 rows of *x-coordinates* while each even column is 18 rows of *y-coordinates,* since each second contains 18 frames of COP data. This matrix of COP data is named *sample_set_index* – where the index is the $n^{th}$ episode of start-end times in *eventType_times* – and saved into a Mat variable called *eventType_[patientNumber].mat*. For each start and end time in *eventType_times*, a new vector is added into *eventType_patientNumber.mat.*

The processes described above were repeated for all events – gross motion, *no_motion*, upper, and lower body motion – for all patients in the study.

# Chapter 4

## Bidirectional Motion Classification

This chapter describes the methods used to develop a discrete classification and real-time detection algorithm that distinguishes between *motion* and *no_motion* events using the sum of distances travelled by the COP as a feature.

## 4.1 Centre of Pressure Trails to Visualize Patient Movements

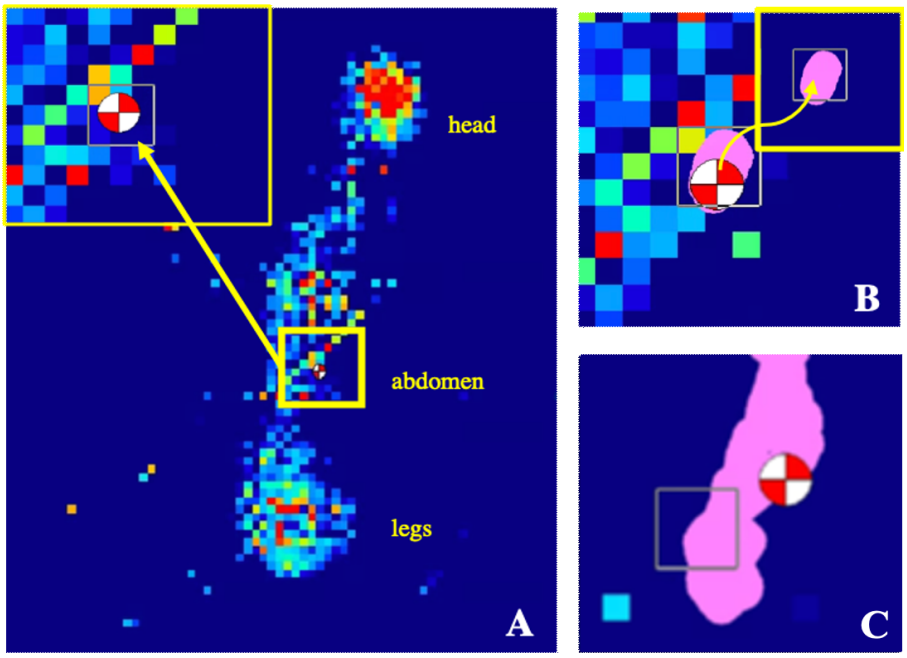



**Figure 7.** PSM data for Patient 2: (A) The start of the study; patient is laying in supine position; the centre of pressure is in the abdominal area and there are no movement trails present. (B) COP movement during a period without patient motion; inset: the pink shaded area shows the area in which the COP travelled. (C) A gross movement episode; the pink shaded area, representing the distance travelled by the COP, is significantly larger than in panel B.

The intuitive underpinnings of the motion detection algorithm developed in this chapter is illustrated in *Figure 7*. The COP is the sum of ground reaction forces between the body and the environment; it can be summarized by applied pressure at all points where the body meets the environment. In this chapter, the distance travelled by the COP during a period of sleep (no movement) is compared to the COP distance travelled during a period of motion which include upper, lower, and gross motion. For visual clarity, *Figure 7* represents the distance travelled by the COP using coloured "COP trails" for Patient 2 during a *no_motion* segment and a *motion* segment.

During a period of *no_motion*, COP movement is bound within a small area, illustrating that the total distance travelled is lower, as compared to the larger area covered by a motion event in *Figure 7b*. These variations suggest that the distances travelled by the COP may be a useful feature for distinguishing between *motion* and *no_motion* events.

## 4.2 Estimation of Distance Travelled by the COP

### 4.2.1 Quantifying COP Travel Dynamics

Since qualitative assessment of the COP trails on the XSensor Pro software showed that the COP on the PSM was moving more dramatically during periods of *motion* than the lack thereof, the actual distance travelled between consecutive COPs across the entire recording period was calculated and graphed over time. The Euclidean distance formula in eq. (9) is used to calculate the distance travelled by the COP between successive seconds. The Euclidean distance is defined as the distance between two points on a two-dimensional plane.

$$COP\_dist(t) = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad \textbf{(9)}$$

Using the coordinate tables made in *Section 3.4.1.1* containing *time* and *(x, y)* coordinates, 18 frames of *x* and *y* coordinates for each second were sliced from the three-column table, and placed side by side in a single 18 rows by 2*s* seconds matrix, where *s* is the number of seconds in the full signal recording. Each column of *x* and *y* coordinates was truncated to a single value by taking the mean, creating a coordinates vector of 1 row by *2s* columns. A vector of *x* coordinates was made by taking every consecutive *x*-value from the coordinates vector at an *odd* index. A vector of y values was made by taking every consecutive *y*-value from the coordinates vector at an *even* index.

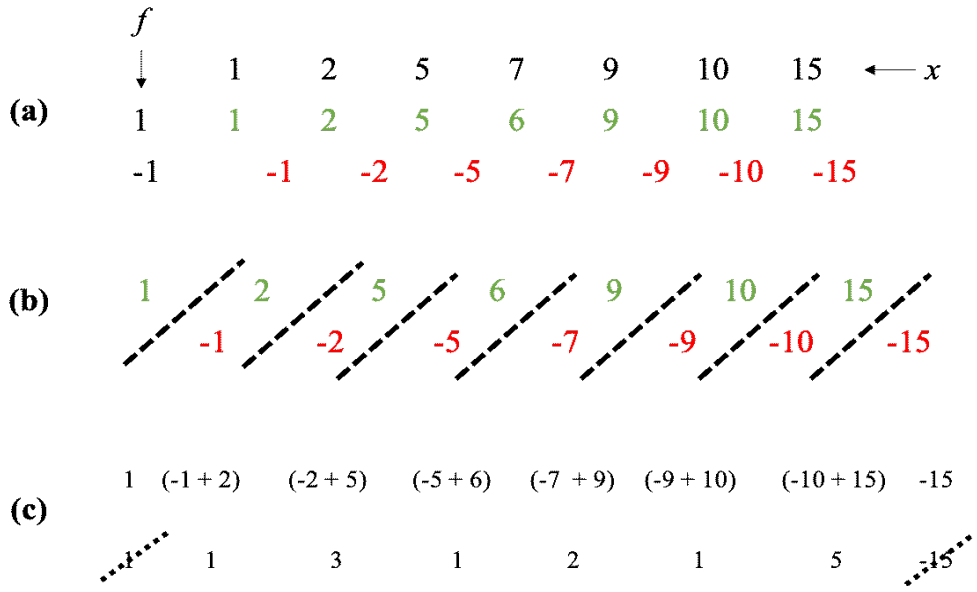



**Figure 8.** An example of using convolution by a filter $f$ = [1 -1] to find first differences between values in an array. In a simple array of length *n*, there should be *n-1* first differences. (a) Each value in $f$ is multiplied to each value in $x$. (b) The sums are computed diagonally to determine the first differences. (c) Since $1 \times x[1] = x[1]$ and $-1 \times x[n] = -x[n]$, these do not serve the purpose of computing first differences and can be discarded from the resultant array.

The first differences were found for *x* and *y* concurrently by linearly convolving the *x* and *y* vectors by a filter of [1 -1] each. If the coordinate vectors and the [1 -1] filter are two sets of polynomial coefficients, convolving the two is equivalent to multiplying polynomial coefficients. Because the linear convolution of the coordinates by [1 -1] results in the first *x* and *y* values remaining and the last *x* and *y* values returning negative, the first and last values are removed, leaving the first differences of the *x* and *y* vectors behind.

An array of distances was computed using eq. (9). The resultant array of distances were then plotted over elapsed and real time; the first vertical panel of *Figure 9* shows the COP travel overviews of three patients.

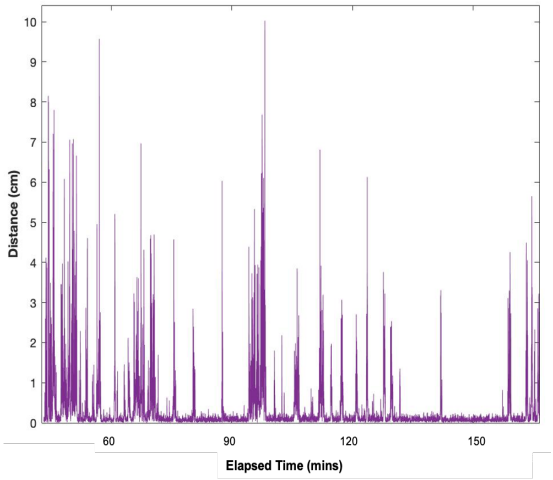
The COP travel overview plots in *Figure 9* showed dramatic peaks and valleys illustrating the distance the COP travels between consecutive seconds over the entire recording period. The vertical panel on the right zooms into a 10 – 15-minute portion of each patient recording where the time stamps are more clearly visible and reflect the accuracy with which the event annotations were recorded to the nearest minute.

Comparing the event annotations to the behaviour on the plots, the periods at which an event was labelled as *motion* matched the peaks in the plot, while the periods at which an event was labelled as *no_motion* matched the valleys. The shorter peaks present in the valleys are reflective of either a *no_motion* or a slight motion event lasting a few seconds; for this reason, a dotted line represents the *no_motion* valleys as some small motion events are also present. When the same analysis was conducted on the remaining four patients whose data are used in this research, similar trends were observed.

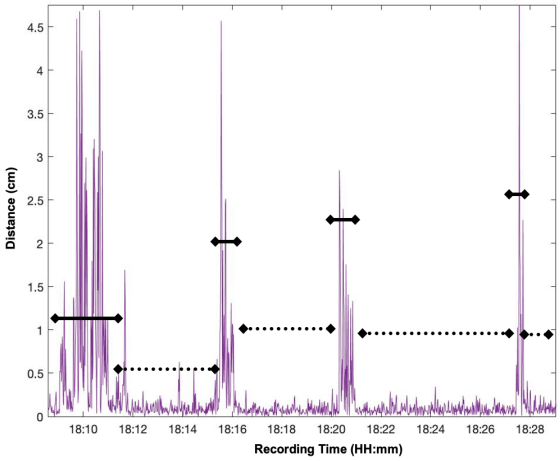
Since the differences in the distance travelled by the COP across consecutive seconds for *motion* versus *no_motion* were observed visually by movement of COP trails in the XSensor Pro V8 software and graphically by computing the actual distances travelled by

the centre of pressure across the recording period, the centre of pressure was deemed a suitable candidate to derive classification features from for use in a motion detection algorithm.

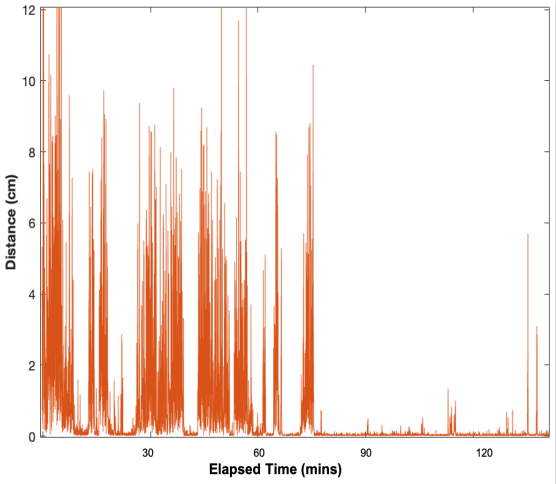
Patient 2: COP Travel Overview (~150 minutes)



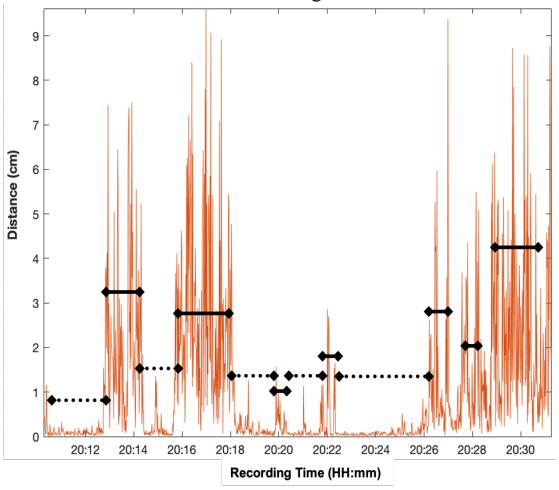

Patient 2: Annotated Segments of COP Travel




Patient 10: COP Travel Overview (~120 minutes)




Patient 10: Annotated Segments of COP Travel



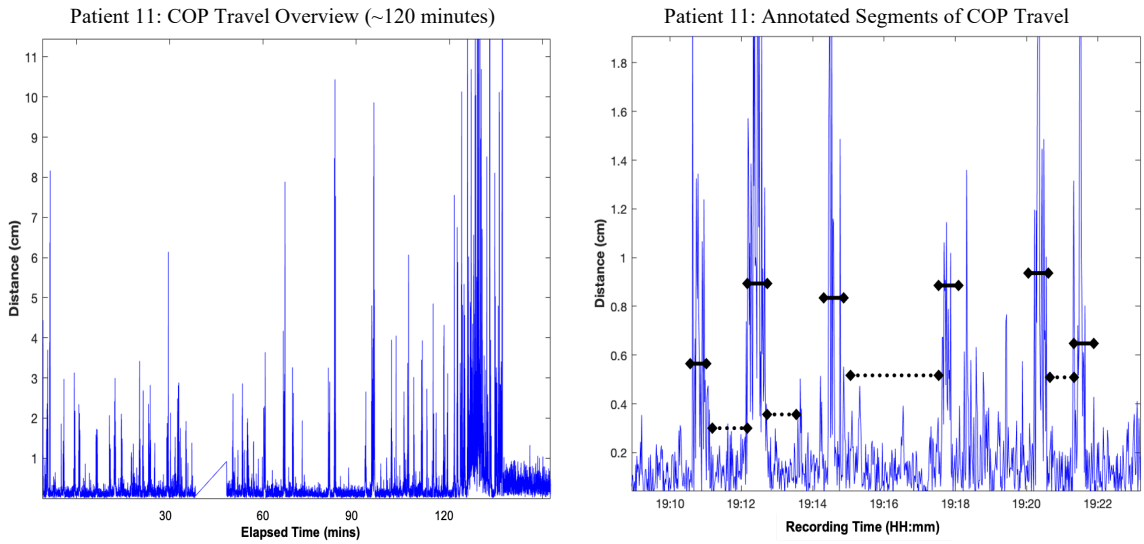



**Figure 9.** Distances travelled by the COP over consecutive seconds across the entire recording period for patients 2, 10, and 11 (left). Annotated segments of COP travel – solid lines represent gross motion events while dotted lines represent periods of *no_motion* and slight motion events (right).

## 4.3 Motion Detection Algorithm using Sum Distances based on Discrete Episodes

Based on the graphed behaviour of the COP in *Figure 9*, it was hypothesized that if the distances within *motion* and *no_motion* event segments were summed, they would present as differentiating features between periods of *motion* and periods of *no_motion*.

### 4.3.1 Developing a Proof-of-Concept Algorithm for Summed COP Distances

This section describes a method to find the optimal number of seconds (*wLen + 1)* of *motion* and *no_motion* recording data to sum per-second COP distances to determine a minimum sum distance threshold that can effectively distinguish between *motion* and *no_motion*. The sum of COP distances is computed over a window of size *wLen*; this has

the effect of smoothing out the per-second COP distances to arrive at a more robust indicator of patient movement. The sum of COP distances over a window starting at time t and of size *wLen* is defined by *Eq* (*10*). The pseudocode to implement this calculation is shown in *Figure 10.*

$$sum\_COP\_dist(t, wLen) = \sum_{\tau=t}^{t-wLen} COP\_dist(\tau) \quad \textbf{( 10 )}$$

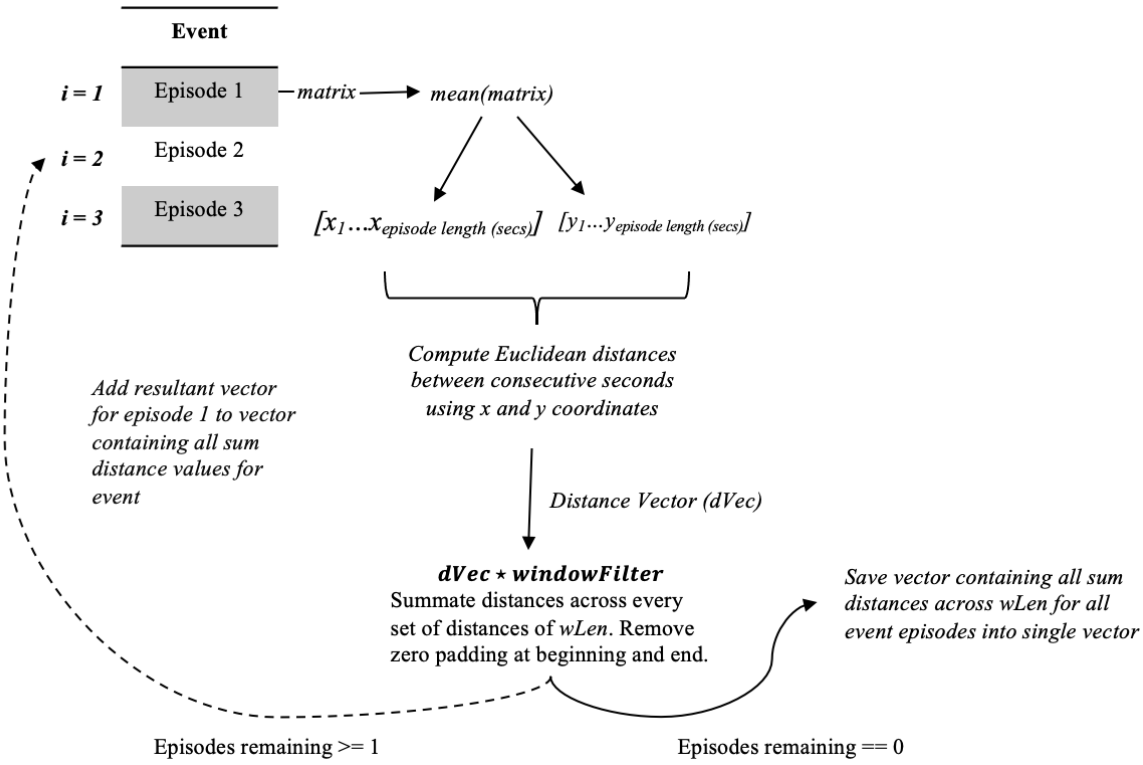


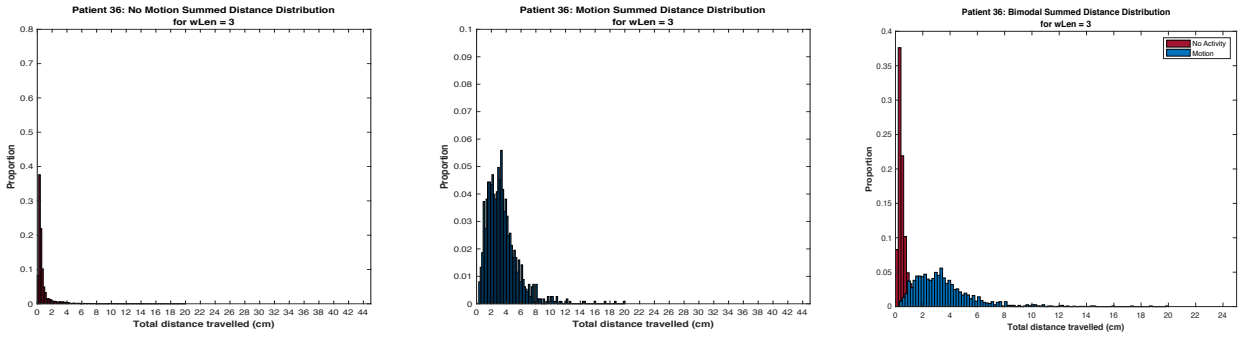

**Figure 10.** Flowchart describing how sum distance vectors were computed over every annotated episode for *motion* and *no_motion* events using a sliding window of *wLen* across each episode.

The summed distances across all annotated recordings of each event – *motion* and *no_motion* – were collected in vectors for each event type respectively to be plotted against each other; that is summed distance vectors of *motion* and *no_motion* events were prepared

separately to be plotted in the same figure. An intermediary step was taken to determine whether the duration of the annotated event episode was at least *wLen +1* seconds long to accommodate enough points for a sliding window, before applying the sliding window sum distance algorithm to a recording and adding its data to the event vectors. Upon passing this condition, the Euclidean distances travelled by the COP between consecutive seconds follows the method described in *Section 4.2.1*, for the entire episode. The COP distances are summed using a sliding window of *wLen*, resulting in one *sum_COP_distance* value for each overlapping window within the episode.

This process is repeated for each annotated episode in the event file that meets the minimum episode length condition of *wLen + 1* seconds.

To demonstrate the differences between the *motion* and *no_motion* classes using the sum COP distance distribution, *Figure 11* shows data from Patient 36 for three values of *wLen*. This parameter was tuned to find a window size that resulted in good separation between the *sum_COP_distance* values for each class.



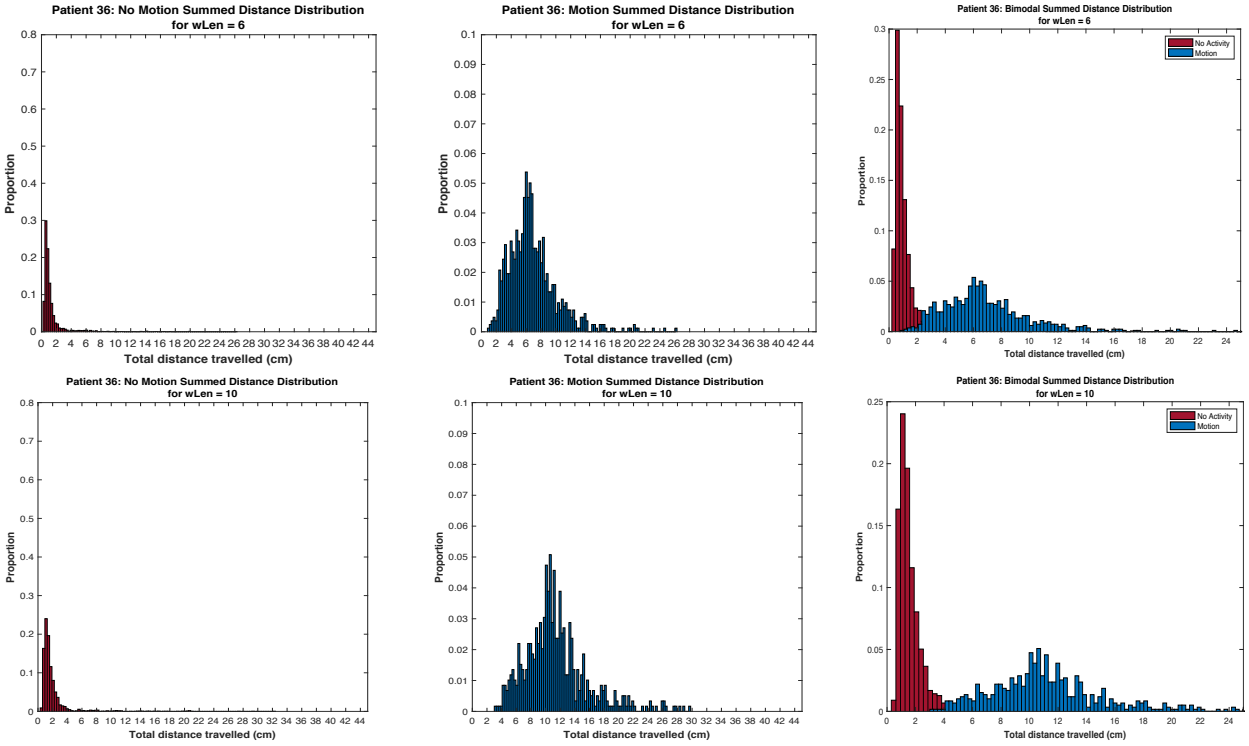



**Figure 11.** Histograms with manually adjusted window sizes to visualize summed distance distributions of *motion* and *no_motion* events in Patient 36. Window size 3 – top, window size 6 – middle, and window size 10 – bottom.

The three sample test window sizes used to explore the potential value of the algorithm were 3, 6, and 10. A window size of three was selected to represent a very small duration – four-seconds – to see how strong the distribution is with very short segments of *motion* and *no_motion* data. To determine whether distinguishability dramatically improves with larger windows, the duration was doubled to test with seven-second-long segments or a window size of six. A window of ten was the third and last window tested in this analysis to understand how great the class separation is with summed COP distances taken from segments greater than 10 seconds. The following are the insights derived from these plots:

a) As *wLen* increases in value, there is less overlap between the two classes.

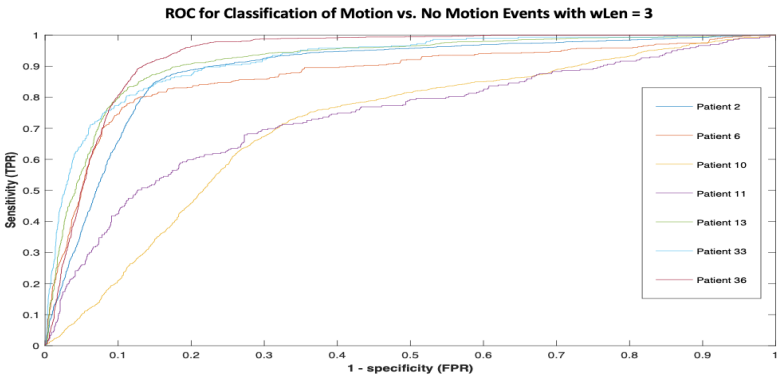
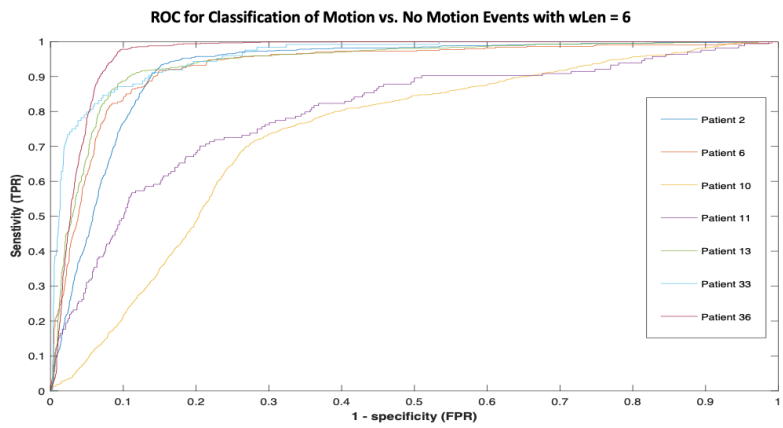
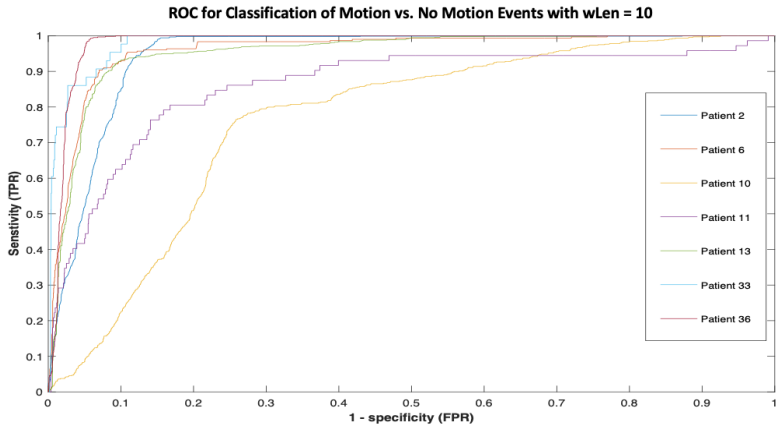
b) For *no_motion* segments, the range of total distance travelled by the COP computed for *wLen +1 seconds* of data using any of the three tested parameter sets appeared to be between 0 and 4 centimeters.

c) For *motion* segments, the range of total distance travelled by the COP computed for *wLen +1 seconds* of data using any of the three tested parameter sets appeared to be greater than 3 – 4.5, and less than 15.

d) As the window size increases, the number of episodes whose data can be considered is reduced, since shorter episodes fail the duration requirement of *wLen*+1 seconds.

While the qualitative assessment in *Figure 11* suggests the selection of a larger *wLen*, a systematic and quantitative parameter tuning was conducted, as follows. The receiver operating curve (ROC) is used to illustrate the trade-off between specificity and sensitivity of a diagnostic test at different thresholds. The overlap of *no_motion* segments into *motion* segments are false positives, while the overlap of motion segments into *no_motion* segments are false negatives. Moving the decision threshold point to the left will increase the number of false positives and moving it to the right will increase the false negatives. At each threshold, there is a corresponding sensitivity (TPR) and 1-specificity (FPR). Depending on how close the ROC curve is to the y-axis, parallel to the x-axis, and distant from the diagonal of the plot, how well the classes are separated and whether the optimal threshold is sufficient to distinguish between classes can be visualized.

**Table 4.** AUC values and ROC plots representing classification quality of the sum distance algorithm with changing *wLen*

| *wLen* | *Patient* | *AUC* | *ROC* |
|---|---|---|---|
| ***3*** | *2* | 0.8881 |  |
| | *6* | 0.8710 | |
| | *10* | 0.7096 | |
| | *11* | 0.7369 | |
| | *13* | 0.9158 | |
| | *33* | 0.9188 | |
| | *36* | 0.9324 | |
| ***6*** | *2* | 0.9220 |  |
| | *6* | 0.9289 | |
| | *10* | 0.7329 | |
| | *11* | 0.7960 | |
| | *13* | 0.9385 | |
| | *33* | 0.9577 | |
| | *36* | 0.9647 | |
| ***10*** | *2* | 0.9459 |  |
| | *6* | 0.9596 | |
| | *10* | 0.7628 | |
| | *11* | 0.8607 | |
| | *13* | 0.9522 | |
| | *33* | 0.9823 | |
| | *36* | 0.9800 | |







To quantify the quality of separation between the classes with the output of the sum distance algorithm using the increasing window sizes, we compute at the ROC-AUC as

shown in *Table 4.* AUC values and ROC plots representing classification quality of the sum distance algorithm with changing *wLen*; the greater the AUC, the better the class separation.

*Table 4* shows both quantitatively and visually that changing the window sizes appears to support the hypothesis that varying the size of the summed distance window can improve the quality of classification. In this proof-of-concept experiment arbitrary window sizes were used, and the AUC was used as an evaluation metric to test whether variations in the size of the observation window would impact the quality of classification.

The window sizes were selected without knowledge of how other window sizes could perform; seeing that Patients 10 and 11 have shallower ROCs to the remaining patients, the next step was to determine the optimal classifying window size for each patient individually to work towards a generalized window that would reasonably suit all patients.

#### *4.3.1.1 Patient-Specific Parameter Optimization for Global Generalizability*

This section discusses the outcomes of using a grid search to compute AUC's for each patient. The best window size is that which produces the highest AUC, however, this section also considers influencing factors such as the ratio between *motion* and *no_motion* events in the evaluation set. A discussion of generalization follows.

In each patient recording, there are more *no_motion* annotations than *motion* annotations because the infant tends to spend most of its time sleeping without movement or disturbances; for this reason, there is class imbalance present in all recordings. Whereas a *no_motion* episode can often be greater than 20 seconds, *motion* episodes are most often limited to <10 seconds. When the window size increases, the minimum duration of a classifiable event should increase which reduces the number of eligible episodes in the

discrete episode training dataset. The AUC at that point is based on a very imbalanced dataset which is not a fair representation of the actual data.

**Table 5.** Median duration of *motion* and *no_motion* episode annotations

| Patient | Median Motion Duration (secs) | Median *No_motion* Duration (secs) |
|---|---|---|
| **2** | 12.0 | 22.0 |
| **6** | 10.0 | 12.0 |
| **10** | 8.0 | 10.0 |
| **11** | 7.0 | 13.0 |
| **13** | 10.0 | 13.0 |
| **33** | 6.0 | 29.0 |
| **36** | 7.0 | 8.0 |
| **Average** | **8.6** | **15.3** |

*Table 5* shows the median duration of *motion* and *no_motion* recordings for each patient and the average median across all patients. Seeing that the average median duration of motion events is 8.6 seconds, approximately 50% of the *motion* data would be omitted from the training set if the *wLen* > 9.

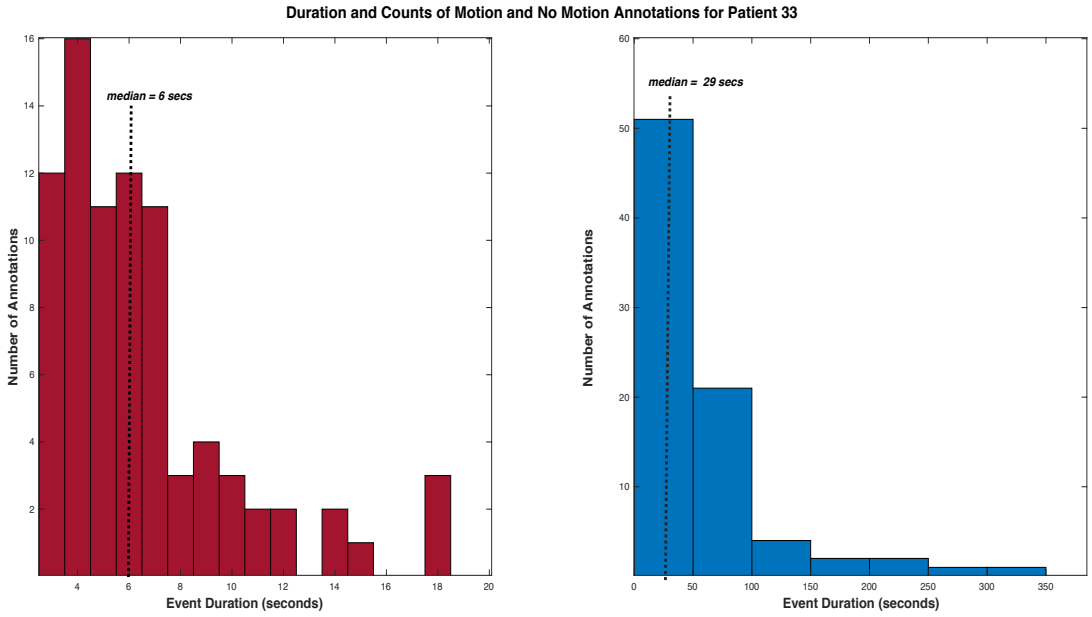



**Figure 12**. Distribution of durations of all *motion* (left) and *no_motion* (right) annotations for Patient 33 with medians labelled (dashed lines).

*Figure 12* shows the distribution of durations of motion and *no_motion* annotations for Patient 33 to visualize the generally short durations of *motion* episodes and the generally longer durations of *no_motion* episodes.

To further illustrate the impact of the *wLen* parameter on the class imbalance between *motion* and *no_motion* episodes, *Figure 13* illustrates the ratio of *motion:no_motion* episodes for each patient, as *wLen* increases, and the AUC as a function of *wLen* for each patient.

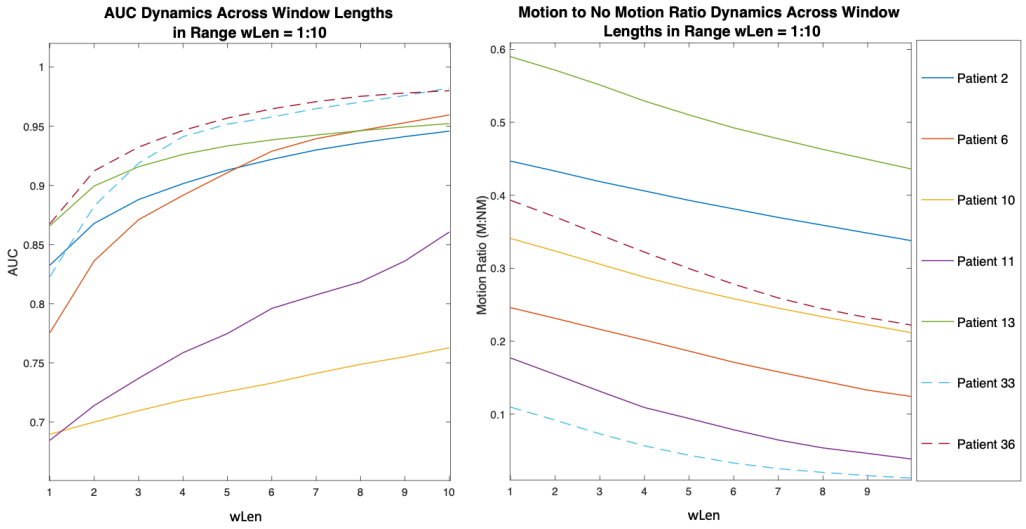



**Figure 13.** AUC and motion ratio (*motion*:*no_motion*) variations as *wLen* increases.

Using the algorithm from *Section 4.2.1,* for each patient the AUC values were computed for all wLens in the range of 1 – 10. For each of the ten window sizes, the plots in *Figure 14* show an inverse relationship between AUC and the motion ratio, where a higher *motion* to *no_motion* ratio produces a lower AUC. This experiment formed the basis of identifying the optimal window size for each patient.

Since it became empirically clear that increases in AUC may be due to increasing imbalance in the data, AUC was no longer the sole metric in determining optimal parameter

sets alone. Given that neonatal motions in the NICU are diverse, it was important to include as much of the observed motion data as possible while achieving the best possible AUC.

For each patient, the *motion*:*no_motion* ratio and AUC at each window size were computed and ordered based on the *motion*:*no_motion* ratio. The minimum and maximum M:NM ratio were averaged. Then, the window size producing a *motion*:*no_motion* ratio closest to the average was selected, and the corresponding AUC was noted. As shown in *Figure 13* and *Table 5*, greater window sizes correspond with larger AUCs, but all motion types are not necessarily represented as shorter motion events are omitted. For this reason, AUC is being traded off to retain the greatest possible amount of motion training data to ensure that the training data is representative of all motion types including brief motions.

**Table 6.** AUC performance and *motion*:*no_motion* ratios across window sizes 4 and 3 *(wLen)* across all patients. AvgA reports the average performance across all patients, while AvgB excludes patients 10 and 11 as outliers (see text).

| | | ***wLen*** | |
|---|---|---|---|
| | | *4* | *3* |
| **Patients** | *2* | 0.90 (0.41) | 0.89 (0.42) |
| | *6* | 0.89 (0.20) | 0.87 (0.22) |
| | *10* | 0.72 (0.29) | 0.71 (0.31) |
| | *11* | 0.76 (0.11) | 0.74 (0.13) |
| | *13* | 0.93 (0.53) | 0.92 (0.55) |
| | *33* | 0.94 (0.06) | 0.92 (0.07) |
| | *36* | 0.95 (0.32) | 0.93 (0.35) |
| **Avgs.** | **AvgA** | 0.87 (0.27) | 0.86 (0.29) |
| | **AvgB** (ex. P10, P11) | 0.92 (0.22) | 0.91 (0.23) |

*Table 6* summarizes the best window size for each patient and the ratio of *motion*:*no_motion* events in brackets. Ideally, a higher *motion*:*no_motion* ratio in the training set suggests a reduced likelihood of false negatives, more diversity in the types of motion events captured, and potential for the parameters to represent most of the motion data.

The average AUC and motion ratio was computed for *wLen* parameters three and four across all patients first (AvgA in *Table 6*) and then again excluding patients 10 and 11 (AvgB in *Table 6*). Patients 10 and 11 were excluded from further consideration during parameter optimization, since their AUCs are lower on average compared to other patients in the set, despite their *motion*:*no_motion* ratios being similar. Upon further examination, patients 10 and 11 were found to have more upper body motion events than other patients, where upper body motion is more difficult to classify than whole body or lower body (see *Chapter 5*).

*Table 6* indicates that *wLen* = 4 results in a greater AUC, while *wLen* = 3 results in a greater *motion*:*no_motion* ratio. Giving preference to the parameter with the highest *motion*:*no_motion* ratio, the ideal window size appears to be *wLen* = 3, suggesting that if an episode is at least three seconds long, it is sufficient to distinguish between *motion* and *no_motion* to an average AUC of 0.86 across all patients (AvgA), and to an AUC of 0.91 across most patients (AvgB).

#### *4.3.1.2 Determination of Decision Threshold*

MATLAB's inbuilt function *perfcurve* was used to create the ROC curves in this thesis. While the ROC curve reports the performance over all possible decision thresholds, it is necessary to select a single decision threshold for deploying the final classification model.

To compute the optimal operating point of the ROC, the slope *(S)* of the cost function is computed using eq. (11) which considers the cost *Cost(N|P)* of misclassifying a positive class as a negative class, the cost *Cost(P|N)* of misclassifying a negative class as a positive class, the total positive classifications $P = TP + FN$, and the total negative classifications $N = TN + FP$.

$$\boldsymbol{S = \frac{Cost(P|N) - Cost(N|N)}{Cost(N|P) - Cost(P|P)} * \frac{N}{P}} \qquad \textbf{( 11 )}$$

The point at which the cost curve with slope *S* meets the ROC curve is the optimal operating point. In this thesis, we use the default zero-one cost function, so the first term in eq. (11) disappears. Therefore, the optimal operating point is a function of N/P, or the class imbalance. Using this point, the threshold by which the optimal point was computed is found, and subsequently, the threshold is used as a cut-off value to calculate classification accuracy by first calculating true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN).

**Table 7.** Optimal thresholds and associated performance accuracies

| *wLen* | *Patient* | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | *2* | | *6* | | *10* | | *11* | | *13* | | *33* | | *36* | |
| | *t (cm)* | *acc* | *t (cm)* | *acc* | *t (cm)* | *acc* | *t (cm)* | *acc* | *t (cm)* | *acc* | *t (cm)* | *acc* | *t (cm)* | *acc* |
| *3* | 0.65 | 0.85 | 0.94 | 0.88 | 21.14 | 0.77 | 10.42 | 0.88 | 0.98 | 0.87 | 1.25 | 0.94 | 1.34 | 0.88 |
| | (0.42) | | (0.22) | | (0.31) | | (0.13) | | (0.55) | | (0.06) | | (0.35) | |
| *4* | 0.90 | 0.86 | 1.22 | 0.89 | 28.94 | 0.78 | 6.77 | 0.90 | 1.70 | 0.88 | 1.39 | 0.96 | 1.94 | 0.90 |
| | (0.41) | | (0.20) | | (0.29) | | (0.11) | | (0.53) | | (0.06) | | (0.32) | |

The decision thresholds for *wLen* = 3 and *wLen* = 4 are presented in *Table 7* along with the ratio of *motion* to *no_motion* episodes. In general, the thresholds ranged between 0.65 cm and 1.94 cm; however, Patients 10 and 11 have thresholds greater than 6, ranging

between 6.77 cm and 28.94 cm. This further supports their exclusion from the *wLen* hyperparameter tuning in the previous section.

*Table 8* displays the results of using various decision thresholds with window sizes three and four. Thresholds for each window size were averaged across all patients, and the overall accuracy is reported when that average threshold is applied to all patients. The same process is repeated two more times across both window sizes: once with the average threshold calculation omitting the threshold found for Patient 10, and once with omitting the thresholds found for both Patients 10 and 11.

**Table 8.** Average decision thresholds and associated accuracies at *wLens* of 3 and 4

| **Group Average** | ***wLen = 3*** | | ***wLen = 4*** | |
|---|---|---|---|---|
| | ***t (cm)*** | ***acc*** | ***t (cm)*** | ***acc*** |
| *All* | 5.2457 | 0.8671 | 6.1229 | 0.8814 |
| *Exclude Patient 10* | 2.5967 | 0.8833 | 2.3200 | 0.8983 |
| *Exclude Patient 10 and 11* | 1.0320 | 0.8840 | 1.4300 | 0.8980 |

In *Table 7*, the algorithm performs considerably better with Patient 11's data than with Patient 10's data across both window sizes in terms of accuracy at the optimal threshold, but its *motion:no_motion* ratios are smaller than the average in the set and compared to those of Patient 10. Including the threshold of Patient 10 causes the most dispersion in the overall average. In *Table 8* when Patient 10's threshold is dropped from the average calculation for both *wLens* 3 and 4, the change in average threshold is between 2.64 cm and 3.80 cm. When Patient 11's data is further removed from the average, the change in the average threshold from this exclusion for both windows is between 0.89 cm and 1.30 cm. While Patient 11's threshold values are high and its motion ratios are less than 0.15 cm across both window sizes, including this patient's threshold as part of the final

parameter calculation may improve the generalizability of the algorithm across patients with more ambiguous data, possibly including Patient 10's data.

The parameter *wLen* = 4 and its corresponding threshold produce the highest overall classification accuracy and produces a negligible increase in accuracy compared to the accuracy at *wLen = 3* when data for Patient 10 and 11 are removed. To create a generalizable threshold that can include anomalous patients like Patient 10 and 11, *wLen* = 4 is selected as the window size.

The final threshold is computed by first averaging the two thresholds within each window size in *Table 8* where data for Patient 10 (Case 1) and data for both Patient 10 and 11 (Case 2) are removed. The two resultant values are then averaged between the two window sizes to compute the final threshold.

The final cut-off threshold, that is the minimum distance the COP must travel over a window to be classified as a *motion* event, is computed as follows:

$$t_{GEN} = \frac{t_{Case\ 1\ (wLen=3)} + t_{Case\ 2\ (wLen=3)}}{4} + \frac{t_{Case\ 1\ (wLen=4)} + t_{Case\ 2\ (wLen=4)}}{4}$$

$$t_{GEN} = \frac{2.60\ cm + 1.03\ cm}{4} + \frac{2.32\ cm + 1.43\ cm}{4}$$

$$t_{GEN} = 1.84\ cm \qquad \textbf{( 12 )}$$

#### *4.3.1.3* Final Motion Detection Generalized Parameter Set

The ultimate parameter set for motion detection is *(wLen* = 4 s*, $t_{GEN}$* = 1.84 cm). In *Table 9*, the AUC is computed for this window size, what remains is to compute the classification accuracy across all patients using the $t_{GEN}$ threshold. For this experiment, Patients 10 and 11 are included to assess their performance at this threshold. *Table 9* displays the

performance of each patient using this parameter set, where patients 10 and 11 have the lowest F1 scores, recall, and precision metrics.

**Table 9.** Motion detection performance across all patients with generalized parameter set *(wLen = 4, 1.84 cm)*

| *Patient* | *AUC* | *TP* | *FP* | *TN* | *FN* | *FPR* | *Precision* | *Recall* | *Acc* | *F1* |
|---|---|---|---|---|---|---|---|---|---|---|
| ***2*** | 0.90 | 870 | 321 | 2588 | 311 | 0.10 | 0.73 | 0.74 | 0.85 | 0.74 |
| ***6*** | 0.89 | 330 | 153 | 2505 | 206 | 0.08 | 0.68 | 0.62 | 0.89 | 0.65 |
| ***10*** | 0.72 | 665 | 970 | 2398 | 304 | 0.09 | 0.41 | 0.67 | 0.71 | 0.51 |
| ***11*** | 0.76 | 108 | 209 | 2001 | 133 | 0.06 | 0.45 | 0.34 | 0.86 | 0.38 |
| ***13*** | 0.93 | 915 | 180 | 1976 | 226 | 0.10 | 0.84 | 0.80 | 0.87 | 0.82 |
| ***33*** | 0.94 | 87 | 54 | 3852 | 134 | 0.03 | 0.62 | 0.39 | 0.95 | 0.48 |
| ***36*** | 0.95 | 930 | 366 | 2773 | 81 | 0.03 | 0.72 | 0.92 | 0.89 | 0.81 |

*Figure 14* displays the ROC and precision-recall curves for each patient's performance with this parameter set.

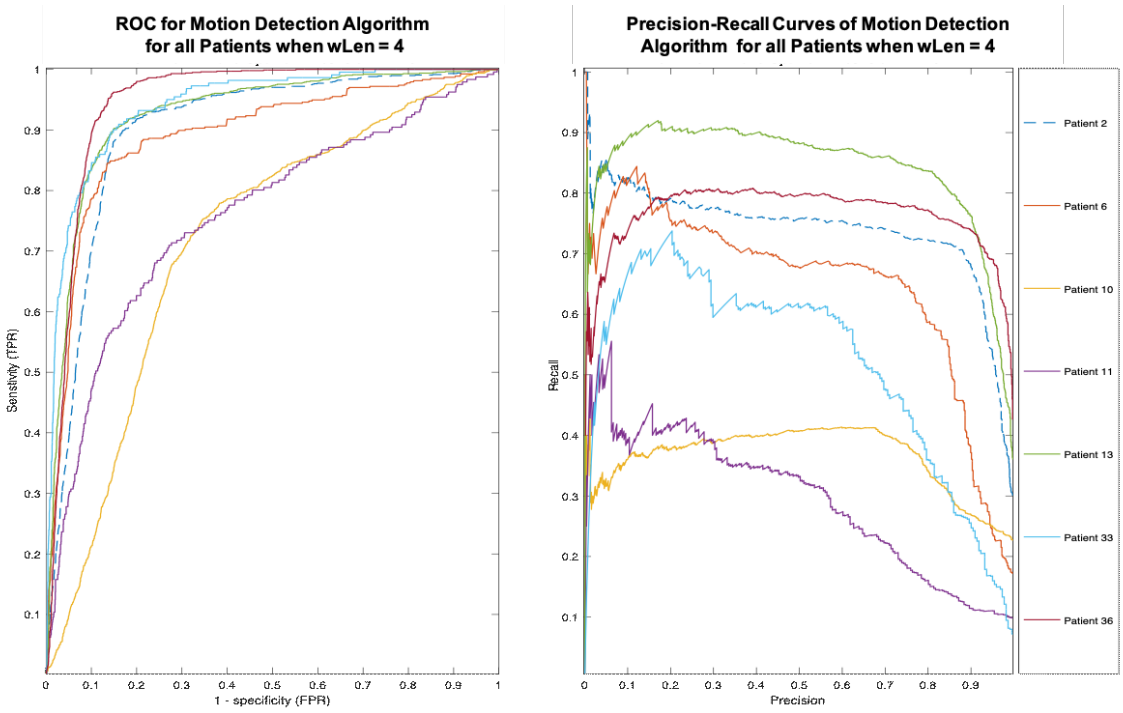



**Figure 14.** ROC and precision-recall curves for algorithm performance using *wLen = 4* across all patients to distinguish between *motion* and *no_motion* events.

The algorithm developed in this section uses discrete episodes of *motion* and *no_motion* in a non-continuous series of events; that is, multiple episodes of *motion* and *no_motion* were identified, and the COP distance was computed over windows within this episodes. Based on the figures and outcomes in this section, there is a difference in the COP distances computed over *motion* and *no_motion* episode across window size variations. The limitation of the work in this section is that, in a real-time detection algorithm, there can be overlap between *motion* and *no_motion* data within a sliding window and there is no limitation on the minimum episode length. For these reasons, a real-time detection algorithm may consider a different minimum sum threshold for classification between a *motion* and *no_motion* event. The determined window size of 4 and cut-off threshold of 1.84 cm.

## 4.4 Real-time Motion Detection Algorithm with Sum

To develop a real-time motion detection algorithm, continuous segments PSM data containing both *motion* and *no_motion* events were selected of varying lengths; some segments are as short as two minutes while others are as long as thirty minutes. The total amount of time used from all patients was six hours and eight minutes, with an average of one hour from each patient.

For each segment, COP coordinates were down-sampled to one coordinate pair per second and were labelled as *motion* or *no_motion* according to the annotated observations. However, since the sum algorithm first finds the distance between consecutive seconds before computing the sum distance algorithm across a window, labels were assigned across a sliding window of two datapoints. If both seconds were *not motion*, then a label of

'*no_motion*' was given. If one of two seconds was *motion*, then a label of '*motion*' was given; the final set of datapoints in the real-time series was the original length minus one.

The Sum_COP_Distance algorithm in *Section 4.3* used data that contained discrete types of episodes – either *motion* or *no_motion*, so the true labels of a window were always either 0 or 1, never a combination. For the real-time case, windows of training data contains a mix of *motion* and *no_motion* episodes. The true labels were created based on the annotations across the entire window using the median event within the window. However, this requires that the window be an odd value. The labelling method is illustrated in *Figure 33* in *Appendix D.* In *Section 4.3,* the 'optimal' parameters for the sum window approach was a window size of 4, which is even; so, the *wLen* parameter required retuning in the present section.

Previously, shorter window sizes were preferred, to ensure that short episodes could be retained in the training data. Given that real-time data is continuous, the real-time algorithm explores longer *wLen* parameter values. The algorithm was tested with window sizes of 5, 7, 9, and 11.

### 4.4.1 Identifying Optimal Parameters for Real-Time Motion Detection

*Table* 10 reports the AUC, threshold, precision, and recall at the optimal threshold which was determined using Equation 9 for odd-numbered window sizes within the range [5,11], across patients 2, 6, 11, 13, 33, and 36, where motion is the positive class. Patient 10 was excluded from analysis here, for the reasons discussed in *Section 4.3*.

**Table 10.** Performance metrics of the COP-sum-distance algorithm for real-time motion detection (*auc* = ROC-AUC, *t* = threshold, *P* = precision, and *R* = recall)

| ID | Mass Class | 5 | | | | 7 | | | | 9 | | | | 11 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **auc** | ***t*** | **P** | **R** | **auc** | ***t*** | **P** | **R** | **auc** | ***t*** | **P** | **R** | **auc** | ***t*** | **P** | **R** |
| **2** | High | 0.93 | 1.46 | 0.87 | 0.84 | 0.94 | 1.63 | 0.85 | 0.90 | 0.95 | 2.02 | 0.85 | 0.92 | 0.95 | 2.75 | 0.85 | 0.92 |
| **6** | High | 0.87 | 1.53 | 0.80 | 0.82 | 0.90 | 2.67 | 0.83 | 0.80 | 0.92 | 3.51 | 0.84 | 0.84 | 0.93 | 4.52 | 0.86 | 0.85 |
| **13** | High | 0.82 | 4.62 | 0.73 | 0.71 | 0.81 | 5.64 | 0.72 | 0.73 | 0.89 | 7.37 | 0.78 | 0.80 | 0.85 | 9.98 | 0.68 | 0.69 |
| **11** | Med | 0.69 | 7.26 | 0.59 | 0.33 | 0.70 | 8.59 | 0.60 | 0.34 | 0.71 | 9.93 | 0.50 | 0.35 | 0.72 | 12.1 | 0.37 | 0.37 |
| **33** | Med | 0.85 | 3.10 | 0.54 | 0.32 | 0.86 | 3.82 | 0.73 | 0.37 | 0.94 | 3.83 | 0.75 | 0.55 | 0.95 | 4.50 | 0.71 | 0.53 |
| **36** | Med | 0.88 | 3.54 | 0.85 | 0.87 | 0.93 | 5.34 | 0.88 | 0.84 | 0.91 | 7.22 | 0.82 | 0.72 | 0.96 | 8.75 | 0.92 | 0.84 |
| Average | | 0.84 | 3.59 | 0.73 | 0.65 | 0.85 | 4.62 | 0.77 | 0.66 | 0.89 | 5.65 | 0.76 | 0.70 | 0.89 | 7.10 | 0.73 | 0.70 |
| Average (ex. P11) | | 0.87 | 2.85 | 0.76 | 0.71 | 0.88 | 3.82 | 0.80 | 0.73 | 0.92 | 4.79 | 0.81 | 0.77 | 0.93 | 6.10 | 0.80 | 0.77 |

*Table* 10 shows relatively consistent AUC performances across the real-time segments tested for each patient in the high mass class category ($m \geq 2500g$). The optimal threshold for each patient in this category also produced relatively high precision and recall scores across all windows except for *wLen* = 11, which suggests that including a window too large can confuse the algorithm when it deals with the problem of multiple events in a single window. Given that motion events are generally shorter than *no_motion* events, the algorithm may tend to over-predict the dominant class for large windows that include both *motion* and *no_motion* data.

The AUC, precision, and recall scores show consistent trends across *wLen* values, but not between subjects. As with *Section 4.3*, Patient 11 is dissimilar from all other patients with low AUC across all windows. Given the anomalous behaviour of Patient 11's data, averages are computed including and excluding their metrics. Removing Patient 11 from the average decreases the average decision threshold by ~1.0 cm, increases the average precision by ~0.03 – ~0.07, and increases the average recall by ~0.10.

While the AUC generally increases as the window size increases, the same cannot be said for precision. As *wLen* and minimum threshold increases, recall also increases; the opposite effect is observed for precision. Interestingly, beyond *wLen* = 9, precision decreases and recall plateaus. Between *wLens* 9 and 11, average recall values are the same, at which point we look at precision – precision is highest at a *wLen* of 9 with an average decision threshold of 4.79 cm excluding Patient 11. For these reasons, the window size that is used for the motion detection algorithm is that of 9.

#### *4.4.1.1* Testing Real-Time Motion Detection Algorithm with Baseline Parameters

The parameters determined in *Section 4.4.1* were created by averaging classifier performance at *wLen* = 9 and across patients; this experiment was important in determining the baseline threshold in distinguishing between motion and *no_motion* events.

*Table 10* reported performance using patient-specific decision thresholds across *wLens* 5,7, 9, and 11, the following section will evaluate patient performance using the determined generalized decision threshold of 4.79 cm which was computed as an average of all patient-specific thresholds across a *wLen* of 9.

*Table 11* evaluates the performance of the classifier on uninterrupted real-time sequences of data containing both *motion* and *no_motion* events across all patients including Matthew's Correlation Coefficient (MCC) and the relative F1-scores. These performance metrics were chosen here since there is substantial class imbalance between *motion* and *no_motion* event types; the MCC and F1-scores are insensitive to unbalanced data, and the MCC is additionally insensitive to asymmetric data.

**Table 11.** Performance metrics of real-time motion classifier using the generalized parameter set (*wLen* = 9, *t* = 4.79 cm)

| | Patient 2 | Patient 6 | Patient 11 | Patient 13 | Patient 33 | Patient 36 |
|---|---|---|---|---|---|---|
| **Precision** | 0.89 | 0.86 | 0.47 | 0.88 | 0.73 | 0.86 |
| **Recall** | 0.68 | 0.61 | 0.32 | 0.70 | 0.24 | 0.70 |
| **AUC** | 0.94 | 0.94 | 0.69 | 0.92 | 0.94 | 0.96 |
| **MCC** | 0.69 | 0.70 | 0.26 | 0.66 | 0.40 | 0.78 |
| **F1 Score** | 0.77 | 0.72 | 0.38 | 0.78 | 0.36 | 0.82 |

Across all patients in *Table 11,* the precision scores are consistently higher than the recall scores. This suggests that the classifier is correct in its prediction of a *motion* event 86% of the time with most patients, with the exclusion of Patients 33 and 11, or at least 73% of the time when including Patient 33. The recall scores are within 0.60 – 0.70; this means that up to 40% of the time, this classification system misclassifies *motion* events as *no-motion* events.

It is important for precision to be high because the cost of incorrectly classifying a *no_motion* event as a *motion* event is higher than incorrectly classifying a *motion* event as a *no_motion* event. If a *no_motion* episode are marked as *motion* – the system may incorrectly gate an alarm by assuming that any alarm was caused by motion artifact. Simultaneously, it is also important for recall to be high because if patient behaviour is incorrectly classified as *no_motion* when the infant is truly moving, this could allow false alarms to pass resulting in nurse fatigue. Clearly, the cost of incorrectly gating a true patient alarm outweighs the cost of a fatigued nurse; for this reason, a classifier with a higher precision than recall is favoured, and reduced recall may be forgiven.

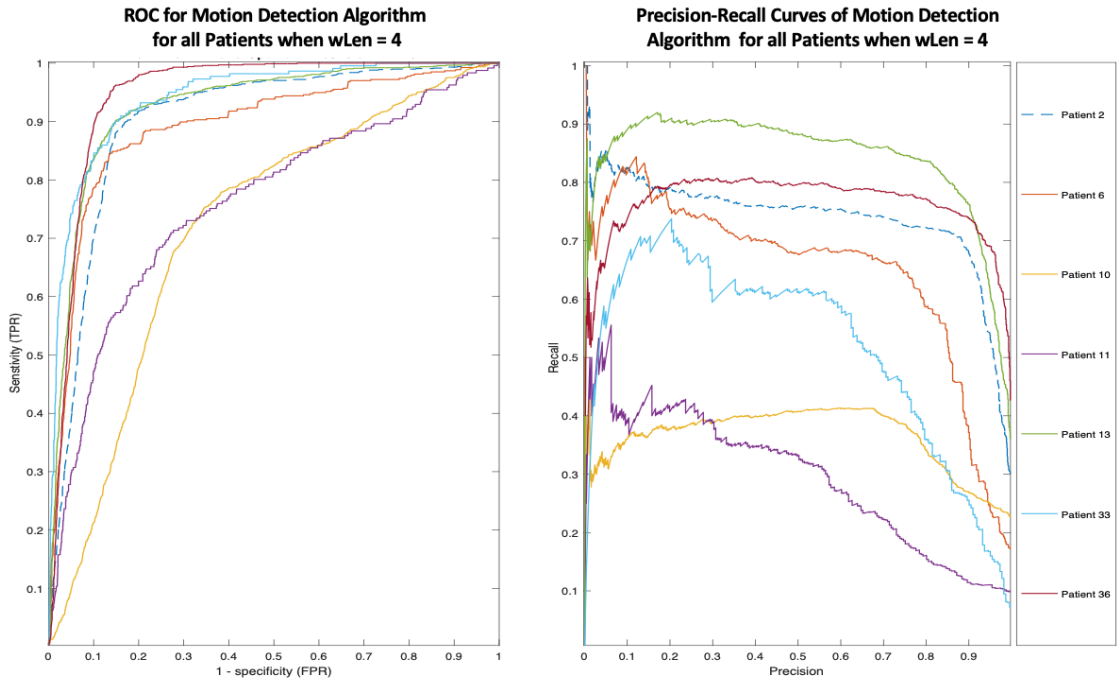



**Figure 15.** ROC and precision-recall curves between patients with real-time classifier using generalized parameters.

*Figure 15* plots the ROC and precision-recall curves for all continuous sequences of data across patients at *wLen* = 9. As per the results in *Section 4.3*, when testing individual patients, Patient 11's ROC curve and AUC score are markedly lower than all other patients. While it is not clearer in the ROC plot, the P-R plot demonstrates that the classifier also struggles to classify Patient 33's data.

A more careful retrospective review of Patient 11's data revealed that several motion events had been mislabelled, likely because of the thick quilt that was covering the patient. The challenge with Patients 11 and 33 was that 40 – 60% of their bodies were covered with a quilt; this did allow for gross motion and upper body motion to be identified with slightly higher degrees of certainty than lower body movement. Patient 11 was 60% covered with a quilt, so upper body motion occurring from the neck down was difficult to identify on camera and in-person but could be captured on the PSM. For this reason, any annotations

labelled as ‘no motion’ may have included upper or lower body motion events which could not be seen on camera, leading to skewed performance metrics.

The reason the data for Patient 11, 33, and 36 were included in this dataset is because during PSM-video timestamp matching with registration points, there was alignment, but every single video annotation was not double checked for alignment after this point. It is possible that the video and PSM become asynchronous at some point in the recording resulting in the video and PSM match to be completely off, while for other patients, this was less of an issue. Patient 36 was uncovered, unlike Patients 11 and 33; it is worth exploring whether this patient had elevated heart rate throughout the recording period, increasing the velocity of thoracic activity, leading to more pronounced COP movement during periods of no motion, confusing the algorithm. As it pertains to Patient 33, it was found that 32 out of 49 continuous monitoring segments were less than six seconds so as per the sliding window, could never be identified as motion – this suggests that very short windows of motion are unidentifiable by this algorithm. Lastly, it was found for the upper and lower body motion segments were completely overlapping, suggesting that hidden upper body events under the blanket were misannotated as no motion.

Throughout this chapter, the data of both Patients 10 and 11 were excluded from analyses at various steps, ultimately excluding Patient 11’s data. The performance of Patient 10 was generally different from all other patients, and as mentioned, it is possible that Patient 10’s data also suffered from asynchrony between the PSM and the video data. However, as it did not perform as poorly as Patient 11 overall, it was included in the parameter optimization steps for the real-time model.

A detailed discussion of the nuances of Patients 11, 33, and 36, considerations of their data, and how adjusting the type and amount of Patient 11’s data can improve classifier

performance is continued in Appendix B. For the uncertainties and nuances described above and in Appendix B, Patients 11, 33, and 36 are excluded from the remaining work.

## 4.5 Conclusions

In this chapter it was found that the distance travelled by the COP during periods of activity by a neonate weighing at least 1500g on a pressure sensor mat is a valid feature to use in distinguishing between discrete periods of *motion* and *no_motion*. Initial proof of concept was developed using discrete episodes of patient moving or at rest. It was found that increasing the window size led to improved performance, however, this improvement was also due to longer window sizes ruling out brief periods of motion from the data analysis. The algorithm was then extended to examine a sliding window of PSM COP data, no longer constrained to discrete periods of *motion*/*no_motion*. The window size hyperparameter was re-examined and a value of *wLen* = 9 (10 seconds) was found to work well for all patients included in the analysis. With a sample of six patients with at least thirty minutes of continuous real-time data per patient, the sum-distance classifier for motion detection produced a precision score of 0.84, a recall of 0.68, and an overall F1-score of 0.76. The classifier requires a window of ten seconds of data to detect whether a motion event is occurring on a second-by-second basis using a sliding window approach. This outcome is double the ideal window required for discrete episodes, in which a minimum window size of five seconds was identified. This chapter addressed the binary question of the absence or presence of motion of any kind*, Chapter 5* will take this question further to differentiate between upper versus lower body motion.

# Chapter 5

## Upper versus Lower Body Motion Detection

In this chapter we examine whether it is possible to differentiate between upper and lower body motion using discrete segments of motion data. A set of seven features are extracted from sequences of COP data, then, multiple SVMs with variations of the training data are trained and tested to identify methods by which ten-second windows of discrete and labelled motion subtypes of upper and lower body motion can be classified.

## 5.1 Lessons from Binary Motion Detection and Literature Review

During the exploratory phase, when developing the motion detection algorithm for baseline *motion* vs. *no_motion* detection, distances travelled by the COP over a sliding window were examined. It was found that using sum distance as a feature for distinguishing *motion* vs. *no_motion* was a valid approach as there was generally a clear threshold between the two. To assess whether sum distance was an appropriate approach for upper and lower body detection in anticipation of future work, summed distances of upper, lower, gross, and *no_motion* segments were plotted on the same histogram to observe differences and similarities. It was found that *no_motion* and upper body motion overlap with each other. Similarly, gross motion events overlap with lower body motion events. In the *motion* data represented an amalgamation of upper, lower, and gross motion events.

In *Chapter 4* it was found that there was a considerable amount of motion data which was classified as *no_motion*, as reflected in the relatively low recall for the motion detection classifier. It is hypothesized that this is because of the dominance of upper body

motion within the motion class, which overlaps with the sum distances of *no_motion* events. To improve the classifier, a simultaneous system must be created to identify the presence of any of the event types present in *Table 12* which displays a breakdown of motion event types in all patients mentioned in this paper.

**Table 12.** Percentage breakdown of motion types within the motion class between patients

| | ***P2*** | ***P6*** | ***P10*** | ***P11*** | ***P13*** | ***P33*** | ***P36*** |
|---|---|---|---|---|---|---|---|
| Upper | 75.20% | 72.43% | 24.02% | 75.95% | 25.56% | 33.84% | 22.52% |
| Lower | 1.51% | 2.99% | 0.72% | 0.32% | 5.30% | 38.02% | 12.67% |
| Gross | 23.30% | 24.58% | 75.26% | 23.72% | 69.14% | 28.13% | 64.81% |

Previous research using the PSM to distinguish between various types of movement episodes has shown that features such as the lateral centre of pressure (LatCOP), longitudinal centre of pressure (LonCOP), number of acute angles (NAA), lateral variance (LatV), longitudinal variance (LonV), and velocity (Ve) are strong features to distinguish between motion subtypes. Specifically with the adult population in bed transition and mobility studies, NAA and LatV were the strongest features identified using an SVM [22], [49]. while other research suggested that LonV is a good feature to distinguish between upper and lower body motion [4], [5], [42].

## 5.2 Developing a Set of Training Features

Each feature is calculated based on COP data within each sliding window of 10 seconds within one annotated episode. These calculations are repeated across each window, in every episode of a motion sub-type, across all seven patients presented in *Table 2*. Each labelled entry in the table is based on results from a 10-second window of COP data in a labelled

movement episode. When looking at classification boundaries for discrete data in *Chapter 4*, summed distances were calculated across sliding windows and added to a single vector where each value was representative of the class from which the window data was collected.

In a previous study, the NAA was defined as the number of acute angles between each set of three consecutive points within a sliding window which are less than 90 degrees. This criterion was successful in previous studies involving adult participants where angles between consecutive COPs that are greater than 90 degrees denoted a motion or movement event, while anything less denoted a no movement event [22]. In each ten-second window, eight angles can be computed. When computing the angles for each ten-second window and labelling it based on the 90-degree rule, it was found that the motion events rarely contained angles greater than 90 degrees, and for this reason there were almost always eight acute angles in every window. We considered that because the research involving this feature was primarily with adults, they would have fewer acute angles in movement events given their physical strength compared to a neonate. For this reason, the definition of an acute angle was adjusted to less than 45 degrees and was computed using eq. (13), where *m* is a movement event at a point in time *t* in seconds, and *x* and *y* are COP coordinates.

$$\boldsymbol{\theta = tan^{-1}\left(\left|\frac{\frac{(y_{t+1}-y_t)}{(x_{t+1}-x_t)}-\frac{(y_{t+2}-y_{t+1})}{(x_{t+2}-x_{t+1})}}{1-\left(\frac{(y_{t+1}-y_t)}{(x_{t+1}-x_t)}\right)\left(\frac{(y_{t+2}-y_{t+1})}{(x_{t+2}-x_{t+1})}\right)}\right|\right) = tan^{-1}\left(\left|\frac{m_{t\rightarrow t+1}-m_{t+1\rightarrow t+2}}{1-(m_{t\rightarrow t+1})(m_{t+1\rightarrow t+2})}\right|\right)} \quad \textbf{( 13 )}$$

The AvLatCOP and the AvLonCOP are the average changes in *x* and *y* values of the COP, respectively, at any given time on the PSM, where lateral movement is side to side (left arm to right arm, *x*) and longitudinal motion is up and down (head to toe, *y*). Given that upper body movement involves vertical stretches, side-to-side wriggles, and arm

movements, it was hypothesized that windows of upper body motion would have larger AvLonCOP and AvLatCOP values as compared to periods of *no_motion* or lower body motion. Lower body movement would have changes in AvLonCOP but lower or consistently low values of AvLatCOP because leg movements are generally up and down, stretching forwards and pulling backwards, as opposed to side to side as per the bed-side observations.

The AvLatCOP and AvLonCOP were calculated for each episode, and a single average was computed separately for each feature over a sliding 10 second window. Similarly, LatV and LonV describe the variance of the *x* and *y* values of the COP within a window. The variance calculation expresses the change or variability of COP values, whose magnitude are expected to differ between upper and lower body motions, gross body motions, and *no_motion*.

Velocity (Ve) – speed – over a window was computed by taking the total distance travelled by the COP in *cm* and dividing it by time, which was 10 seconds.

$$\boldsymbol{d(t)} = dist\big(COP(t+1), COP(t)\big)$$
$$\boldsymbol{Vel} = \frac{\sum_{t=t_1}^{t_1+wLen-1}(\boldsymbol{d_t})}{\boldsymbol{wLen}} \qquad \textbf{( 14 )}$$

As shown in *Chapter 4*, COP movement trails are longer during periods of motion and shorter during periods of *no_motion*. It was also observed that *no_motion* and upper body motion have similar trajectories while lower body and gross motion also share similar trajectories; this suggests that upper body and lower body motion may have very different movement velocities. It is expected that upper body and *no_motion* may be difficult to distinguish, while lower body and gross motions may also be difficult to distinguish.

Therefore, a single full feature table was created with seven features computed for each 10 second window across all event types and labelled: *upper body motion, lower body motion, no_motion, and gross motion.* To summarize, the seven features in this table were: *NAA, AvgLatCOP, AvgLonCOP, LatV, LonV, and Vel* in that order, where the first column was the labels. A total of 27, 673 rows of discrete ten-second event data were created, with four movement event types.

In this chapter, all patient events are evaluated together as opposed to separately since the intention is to incorporate these findings into a generalizable real-time motion detection algorithm that is applicable to all neonates in the NICU. The primary groups of interest for this chapter are upper versus lower body detection. While the algorithm in *Chapter 4* is able to distinguish between *motion* and *no_motion*, we have yet to demonstrate the ability to differentiate between upper vs. lower body movements.

Across all patients, the class imbalance between upper versus lower body events is large; for this reason, this chapter necessarily requires synthetic oversampling of the minority class using a synthetic minority oversampling technique (SMOTE) as described in *Chapter 2*. Since this chapter will use leave-one group out validation (LOOV), baseline training data were created by combining all the feature data for upper and lower body movement across all combinations of three out of four patients.

Patient 11 is not included in this chapter as it has very few lower body motion annotations and none of them are ten seconds in length. Patients 33 and 36 are not included because of the noise and suspected annotation inaccuracy in their signal data as found in *Chapter 4*. The LOOV testing data associated with each group derives from the patient withheld from the training set and contains both upper and lower body testing data.

### 5.2.1 Data Exploration and Feature Selection for UL Models

This section explores feature importance and feature selection for use in upper versus lower body (UL) models. The exploration and selection methods described here include out-of-bag error and sequential forward selection.

#### *5.2.1.1 Out-of-Bag Error (OOB-error)*

Random forest classifiers can be used to identify the most important features in a feature set by evaluating error based on samples not used to train the tree (i.e., out-of-bag error; OOB-error). Each classification decision tree is trained with all features in the training set, and the OOB-error is averaged across all the trees in the forest.

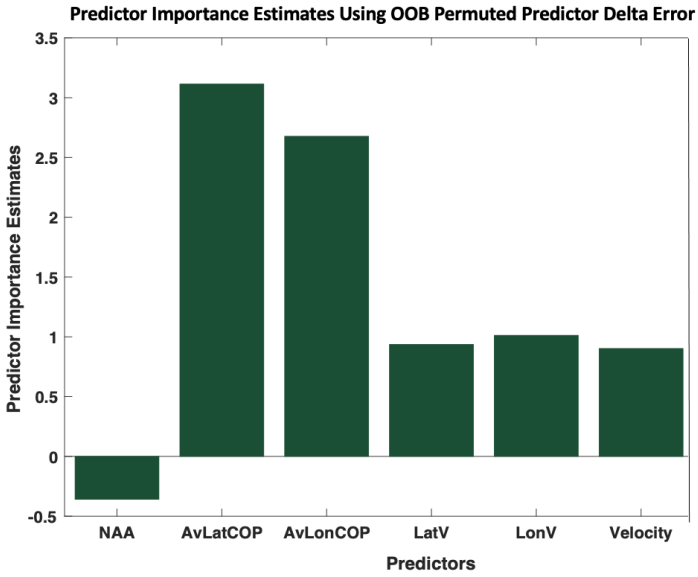



**Figure 16.** Out-of-bag predictor importance for the classification of upper versus lower body movement.

To compute feature importance, the values of a feature are permuted across all samples, the misclassification error is computed, and that error is then compared to the averaged error across all trees computed with the original, non-permuted data. The features that exhibit the largest differences between the original and permuted OOB-error are more

important. In MATLAB, the function used to compute the error is *OOBPermutedPredictorDeltaError* and is used with the *TreeBagger* function.

A simple decision forest model was created using fifty trees and incorporating all feature data across all patients for upper vs. lower body movement and using MATLAB's *OOBPredictorImportance* function to explore the importance of all features in the dataset. The results shown in *Figure 16* suggest that removing AvLatCOP or AvLonCOP from a feature set produces the highest misclassification error, therefore AvLatCOP and AvLonCOP are two of the most important features in distinguishing between upper versus lower body movement. Surprisingly, it appears that NAA has a negative importance, indicating that it is entirely unhelpful to differentiating upper vs. lower body movements, for this reason it is excluded from the training data in further experiments to study feature importance and train SVMs.

#### *5.2.1.2 Sequential Forward Selection (SFS) with SVM*

Sequential feature selection was conducted using MATLAB's s*equentialfs* function. SFS is a bottom-up search strategy that uses forward feature selection to sequentially build a subset of features from the training data that best predict the training labels. This is done by sequentially adding individual features to a classification model, in this case an SVM, to minimize the misclassification error using 5-fold validation until there is no further improvement in prediction [56].

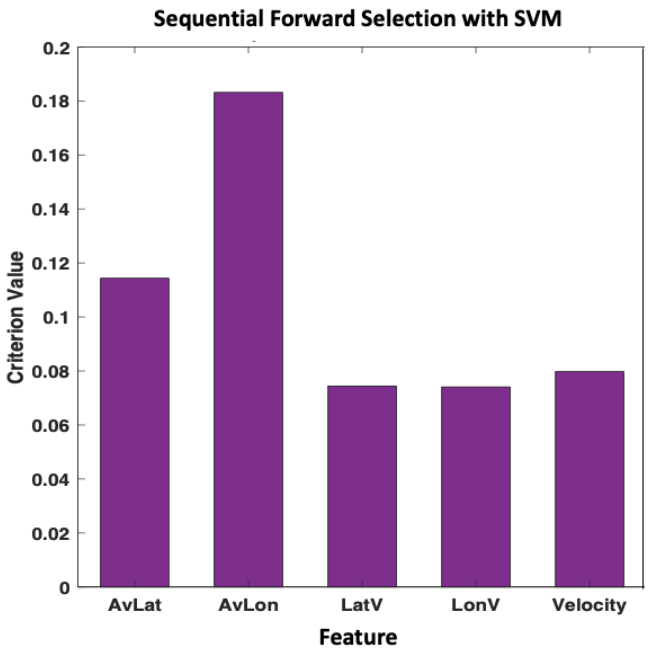



**Figure 17.** Features ordered by criterion values obtained from sequential forward selection with SVM. The redundancy of LatV and LonV are shown. The top four features are: AvLat, AvLon, LatV *or* LonV, and Velocity.

Overall, this analysis demonstrates the relative relevance of each of the features in how well they can distinguish between UL motion and the impact their absence from a feature set makes on the misclassification error. The OOB-error and SFS methods summarize the features that are effective in distinguishing between upper and lower body movement. Given these analyses, the features used in all subsequent models are AvLat, AvLon, LonV, and Velocity. Despite both LonV and LatV having similar behaviour, LonV was selected since upper body versus lower body movement is expected to have stronger variations longitudinally as demonstrated by the LonV importance estimate in *Figure 17.*

## 5.3 SVM for Upper vs. Lower Body Motion Detection

SVM classifiers have been previously shown to be effective when used with features derived from the PSM COP; however, these results have mainly been found in studies involving adult patients; as such, the classification of upper or lower body movement using PSM data collected from neonates has not been previously attempted.

### 5.3.1 Training and Test Data

Patients 2, 6, 10, and 13 were included in training this classifier. Patient 11 had no distinct episodes of lower body motion that were at least 10 seconds long, and Patient 36 and Patient 33's upper versus lower body data were completely unrecognizable when tested with a dummy SVM. Interestingly, Patients 33 and 36 were the only patients in incubators, which may point to a limitation of the upper vs. lower (UL) motion detection system developed here. While Patient 10 had lacklustre performance in the motion detection algorithm in *Chapter 4,* it was included here to maximize dataset size.

*Table 13* breaks down the characteristics of the patients used in this study and the amount of data that was available for use in each movement category. Most of the movement across all patients was upper body motion, however, few motion events were a complete 10 seconds in duration so this limited the number of events that could be included in this study. Furthermore, to calculate features like NAA for the initial feature analysis, at least five seconds of COP data were required, since this would mean a maximum of three acute angles per window. Overall, there is class imbalance between upper and lower body motion, where upper body motion is generally the majority class, except in the case of Patient 10 where it is the minority class. The frequency of movement during periods of monitoring is low; over a two-hour observation window for all patients combined there

were approximately one hour and twenty minutes of upper body movement and approximately 20 minutes of lower body movement.

**Table 13.** Patient characteristics for train/test data in upper vs. lower body motion classifier

| Patient | Bed Type | Weight | # Upper Body (and total time) | # Lower Body (and total time) |
|---|---|---|---|---|
| *2* | Crib | ≥ 2500g | 1056<br>0:17:45 | 294<br>0:05:03 |
| *6* | OVW | ≥ 2500g | 484<br>0:08:14 | 94<br>0:01:43 |
| *10* | Crib | ≥ 2500g | 322<br>0:05:30 | 586<br>0:09:55 |
| *13* | Crib | ≥ 2500g | 2955<br>0:49:15 | 53<br>0:01:02 |
| | | Total Duration | 4817<br>1:20:26 | 1027<br>0:17:17 |

### 5.3.2 SVM Training Architectures

The basic model of the SVM for binary classifier was trained using the *fitcsvm* classification SVM function in MATLAB. The kernel selected for this problem was the radial basis function (rbf), as it is known to be effective for both linearly and non-linearly separable problems. An outlier fraction of 0.30 was set to account for any mixed motion event involving both upper and lower body motion, any no movement event misclassified as movement, any lower body movement misclassified as upper body movement and vice versa, and to minimize other annotation inaccuracy errors overall. Given the data is not separable with a linear hyperplane, a soft margin was used that would separate many but not all datapoints. To solve the quadratic programming problem that arises with the soft margin, a sequential minimal optimization solver (SMO) was selected. Auto hyperparameter optimization was done with *bayesopt* in MATLAB for each patient separately during leave-one-subject-out-validation (LOOCV), which iterates to maximize

the acquisition function – *expected-improvement-plus* – and minimize the objective function. The model also standardizes the training feature data before training the classifier using their weighted means and standard deviations.

Leave-one-subject-out-validation (LOSOV) in this application is the process of training a model using all patients in a training set but one, where the data of the withheld patient is used as a test set. After the model has been trained with all combinations of train and test patients, the averaged performance metrics of the classifier are computed, and the classifier is finally trained on all the patients in the train set.

To solve this problem, the following models were trained and tested based on lower body motion being the positive class:

- SVM 1: *Class Imbalanced*
- SVM 2: *SMOTE applied to training data only*
- *SVM 2.1: SMOTE applied to both Test and Train*
- *SVM 2.2: SMOTE on Train and Majority Undersample on Test*
- SVM 3: *20/50 Test to Train, SMOTE on Train, Imbalanced Test*
- SVM 3.1: *20/50 Test to Train, SMOTE on Train, Majority Undersample on Test*

#### *5.3.2.1 SVM 1 - Class Imbalanced Model*

To evaluate data quality, separability of data, and the ability of the SVM to classify naturally imbalanced data, a raw model was trained with features AvLat, AvLon, LonV, and Velocity. Across all training sets of three patients each, training data had between 1081 – 2127 samples of upper body motion, and 710 – 1136 samples of lower body motion, and the class imbalance ratio of *upper:lower* in testing data ranged between 0.19 – 0.55, where only one patient had a class imbalance ratio greater than 0.50. *Figure 18.* presents a

flowchart with the feature selection process, the model parameters, and the training and testing system.

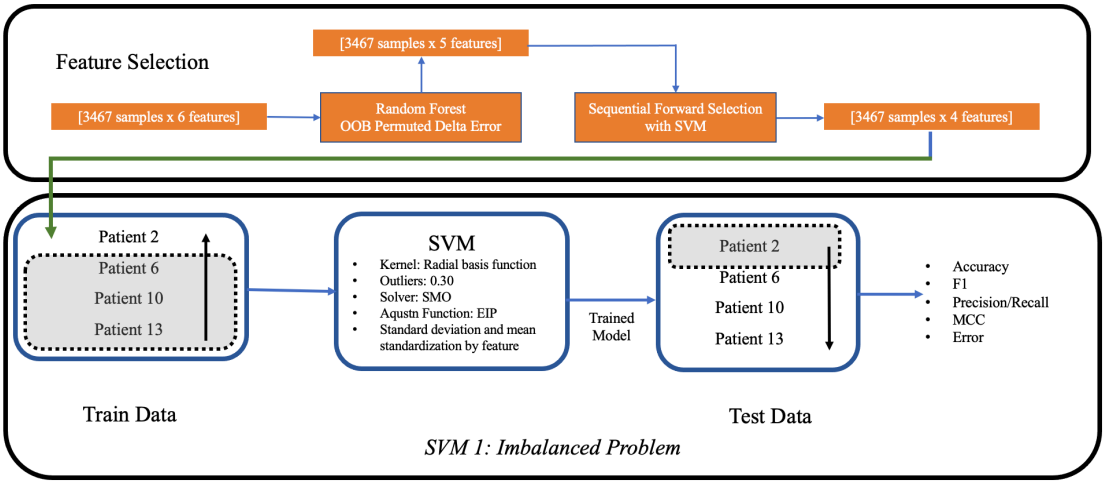



**Figure 18.** Flowchart for model trained with imbalanced train and test data

The best features selected from the feature selection pipeline were AvLat, AvLon, LonV, and Velocity. To visualize the SVM hyperplane, pairs of features are visualized in *Figure 21.* While the model performed well on the training data, they failed on the novel test data because the distributions and location of test points were vastly different from those of the train. Test results for each withheld patient in this model are shown in *Table 14*.

**Table 14.** SVM-1 Imbalanced Model Results

| Model | LOO-Patient | Acc | Prec. | Rec. | F1 |
|---|---|---|---|---|---|
| SVM 1: *Imbalanced Test and Train* | 2 | 0.23 | 0.23 | 1.00 | 0.37 |
| | 6 | 0.44 | 0.20 | 0.79 | 0.32 |
| | 10 | 0.49 | 0.74 | 0.50 | 0.59 |
| | 13 | 0.29 | 0.23 | 0.50 | 0.31 |
| | **Overall** | **0.36** | **0.35** | **0.70** | **0.40** |

SVMs and machine learning classifiers in general have the most unbiased performance when the training data contains the least class imbalance. A sample set of distributions are

shown for the test results of Patient 2 in *Figure 19,* and the ROC curve for overall test performance is shown in *Figure 20.*

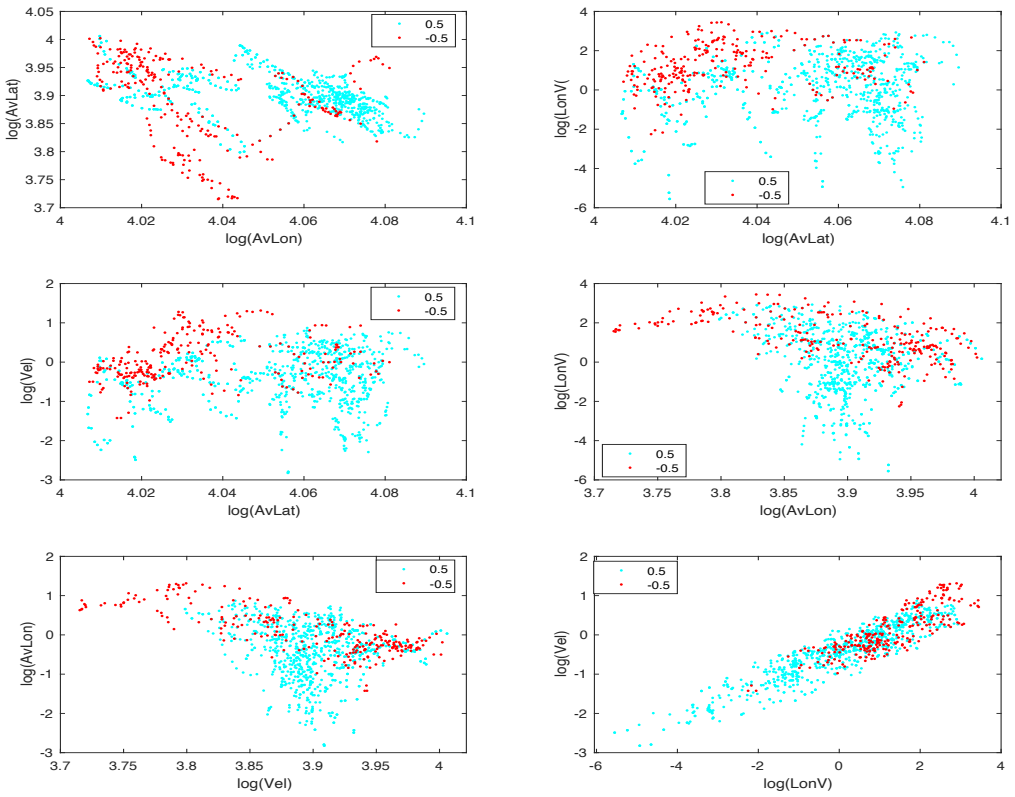



**Figure 19.** Class separation across features with test Patient 2. Lower body motion is more widely distributed and less clustered as in the train data.

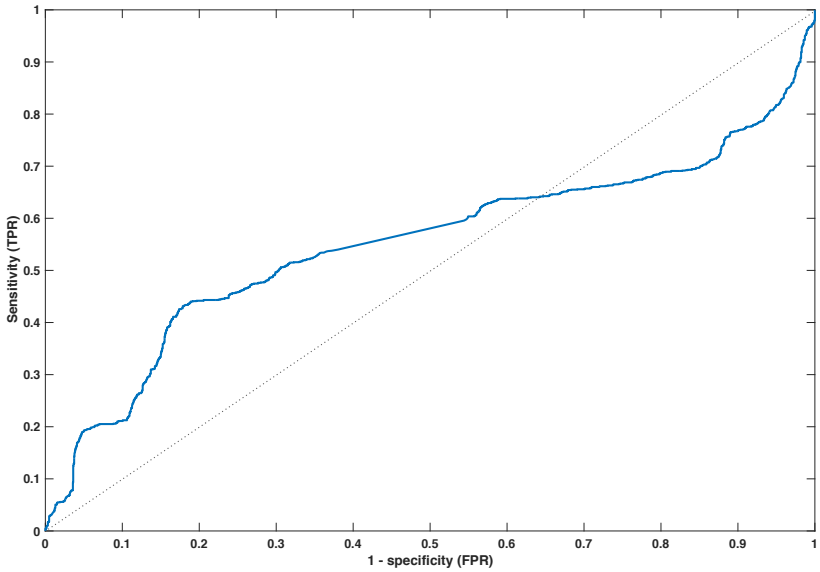



**Figure 20.** Patient ROC for overall test performance of imbalanced train data on "SVM1".

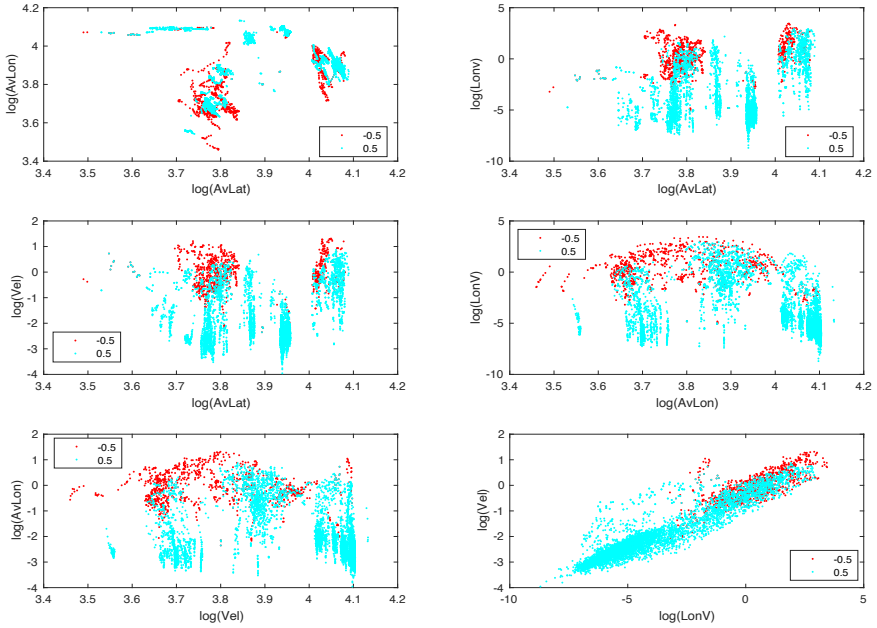



**Figure 21.** Class separation with six different binary combinations of the four selected features in the SVM. The best separation appears with LonV and Velocity, and the poorest appears with AvLat and AvLon.

The results in *Table 14* show a much higher recall than precision overall – recalling that lower body is the positive class, this means that most lower body motion is being correctly detected, however many upper body motions are being incorrectly classified as lower body, leading to many false positives. Precision is very low, which suggests that upper body motion is being misclassified as lower body motion, leading to a higher false positive rate. The ROC curve in *Figure 20* falls both above and below the line where AUC = 0.50. This behaviour shows that at all or most thresholds used to plot the ROC, the TPR and FPR are close in value, so a high recall/sensitivity suggests a low precision due to the high FPR.

#### *5.3.2.2 SVM 2, SVM 2.1, SVM 2.2 - Class Balanced Models*

In this section, the impact of the class imbalance issue from *Section 5.3.2.1* is explored and the model is improved with SMOTE and majority under-sampling. The flowcharts for these models are shown below in *Figure 22*. Three methods of balancing the model were examined: SVM2 - applying synthetic minority oversampling (SMOTE) which uses 5-NN to grow the training set, SVM 2.1 – applying SMOTE to both the train and test sets, and SVM 2.2 - using majority under-sampling on the test set to balance classes. The internal structure of the SVM remained unchanged. Since feature selection was originally conducted with an unbalanced train set, feature selection was redone every time the model is retrained in the absence of a leave-out patient; the optimal features were consistently the same as when the model had class imbalance in *Section 5.3.2.1*. Once the model was trained with SMOTE for all leave-out patients, with the test sets still unbalanced, SMOTE (SVM 2.1) and under-sampling (SVM 2.2) was applied to the test data and re-tested on the SMOTE trained SVM (SVM 2), as shown by the shaded portion of panels two and three in *Figure 22*.

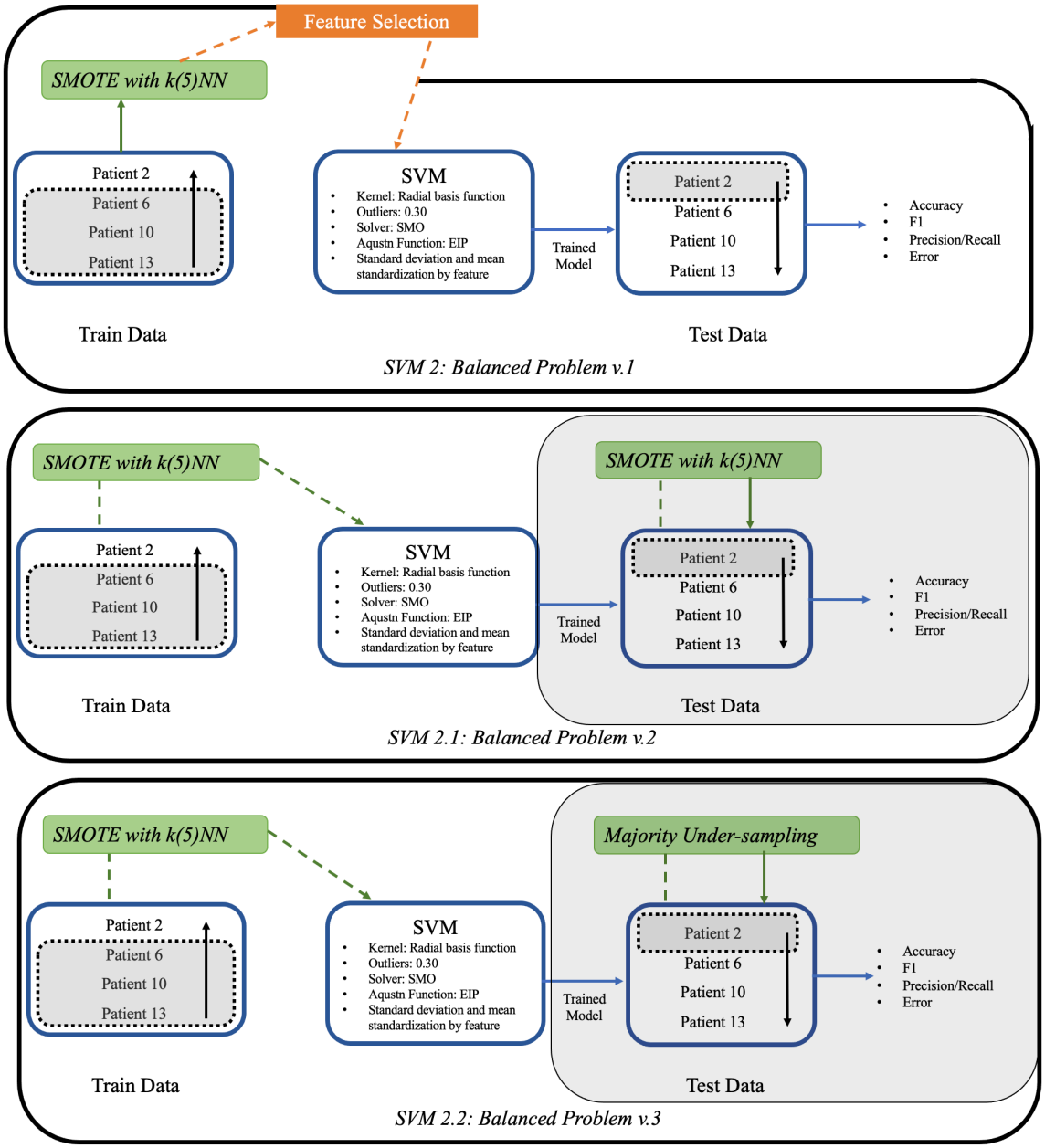



**Figure 22.** SMOTE on train (top), SMOTE on test and train (middle), SMOTE on train and majority under sampling on test (bottom).

The performance metrics for SVM 2: SMOTE on Train where the testing data are unbalanced, are presented below in *Table 15*.

**Table 15.** Test and overall metrics of balanced SVM classifiers with imbalanced test data

| Model | LOO-Patient | Acc | Prec. | Rec. | F1 |
|---|---|---|---|---|---|
| SVM 2: *SMOTE on Train, Imbalanced Test* | 2 | 0.24 | 0.23 | 1.00 | 0.37 |
| | 6 | 0.33 | 0.17 | 0.80 | 0.28 |
| | 10 | 0.57 | 0.72 | 0.70 | 0.71 |
| | 13 | 0.34 | 0.33 | 0.49 | 0.01 |
| | **Overall** | **0.37** | **0.36** | **0.75** | **0.34** |

The performance for Patient 10 increased the most, increasing recall by 0.20, with precision unchanged. Patient 10 was swaddled but not covered, so lower body movement was easier to annotate by clear observation or observational inference. Adding SMOTE to the training set also improved the decision boundaries across feature sets since more data was added to the training dataset to represent lower body motion. This may also be why the classification accuracy of Patient 10 and 13 improved and recall for Patient 6 improved. The training data being SMOTE-enhanced, while not entirely made up of real patient data, paints a balanced picture of the two classes and the test results are effectively more representative of the classifier's unbiased ability to distinguish between upper and lower body movement.

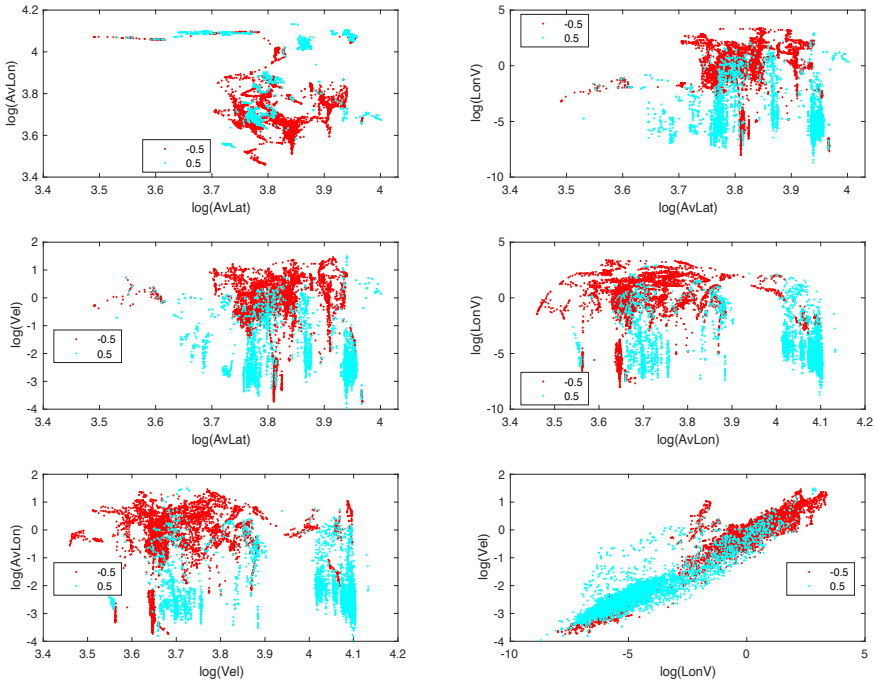



**Figure 23.** Class distribution across features after SMOTE applied to training data.

In *Figure 24,* the ROC for test performance on SVM 2 is less optimistic than test performance with SVM 1, but this may be because of the class imbalance on the overall dataset. The optimal threshold is at an FPR of 0.04 and a TPR of 0.13 to maximize the precision and the recall both. Scaling the threshold anywhere further along the ROC curve would compromise the precision which is already very low at the existing threshold. Test performance and metrics can be skewed if the test set is unbalanced while the train data is balanced. To test this assumption, SMOTE and majority under-sampling techniques were applied to the testing data and used as input for the SMOTE trained classifier.

With the application of the aforementioned techniques and as shown in the results in *Table 16* below, there is a large performance improvement in classification accuracy, the overall precision, and the recall.

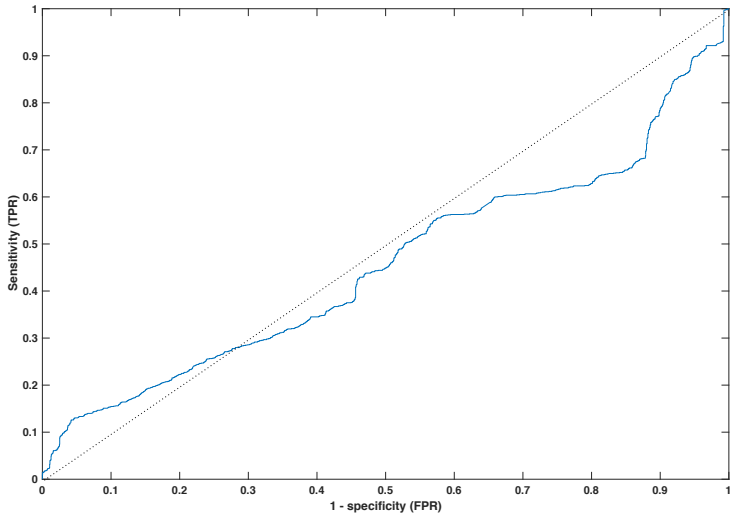



**Figure 24.** ROC for test performance on SMOTE trained SVM (SVM 2).

**Table 16.** Test and overall metrics of balanced SVM classifiers with balanced test data

| Model | LOO-Patient | Acc | Prec. | Rec. | F1 |
|---|---|---|---|---|---|
| SVM 2.1: *SMOTE on Test and Train* | 2 | 0.50 | 0.50 | 1.00 | 0.67 |
| | 6 | 0.53 | 0.52 | 0.83 | 0.64 |
| | 10 | 0.42 | 0.45 | 0.70 | 0.55 |
| | 13 | 0.50 | 0.50 | 0.95 | 0.65 |
| | **Overall** | **0.49** | **0.49** | **0.87** | **0.63** |
| SVM 2.2: *SMOTE on Train, Majority Undersample on Test* | 2 | 0.50 | 0.50 | 1.00 | 0.67 |
| | 6 | 0.56 | 0.54 | 0.80 | 0.63 |
| | 10 | 0.42 | 0.45 | 0.69 | 0.55 |
| | 13 | 0.49 | 0.45 | 0.94 | 0.65 |
| | **Overall** | **0.49** | **0.49** | **0.86** | **0.63** |

Furthermore, the F1-score also improves two-fold when SMOTE is applied to Test and Train versus when Test data was imbalanced without SMOTE, however its sensitivity to class imbalance is a large contributor to this outcome. The accuracy slightly increases when majority under-sampling is applied, and the recall decreases slightly with no impact on precision. The fairest representation of the classifier performance on real test data is with the majority under-sampled dataset because it is balanced and contains only real data as opposed to synthetic data. The limitation of the under-sampled testing is that the diversity

of the test set is reduced. In all, balancing the test data against a balanced classifier produced improved metrics, but did not bring accuracy up to an acceptable standard, which should be at minimum 70%.

#### *5.3.2.3* SVM 3 – Leaky Group-Fold Cross Validation with Sample Imputation

The results with SVM 1 and SVM 2 showed some improvements through attempts to address class imbalance in the training and testing datasets. However, despite these improvements, the overall accuracy and F1-scores have not reached at least 70% and the precision remains below 0.50.

Selzler *et al.,* (2021), in their work on anxiety level classification from physiologic data, applied a leaky group-fold cross-validation with sample imputation, where limited subject-specific test data are included in the training set to determine how much data are required to be manually annotated and imputed into the training data before the classifier can be deemed useful for the rest of a patient's data [57]. This same concept was explored here for movement classification in neonatal patients.

Given the training time required by SVM classifiers in this application, only a limited exploration was conducted in this thesis. 50% of the existing test data for the minority class (lower body) and 20% of the majority class (upper body) are removed from the training set and incorporated into the train set; the remaining test data are preserved for testing. 50% was chosen because the class imbalance between upper and lower body is large for some patients, such as Patient 6 who only has 61 samples of lower body data, corresponding to approximately one minute of annotated movement. The distribution of event durations per patient is shown in *Table 12* and *Table 13.* For the majority class, only 20% of the patient's data were included since the diversity of upper body motion data is greater in the train set. The models in *Figure 25* illustrate these testing variations and imputation processes.

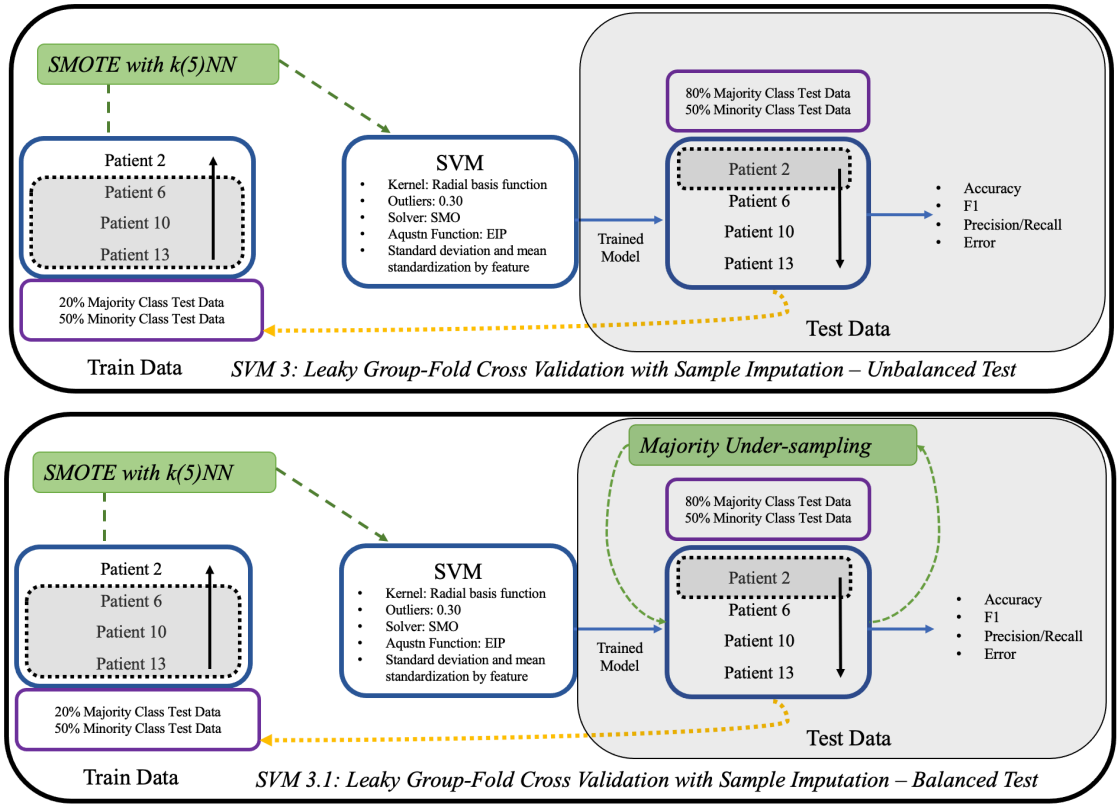



**Figure 25.** Leaky group-fold cross validation with sample imputed in balanced and imbalanced test data experiments.

In one set of results, SMOTE was applied to the training data after imputing some test data into the train set, the same features were extracted as determined in the previous feature selection steps, and the test data was left imbalanced. In another experiment, a trained classifier with imputed test data was used to test on majority under sampled test data. The amount of patient data that was added in minutes is shown in *Table 17* for each class, the results of these experiments are shown in *Table 18*.

**Table 17.** Amount of data added to train set from each patient for SVM 3 and SVM 3.1

| | Patients | | | | Avg Time |
|---|---|---|---|---|---|
| | **2** | **6** | **10** | **13** | |
| **Upper Body** | 9 mins | 4 mins | 3 mins | 24 mins | ~ 10 mins |
| **Lower Body** | 3 mins | 1 min | 5 mins | 31 secs | ~ 3mins |
| **Total** | 12 mins | 5 mins | 8 mins | 24.5 mins | 13 mins |

**Table 18.** Test results of SVMs with imputed subject data in training set

| Model | LOO-Patient | Acc | Prec. | Rec. | MCC | F1 |
|---|---|---|---|---|---|---|
| SVM 3: *Leaky Group-Fold Cross Validation with Sample Imputation – Imbalanced Test* | 2 | 0.81 | 0.45 | 0.87 | 0.53 | 0.59 |
| | 6 | 0.75 | 0.23 | 0.77 | 0.36 | 0.41 |
| | 10 | 0.90 | 0.89 | 0.97 | 0.78 | 0.93 |
| | 13 | 0.98 | 0.94 | 1.00 | 0.96 | 0.97 |
| | **Overall** | **0.86** | **0.63** | **0.90** | **0.66** | **0.73** |
| SVM 3.1: *Leaky Group-Fold Cross Validation with Sample Imputation – Balanced Test* | 2 | 0.78 | 0.78 | 0.71 | 0.52 | 0.75 |
| | 6 | 0.81 | 0.85 | 0.74 | 0.71 | 0.79 |
| | 10 | 0.93 | 0.96 | 0.91 | 0.86 | 0.93 |
| | 13 | 0.99 | 0.99 | 1.00 | 0.99 | 0.99 |
| | **Overall** | **0.87** | **0.90** | **0.84** | **0.75** | **0.87** |

The results in *Table 18* show that adding a percentage of test patient data into the train set can improve overall classification of upper versus lower body motion. The last column of *Table 17* shows us that even though 50 and 20 percent of the test data were added, only an average of 10 minutes of upper body data across patients and 3 minutes of lower body data across patients were added to achieve the results in *Table 18*. The performance overall across all patients can be visualized with the ROC curve in *Figure 26*. Interestingly, more upper body data needed to be added into the classifier than lower body data to improve metrics even though lower body data was the minority class across three out of four

patients; this resulted in fewer false positives and higher precision scores, which in turn impacted the accuracy, MCC, and F1-scores.

These results indicate that including a small amount of subject-specific data in the training set of a UL motion classification model dramatically increases classification accuracy. The next section describes how such an approach could be implemented in a clinical setting.

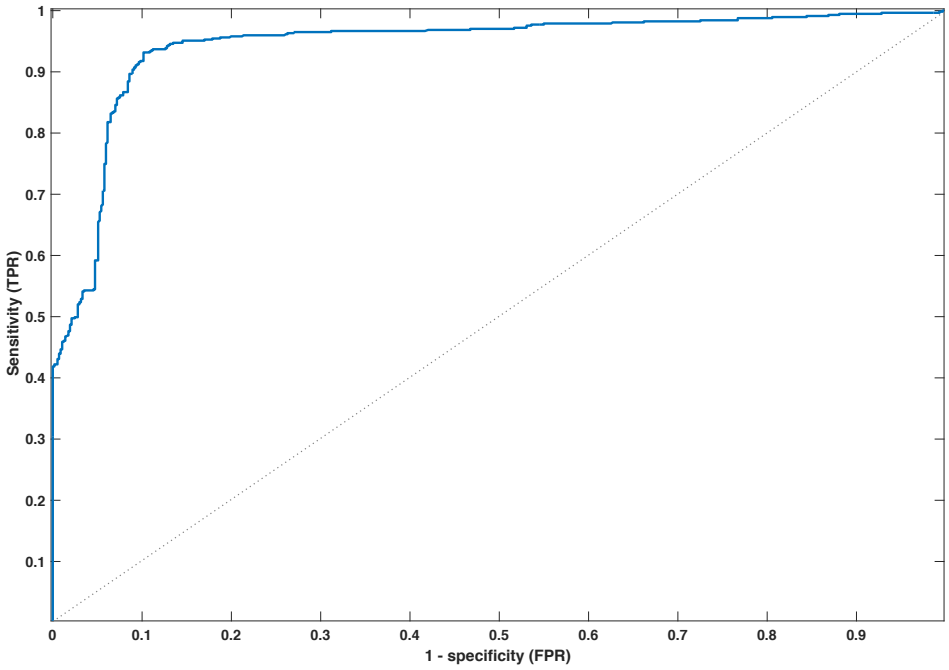



**Figure 26.** ROC curve for the leaky SVM imbalanced model with subject data imputation for all patients.

### 5.3.3 Integration of UL Classification into Real-Time Motion Detection Algorithm

In a real-time model application, this upper and lower motion detection algorithm would operate in conjunction with the baseline *motion* versus *no_motion* detection algorithm described in *Chapter 4*. However, given the most successful model explored in

this chapter required the imputation of at least 10 minutes of upper body motion data and three minutes of lower body motion data per patient, a real-time application of this model would need to be retrained with ~13 minutes of hand-annotated upper and lower body motion data to calibrate the algorithm for each new patient. For this, nurses or parents would require a user-friendly interface containing the monitoring system to capture various movement types and feed them into the model. The added benefits of the retraining piece is that the calibration data would augment the diversity of training data, improve the robustness of the patient types that can be monitored successfully, and improve overall reporting metrics over time. The retraining would impact the upper versus lower body motion detection algorithm and would require up to an hour and a half of training time. When the classifier is fully trained, a ten second segment of motion data can be classified instantaneously with no time delay.

For each episode classified as *motion* with the baseline algorithm, the UL motion classification algorithm would be applied on that ten second window to further refine whether the detected motion was upper or lower body. In the case of gross motion, both upper and lower body movement is occurring as opposed to upper or lower body motion independently. Gross body motion – that is a combination of upper and lower body motion – is also a common movement event that can take place; however, this was not included in the binary upper-lower motion classification model since this model focuses on detecting motion directionality; as such only upper-body-exclusive and lower-body-exclusive motion events were included in the training data. However, if this algorithm were applied on a gross motion event, the pattern of movement would more closely associate with lower body motion and would suggest the silencing of a false alarm. To experiment, during the exploratory analysis phase of baseline motion detection, the distributions of the distances

travelled by the COP during periods of gross and lower body motion entirely overlapped, while gross body motion was entirely separated from upper body motion. This has been empirically suggested in past research as well; in their work on limb detection based on PSM movement, Adami *et al.,* [5] found that when an event included lower and upper body movement, the impact of upper body movement on the centre-of-mass was not substantial leading to a classification 'error' that would classify the event as lower body motion. Gross body being labelled as lower body motion is not detrimental to the algorithm's usefulness since the leg movement is a source for false alarms in the NICU owing to the positioning of the $SpO_2$ sensor.

## 5.4 Summary of Results and Conclusion

For sufficient classification of upper and lower body motion from PSM data collected in the NICU and based on the results of experiments conducted on four crib-based infants weighing ≥ 2500 g, a maximum of thirty minutes of data may be required to be manually imputed into the training data of a classifier and retrained in preparation for continuous monitoring of a new patient. On average, ten minutes of upper body data and three minutes of lower body data for a swaddled, infant weighing at least 2500g in a crib are required to be able to effectively detect lower body motion from upper body motion. In the first experiment, an SVM was trained using leave-one-patient-out cross-validation with imbalanced test sets and imbalanced train sets; the performance was indubitably poor because classification models like SVMs and Random Forests perform best with balanced train data. With a balanced train set and an unbalanced test set, the test performance improved only marginally, with most improvement seen in recall; this meant the classifier was predicting most events as lower body motion and was insensitive to upper body events.

Then, balancing the test data with SMOTE and majority under-sampling approaches improved the test metrics and presented a precision closer to 0.50 and further improved recall scores. This, however, did not solve the problem of the model being hypersensitive to upper body motion despite the improvements in the metrics. For this reason, an approach that would require manual addition of subject-specific data for upper and lower body motion was examined and shown to produce much higher performance accuracies with both imbalanced and under-sampled test sets. Four of the most useful features for distinguishing between upper and lower body motion were average lateral COP (AvLat), average longitudinal COP (AvLon), velocity (Vel), and longitudinal COP variance (LonV). AvLat and AvLon were shown to be most sensitive to upper body motion, while Vel and LonV were of the features most sensitive to lower body motion; together these features worked together in the SVM to produce hyperplanes that would separate upper and lower body data in a test set.

Because manually annotated subject data is required for successful performance of this algorithm, the classifier cannot be said to be generalizable with new patients. The cause for these discrepancies may include mislabelled motion annotations due to the mismatch between time stamps in the PSM data versus the video data which required manual revision of annotated time stamps. It was also found that the data import algorithm which was used to transform data from the .csv file produced by XSensor Pro to .mat files may have dropped frames; however, these frames were dropped because they were damaged, noisy, or unreadable because of NaN signals or errors with the technology. The data processing model accounts for missing frames, but in the process of extracting data between time intervals, the missing frames may have skewed the overall accuracy of the annotation.

In these experiments, summarized in *Table 19*, the models with no data imputation did not work well, but when new patient data was imputed into the training data before training the SVM, the performance significantly improved as compared to SVM 1.

**Table 19.** Summary of SVM classifier performances for UL motion detection

| **Model** | **Acc** | **Prec.** | **Rec.** | **F1** |
|---|---|---|---|---|
| *SVM 1* | 36.17 | 0.35 | 0.70 | 0.40 |
| *SVM 2* | 36.85 | 0.36 | 0.75 | 0.34 |
| *SVM 2.1* | 48.72 | 0.49 | 0.87 | 0.63 |
| *SVM 2.2* | 49.37 | 0.49 | 0.86 | 0.50 |
| *SVM 3* | 86.33 | 0.63 | 0.90 | 0.73 |
| *SVM 3.1* | 87.29 | 0.90 | 0.84 | 0.87 |

# Chapter 6

# Discussion and Conclusions

This chapter details a discussion of all ideas and findings this thesis has explored and analyzes the benefits and limitations of this research; furthermore, this chapter also details and justifies items for future work.

## 6.1 Discussion and Conclusions

This thesis used data science methods on primary ballistographic signal data from a PSM in conjunction with RGB-D video data collected from seven neonates at the Children's Hospital of Eastern Ontario (CHEO) in Ottawa, Ontario to detect baseline motion, upper, and lower body motion. The primary objective of this work was to develop working algorithms that would be able to distinguish motion activity from periods of *no_motion* to gate false alarms from monitoring modalities that are impacted by motion-based noise artifacts. False alarms cause unnecessary and frequent action by nurses as well as an overload of alarm sounding impacting the mental and physical health and agility of nurses overall. The secondary objective of this thesis was to break motion classifications into their subtypes – upper or lower body motion, with lower body being the region of interest. Lower body was preferred as the positive class since the $SpO_2$ sensor is typically wrapped around the foot of an infant given the thinness of the skin, and is notoriously known for false alarms in the NICU [58], [59].

In *Chapter 4,* a baseline motion detection algorithm was developed that evaluates the sum distance travelled by the COP over a ten second window and classifies the event as motion or *no_motion* based on a given threshold of sum movement. The model was

validated using a leave-one-subject-out validation model where the optimal threshold for each patient was evaluated across all other patients. The final model achieved a precision 0.81 and recall of 0.77 across all patients. When the parameters were tested with a real-time sliding window algorithm, a precision of 0.85 and recall of 0.68 were achieved. The limitation of the model was that its inability to distinguish between *motion* versus *no_motion* in situations where patients had an abundance of upper body motion events that were very subtle and whose COP trajectories exhibited low amplitudes, as such they were similar to *no_motion* events. Other patients for which the model had high error were those who were partially covered and swaddled *or* swaddled with multiple layers of blankets that created occlusion during the data collection and annotation process.

Furthermore, as it pertains to annotations and environmental conditions at the time of monitoring; whether an infant was covered with a blanket could have impacted the annotation quality as well. If an infant was covered across the incubator or crib with a blanket, minor movements of the feet or the hands could not be seen on video or in real-time. Upper body motion could only be identified if the head was moving, if the arms were moving out the sides of the blanket, or if the arm movement could be seen poking through the blanket. Similarly, lower body motion could only be identified if leg movement could be seen poking from under the blanket or if the legs were seen moving out the sides of the blanket. The movements under the blanket that were not captured within the frame rate of the video and in real-time monitoring were picked up on the PSM. This may have led to motion events being incorrectly misclassified as no motion events in the annotations, and correctly as motion events in the model, yet still skewing the metrics since it is technically a misclassification by the expected label.

An algorithm that can detects lower movement can not only contribute to gating an $SpO_2$ false alarm but may also form the basis of clinical intervention detection such as diaper changes where movement and activity is largely in the lower body region. The problem of distinguishing between upper versus lower body motion was attempted in *Chapter 5* using SVMs as classifiers, and random forests and sequential forward selection (SFS) with SVMs as feature selection and extraction methods. Of the six models used to improve the precision and recall in detecting lower body motion from upper body motion, balancing the test and train sets with SMOTE or majority under sampling, and applying sample imputation into the training data significantly improved classifier performance from the first raw model that used an imbalance dataset for training and testing. The latter approach would require collecting a few minutes of hand-annotated motion data for a new patient prior to training a patient-specific UL motion classifier.

A note to make about the upper body motion classifications is that annotations were not discriminated based on the magnitude of movements. A slight movement of an arm is classified as upper body motion, and so is a large movement of both the arms and head. A frequent misclassification that occurred was upper body motion being misclassified as lower body motion; it is possible that the classifier had learned lower body motion well, but it is also possible that the magnitude and type of upper body motion events varied across patients.

Lastly, this work prioritized the SVM as it has had the most success in literature for motion detection, however it suffers from a scalability problem where memory use and computational time are both very large especially when kernels are nonlinear and datasets are large. Since the most successful method of upper versus lower body detection involved an SVM that would need to be retrained for each new patient, it would require up to a full

day to train the model before it is useful to the patient. To improve scalability, a parallel SVM algorithm may be implemented that reduces memory use by loading only essential data to each machine to perform parallel computation [60]. Transfer learning is when a model is partially trained with a subset of data and completely trained with new data before running a set of predictions; transfer learning has been scarcely explored in literature with SVMs but a deep learning model such as a long-short-term-memory network may be able to achieve this and should be explored.

While the research on baseline motion detection with neonates in the NICU has been explored in the past, the foundations of building a classifier that can distinguish between upper versus lower body motion had not been investigated prior to this research. In past research it was found that the number of acute angles was a significant feature for distinguishing between upper and lower body motion [37]–[39]. In this work, redefining and adjusting this feature to accommodate smaller movements of neonates did not demonstrate value for motion directionality classification. Additionally, the pressure sensor mat was only used with the adult and geriatric populations, while thin pressure sensor arrays were used for pose and motion detection for neonates as opposed to motion directionality classification. Furthermore, unlike past research with neonates, this research was conducted in natural environmental conditions of the NICU, the infants and their environments were not manipulated for data collection purposes [23]. Lastly, the subjects in this study were both covered and uncovered; past studies with either infant and adult populations have required subjects to be uncovered for study purposes involving pose detection, transition detection, and motion detection – the ability to detect motion with partially covered patients, and empirically investigate barriers to motion detection with covered patients is a novel component of this work.

The insights and methods in this work can be used to create a multi-class classification model with layered class validation algorithms to improve accuracy, precision, and recall metrics across classes, improve false alarm gating, tackle the problem of nurse alarm fatigue, and collect insights towards conditions with motion-related patterns such as restless leg syndrome and epilepsy that may not otherwise be caught.

A factor which may have influenced the performance of the unimputed SVMs is the cost matrix which was set to the default where Cost($i,j$) = 1 if $i$~= $j$, and Cost($i,j$) = 0 if $i$ == $j$, where $i$ is the actual class and $j$ is the predicted class; that is the cost of misclassifying lower and upper body motion were equally costly.

## 6.2 Future Work

- Baseline Motion Detection
    - The patients used for studying baseline motion detection were patients 2 (covered with blanket), 6 (covered with blanket), 10 (swaddled), 11 (covered with blanket), 13 (mixed coverage), 33 (swaddled), and 36 (uncovered); the patients with which motion detection was most effective was with patients 2, 6, and 13 out of which only patient 13 was swaddled for a portion of the recording time. The performance of swaddled patient 10 and 33 was generally poor and this may be due to lower amplitude pressure signals. It would be useful to study the efficacy of classifier performance on each coverage type with a larger subset of patients to identify ways to optimize the model for various types of environmental conditions, such as swaddle-type or the lack thereof.

- Collect annotated data from those patients who had no blankets covering them or swaddling them and whose movements are clear in the video.
    - *Justification:* While specifically using patient's data who are uncovered could reduce the diversity of the data and possibly impact the degree of generalizability, the clarity of patient movement and the lack of reason to make observational inferences may improve the truth value in determining whether the patient is surely moving or not moving even when covered or under a blanket. This motion detection algorithm produced classifications on new data in less than a second, suggesting that it could be deployed in real-time.
    - *Justification:* It should also be noted that swaddling could impact signal strength similar to how infant weight could impact signal strength. An experiment could consider testing this algorithm separately on a weight-based study, swaddle-based, or visibility-based study.
- Consider the use of other deep learning models, such as a convolutional neural network (CNN) combined with a long-short-term memory model (LSTM), that can take event-segmented frames of pressure maps from a labelled time-series of PSM data to distinguish between motion and no motion events.
    - CNN-LSTM – this model can be used for supervised machine learning to apply classifications to sequences of images (videos). In this case, segments of moving pressure map data could be re-

processed into low-definition video data to optimize processing time and be used for classification in real-time. The CNN component is used for feature extraction while the LSTM would be used for sequence classification.

- Upper versus Lower Body Motion Detection
    - Expand feature-set using feature engineering methods such as cross-feature multiplication prior to conducting feature selection.
        - *Justification:* The features in this study were selected from studies that examined their performance in detecting movement on a PSM with young adult and geriatric populations; since these features and methods had not been used in a neonatal application before, this problem was attempted using field knowledge.
    - Consider optical flow methods with pressure map images by isolating centre points on the upper and lower body as opposed to the COP alone.
        - Other studies have used features that compute the number of active sensors at a given period to evaluate the magnitude of movement given the weighted sum of active sensors.
        - The process of extracting pressure maps, segmenting the images by frames so that it crops only the ROI, and then computing these features would be a time-consuming training process and possibly inefficient. Nevertheless, it would be interesting to see how non-COP approaches would fare for the problem of upper versus lower body detection.

        - A good model for this process could combine strong image classification models such as the CNN. This method is computationally expensive if models are not optimized, so optimization methods and data quality improvement methods must be explored first.
        - Digital signal processing approaches have been explored in the past to study movement detection and classification with the neonatal population using PSMs, however, the distinction between upper versus lower body motion using unique patterns from extracted BSG signals have not yet been attempted in literature for the neonatal population.
    - Optimize the SVM classifier training time by using parallel computing methods or a transfer learning approach with a new model architecture
    - Separate upper body motion into mixed upper body motion and segmented upper body motion to improve quality of motion detection between.
    - Collect annotated data from those patients who had no blankets covering them and whose movements are clear in the video; this will require further validation between the PSM and video time stamps and a revision of the PSM to MATLAB import script to account for dropped or missing frames.
    - One of the major problems with motion detection in *Chapter 4* was the suspicion that upper body motion was being misclassified as *no_motion* and vice versa because of the overlap observed with their sum distances.

- Develop a user-interface for nurses, other care staff members, and parents of patients to be able to collect calibration data for the monitoring system prior to its use with new patients
    - The current Temporal Event Annotation (TEA) app is specific to research applications and is more complex in that it contains buttons for multiple event types. This app should only have four pages: calibrate, train model, begin monitoring, and monitoring history. The calibration page should turn the PSM on and have the buttons: ‘upper body motion’, ‘lower body motion’, ‘gross motion’, whereas if nothing is selected, the system records the time as ‘no motion’. Each button should be pressed twice – once to mark the start and once to mark the end of the event.
    - The time stamps should be stored in .xlsx workbooks with different tabs for each event-type as opposed to JSON files; this is how the event data is extracted from the continuous PSM signals to label and train the model.
    - The data pre-processing piece involved consumed XSensor data using a Python script, converted it into MAT-usable files, etc. as described in Chapter 3. This process should be optimized for real-time use so that the data can be prepared to re-train the SVM which would already be a time expensive process.

## 6.3 Summary of Contributions

- Supported the process of collecting observational patient data at CHEO for the latter third of the data collection phase.
- Developed a baseline real-time motion detection algorithm that is effective across patients weighing at least 1500g at the time of measurement and laying in a crib, overhead warmer, or incubator.
- Developed a working model for lower versus upper body motion detection where lower body motion is the target class. Determined limitations within the data and raised magnitude of movement as a misclassification factor within upper body motion detection. Demonstrated that inclusion of minimal patient-specific calibration data substantially improved upper vs. lower body motion classification.
- Published a conference paper with Medical Methods and Applications (MeMeA) Symposium 2020 and won Best Student Paper Award for my work in using a PSM to distinguish between motion and *no_motion.*

# References

[1] V. Sekar, K. Manikandan, and N. Kumar, “Novel fall detection algorithm for the elderly people,” *2014 Int. Conf. Sci. Eng. Manag. Res. ICSEMR 2014*, pp. 2–4, 2014.

[2] D. R. Lowne and M. Tarler, “Designing a low-cost mattress sensor for automated body position classification,” *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, vol. 7 VOLS, pp. 6437–6440, 2005.

[3] M. H. Jones, R. Goubran, and F. Knoefel, “Identifying movement onset times for a bed-based pressure sensor array,” *IEEE Int. Work. Med. Meas. Appl. MeMeA 2006*, vol. 2006, no. April, pp. 111–114, 2006.

[4] A. M. Adami, A. G. Adami, T. L. Hayes, and Z. T. Beattie, “A system for assessment of limb movements in sleep,” *2013 IEEE 15th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2013*, no. Healthcom, pp. 419–423, 2013.

[5] A. M. Adami, M. Pavel, T. L. Hayes, A. G. Adami, and C. Singer, “A method for classification of movements in bed,” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 7881–7884, 2011.

[6] N. Foubert, A. M. McKee, R. A. Goubran, and F. Knoefel, “Lying and sitting posture recognition and transition detection using a pressure sensor array,” *MeMeA 2012 - 2012 IEEE Symp. Med. Meas. Appl. Proc.*, pp. 65–70, 2012.

[7] C. Van Pul, H. P. M. E. V. D. Mortel, J. J. L. V. D. Bogaart, T. Mohns, and P. Andriessen, “Safe patient monitoring is challenging but still feasible in a neonatal intensive care unit with single family rooms,” *Acta Paediatr. Int. J. Paediatr.*, vol. 104, no. 6, pp. e247–e254, 2015.

[8] Y. Bitan, J. Meyer, D. Shinar, and E. Zmora, “Nurses? reactions to alarms in a neonatal

intensive care unit," *Cogn. Technol. Work*, vol. 6, no. 4, pp. 239–246, 2004.

[9] S. Lawless, "Crying wolf: false alarms in a pediatric intensive care unit.," *Crit Care Med.*, vol. 22, no. 6, pp. 981–5, 1994.

[10] K. Lewandowska, M. Weisbrot, A. Cieloszyk, W. Mędrzycka-Dąbrowska, S. Krupa, and Dorota Ozga, "Impact of Alarm Fatigue on the Work of Nurses in an Intensive Care Environment—A Systematic Review," *Int J Env. Res Public Heal.*, vol. 17, no. 22, 2020.

[11] W. PC, W. AH, D. D, and L. J, "Development and validation of a scale to measure work-related fatigue and recovery: the Occupational Fatigue Exhaustion/Recovery Scale (OFER)," *Occup Env. Med*, vol. 47, no. 6, pp. 594–606, 2005.

[12] S. A.K., T. A.F., G. T.M., R. C.C., and R. K.M., "Nurses' Perceptions and Practices Toward Clinical Alarms in a Transplant Cardiac Intensive Care Unit: Exploring Key Issues Leading to Alarm Fatigue," *JMIR Hum. Factors.*, vol. 3, no. 2, 2015.

[13] D. Ostojic, S. Guglielmini, S. Guglielmini, V. Moser, and J.-C. Fauchere, "Reducing False Alarm Rates in Neonatal Intensive Care: A New Machine Learning Approach," *Adv. Exp. Med. Biol.*, vol. 1232, 2020.

[14] Z. Ren *et al.*, "Analyzing center of pressure progression during bed exits," *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014*, pp. 1786–1789, 2014.

[15] Z. Huang, Y. Liu, Y. Fang, and B. K. P. Horn, "Video-based Fall Detection for Seniors with Human Pose Estimation," *4th IEEE Int. Conf. Univers. Village 2018, UV 2018*, pp. 18–21, 2019.

[16] A. Bekele *et al.*, "Real-time Neonatal Respiratory Rate Estimation using a Pressure-Sensitive Mat," *MeMeA 2018 - 2018 IEEE Int. Symp. Med. Meas. Appl. Proc.*, pp. 2–6, 2018.

[17] O. Permatasari, S. U. Masruroh, and Arini, “A prototype of child monitoring system using motion and authentication with Raspberry Pi,” *Proc. 2016 4th Int. Conf. Cyber IT Serv. Manag. CITSM 2016*, pp. 1–6, 2016.

[18] A. Bekele *et al.*, “Real-time Neonatal Respiratory Rate Estimation using a Pressure-Sensitive Mat,” *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA), Rome, Italy*, pp. 2–6, 2018.

[19] E. Koch and A. Dietzel, “Stretchable Sensor Array for Respiratory Monitoring,” *19th Int. Conf. Solid-State Sensors, Actuators Microsystems*, pp. 2227–2230, 2017.

[20] J. J. Liu *et al.*, “BreathSens: A Continuous On-Bed Respiratory Monitoring System with Torso Localization Using an Unobtrusive Pressure Sensing Array,” *IEEE J. Biomed. Heal. Informatics*, vol. 19, no. 5, pp. 1682–1688, 2015.

[21] A. Sivanantham, “Measurement of heartbeat, respiration and movements detection using Smart Bed,” *2015 IEEE Recent Adv. Intell. Comput. Syst. RAICS 2015*, no. December, pp. 105–109, 2016.

[22] S. Bennett, Z. Ren, R. Goubran, K. Rockwood, and F. Knoefel, “In-Bed Mobility Monitoring Using Pressure Sensors,” *IEEE Trans. Instrum. Meas.*, vol. 64, no. 8, pp. 2110–2120, 2015.

[23] R. Joshi *et al.*, “A ballistographic approach for continuous and non-obtrusive monitoring of movement in neonates,” *IEEE J. Transl. Eng. Heal. Med.*, vol. 6, no. September, pp. 1–10, 2018.

[24] I. Scott, S. Carter, and E. Coiera, “Clinician checklist for assessing suitability of machine learning applications in healthcare,” *BMJ Heal. Care Inform.*, vol. 28, no. 1, 2021.

[25] N. B. Karayiannis, G. Tao, J. D. Frost, M. S. Wise, R. A. Hrachovy, and E. M. Mizrahi, "Automated detection of videotaped neonatal seizures based on motion segmentation methods," *Clin. Neurophysiol*, vol. 117, no. 7, pp. 1585–1594, 2006.

[26] Y. S. Dosso, S. Aziz, S. Nizami, K. Greenwood, J. Harrold, and J. R. Green, "Video-Based Neonatal Motion Detection," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2020-July, pp. 6135–6138, 2020.

[27] S. Kalitzin, G. Petkov, D. Velis, B. Vledder, and F. Lopes Da Silva, "Automatic segmentation of episodes containing epileptic clonic seizures in video sequences," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 12, pp. 3379–3385, 2012.

[28] A. Heinrich, D. Geng, D. Znamenskiy, J. P. Vink, and G. De Haan, "Robust and sensitive video motion detection for sleep analysis," *IEEE J. Biomed. Heal. Informatics*, vol. 18, no. 3, pp. 790–798, 2014.

[29] M. C. Chang *et al.*, "In-bed patient motion and pose analysis using depth videos for pressure ulcer prevention," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Septe, pp. 4118–4122, 2018.

[30] X. Hu, K. Naya, P. Li, T. Miyazaki, K. Wang, and Y. Sun, "Non-Invasive Sleeping Posture Recognition and Body Movement Detection Based on RFID," *Proc. - IEEE 2018 Int. Congr. Cybermatics 2018 IEEE Conf. Internet Things, Green Comput. Commun. Cyber, Phys. Soc. Comput. Smart Data, Blockchain, Comput. Inf. Technol. iThings/Gree*, pp. 1817–1820, 2018.

[31] S. Hu, W. Rueangsirarak, M. Bouchee, N. Aslam, and H. P. H. Shum, "A motion classification approach to fall detection," *Int. Conf. Software, Knowl. Information, Ind. Manag. Appl. Ski.*, vol. 2017-Decem, pp. 1–6, 2018.

[32] A. Zhang *et al.*, "Automated tremor detection in Parkinson's disease using accelerometer signals," *Proc. - 2018 IEEE/ACM Int. Conf. Connect. Heal. Appl. Syst. Eng. Technol. CHASE 2018*, pp. 13–14, 2019.

[33] J. M. Fisher, N. Y. Hammerla, T. Ploetz, P. Andras, L. Rochester, and Richard W Walker., "Unsupervised home monitoring of Parkinson's disease motor symptoms using body-worn accelerometers," *Parkinsonism Relat. Disord.*, vol. 33, pp. 44–50, 2016.

[34] C. Marcroft, A. Khan, N. D. Embleton, M. Trenell, and T. Plötz, "Movement recognition technology as a method of assessing spontaneous general movements in high risk infants," *Front. Neurol.*, vol. 6, no. JAN, p. 284, 2015.

[35] P. Jallon, S. Bonnet, M. Antonakios, and R. Guillemaud, "Detection system of motor epileptic seizures through motion analysis with 3D accelerometers," *Proc. 31st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. Eng. Futur. Biomed. EMBC 2009*, pp. 2466–2469, 2009.

[36] N. Foubert, A. M. McKee, R. A. Goubran, and F. Knoefel, "Lying and sitting posture recognition and transition detection using a pressure sensor array," *MeMeA 2012 - 2012 IEEE Symposium on Medical Measurements and Applications, Proceedings*. pp. 65–70, 2012.

[37] G. M. Lyons, K. M. Culhane, D. Hilton, P. A. Grace, and D. Lyons, "A description of an accelerometer-based mobility monitoring technique," *Med. Eng. Phys.*, vol. 27, no. 6, pp. 497–504, 2005.

[38] K. Culhane, G. Lyons, D. Hilton, P. Grace, and D. Lyons, "Long-term mobility monitoring of older adults using accelerometers in a clinical environment," *Clin. Rehabil.*, vol. 18, no. 3, p. 335, 2004.

[39] S. H. Lee, H. D. Park, S. Y. Hong, K. J. Lee, and Y. H. Kim, "A Study on the Activity Classification Using a Triaxial Accelerometer," *Annu. Int. Conf. IEEE Eng. Med. Biol. - Proc.*, vol. 3, pp. 2941–2943, 2003.

[40] X. Zhu *et al.*, "Real-time monitoring of respiration rhythm and pulse rate during sleep," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2553–2563, 2006.

[41] E. J. Pino, A. Dorner De La Paz, P. Aqueveque, J. A. P. Chavez, and A. A. Moran, "Contact pressure monitoring device for sleep studies," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, pp. 4160–4163, 2013.

[42] A. M. Adami, A. Adami, C. M. Singer, T. L. Hayes, and M. Pavel, "A System for unobtrusive monitoring of mobility in bed," *Proc. 11th IEEE Int. Conf. Comput. Sci. Eng. CSE Work. 2008*, pp. 13–18, 2008.

[43] R. Murthy, I. Pavlidis, and P. Tsiamyrtzis, "Touchless monitoring of breathing function," *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, pp. 1196–1199, 2004.

[44] M. Cretikos, J. Chen, K. Hillman, R. Bellomo, S. Finfer, and A. Flabouris, "The objective medical emergency team activation criteria: A case-control study," *Resuscitation*, vol. 73, no. 1, pp. 62–72, 2007.

[45] D. Townsend, M. Holtzman, R. Goubran, M. Frize, and F. Knoefel, "Effect of windowing on central apnea detection," *2010 IEEE Int. Work. Med. Meas. Appl. MeMeA 2010 - Proc.*, pp. 117–121, 2010.

[46] K. Banno and M. H. Kryger, "Sleep apnea: clinical investigations in humans," *Sleep Med*, vol. 8, pp. 400–426, 2007.

[47] S. L. Bennett, R. Goubran, K. Rockwood, and F. Knoefel, "Monitoring the relief of pressure points for pressure ulcer prevention: A subject dependent approach," *MeMeA 2013 - IEEE Int. Symp. Med. Meas. Appl. Proc.*, pp. 135–138, 2013.

[48] J. Elfehri, F. Boussu, V. Koncar, and C. Vasseur, “Novel approach of ulcer prevention based on pressure distribution control algorithm,” *2011 IEEE Int. Conf. Mechatronics Autom. ICMA 2011*, pp. 265–270, 2011.

[49] M. Taylor *et al.*, “Assessing standing stability of older adults using pressure sensitive arrays,” *MeMeA 2012 - 2012 IEEE Symp. Med. Meas. Appl. Proc.*, pp. 10–14, 2012.

[50] J. Light, S. Cha, and M. Chowdhury, “Optimizing Pressure Sensor Array Data for Smart Shoe Fall Monitoring System,” *2015 IEEE SENSORS*, pp. 1–4, 2015.

[51] F. Q. Al-Khalidi, R. Saatchi, D. Burke, H. Elphick, and S. Tan, “Respiration rate monitoring methods: A review,” *Pediatr. Pulmonol.*, vol. 46, no. 6, pp. 523–529, 2011.

[52] S. Davis *et al.*, “Continuous non-contact vital sign monitoring in neonatal intensive care unit,” *Healthc. Technol. Lett.*, vol. 1, no. 3, pp. 87–91, 2014.

[53] C. H. Lund, L. B. Nonato, J. M. Kuller, L. S. Franck, C. Cullander, and D. J. Durand, “Disruption of barrier function in neonatal skin with adhesive removal,” *J. Pediatr.*, vol. 131, no. 3, pp. 367–372, 1997.

[54] A. Bekele, J. Samuel, S. Nizami, A. Basharat, R. Giffen, and J. R. Green, “Ontology Driven Temporal Event Annotator mHealth Application Framework,” *Proc. 28th Annu. Int. Conf. Comput. Sci. Softw. Eng.*, pp. 309–314, 2018.

[55] J. Zhao, F. Gonzalez, and D. Mu, “Apnea of prematurity: From cause to treatment,” *Eur. J. Pediatr.*, vol. 170, no. 9, pp. 1097–1105, 2011.

[56] Matlab, “Sequential Forward Selection,” 2018. [Online]. Available: https://www.mathworks.com/help/stats/sequentialfs.html. [Accessed: 10-Aug-2021].

[57] R. Selzler, A. D. C. Chan, and J. R. Green, “Impact of Subject-specific Training Data in Anxiety Level Classification from Physiologic Data,” in *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 2021, pp. 1–6.

[58] T. E. Bachman, T. E. Bachman, N. P. Iyer, C. J. L. Newth, P. A. Ross, and R. G. Khemani, “Thresholds for oximetry alarms and target range in the NICU: An observational assessment based on likely oxygen tension and maturity,” *BMC Pediatr.*, vol. 20, no. 1, pp. 1–8, 2020.

[59] N. Phattraprayoon, S. Sardesai, M. Durand, and R. Ramanathan, “Accuracy of pulse oximeter readings from probe placement on newborn wrist and ankle,” *J. Perinatol.*, vol. 32, no. 4, pp. 276–280, 2012.

[60] E. Y. Chang, H. Bai, K. Zhu, H. Wang, J. Li, and Z. Qiu, “PSVM: Parallel Support Vector Machines with incomplete cholesky factorization,” *Scaling up Mach. Learn. Parallel Distrib. Approaches*, vol. 9780521192, no. 2, pp. 109–126, 2011.

# Appendix A

Thirty-three infants' data with weight characteristics across three categories were collected across three different observation conditions. Sex and gestational age were not controlled for in this study. *Table 20* describes all patients in this study.

**Table 20.** Descriptive characteristics for all patients in NICU study

| Patient ID | Bed Type | Weight Class | Weight (g) | Sex | Gestational Age |
|---|---|---|---|---|---|
| 22 | overhead warmer | *<1500g* | 1391 | female | 25 weeks, 2 days |
| 38 | | | 1430 | female | 31 weeks, 4 days |
| 35 | | | 1220 | male | 31 weeks, 1 day |
| 1 | | *1500g ≤ w ≤ 2500g* | 2465 | male | 35 weeks, 7 days |
| 23 | | | 1690 | male | 24 weeks, 2 days |
| 26 | | | 2045 | male | 32 weeks, 4 days |
| 27 | | | 2300 | male | 37 weeks, 1 day |
| 29 | | | 1930 | female | 36 weeks, 4 days |
| 6 | | *≥ 2500g* | 3504 | male | 41 weeks, 3 days |
| 14 | | | 2800 | female | 37 weeks, 2 days |
| 15 | | | 3250 | female | 39 weeks, 0 days |
| 16 | | | 3460 | female | 39 weeks, 1 day |
| 17 | | | 2765 | male | 35 weeks, 2 days |
| 8 | crib | *1500g ≤ w ≤ 2500g* | 2340 | female | 37 weeks, 4 days |
| 28 | | | 2055 | female | 33 weeks, 5 days |
| 11 | | | 2415 | female | 28 weeks, 5 days |
| 30 | | | 2310 | male | 32 weeks, 5 days |
| 32 | | | 1780 | male | 27 weeks, 1 day |
| 2 | | *≥ 2500g* | 3620 | male | 31 weeks, 6 days |
| 5 | | | 3480 | male | 30 weeks, 0 days |
| 13 | | | 4435 | male | 40 weeks, 1 day |
| 9 | | | 2520 | female | 37 weeks, 3 days |
| 10 | | | 3530 | male | 26 weeks, 5 days |
| 18 | incubator | *<1500g* | 1290 | female | 33 weeks, 1 day |
| 21 | | | 1110 | male | 31 weeks, 5 days |
| 24 | | | 1280 | male | 30 weeks, 0 days |
| 25 | | | 889 | male | 26 weeks, 6 days |
| 19 | | *1500g ≤ w ≤ 2500g* | 1604 | male | 31 weeks, 1 day |
| 37 | | | 1720 | male | 31 weeks, 2 days |
| 36 | | | 1605 | male | 31 weeks, 2 days |
| 31 | | | 1545 | male | 28 weeks, 1 day |
| 34 | | | 1727 | female | 36 weeks, 5 days |
| 33 | | | 1825 | female | 36 weeks, 4 days |

# Appendix B

In *Section 4.4* the challenges with Patient 11's data and its use in the motion detection algorithm are briefly described. The analysis of optimal window lengths and decision thresholds are explored while including and excluding Patient 11. This section does a deep dive on Patient 11's data, the observation environment and any annotation or recording ambiguities.

## *Binary Classification Performance Across Patients and an Exploration of Patient 11*

To better understand the types of errors being made by the classifier, the confusion matrices and performance metrics including and excluding Patient 11, across all patients are shown in *Figure 27.*

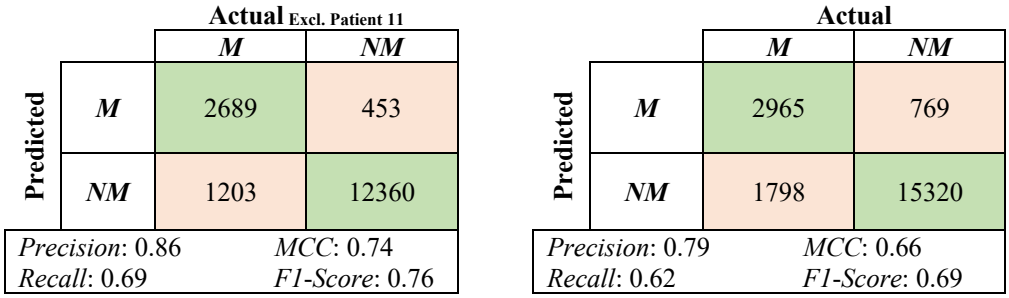



**Figure 27.** Confusion matrices and performance metrics pooled across all patients, including (left), and excluding (right) patient 11 data.

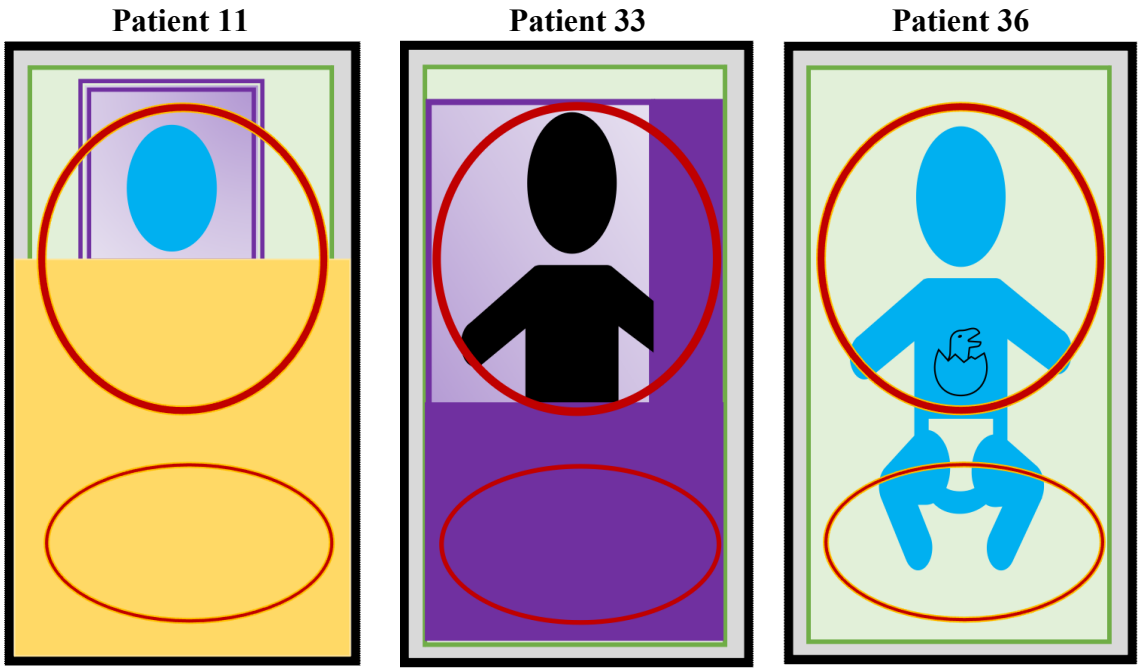
Each value in the matrix represents classification performance for a window of size *wLen* = 9. Looking at the confusion matrix on the right, it appears that the biggest changes – after including Patient 11's performance in the matrix – are reflected in the false positive, false negative, and true negative counts. Between the two, there are more false positives added than false negatives when Patient 11 is included, leading to a greater reduction in precision than in recall, both of which show a reduction. Overall, the precision of this classifier is 0.86 and the recall is 0.69; the classifier is fair at avoiding false positives; in other words, it is better than chance at avoiding the

misclassification of a *no_motion* event as a motion event. The MCC score is 0.74; since the MCC can range from -1 to 1 where -1 is an extremely poor classifier. The classification accuracies are 0.90 and 0.84 for the confusion matrices excluding and including the performance of Patient 11 on the classifier, respectively. Given the data is imbalanced and the cost of the classifier predicting false positives is greater than the impact of the classifier predicting false negatives, the technical accuracy is not a fair representation of classifier performance, and the F1-score is preferred. Furthermore, the importance of precision and recall is recognized for clinical application – since the F1-score gives equal importance to both precision and recall, it is the fairest estimator of the classifier's overall performance with relation to both those metrics.

Looking at *Table 10,* the patient data that the classifier produces the best results with are Patients 2, 6, and 13 – all of which are in the higher weight class category with masses greater than 2500 grams. Patients 11, 33, and 36 share the same middle weight category weighing $1500g \leq w < 2500g$ but have more variations when tested with the classifier, with Patient 11's performance metrics being the most anomalous from the other two patients. To understand the differences between Patients 11, 33, and 36, the case report form for each patient was reviewed which contains information on the patient's weight and bed type. The video data was also reviewed to see the cushioning under and around the baby that could impede with receiving a stronger signal with the PSM.

Reflecting on the behaviour of Patient 11 in *Figure 14* and the real-life observation conditions for Patient 11, as well as the fine-tuning of annotations that were conducted post-collection, the challenge with Patient 11 is that the infant is 60% covered with a quilt. In contrast, Patients 33 and 36 have exposed abdominal areas and no double swaddle, if swaddled at all. These conditions are illustrated in *Figure 28.*

Upper body motion was noted as any movement observed of the head and 'pushing through' behaviour in the abdominal region by what was understood to be the arms moving in any direction. Lower body motion was noted as any 'pushing through' behaviour in the pelvic area and below, which was understood to be the legs moving up and down. The thick quilted blanket was placed directly on the infant but due to the rigidity of the blanket, it appears to barely hug the infant. If there was any movement occurring under the blanket that did not involve arms or legs moving up and down, instead a 'wriggling' or 'fidgeting' behaviour, this would not have been observed clearly and that portion of the event was annotated as '*no_motion*'.




Legend: *Black border: mattress, gray border: sensor mat, green border: bedsheet, purple (single border): single swaddle, purple (double border): double swaddle, yellow square: opaque blanket/quilt, wearing clothing (excluding diaper): blue infant*

**Figure 28.** In-bed conditions for patients 11, 33, and 36. The red circles represent the areas of the infant observed to determine upper, lower, and gross movement. Patient 11 was in a crib, while patients 33 and 36 were in incubators.

All of Patient 11's data was broken down into four sets of continuous and un-interrupted mixed motion and *no_motion* real-time data – this included gross, upper, lower, and *no_motion* segments.

To understand whether any of the data was poor or inaccurate, the performances on each of the four sets of continuous data were plotted in *Figure 29.* It appears that there are two sets of continuous data, sets 1 and 2 that appear to have the best precision-recall curves and AUC curves, whereas set 3 and 4 have lower curves. There is no relationship between the duration of the segment and the number of associated datapoints to the performance of the classifier because in both pairs of curves there is a segment with less than ten minutes and greater than ten minutes. The names of the sets match their order in time, that is the first set contains time series data that precedes set two and set two contains time series data that precedes set three. It is interesting then to note that the first ~25 minutes of data have the strongest curves while the last 45 minutes have the weakest curves. The precision-recall curves are overall shallow and close to the 0.50 chance cut-off.

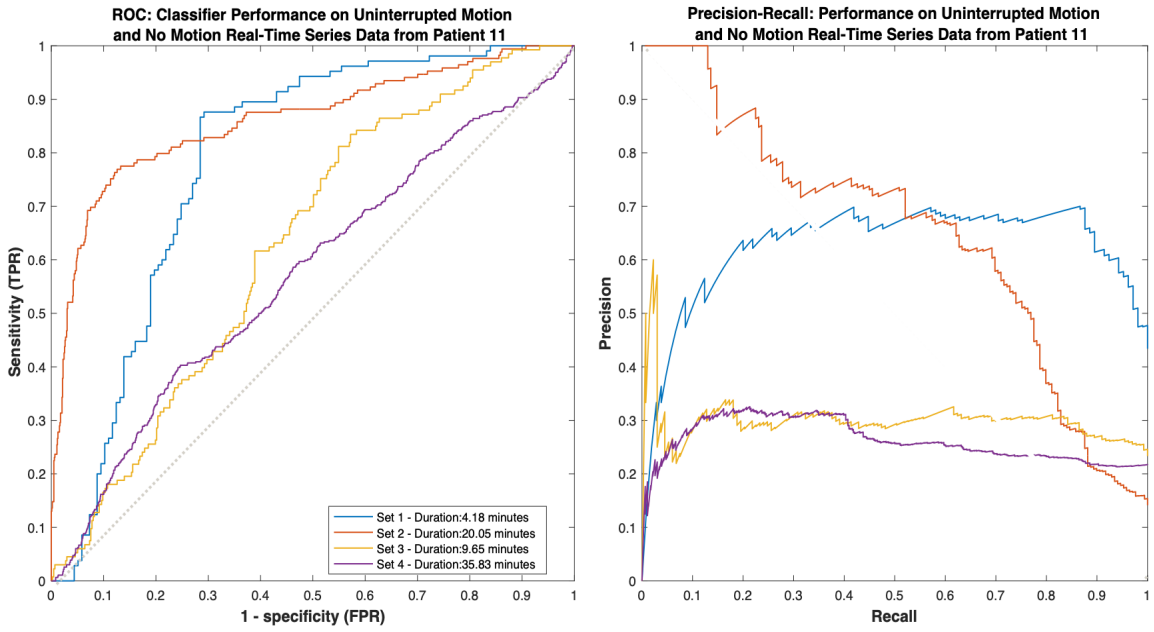



**Figure 29.** Classifier performance on individual series of uninterrupted motion-*no_motion* datasets from Patient 11.

When aligning the pressure sensor data with the video data during the post-collection annotation review, the video data was found to be 13 minutes and 29 seconds ahead of the PSM using the validation protocol with registration points as described in *Section 3.3.* Based on this, all PSM timestamps for Patient 11 were matched and identified, however, not every single annotation was reviewed manually given the registration points were meant to be used to verify synchronicity between the PSM and video data; spot checks across the annotations were done to validate matching.

The plots for Patient 11 suggest that the events in the latter half of the annotated datasets were not accurately annotated, and given the low precision, events labelled as '*no_motion*' were 'inaccurately' classified as motion. Based on this information, both motion and *no_motion* events in sets 1 and 2, and 3 and 4 were reviewed in the source video. It was found that the *no_motion* events in sets 1 and 2 were properly labelled as per the synchronization difference identified in the annotation and validation steps; however, the *no_motion* events in sets 3 and 4 were indeed mislabelled as being entirely *no_motion* and the motion events were mislabelled as being entirely motion. The algorithm, however, was correct to identify these segments as motion based on the baseline parameters that produced favourable results when the classifier was testing with patients 2, 6, 13, 33, and 36. The synchronization difference in time was identified correctly across patients 2, 6, 13, 33, and 36 and partially correctly for Patient 11 as per the current results.

It is possible that the sensors may have picked up noise from the environment, that the heart rate could have caused an elevated velocity of thoracic movement that could not be seen on camera or necessarily in person given the coverage and thickness of the blanket. Additionally, there were changes in video contrast during segments labelled '*no_motion*' that could have suggested slight movements in the legs or lower abdominal areas that shifted the blanket, but while slight movement of the COP could sometimes be seen on the PSM, movement was not clear in real-time observation

or video data. Moreover, during data transfer and given the size of the video files per patient, it is possible that frames were dropped or skipped at or after a certain point which would negate the time stamp synchronization. Lastly, it is also possible that the synchronicity between the video and PSM was further separated due to a software error that impacted the way the PSM and video data were post-processed.

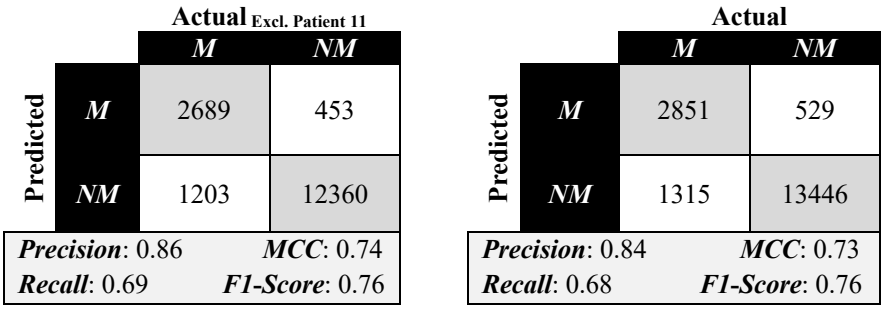



**Figure 30.** Performance metrics across patients including (left) and excluding (right) revised Patient 11 data.

Given the findings above as they pertain to Patient 11, a general set of metrics can be re-computed including only the first two continuous series from Patient 11. *Figure 30* displays the confusion matrices including and excluding Patient 11's data post-revision. Adding Patient 11's revised data to this classifier now preserves the performance of the classifier overall by producing metrics that are closer to the overall average when considering the other patients in the dataset. The F1-score is unaffected, the precision and recall are marginally reduced by 0.02 and 0.01 respectively.

*Figure 31* shows that the ROC curve for Patient 11 is no longer deviant, similarly the precision-recall curve for Patient 11 is no longer below 0.50 and presents a closer curve to the other patients in the dataset. Patient 33 is another patient whose data presented many errors in classification; the precision-recall curve almost matches the 0.50 line while the AUC is still similar to other patients. While the data for Patient 33 contains both motion and *no_motion* events and there is an hour of

data present with 49 continuous motion segments out of 108 total segments, the duration of motion events total to 4 minutes and 46 seconds. The AUC is poor at recognizing this class imbalance. Every two seconds, one distance datapoint is created for summing over a window size of 9, which means that the window must contain at least 10 seconds of data. Every 10 seconds of data must contain at least 6 seconds that are labelled as motion since six seconds produce five datapoints within a nine-point window, and a minimum of 5 motion or *no_motion* labels are required to assign a label for testing.

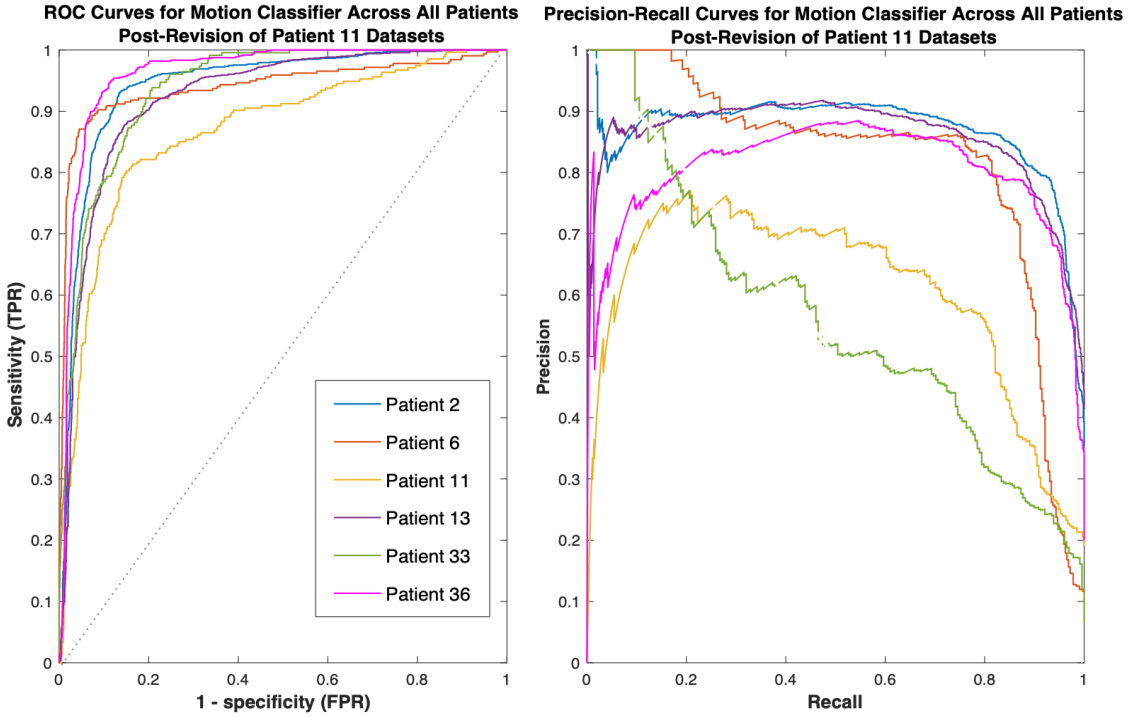



**Figure 31.** ROC and precision-recall curves between patients including revised Patient 11 data with real-time classifier using generalized parameters.

Thirty-two out of fourty-nine of the continuous motion segments are smaller than 6 seconds so they are never identified as motion in the labelling scheme. When those portions of COP data show motion-like behaviour, the assigned label will be motion which is technically correct, but not per the expectations of the mode-based label. When smaller segments of motion occur that are four

seconds long or less, there would be more *no_motion* points to correctly classify that segment as mostly *no_motion* even if motion-like behaviour was present. The performance of the classifier on Patient 33 suggests that shorter segments of motion are more difficult to identify and are deemed as negligible motion events unless followed by another short segment of motion within the same window, if using longer windows.

Of the four minutes of motion data, 1 minute and 36 seconds of the data is upper body motion whose sum distances overlap with the sum distances of *no_motion* segments. These results were found when plotting histograms to determine the threshold between motion and *no_motion*, as can be seen in *Figure 32.* For this reason, it is possible that upper body motion segments were also classified as *no_motion*, adding to the false negative count for Patient 33 and contributing to the low recall.

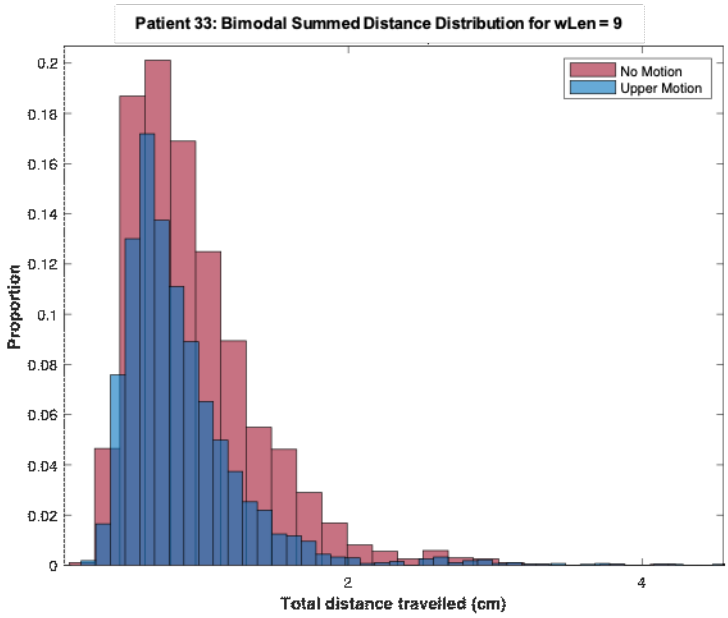



**Figure 32.** Sum distance histogram plot for episodes of *no_motion* and *motion* for *wLen = 9.*

# Appendix C

This section describes the annotation protocol for the events explored in this study (*motion, upper body motion, lower body motion)* and how any additional event types were annotated for use in future work.

**Table 21.** Complete set of annotations for all patient events during observation period

| Event Type | Description | Inclusions or Exclusions | |
|---|---|---|---|
| Motion (gross) | No dominant limb movement; both arms, both legs, and head in sporadic motion | • **Exclusion:** Gross twitches lasting between 1 – 2 seconds not considered as a motion event<br>• **Exclusion:** Pauses lasting 1 – 2 seconds between motion events included in sum motion event | |
| Upper and Lower Body Motion (UBM and LBM) | Specific limb movements – differentiate between arms, legs, and head movement | Lower Body Motion (LBM) | • Legs: During observation, both legs raise or move under a blanket; annotations of specific leg movements come from 'gross' annotation, when legs move independently from gross annotation, we do not specify which leg. Labelled 'legs'. |
| | | Upper Body Motion (UBM) | • Head: Head moving up and down or side to side. Labelled 'head'<br>• Arms: Left and right arm treated as 'arms'. |
| *No_motion* (gross) | No movement of limbs, no internally induced events (i.e., yawn, sneeze, etc.). Typically, a patient is resting or sleeping. | • **Inclusive of:** Gross twitches that last between 1 – 2 seconds; these could be representative of brief sleep apnea, but this is not clinically concerning unless it lasts 15 seconds or more [55]<br>• **Inclusive of:** Hand wiggle, wrist flexation, and brief head nods lasting between 1 – 3 seconds | |
| Internally Induced Motion (IIM) | PSM activity induced by sneezes, yawns, and cries; coughs are not heard or detected using the PSM or video data | • Duration of sneeze, yawn, and cries vary – all are included. | |
| Externally Induced Motion (EIM) | PSM activity induced by environmental noise. | • **Inclusive of:** Parent patting, caressing, or kissing the child; patient or parent touching bed; nurse touching or readjusting the patient; patient removed and replaced on bed<br>• **Inclusive of:** Clinical interventions and routine clinical care; 'referencing' events during study | |
| Clinical Intervention | Routine clinical care | • Diaper change, bottle feeding, $SpO_2$ sensor, giving a soother | |

# Appendix D

This section illustrates how the distances travelled between consecutive seconds of movement in mixed event episodes were labelled for real-time baseline motion detection. Further, this section illustrates how summed windows of those distances were labelled to compute real-time model metrics. As can be seen in *Figure 33*, odd numbered windows allow for a median event to be identified while even numbered windows still produce ambiguity between labels.

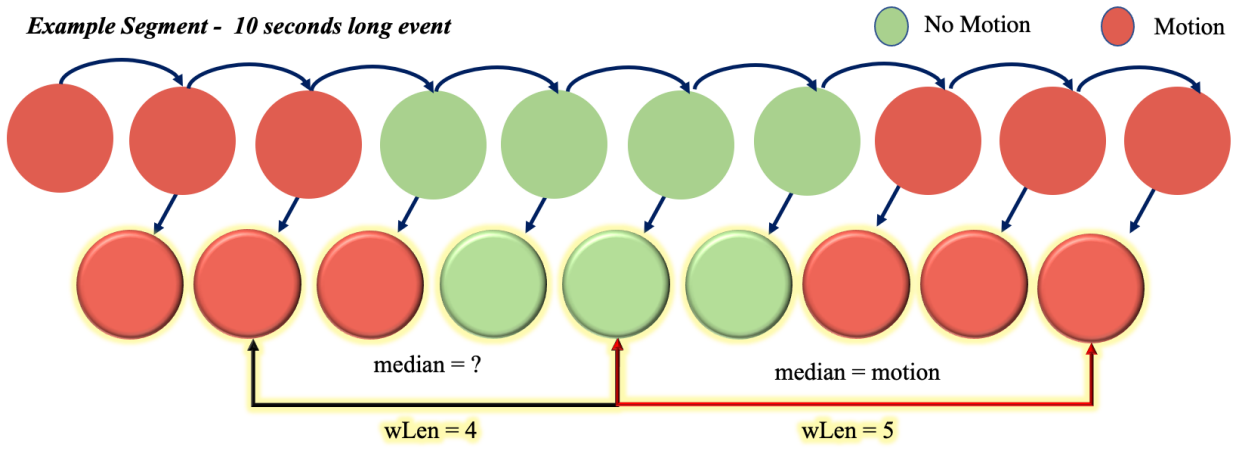



**Figure 33.** Median labelling approach for real-time motion detection model metrics.

# Copyright Acknowledgements