



Automated detection of Patient-Ventilator Asynchrony in Mechanically Ventilated Patients

Melissa de Bie

Master Thesis Technical Medicine

Intensive Care Unit LUMC

AUTOMATED DETECTION OF PATIENT-VENTILATOR ASYNCHRONY IN MECHANICALLY VENTILATED PATIENTS

Melissa de Bie

Student number : 4547969

12 04 2024

Thesis in partial fulfilment of the requirements for the joint degree of Master of Science in

Technical Medicine

Leiden University | Delft University of Technology | Erasmus University Rotterdam

Master thesis project (TM30004 ; 35 ECTS)

Dept. of Biomechanical Engineering, TUDELFT
Intensive Care Unit, LUMC

Supervisor(s):

Dr. A. Schoe

Dr. D.M.J. Tax

Thesis committee members:

Dr. A. Schoe

Dr. D.M.J. Tax

Dr. D. van Westerloo

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Preface

This thesis came about after a lot of challenges, doubts and hard work. After a year of doing research in the ICU at LUMC, I am happy and proud to finally have my thesis in front of me.

I would like to express my gratitude to my supervisors for their guidance throughout this journey. Bram was always available to answer my questions and was flexible with my schedule, which greatly contributed to my progress. I am grateful to David for his patience in explaining complex technical concepts in a comprehensible manner. His efforts have been invaluable in shaping my understanding.

I am also thankful to Petra and Willem for their unwavering support and dedication. They spent countless hours assisting me with labelling tasks and were continuously available for guidance.

Additionally, I extend my thanks to my fellow peers, whose friendship and encouragement have been a constant source of motivation. I would like to express my gratitude to Imane, Janno, Suus, Friso, and especially Floor for making this journey more enjoyable.

Additionally, I would like to thank my family and friends for their unwavering support throughout this challenging yet fulfilling period. They were always there to listen, even if they did not fully understand the complexities of my work.

As I reflect on this chapter of my life, I look forward to the future and hope to use my technical background to contribute to the improvement of healthcare.

Contents

Preface	2
List of Abbreviations	5
Abstract.....	6
Introduction	7
Background	9
Medical Background	9
Technical Background.....	12
Methods.....	17
Data collection	17
Data Labelling	17
Data pre-processing in Python:.....	17
Mahalanobis distance	21
Unsupervised machine learning: create pre-annotations	22
CNN Model.....	24
Data input	25
Final Validation	28
Results.....	29
Data collection	29
Unsupervised Machine learning	29
Supervised machine learning.....	31
Adjusting Scaling Factors for Improved Prediction.....	35
Validation by Clinicians	36
Discussion	38
Limitations	38
Recommendations.....	40
Further perspectives.....	41
Conclusion.....	41
References	42
Appendix.....	45
A. Data Annotation Protocol	45
B. PVA Types per Patient.....	47
C. AUC performances from the ALL model.....	48
D. Other Artefacts.....	50

E. Performances of clinical validation by experts.....51

List of Abbreviations

AI - Artificial Intelligence
ARDS - Acute Respiratory Distress Syndrome
ASV - Adaptive Support Ventilation
AUROC - Area Under the Receiver Operating Characteristic
CNN/ConvNet - Convolutional Neural Network
COPD - Chronic Obstructive Pulmonary Disease
CV - Cross-Validation
DT - Double Triggering
ETS - Expiratory Trigger Sensitivity
FN - False Negative
FP - False Positive
GAN:
ICU - Intensive Care Unit
IEE - Ineffective Inspiratory Effort
LOOCV - Leave-One-Out-Cross Validation
LUMC - Leiden University Medical Centre
ML - Machine Learning
MV - Mechanical Ventilation
OVR - One-vs-Rest Strategy
Paw - Airway Pressure
PCA - Principal Component Analysis
P-CMV - Pressure Controlled Mandatory Ventilation
Pcontrol - Adjustable pressure above PEEP
PCV - Pressure Controlled Ventilation
PEEP: positive end-expiratory pressure
Pes: Oesophageal pressure
PL: Transpulmonary pressure
PRamp - Pressure Ramp
PSV: pressure support ventilation
PVA: Patient-ventilator Asynchrony
ReLU: Rectified Linear Unit
ROC: receiving operation characteristic
ROI: region of interest
RR: respiratory rate
SPON: spontaneous ventilation
TN: true negative
TP: true positive

Abstract

Introduction

Patient-ventilator asynchrony (PVA) poses a significant challenge in the management of mechanically ventilated patients, contributing to adverse clinical outcomes. Current methods of detecting PVA rely on visual assessment by clinicians, leading to subjectivity and inconsistency. Therefore, there is a need for automated techniques to identify PVA accurately and efficiently. In this study, we explore the application of supervised and unsupervised machine learning algorithms to develop an automatic detection system for PVA.

Methods

This study was conducted at the ICU of the LUMC in Leiden, the Netherlands. Patients eligible for inclusion were mechanically ventilated with an esophageal balloon inserted. Data collected included flow, Paw, and Pes curves, which were labelled using an open-source data labeling platform and processed in Python. Supervised CNN models were trained for different ventilation modes, while unsupervised techniques, utilizing Mahalanobis distance, were explored for data pre-labeling. The discriminative capability of the models was assessed using AUROC values.

Results

25 patients were included in this study and labelled by clinicians. Using an unsupervised machine learning technique based on the Mahalanobis distance for data pre-labeling, a threshold of 3.5 was selected, resulting in a 95% accuracy in correctly identifying normal breaths. Creating different CNN models for automating the detection of PVA the results demonstrate the discriminative capability of the various models across all ventilation modes, PSV and PCV ventilation. They can differentiate between normal and abnormal breaths, as indicated by the AUROC values of $0.85(\pm 0.08)$, $0.83(\pm 0.12)$, and $0.80(\pm 0.28)$ respectively.

Discussion

This study investigated the application of machine learning techniques to analyse ventilation data in critical care settings. Through a combination of unsupervised and supervised learning methods, we have explored the automation of the labelling process and the development of predictive models for identifying patient-ventilator asynchronies (PVAs).

Introduction

In the intensive care unit (ICU) mechanical ventilation plays a crucial role in supporting critically ill patients with respiratory difficulties (1). Ventilation can partially assist a patient's breathing, and take over completely if necessary. The purpose of ventilation is to provide optimal respiratory support to patients. However, it is sometimes seen that the ventilation and breathing of patients are not properly synchronised to each other. This leads to patient-ventilator asynchrony (PVA)(2). This phenomenon happens when the timing of the ventilator-assisted breaths mismatches the patient's respiratory effort (3-5). There are different types of PVAs occurring during the different phases of inhalation and exhalation (6).

Currently, the diagnosis of PVA relies on visual analysis of ventilator waveforms, specifically pressure and flow curves (7), for this continuous bedside monitoring of the ventilator is required, which is not feasible in clinical practice. Consequently, there is an underdiagnosis of PVA, leading to uncertainty regarding its incidence (8). The sensitivity of visual analysis based on flow and volume curves is low, ranging from 16-28% (9).

PVA has potential negative effects on clinical outcome measures. It is known to have an association with a higher work of breathing, leading to excessive strain on the respiratory muscles. This, in turn, can lead to more dyspnoea and discomfort (4, 10). Several studies have described the effect of PVA on clinical outcomes. These show that PVA may be associated with prolonged mechanical ventilation and ICU length of stay. However, the effects of PVA on mortality have been inconsistent. This suggests that further research is needed to determine its role in mortality outcome (11). In addition, it appears that different types of asynchronies have different effects on clinical outcome measures. Specifically for reverse triggering, it is hypothesised that there is an association between signs of patient improvement and reverse triggering. Additionally, it has been observed that PVA tends to occur in clusters and that these clusters have the greatest impact on clinical outcome measures (12, 13). To identify the occurrence of clusters of PVA, continuous monitoring is necessary.

Automatic detection algorithms can assist in the diagnosis of PVA using the ventilator waveforms of mechanically ventilated patients. Presently, two types of algorithms have been developed in a research-based environment: rules-based and machine learning algorithms. However, no algorithm is currently suitable for clinical use as all are still in a research setting (14). Current algorithms developed consist of data solely based on ventilator's flow and pressure curves (Paw). However, the use of reference signals from patients, such as oesophageal pressure measurement, which involves placing a balloon in the oesophagus to measure lung and chest wall mechanics and transpulmonary pressure during mechanical ventilation, or diaphragmatic electrical activity, can provide more information about the patient's respiratory effort and lead to increased PVA detection (5, 15). The addition of oesophageal pressure (Pes) to the algorithm leads to the diagnosis of more PVAs and a deeper understanding of the patient's ventilation process (16, 17).

Automatic detection of PVA with the use of an algorithm brings several potential benefits. An algorithm allows continuous monitoring, which also leads to better accuracy of diagnosis and a reduction in workload. This, in turn, allows adequate response to the patient's ventilator strategy when detecting many asynchronies. In addition, with continuous monitoring, data can be collected, allowing insightful mapping of the clinical effect of the occurrence of PVAs.

This study aims to develop an automatic detection algorithm capable of recognizing PVA in mechanically ventilated patients using flow, Paw, and Pes curves. To achieve this, two machine learning techniques are investigated: supervised and unsupervised machine learning.

Supervised machine learning techniques are well known in medical research, especially when dealing with large labelled datasets (18). Convolutional neural networks (CNNs), a subset of supervised learning, are particularly suited to our clinical problem. CNNs excel at automatic identifying relevant patterns in labelled data, using the large amount of underlying information available and can develop an algorithm based on these patterns. The algorithm can then automatically recognise new unlabelled data and label it based on the data in the algorithm (19).

Unsupervised machine learning offers an interesting alternative, as it does not require labelled data and is known for its ability to recognise patterns without the labelled data, as example mostly used to detect anomalies, in line with the aim of this study. Unsupervised machine learning relies on identifying groups of data based on similarities and differences.

In summary, to gain more insight into PVA detection and its clinical consequences, the aim of the study is to develop an detection algorithm that can automatically recognise PVA in mechanically ventilated patients based on the flow, Paw and Pes curve. This research explores opportunities within supervised and unsupervised machine learning to address this clinical problem.

Background

Medical Background

Mechanical ventilation and ventilation modes

As stated in the introduction, numerous patients in the ICU require mechanical ventilation due to their respiratory needs. Mechanical ventilation is used in different modalities, during this study three different modalities are used including, pressure support ventilation (PSV), pressure controlled ventilation (PCV), and (semi-)closed loop ventilation. In this study, the C-6 mechanical ventilators manufactured by Hamilton Medical are used.

With PSV, the patient triggers every breath. PSV provides an amount of pressure (P_{supp}) during inspiration, to reduce the effort of breathing and support the spontaneously breathing patient. To use this modality, you must also set the timing of the inspiration by adjusting the expiratory trigger sensitivity (ETS). This setting determines the duration of inspiration relative to expiration (20).

In PCV ventilation, all breaths are pressure-controlled and mechanical, regardless of whether they are triggered by the patient or the ventilator. This means that pressure is delivered at a constant level with the volume depending on the pressure settings, inspiration time, and the resistance and compliance of the patient's lungs. In this context, $P_{control}$ refers to the pressure set on top of the positive end-expiratory pressure (PEEP), while frequency and the inspiration expiration ratio (I:E) define the breathing cycle. The pressure ramp (PRamp) setting determines how quickly the ventilator increases the delivered pressure to reach the desired level (20).

The study also used (semi)closed-loop ventilation, a ventilation method that continuously monitors the patient's status and adapts to their needs. Several forms of this modality are available, including adaptive support ventilation (ASV), a ventilation mode that maintains a preset minimum minute ventilation (the total volume of air that is breathed in and out by a person in a minute). This mode assumes an optimal breathing pattern and adjusts inspiratory pressure and machine frequency based on changing patient characteristics (20, 21).

Diagnosis of PVA

Currently, the method used to detect PVA is by visually examining the ventilation curves at the bedside using pressure and flow curves. However, research has shown that clinicians in intensive care units identify only a third of all PVAs, suggesting that the true incidence of PVAs is underestimated.

Inexperienced staff are even less likely to recognise PVA (9). To improve and simplify the diagnosis of PVA, adding a reference signal from the patient is important. There are two options for reference signals in mechanically ventilated patients: the oesophageal pressure (P_{es}) or the electrical activity of the diaphragm (EAdi) (22). The oesophageal pressure (P_{es}) was used in this study.

Oesophagus pressure (P_{es})

As described above, adding the P_{es} improves the diagnosis of PVA, and therefore, the P_{es} curve plays a significant role in the context of ventilation (16). The P_{es} curves provides valuable information regarding the patient's respiratory effort and the initiation of inhalation. The oesophageal pressure curve is particularly useful in detecting patient-ventilator asynchronies, as it serves as a reliable marker to identify abnormalities in the timing and coordination between the patient's respiratory effort and the mechanical support provided by the ventilator. The P_{es} measurements enhances the accuracy and effectiveness of

asynchrony detection during ventilation (23). In the LUMC, an oesophageal balloon is inserted into each patient who is predicted to require mechanical ventilation for over 24 hours.

Types of Patient-Ventilator Asynchronies (PVA)

Patient-ventilator asynchrony occurs when there is a mismatch between the patient and the ventilator. There are several types of asynchronies: *early cycling* also known as *short cycling* or *premature cycling*, *delayed cycling* also known as *late cycling* or *prolonged cycling*, *auto-triggering*, *ineffective triggering* also known as *ineffective inspiratory effort (IEE)*, *double triggering*, *reverse triggering*, and *flow asynchronies*. The asynchronies can be divided in the following categories: *cycling asynchronies*, *trigger asynchronies* and *flow asynchronies*. Examples of the different asynchronies can be seen in figure 1.

Cycling asynchronies: Early cycling and delayed cycling

Early cycling occurs when expiration of the ventilator begins before the inspiratory effort of the patient is complete. This can be easily visualised using the Pes curve, see figure 1. Where the expiratory phase begins before the inspiratory effort is complete. This form of asynchrony is more common in PCV than in PSV. In PCV, the inspiratory time is set, whereas in PSV, the expiratory trigger sensitivity (ETS) is defined. This value represents a percentage of the peak inspiratory flow at which the ventilatory cycles from inspiration to exhalation (24, 25). However, it is important to note that in PSV, higher percentages of the set peak flow can lead to more early cycling, this results in a shorter inspiratory time and therefore may the ventilator terminate inspiration before the patient has completed their full breath (26). This asynchrony can lead to double triggering (see trigger asynchronies) (6, 27).

Delayed cycling occurs when the ventilator's inspiratory time is longer than the inspiratory effort of the patient. Mechanical insufflation continues after the inspiratory effort of the patient has stopped, or even during expiration. This can be seen with an upward trend in the Pes curve during the mechanical breath, indicating that the inspiration effort has stopped. This can be caused by a prolonged inspiration time set by the ventilator, or an air leak. Risk factors for delayed cycling are chronic obstructive pulmonary disease (COPD) and asthma and can contribute to hyperinflation in these patients (6). In PSV, this asynchrony occurs when the ETS is set too low; by increasing the ETS, the expiratory phase is prolonged and the inspiratory phase is shortened (24).

Trigger asynchronies: Auto-triggering, Ineffective triggering, Double triggering, Reverse triggering

Auto-triggering occurs when the ventilator delivers a mechanical breath without an inspiratory effort. This can occur from factors related to the ventilator system or related to the patient. Ventilator-related causes include air leaks within the system, excessively high trigger sensitivity, with the threshold for a trigger set to a low value, or due to water accumulation in the ventilator tubing. On the other hand, patient-related factors can be intrathoracic pressure fluctuations induced by cardiac activity. The Pes curve serves as a visual tool for detecting auto-triggering, where an absence of a triggered breath can be seen (6, 27).

With *ineffective triggering*, there is an inspiratory effort from the patient without the machine administering mechanical breathing. Several factors can contribute to ineffective triggering, including low trigger sensitivity with a high trigger threshold value, respiratory muscle weakness, reduced respiratory drive, inadequate programmed positive end-expiratory pressure (PEEP) and dynamic hyperinflation (auto-PEEP) (2, 27, 28).

Double triggering occurs when during one inspiratory effort from the patient two mechanical breaths are delivered. Double triggering leads to breath stacking, which is known to have adverse effects. Factors that can cause double triggering are a short ventilator inspiratory time. Risk factors for double triggering are patients with acute respiratory distress syndrome (ARDS) and protective ventilation (2, 27).

Reverse triggering occurs when mechanical breath delivered by the ventilator triggers a neural response in the patient, leading to involuntary patient effort and diaphragmatic contraction. This results in a sequence of two consecutive mechanical breaths without adequate expiratory time. The first breath is initiated by the ventilator, based on controlled ventilation, the second breath is initiated by the patient due to a reflex contraction of the diaphragm (17). ARDS is a known risk factor for reverse triggering. If it occurs periodically or synchronously with different patterns it is mostly seen in awake sedated patients (27).

Flow asynchronies occur when the flow generated by the ventilator does not meet the flow required by the patient. This type of asynchrony occurs mainly in volume-controlled ventilation (2). As volume-controlled ventilation is not used on the ICU at the LUMC, it will not be discussed further here.

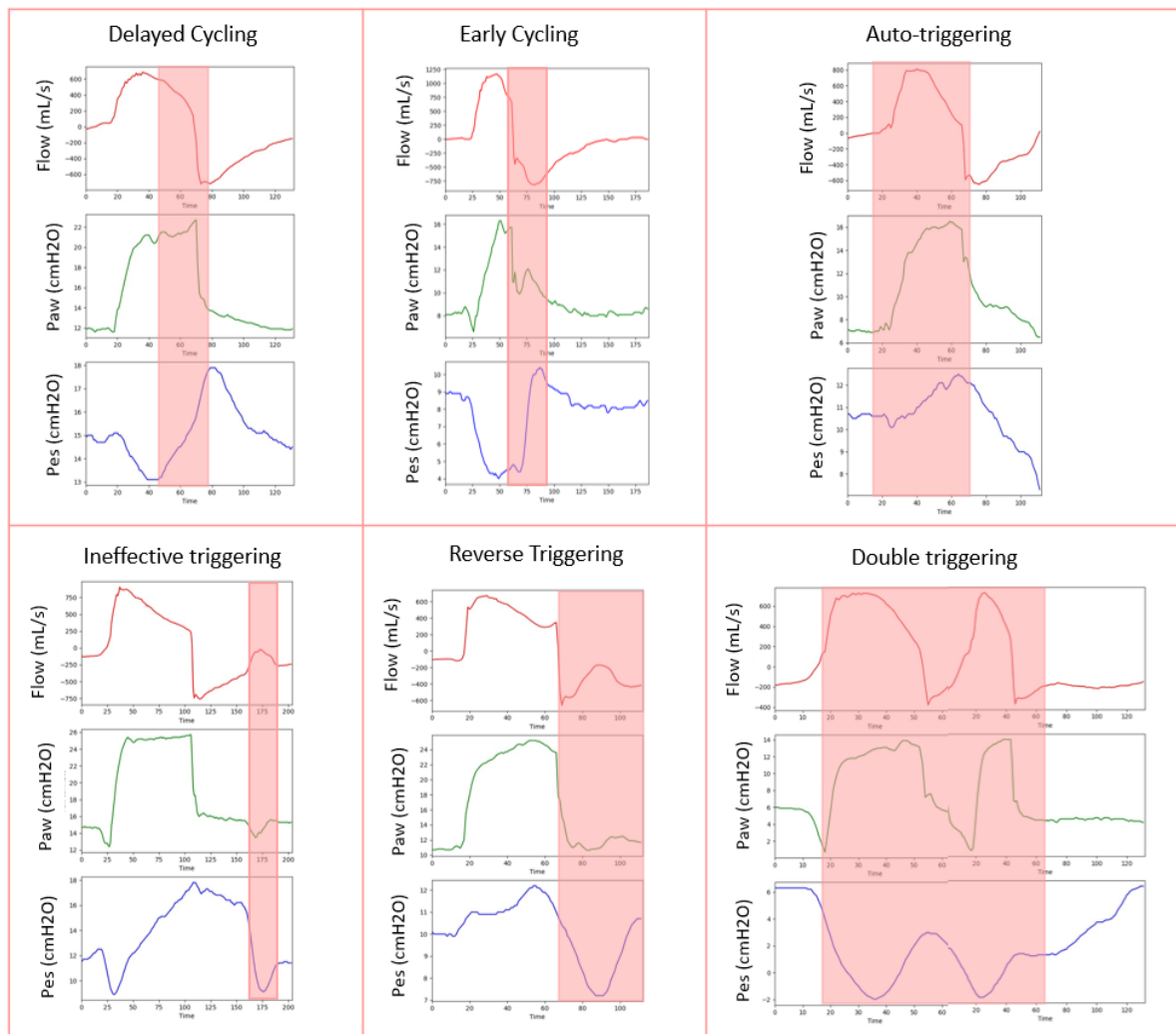


Figure 1. Examples of different forms of Patient-Ventilator Asynchrony, with the asynchrony marked in pink.

Technical Background

Artificial intelligence (AI) is a rapidly advancing field of computer sciences that focuses on creating intelligent agents that can perform tasks that typically require human-level intelligence (29). Machine Learning (ML) is a subfield of AI that focuses on the development of algorithms and statistical models that enable computer systems to learn from and make predictions or decisions based on input data (30). Machine learning can be broadly categorized into two types: supervised and unsupervised techniques.

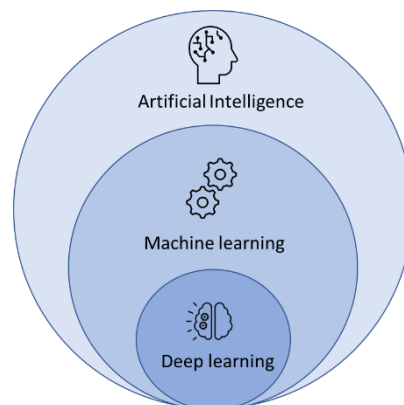


Figure 2. Fields of Artificial Intelligence

Supervised learning

In supervised learning, the algorithm is trained using labelled data, which has been annotated or categorized by humans. The labelled data helps the algorithm to learn to recognize specific patterns in each class, making it capable of performing predictions on new, unseen data. The disadvantage of supervised learning is that it requires labelled data, which is known as a time consuming process (31).

Deep learning

Deep learning is a subset of machine learning that uses multiple layers of learning. Each layer has its own function within the neural network, and in deep learning, more layers are used to extract key information from the data, making it a more advanced learning technique. Unlike classical machine learning, where the user has to select the important features, deep learning automatically selects the features using the multiple layers. This is a significant advantage of deep learning (32).

Convolutional neural network (CNN)

CNN also referred to as ConvNet is a supervised deep learning technique that is used for solving complex problems. CNN is a technique that is mainly used for classification based on contextual information (33). And is mostly used in image classification, object recognition and speech recognition (33, 34).

A typical CNN model has a single input and output layer with multiple hidden layers. With A CNN model is built of several components, including a *convolutional layer*, *pooling layer*, *activation function* and *fully connected layer*.

To address the aim of this study, a CNN model was developed for image classification. The model takes images as input and predicts the content of the image. For the purposes of this thesis, the input data to the CNN was assumed to be an image. To do this, the existing patient data from the ventilator must be converted into an image using resizing techniques, which will be explained in more detail in the Methods section.

Convolutional layer

The first step of a CNN involves the convolutional layer. Feature extraction is a crucial step within the *convolutional layer*, wherein the most informative features for classification of the input data, in this case an image are identified and extracted. For the feature extraction in the CNN model a convolutional operations takes place which involves moving a kernel across the input data, a 2D image, enabling the extraction of distinctive features. This process is based on convolutions between the kernel and the input data, which highlight relevant patterns within the image (35, 36).

In image processing, convolution is a mathematical operation between the input image, often represented as 'f', and a small mathematical matrix, typically referred to as a 'kernel' or 'filter', denoted as 'w'. The image is multiplied with the kernel to extract the most important features.

The mathematical expression for the convolution operation is shown in equation (1) (37) (38):

$$g = f * w \tag{1}$$

where 'g' represents the resulting filtered image, also known as the feature map that highlights certain features of the input image, 'f' is the original input image, and 'w' is the kernel or filter (39). It is important to note that this seemingly simple equation conceals a complex process, which is demonstrated in equation (2).

$$g(x, y) = \sum_m^{\infty} \sum_n^{\infty} f(x - m, y - n)w(m, n) \tag{2}$$

Equation 2 explains how each pixel in the filtered image is formed by considering its surroundings in the original image and weighting them by the values represented in the convolution kernel.

The convolution process involves using a small kernel and systematically sliding it over the image. The convolution process involves multiplying the kernel's values with the pixel values in the image at each position and then adding them together to obtain a filtered image. The schematic overview of a convolution operation can be seen in figure 3.

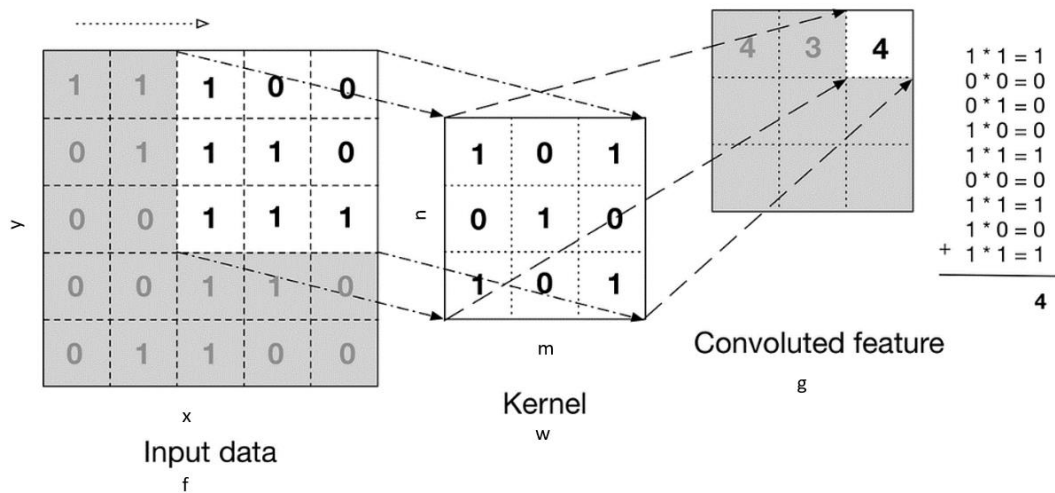


Figure 3. Schematic overview of a convolution operation (1).

Activation Function

After the convolutional layer, a non-linear *activation function* such as ReLU (Rectified Linear Unit) is applied to introduce non-linearity into the neural network. By introducing non-linearity into the model, the CNN model becomes capable of learning and representing more complex and nonlinear patterns in the data (40).

Pooling Layer

After the activation function, a pooling layer is applied to reduce the dimensions of the extracted features, which helps reduce computation time (41). Various types of pooling layers exist, with max pooling being a common choice for CNNs. This method reduces dimensionality by retaining only the most significant information from the input data. Typically, a window size of 2x2 is used for max pooling. This window moves over the input data, retaining only the highest value within the window. As the pooling window moves across the input feature map, it selects the highest value within the window to generate the output. This technique reduces dimensionality while retaining the most important information (33, 39). The process of Max pooling with the use of a 2x2 kernel can be seen in figure 4.

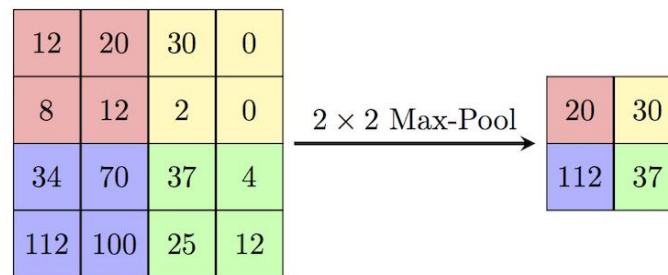


Figure 4. Example of an application with an 2x2 max-pooling

(42)

After the application of convolutional and pooling operations, the next step in the CNN-process are the fully connected layers also known as the hidden layers. This takes the output of the convolutional and pooling layers and predicts the best label to the image. Before entering the fully connected layers, the data is flattened into a 1-dimensional array for preprocessing (36).

Overview of a CNN model is visualized in figure 5.

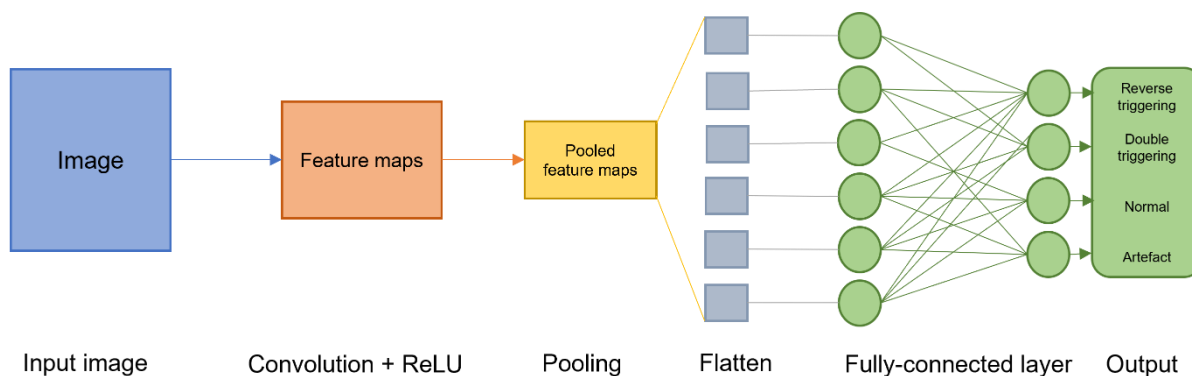


Figure 5. Schematic overview of a CNN model

During training, the CNN network determines the optimal parameters for achieving the best model performance using backpropagation and a loss function. The loss function compares predicted and actual outcomes, with a high loss indicating incorrect predictions and a low loss indicating accurate predictions. Backpropagation is the process by which the model learns, iteratively adjusting its

parameters based on the gradient of the loss function. This process improves the model's performance to achieve the highest possible accuracy (43).

A great advantage of CNNs lies in their ability to automatically learn relevant features from raw image data, eliminating the need for manual feature selection. The automatic feature extraction improves the model ability to handle complex visual patterns and variations, making CNNs highly effective in image classification tasks.

Unsupervised learning

In contrast to supervised learning, unsupervised learning aims to discover patterns, structures or relationships in unlabelled data. It explores data without predefined labels. Unsupervised machine learning can be broadly categorized into four main techniques (44):

1. Clustering is a method in unsupervised machine learning that groups similar data points together based on their characteristics. This helps identify patterns or groupings in the data that may not be immediately apparent. Based on this information data that is similar to each other can be grouped together.
2. Dimensionality reduction techniques, like principal component analysis (PCA) These techniques aim to reduce the number of features or dimensions in a dataset while preserving as much information as possible. This can be useful for tasks such as data visualization, feature selection, or noise reduction. This can be useful for tasks like data visualization and noise reduction (45).
3. Generative models, These models can learn the underlying distribution of the data and generate new samples that resemble the original data. This can be useful for tasks such as data generation, anomaly detection, and data augmentation (46).
4. Anomaly Detection, this technique involves identifying data points that deviate significantly from the norm or expected behavior. Anomaly detection methods can be statistical or machine learning-based and are used to identify outliers or anomalies in the data.

There are several types of techniques that can be used to detect anomalies, one of which is distance-based techniques. Distance-based techniques involve the determination of the similarity of the data points with respect to each other. Data points that are similar to each other are considered to be close together in the feature space, while data points that are dissimilar are far apart. Outliers, or anomalies, are data points that are significantly different from the rest of the data, and therefore have the least similarity to other data points (47).

This study will utilize a CNN algorithm that requires labelled breaths images as input data to recognize PVA. Labelling breaths and distinguishing normal from abnormal breaths is a time-consuming task due to the prevalence of normal respiratory data compared to PVA data. In the context of this study, unsupervised machine learning techniques can provide an option for automatic labelling, making the labelling process less time-consuming. Automated data labelling is best achieved through the use of anomaly detection methods, as PVAs are considered abnormal breaths and therefore behave like anomalies.

The main objective of this master's thesis is to develop a supervised CNN machine learning algorithm capable of detecting patient-ventilator asynchrony automatically. Additionally, as a secondary research objective is to identify an unsupervised machine learning technique for data labelling, with the goal of

automating labelling process for a subset of the data and, as a result, reducing the time necessary for labeling the entire dataset.

Methods

Data collection

This study was conducted at the ICU of the LUMC in Leiden, the Netherlands. Patients who were mechanically ventilated and had an esophageal balloon inserted were eligible for inclusion. Esophageal balloon placement is contraindicated in patients with specific conditions, including oesophageal varices grade 3 or higher, recent gastrointestinal bleeding (<1 month, relative), thrombocytopenia < 50 (relative), gastroesophageal malignancy, history of gastroesophageal surgery, other diseases of the oesophagus or thorax that may lead to complications or unreliable measurements, and pneumonectomy (48). Once eligible patients were identified, a memory box was connected to the ventilator to record relevant data, including Paw, flow, volume and Pes curves, along with additional information of the ventilator, such as time, breath number and the timing of inspiration and expiration. Each patient's data was recorded at least once per admission, providing up to 5 hours of available data per recording. Ethical approval for this study was obtained from the non-WMO committee of division 1 of the LUMC, protocol number 2022-061, and informed consent was obtained from the patient's legal representative.

Data Labelling

In order to prepare the ventilator data to be used for the training of the machine learning model, the captured patient data was labelled using the open source data labelling platform Label-Studio (49). This platform allows for simultaneous labelling of all the respiratory curves, including the Paw, Pes and flow curves. To ensure accuracy and consistency in the labelling process, a data annotation protocol was written and followed during the labelling process. The data annotation protocol can be found in Appendix B. An hour of data with the most asynchronies, referred to as the region of interest (ROI), was extracted from each patient's recording. This data was then annotated by a single clinician following the developed protocol. The data was categorized into different labels representing different types of asynchrony. These labels include *reverse triggering*, *auto-triggering*, *ineffective effort during expiration (IEE)*, *early cycling*, *delayed cycling*, *double triggering*, *cough*, *peristalsis*, and *other artefacts*. Data classified as 'normal' was not labelled during the label process in Label-Studio, but was automatically labelled as 'normal' when loaded into Python. To ensure the highest quality of labelling, four clinicians with expertise in PVA performed the annotation process. Once the annotation was completed the time series data was downloaded from Label-Studio and further processed in Python.

Data pre-processing in Python:

After annotating the time series dataset, the data is retrieved from Label-Studio and stored in a CSV file, which is imported into Python. This dataset now includes the annotations provided by clinicians, where each time series data point has a label. In addition to the Paw, pes and flow curves the dataset is added with information from Hamilton Medical. This information consists of the breath number, where each breath is numbered since the start of ventilation, the moment of inspiration or expiration, and whether the breath is initiated by the patient or delivered as a controlled breath by the mechanical ventilator.

This research is a follow-up to an earlier study on developing an algorithm to automatically detect PVA. In the previous study, we chose to use a CNN algorithm to automatically detect the PVA, this was done using images of a single breath as data input. To create this image, the existing patient data must first be cut into individual breaths from which an image can be created. The further segmentation process is further described below.

Step 1: Data segmentation

The previous study on automatic PVA detection already used an algorithm to split patient data into separate breaths. Further visual analysis shows that this algorithm did not work optimally with multiple breath cycles where it was supposed to be one breath cycle. The basis of this algorithm was to detect the start and end of an inspiration in order to frame a single breath in an image, based on flow acceleration and the moment the flow switched from positive to negative or vice versa. However, Figure 6 shows that the algorithm failed to detect the start of inspiration. This algorithm led to incorrect splitting where multiple breaths were combined where they should have been one breath. This can be seen in figure 7.

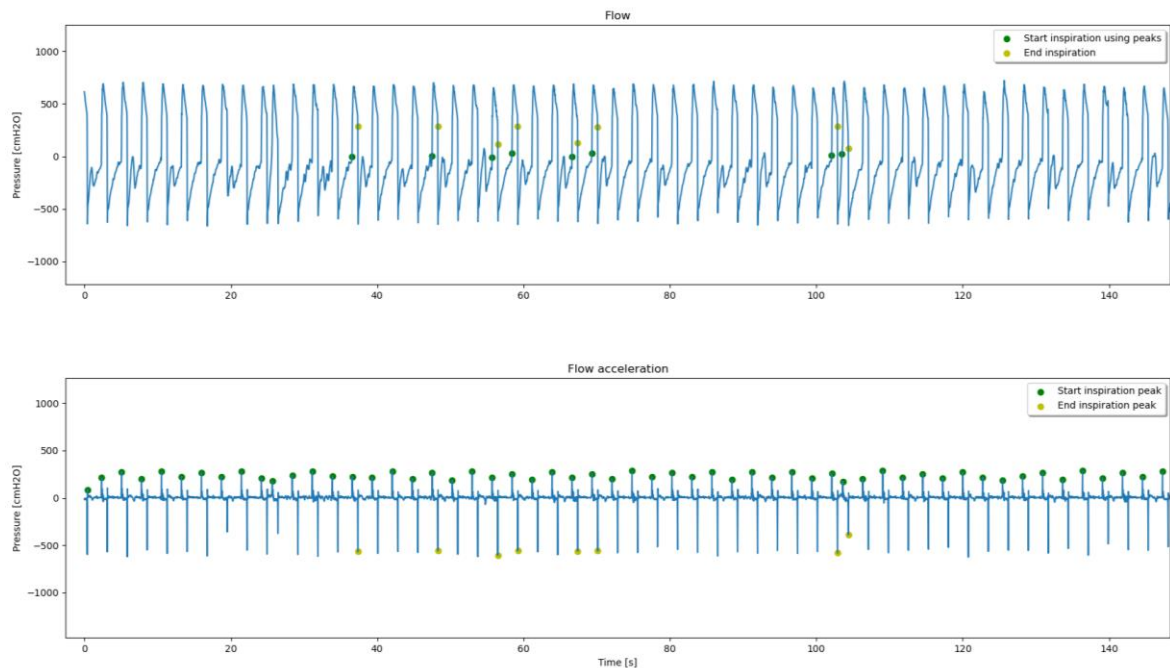


Figure 6. Problems detecting Inspiration with the time series data

To solve this problem, a new method of splitting the data into separate breaths was investigated and a method using information from the Hamilton Medical ventilator was chosen. The ventilator registers when inspiration starts and stops, based on this information the data can be segmented into individual breath cycles. The exact method used to determine the start of inhalation and exhalation by the ventilator is not entirely clear, but it is based on information from when the ventilator starts and stops the mechanical ventilation administered.

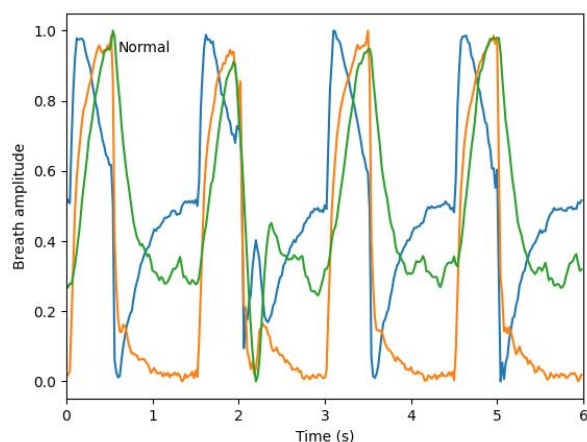


Figure 7. Multiple breaths captured in a single breath frame

$$start = start_insp - \left(\frac{1}{10} * \frac{60}{RR} * 50 \text{ Hz} \right) \quad (3)$$

In equation 3, the variable 'start' represent the indices of the start of the inspiration used in this algorithm. 'start_insp' represents the starting point provided by the Hamilton Medical ventilator, and 'RR' denotes the most frequently observed respiratory rate during the data recording period. This equation calculates the appropriate starting point for each data window, ensuring that each respiratory cycle depicted captures one complete breath cycle along with relevant contextual information. An example of a single breath cycle is shown in Figure 8.

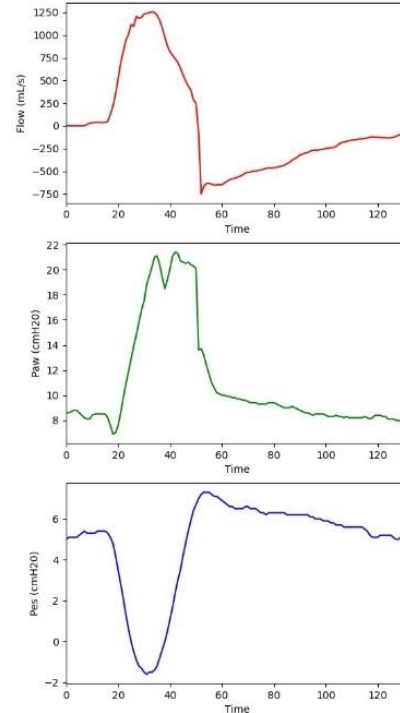


Figure 8. Example of an segmented breath

Step 2: Scaling the data

The time series data is now segmented into individual breaths and includes the information from Hamilton whether the breath is a patient-triggered or a ventilator-provided breath, based on the information given by the ventilator.

A single breath cycle consists of the Paw, flow, and Pes curves. These curves have different units of measurement, resulting in variations in their magnitudes when plotted together in a single image. To optimize the visual representation of these curves and retain their detail within the image, a normalization process is applied. This approach effectively scales the data to have a standard deviation of 1 while centering it around the mean value. By doing so, all three curves can be clearly visualized within the same image, an example can be seen in figure 9.

During the patient recording, regular patient care continued, so it is possible that during the recording, the oesophageal balloon was temporarily disconnected for patient care. During the disconnection or deflation of the balloon, there is no accurate measurement present of the Pes signal. Therefore, signals where the Pes curve is 0 or is non-informative are removed.

Step 3: Converting signals to images.

After scaling the time series of the respiratory signals, the next step in the pre-processing phase relates to visualisation, i.e. converting the 1D time series signals into 2D images. This is done using line plots where the horizontal x-axis represents time and the vertical y-axis represents the scale values of the Paw, Pes and Flow curves, so that each image has a size of Lx3x1, where L is the time length of the breath. The three channels represent the three different curves, the Paw, the Pes and the Flow curve. An example of the visualisation of the image can be seen in figure 9.

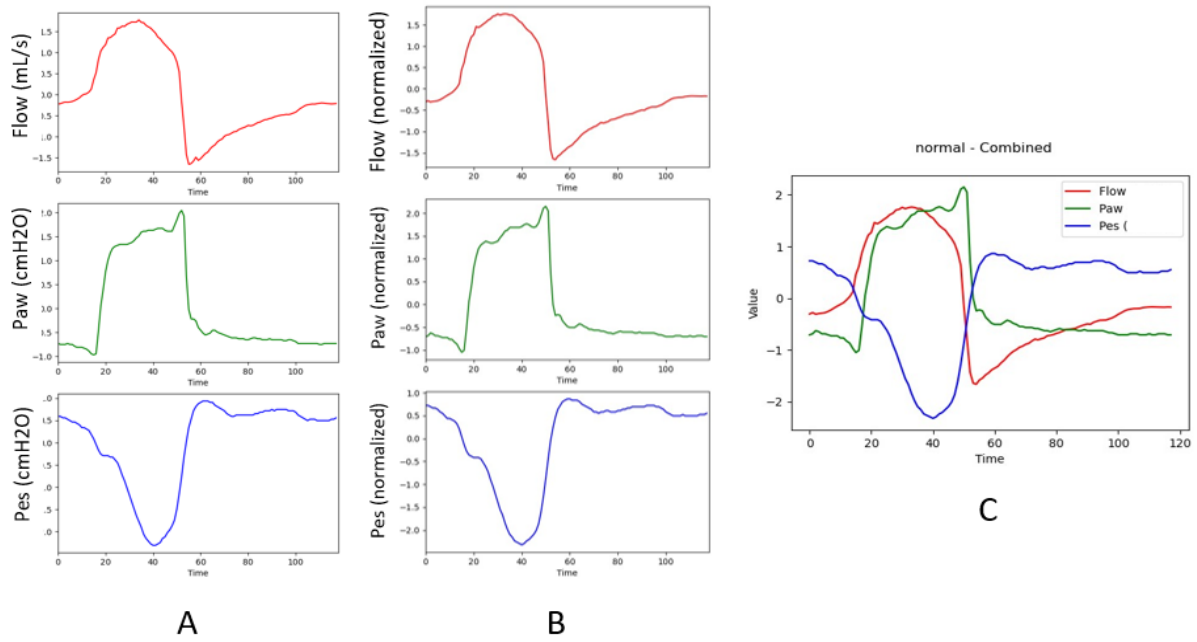


Figure 9. Preprocessing process. A = original breath, B = breath with the normalized curves, C = normalized curves combined into one image
Paw = pressure, Pes = Oesophageal pressure

The dataset consists of respiratory data from multiple patients, resulting in varying respiratory rates and breaths cycle sizes. To ensure uniform image sizes for machine learning model training, a zero-padding technique is applied. This adjust the breath image size to match the largest image in the dataset creating consistent dimensions. The smaller images are filled with 'empty' information at the edges so that this does not affect the rest of the process. This means that zeros are added to the sides. This enhances the model's ability to effectively analyse respiratory patterns.

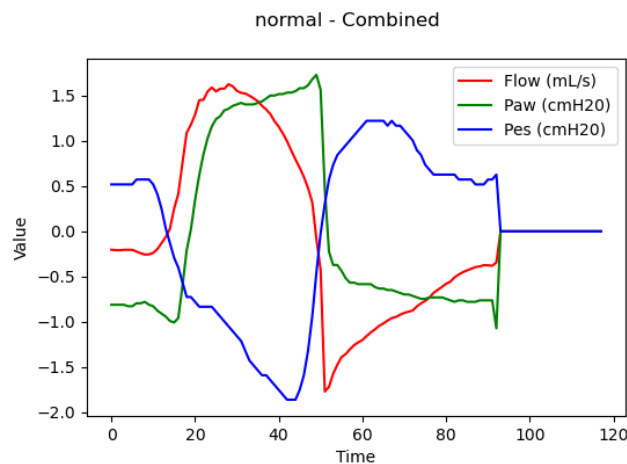


Figure 10. data input of breath image with zero padding at the end of the curve

Mahalanobis distance

During this research, the Mahalanobis distance was used several times. This statistical metric measures similarity/dissimilarity between a data point (breath) and distribution. In the context of multivariate data, where different variables contribute to the overall distribution, the Mahalanobis distance is particularly useful. The formula used to calculate the Mahalanobis distance can be found below (50):

$$D = \sqrt{(x - m)^T \cdot C^{-1} \cdot (x - m)} \quad (4)$$

In equation 4:

- D corresponds to the mahalanobis distance
- x represents the vector of data
- m represents the vector of mean values
- C^{-1} represents the inverse covariance matrix
- T indicates the transpose operation

The Mahalanobis distance serves a dual purpose: it detects outliers within the dataset and quantifies the dissimilarity or proximity of data points to the center of the data distribution. In this context, the center of the data distribution represents the mean of the data. The Mahalanobis distances takes into account the covariance structure of the data and calculates how many data and standard deviations away a data point is from the mean. Data points that are closer to the mean have lower Mahalanobis distances, while those farther away have higher distances. This distance metric is commonly used for anomaly detection and for identifying data points that deviate significantly from the expected data distribution. An example of the Mahalanobis distance of the breaths and their distribution is shown in Figure 11.

By using the Mahalanobis distance for each breath, a clear distinction can be made between normal breaths and those containing artefact/asynchronies. This distinction is achieved by identifying breaths with the largest Mahalanobis distance as potential outliers, while also recognising normal breaths based on those with the smallest Mahalanobis distance.

The breath datapoints can be effectively visualized in the feature space using PCA and the Mahalanobis distance. This visualisation allows for a clear representation of the distribution of breaths along with their corresponding labels. Figure 11 presents an illustrative example of this distribution, providing valuable insights into how different breaths are spatially distributed with their respective labels.

It can be seen that the centre of the distribution is normally labelled breaths. The artefact labelled breaths are in the plots seen as outliers. In addition to the normal distribution there is also a group of breaths labelled as reverse triggering. This shows that the labels are clustered together and that the labels are distributed as their own distribution. This leads to the hypothesis that an unsupervised machine learning technique can be used to distinguish between the different types of PVA.

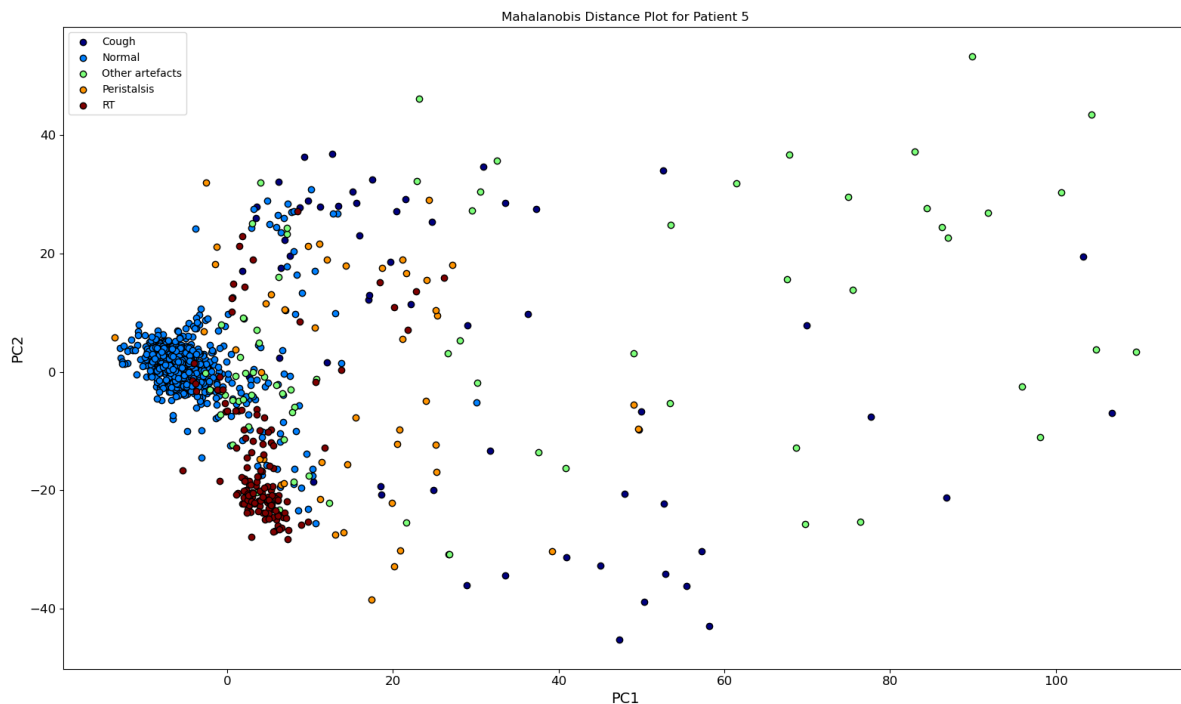


Figure 11. Breaths as datapoint visualised using Principal Component Analysis and the Mahalanobis distance

Unsupervised machine learning: create pre-annotations

A notable disadvantage of this study is that there is no database available for patient-ventilator asynchrony data. As a result, all data collection and labelling tasks have to be undertaken by ICU researchers at the LUMC. This is particularly labour intensive in a healthcare setting where time is already a limited resource. To speed up some of the time-intensive labelling process, the use of an unsupervised machine learning algorithm was explored, with the use of the Mahalanobis distance, to determine the feasibility of partially automating the labelling process.

In automating the labelling process, there is a particular interest in labelling 'normal' breaths in advance or automatically, as most of the data is 'normal', causing it to be the most time-consuming labelling activity. It is therefore investigated whether the Mahalanobis distance can be used as an unsupervised machine learning algorithm to automatically distinguish between normal and abnormal breaths. This is possible because most of the breaths in the dataset are normal, and so this becomes the centre of the data, so based on the Mahalanobis distance formula above, the normal breaths get the smallest Mahalanobis distance. To automatically distinguish between the normal and abnormal breaths, a threshold of Mahalanobis distance is searched for, below which the breaths are always normal.

To establish a threshold, we utilise data from patients that have already been manually labelled by clinicians. It was visually verified that breaths labelled by clinicians were actually normal breaths. By using different thresholds, we determined the range within which 95% of breaths were classified as normal. This percentage was chosen to avoid missing too many normal breaths, which would require manual labelling. If the threshold is lowered significantly, the Mahalanobis distance may label asynchronous breaths as normal. It is important to avoid both of these situations. Various thresholds were tested using a histogram of the labelled data, which displayed the most frequent Mahalanobis distance per label. In particular, the Mahalanobis distances of the normally labelled breaths were considered.

To determine the threshold, the following formula has been used:

$$D95 = \frac{(\sum D_{normal} < thr)}{\sum N_{normal}} \times 100\% \quad (4)$$

- $D95$ represents the Mahalanobis distance threshold where 95% of the normal breaths are labeled as normal
- D_{normal} is the Mahalanobis distance of normal breaths of a patient labelled by clinicians
- Thr is a chosen threshold for the mahalanobis distance
- N_{normal} is the total count of normal breaths of a patient labelled by clinicians

Equation 4 establishes the percentage of breaths that can be accurately pre-labelled using the unsupervised machine learning algorithm. This is achieved by dividing the number of breaths with a Mahalanobis distance less than the threshold by the total number of normal breaths. This process is repeated for several thresholds, and the threshold that prelabels more than 95% correctly is used for prelabelling the breaths.

Labelled data

Visual analysis of the recorded patient data revealed that certain breaths were inaccurately labelled, particularly those labelled with the label 'normal'. These breaths had visual artefacts indicating possible mislabelling. The Mahalanobis distance was used to assess the abnormality of the breaths. From the breaths that were labelled *normal*, it was also found that the breaths with the visual artefacts had a Mahalanobis distance greater than 10, which is a clear deviation from the normal distribution. This can also be seen by creating a histogram of all the Mahalanobis distance of all the breaths and their corresponding labels, as shown in figure 12. This histogram shows that all breaths labelled as *normal* exhibit Mahalanobis distances falling within the range of 0 to 5. It is presumed that any breaths with a distance exceeding 10 deviates significantly from the norm and is considered an anomaly. This assumption aligns with visual evidence, as depicted in figure 13. In order to train the machine learning model with the correct data, the breaths that are anomalies, with a Mahalanobis distance >10 and are labelled 'normal' by the clinicians were changed to the label 'other artefacts'.

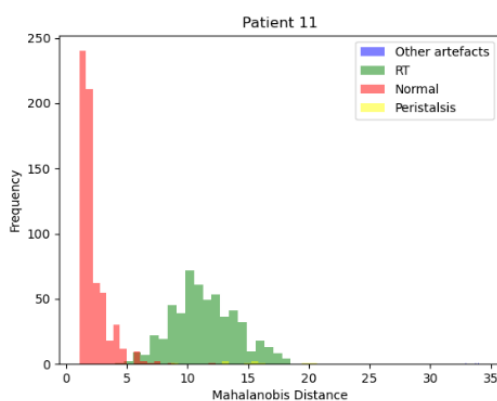


Figure 12. Histogram of the mahalanobis distance of all breaths from patient 11.

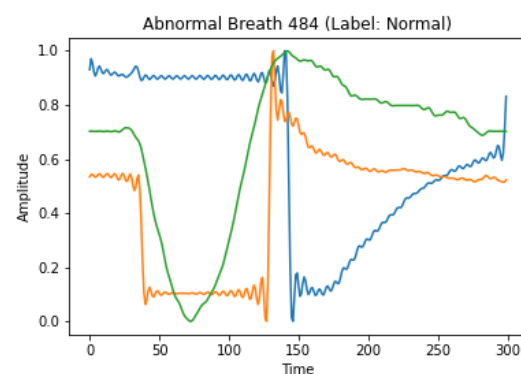


Figure 13. Wrong label to a normal breath, showing a breath with artefacts

CNN Model

Architecture

A CNN model consists of multiple layers, which are described in the technical background section. In this study, an existing CNN model built on TensorFlow was used, and its architectural details are shown below. The architecture of the CNN model remains unchanged, with the exception of the input data format. The new input data format changed from the conventional 300x3x1 dimension to Lx3x1, where L is the time length of the largest breath.

The CNN model consists of four blocks, each containing a 2D convolutional layer, a batch normalization layer, and an activation function that uses ReLU. Additionally, pooling is included in each block. After these four blocks, there is a fully connected layer with a softmax filter, followed by a dropout layer.

A visualisation of the architecture of the CNN model is shown in figure 14 and more information on the layers of the CNN in table 1.

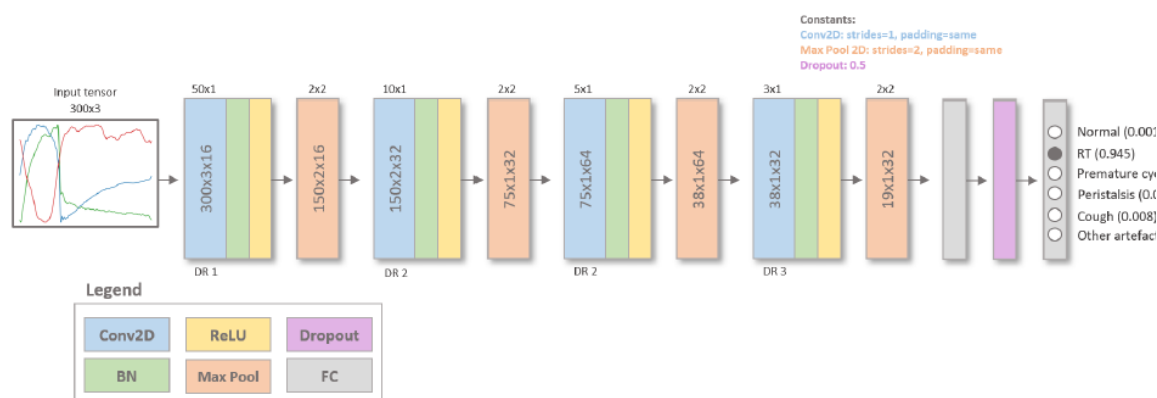


Figure 14. Schematic overview of the architecture of the CNN model

Table 1. Overview of the different layers of the CNN model

Layers	Types	Dilation rate	Activation function	Output shapes	Size of kernel	No. of kernels	Stride	No. of parameters
0	Input	-	-	300 x 3	-	-	-	0
1	2D Convolution	1	ReLU	300 x 3 x 16	50 x 1	16	1	816
2	Batch Normalization	-	-	300 x 3 x 16	-	-	-	1200
3	2D Max Pooling	-	-	150 x 2 x 16	2 x 2	-	2	0
4	2D Convolution	2	ReLU	150 x 2 x 32	10 x 1	32	1	5152
5	Batch Normalization	-	-	150 x 2 x 32	-	-	-	600
6	2D Max Pooling	-	-	75 x 1 x 32	2 x 2	-	2	0
7	2D Convolution	2	ReLU	75 x 1 x 64	5 x 1	64	1	10304
8	Batch Normalization	-	-	75 x 1 x 64	-	-	-	300
9	2D Max Pooling	-	-	38 x 1 x 64	2 x 2	-	2	0
10	2D Convolution	3	ReLU	38 x 1 x 32	3 x 1	32	1	6176
11	Batch Normalization	-	-	38 x 1 x 32	-	-	-	152
12	2D Max Pooling	-	-	19 x 1 x 32	2 x 2	-	2	0
13	Fully connected	-	ReLU	256	-	-	-	155904
14	Fully connected	-	Softmax	6	-	-	-	1542

Data input

Significant differences exist in the normal breathing patterns between different ventilation modes. Figure 15 and 16 show the PSV (spontaneous) and PCV (P-CMV) normal breathing patterns, respectively. To train the machine learning algorithm accurately, different machine learning models are developed for each ventilation mode. This can be achieved by categorising the patient's breaths according to data from the Hamilton ventilator, distinguishing between a mechanical breath (PCV) and a supporting breath (PSV).

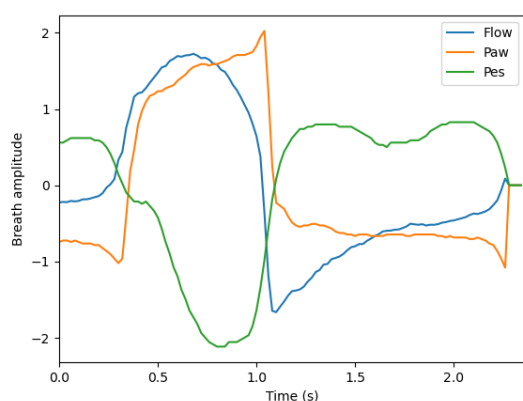


Figure 15. PSV – normal breath

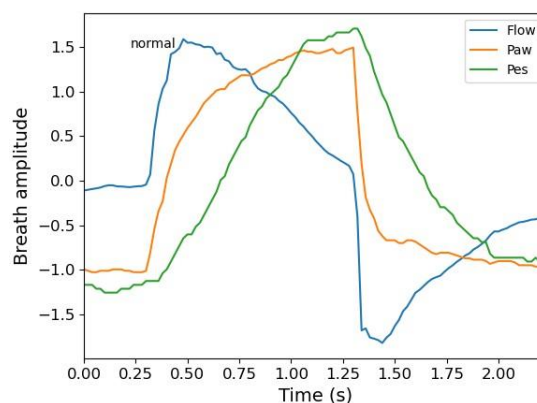


Figure 16. PCV – normal breath

As a result, the patient data is classified into two main groups: PCV data or PSV data. This division serves as the basis for creating two separate and specialised machine learning models. One model deals exclusively with PCV data while the other one is specifically designed for PSV breathing data. In addition to the PCV and the PSV modes, the LUMC ICU also uses a (semi)closed-loop ventilation mode, specifically the Adaptive Support Ventilation (ASV) mode. ASV allows the patient to initiate breaths independently while also providing controlled mechanical breaths if the patient's spontaneous breathing is insufficient. Due to the use of both controlled and supported breaths in this ventilation mode, the development of a machine learning model for this specific ventilation mode is challenging, as the data provided by Hamilton is not sufficient to differentiate between ASV mode and PSV or PCV. Therefore, patient-triggered breaths are part of the spontaneous group and machine-triggered controlled breaths are part of the P-CMV group. This may lead to the division of breaths from a single patient in both the spontaneous and P-CMV groups.

In addition to the PSV and PCV models, a further model has been developed using the entire dataset (ALL), consisting of all breaths of all patients, both spontaneous and controlled. Therefore, a total of three models have been created, namely ALL, PSV, and PCV.

Cross-validation

Following the completion of data pre-processing, the dataset undergoes division into distinct groups for model training. To minimize the risk of overfitting, the Leave-One-Out-Cross Validation (LOOCV) method

is used. With this method the dataset is divided into multiple groups, with each cross-validation iteration involving one group serving as the validation set while the model is trained on the remaining groups (51).

In a more specific context, during a

LOOCV fold, the data of a particular patient is intentionally excluded from the training process to prevent the model's exposure to that specific patient's breaths. Subsequently, the model's performance is then assessed using the withheld patient data to evaluate its generalization capabilities to previously unseen examples (52).

In addition to the validation set for each fold, a distinct test dataset was also created, which is entirely excluded from the model's training. The model's performance on this test set is subsequently evaluated.

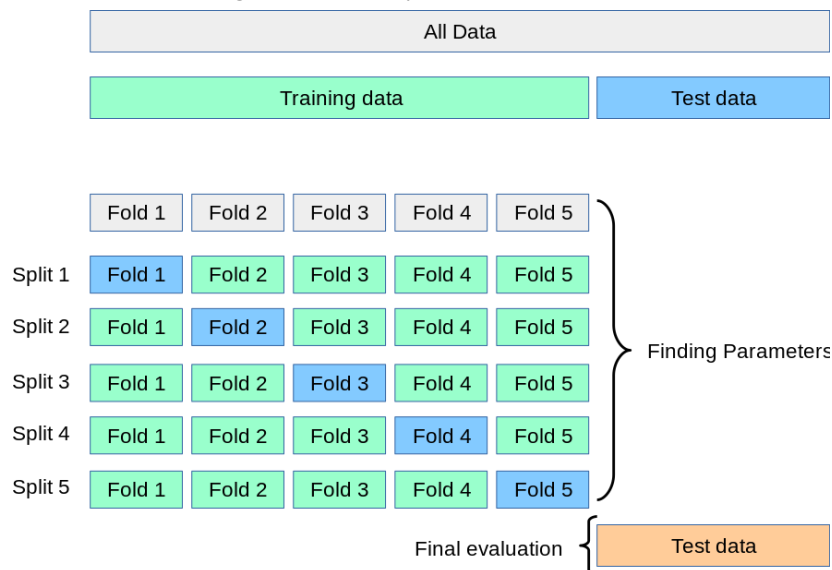


Figure 17. Schematic overview of Leave-One-Out-Cross Validation
The process of a leave-one-group-out cross-validation is shown in Figure 17 (52).

Performance

A number of different performance measures are used to assess the performance of the model. To understand the principles of these performance metrics, it is first necessary to introduce the following concepts that are shown in table 2:

Table 2. Confusion matrix based on normal breathing an Patient-Ventilator Asynchrony (PVA)

	PVA	Normal breathing
Model predicts PVA	True Positive (TP)	False Positive (FP)
Model predicts Normal breathing	False Negative (FN)	True Negative (TN)

The following performance metrics were used:

Sensitivity, also known as the true positive rate, represents the proportion of actual positive cases (PVA) that the model correctly identifies. It measures the model's ability to correctly classify PVA cases as positive. So sensitivity shows the model ability in correctly identifying positive cases, a crucial aspect in a clinical context. The formula to calculate the sensitivity can be seen in formula 7 (53).

$$Sensitivity = \frac{True\ positive}{(True\ positive + False\ negative)} \quad (7)$$

Specificity: Specificity shows the model's ability to accurately identify negative cases, which is of great importance in clinical applications. In this context, how good is the model's ability to correctly classify normal breaths as normal? The formula to calculate the specificity is seen in formula 8 (53).

$$Specificity = \frac{True\ negative}{(True\ negative + False\ positive)} \quad (8)$$

AUROC (Area Under the Receiver Operating Characteristic Curve): The AUROC is a performance metrics that is used to evaluate classification models. It assesses the

model's capability to distinguish between positive and negative cases, the discrimination of the model, and is widely recognized as a golden standard for evaluating model performance. In the context of PVA and normal breathing, the AUROC measures how well the model can discriminate between these two categories. An example of the ROC curve can be seen in figure 18. The AUROC is based on sensitivity (y-axis) and 1-specificity (false positive rate) (x-axis).

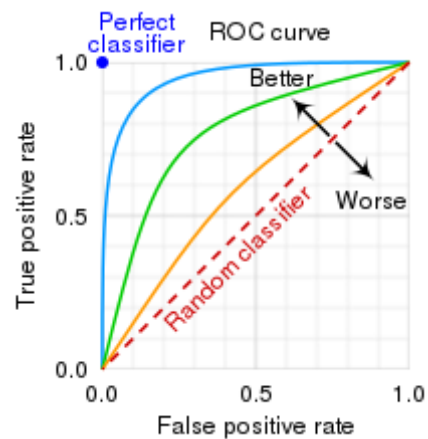


Figure 18. Example of an ROC curve

1-specificity, is known as the false positive rate, which is the proportion of actual negative cases (normal breathing) that the model incorrectly classifies as positive (PVA). It measures the tendency of the model to misclassify normal breathing as PVA (54).

The AUROC is the area under the ROC curve and can be and can be classified into the following performance classes (54, 55) (56).

- AUROC of 0.5 suggests no discrimination
- AUROC < 0.6 poor discrimination
- AUROC of 0.6-0.75 possibly helpful
- AUROC > 0.75 clearly useful.

F1-score: The F1-score combines two crucial metrics: precision (also known as positive predictive value), which measures the model's ability to make correct positive predictions, the proportion of correctly identified positive cases out of all cases predicted as positive. And recall also known as the sensitivity captures the model's ability to find all the positive cases, measuring the proportion of correctly identified positive cases out of all actual positive cases.

The F1-score is displayed in a score between 0 and 1. This metric proves particularly valuable in scenarios with imbalanced datasets, such as those in this study where normal breaths are far more frequent than PVA instances.

$$F1 = \frac{2 * (precision * recall)}{(precision + recall)} = \frac{2 * TP}{2 * TP + FP + FN} \quad (5)$$

Accuracy: Accuracy shows the proportion of correctly classified cases out of the total predictions made by the model.

$$Accuracy = \frac{(True\ positive + True\ negative)}{(True\ positive + True\ negative + False\ negative + False\ positive)} \quad (6)$$

Final Validation

During this research, several machine learning models were created and compared. To validate the model's performance, new patient data that is not previously seen by the model is predicted. These predictions, generated for individual breaths with the new data, are loaded into Label-Studio, the annotation program. This approach enables clinicians, who have expertise in data labelling, to evaluate the accuracy of the model in classifying unseen data. As clinicians are presently the most widely used and only validated option for detecting PVA, it is important to consider their objectivity and precision in this task.

Results

Data collection

Between January 2023 and August 2023, a total of 25 patients were included and labelled. Among them, 12 were ventilated in PSV mode, 6 in PCV, 5 in ASV mode ((semi)closed-loop ventilation), and 2 received ventilation through both PCV and ASV during the recording. Clinical characteristics are detailed in Table 3. From these patients, a total of 39.856 breaths were analysed. Details on the PVA types and their respective quantities per patient are available in Appendix C.

Table 3. Patient characteristics

Patient	Age (years)	Gender	Reason for admission	ICU length of stay (days)	Length of MV (days)	Ventilation mode
1	40	Female	Cardiac disease	17	17 d	SPON
2	48	Male	Cardiac disease	67	50	SPON
3	75	Female	Sepsis	28	28	ASV
4	61	Male	Postoperative	19	15	SPON
5	65	Female	Sepsis	56	44	SPON
6	57	Female	Respiratory disease	7	7	P-CMV/ASV
7	55	Male	Respiratory disease	7	6	P-CMV
8	44	Male	Cardiac disease	50	46	SPON
9	39	Female	Acute liver failure	8	8	SPON
10	58	Male	Cardiac disease	81	57	SPON
11	73	Male	Respiratory disease	6	6	P-CMV
12	56	Male	Cardiac arrest	46	46	SPON
13	64	Male	Respiratory disease	11	9	SPON
14	61	Male	Respiratory disease	6	6 *T	SPON
15	56	Female	Respiratory disease	51	47	i-ASV
16	73	Male	Cardiac arrest	34	26	SPON
17	75	Male	Respiratory disease	46	34	P-CMV
18	69	Male	Cardiac disease	27	27	P-CMV
19	62	Male	Liver disease	22	4	i-ASV
20	67	Male	Liver disease	120	72	i-ASV
21	64	Male	Cardiac disease	2	2	P-CMV
22	66	Male	Respiratory disease	18	18	i-ASV
23	61	Male	Postoperative	83	83	Spon
24	58	Male	Respiratory disease	19	3	P-CMV
25	44	Male	Respiratory disease	7	6	P-CMV

SPON = PSV (supportive ventilation), ASV = (semi)closed-loop ventilation, i-ASV = (semi)closed-loop ventilation, P-CMV = PCV (controlled ventilation)

Unsupervised Machine learning

The Mahalanobis distance was used as an unsupervised machine learning algorithm to partially automate the labelling process.

A first step is to determine the threshold for distinguishing between normal and abnormal breaths. An essential step in this process was the analysis of Mahalanobis distance histograms per patient generated from previously labelled data. These histograms were constructed based on the Mahalanobis distance calculated from the mean and covariance of normal breaths from each patient. The histograms show all the Mahalanobis distances of all the breaths of a patient. To determine an appropriate threshold for pre-labelling breaths as normal, different thresholds were investigated. The percentage of breaths correctly classified as normal was determined using the formula given in the Methods section.

Data from 13 patients were used because the data from these patients had already been manually labelled and were therefore available and could be used to determine the threshold; this threshold could then be used in the study and it was not necessary to label all the data that had not yet been labelled. The new unsupervised machine learning technique was then applied to additional patients.

Table 4. *Threshold Mahalanobis distance*

Patients/Threshold	5	4,5	4	3,5	3
1	97.07	97.99	99.15	99.48	99.69
2	98.64	99.27	99.56	99.88	100.0
3	93.75	94.77	94.59	95.24	97.66
4	99.31	99.40	99.59	99.51	99.45
5	99.85	99.91	99.94	99.97	99.96
6	99.55	99.55	99.64	99.87	100.0,
7	99.79	99.79	99.79	99.79	99.78
8	98.45	98.67	98.89	99.10	99.43
9	99.66	100.0	100.0	100.0	100.0
10	99.94	100.0	100.0	100.0	100.0
11	99.84	99.84	100.0	100.0	100.0
12	16.73	40.82	76.19	96.0	100.0
13	96.60	97.08	97.57	97.93	98.17

After analysing the results, a threshold of 3.5 was selected, resulting in a 95% accuracy in correctly identifying normal breaths, as shown in Table 4.

After applying this threshold, the process of pre-annotating new unlabelled data becomes possible. When unlabelled data is loaded and pre-processed, any breath with a Mahalanobis distance less than 3.5 is assigned the 'normal' label. In order to use the pre-annotations, the labels must be constrained to a format compatible with Label-Studio and aligned with the structure of the original dataset. Labels originally assigned to whole breaths were transformed into labels assigned to individual data points. These modifications allowed the data to be imported into Label-studio with pre-annotated information (Figure 19).

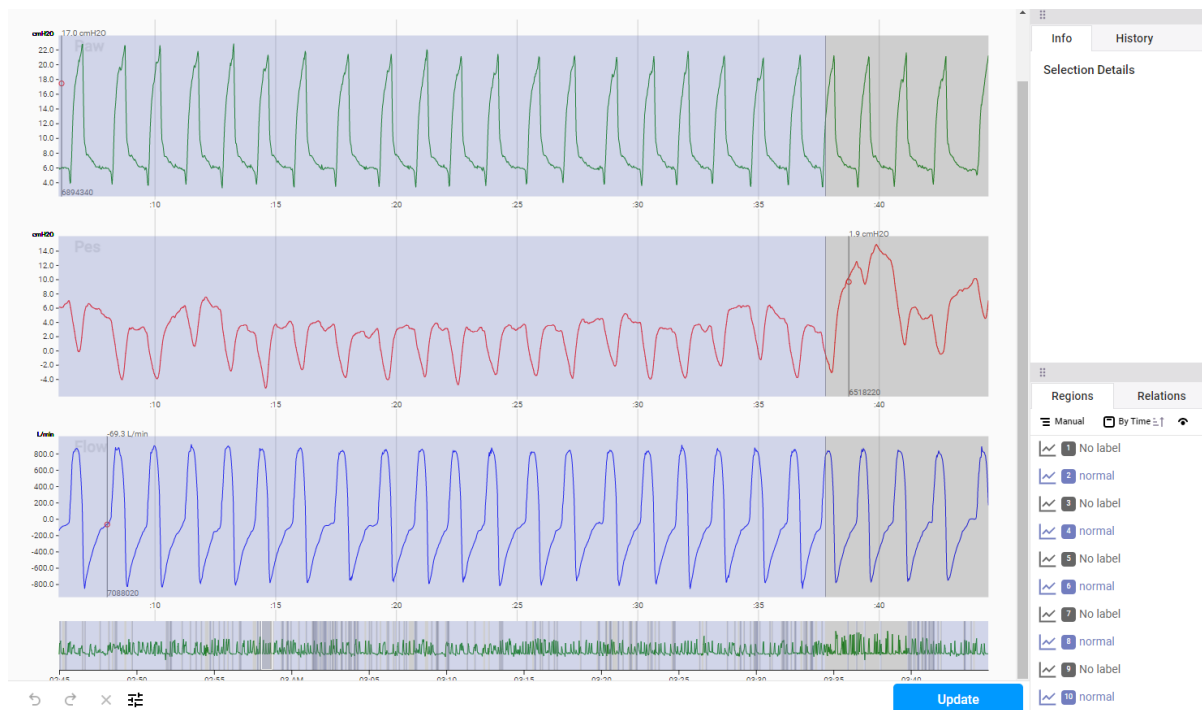


Figure 19. Pre-annotations using unsupervised machine learning

In Figure 19, it is observed that only breaths that are likely to be abnormal require labelling. The pre-annotated data is accurately labelled as normal, resulting in a reduced labelling time.

Supervised machine learning

As described in the methods three different CNN models were built during this study based on the different ventilation modalities. Table 5, 6 and 7 presents the performance metrics of all different models (ALL, PSV and PCV), including AUC, F1 score, accuracy, specificity and sensitivity.

Table 1. Performance of the supervised machine learning models

Ventilation modes: ALL	AUC	F1	Accuracy	Specificity	Sensitivity
Auto triggering*	0.69 (0.09)	0.84 (0) *	0.99 (0.02)	0.99 (0.01) *	0.83 (0) *
Cough	0.84 (0.08)	0.48 (0.22)	0.98 (0.02)	0.99 (0.01)	0.48 (0.25)
DT*	0.91 (0.19)	0	0	0	0
Delayed cycling	0.61 (0.04)	0.66 (0.43)	0.97 (0.03)	0.96 (0.05)	0.65 (0.44)
IEE*	0.41 *(0.11)	0.86 (0) *	0.98 (0.03)	0.94 (0.02) *	0.9 (0) *
Other artefacts	0.81 (0.35)	0.6 (0.2)	0.98 (0.02)	0.99 (0.02)	0.61 (0.21)
Peristalsis	0.83 (0.23)	0.34 (0.27)	0.98 (0.02)	0.99 (0.01)	0.42 (0.2)
Premature cycling	0.67 (0.11)	0.18 (0.34)	0.99 (0.01)	0.99 (0.01)	0.79 (0.1)
RT	0.73 (0.13)	0.43 (0.44)	0.98 (0.01)	0.99 (0.01)	0.76 (0.28)
normal	0.84 (0.13)	0.96 (0.04)	0.96 (0.03)	0.72 (0.19)	0.97 (0.07)

*Based on limited evidence, interpretation should be approached with caution.

Table 2. Performance of PCV model

Ventilation modes: P-CMV	AUC	F1	Accuracy	Specificity	Sensitivity
Cough	0.91 (0.17)	0	0.65 (0.46)	0	0
Other artefacts	0.84 (0.31)	0.43 (0.36)	1 (0)	0.88 (0.31)	0.42 (0.36)
Peristalsis	0.85 (0.08)	0	0.97 (0.04)	0	0
RT	0.82 (0.12)	0.64 (0.39)	0.97 (0.03)	0.96 (0.05)	0.67 (0.41)
normal	0.86 (0.12)	0.85 (0.28)	0.91 (0.14)	0.77 (0.26)	0.86 (0.29)

Table 3. Performance of the PSV model

Ventilation modes: SPON	AUC	F1	Accuracy	Specificity	Sensitivity
Auto triggering	0.74 (0.09)	0	0.97 (0.05)	0	0
Cough	0.87 (0.09)	0.3 (0.09)	0.94 (0.09)	0.98 (0.02)	0.14 (0.12)
DT	0.93 (0.21)	0	0.99 (0)	0	0
Delayed cycling	0.54 (0.04)	0	0.85 (0.28)	0	0
IEE	0.49 (0.03)	0	0.91 (0.12)	0	0
Other artefacts	0.79 (0.20)	0.25 (0.15)	0.78 (0.22)	0.77 (0.25)	0.52 (0.2)
Peristalsis	0.81 (0.09)	0.25 (0.18)	0.95 (0.04)	0.98 (0.02)	0.27 (0.29)
Premature cycling	0.68 (0.04)	0	0.96 (0.05)	1 (0)	0
normal	0.80 (0.23)	0.74 (0.29)	0.77 (0.19)	0.62 (0.17)	0.79 (0.25)

The results indicate that patient-ventilator asynchronies do not occur uniformly across ventilation modes. As seen in the results, there are 5 classes in the PCV mode: Cough, Other artefacts, Peristalsis, RT and normal. This is because controlled ventilation requires less or no inspiratory effort from the patient, so there are fewer instances of asynchrony.

Furthermore, there is considerable variability in the frequency of asynchrony events among patients. Thus, great care is necessary when interpreting the AUC score for auto triggering, double triggering (DT), and IEE.

AUROC curves for Normal breaths

Figures 20, 21 and 22 illustrate the AUROC curves for three distinct models: one trained on all ventilation modes, one specific for PSV and one for PCVV ventilation. These AUROC curves show the classification performances of normal breaths.

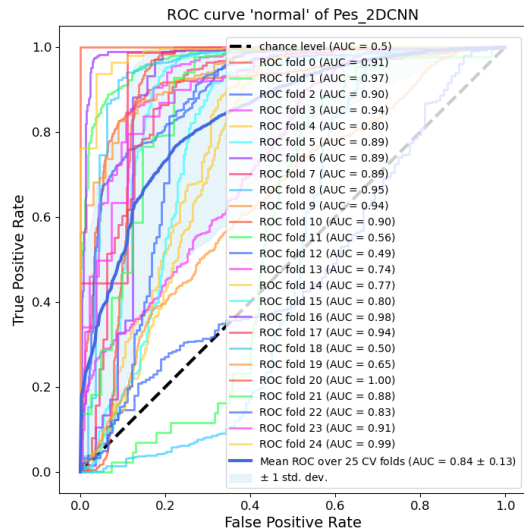


Figure 20. Performance for the normal breaths, model = ALL

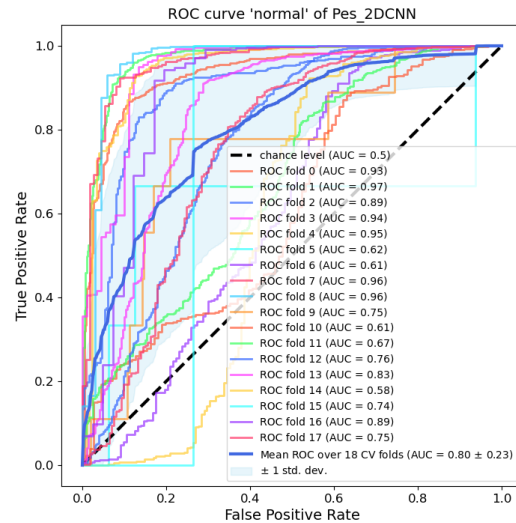


Figure 21. Performance for the normal breaths, Model = SPON

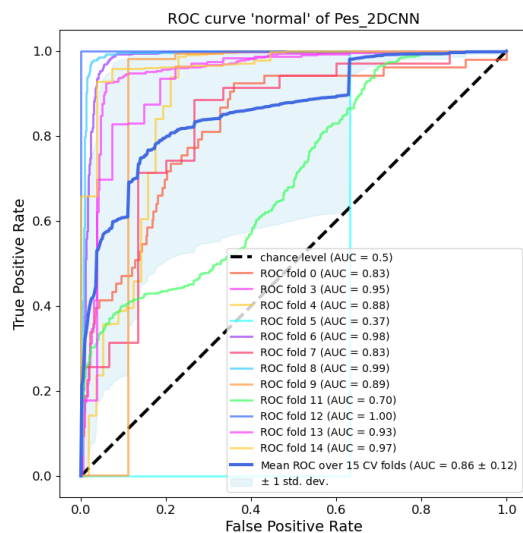


Figure 22. Performance for the normal breaths. Model = P-CMV

These results demonstrate the discriminative capability of the various models across all ventilation modes, PSV and PCV ventilation. They can differentiate between normal and abnormal breaths, as indicated by the AUROC values of $0.85(\pm 0.08)$, $0.83(\pm 0.12)$, and $0.80(\pm 0.28)$ respectively. However, it can be observed that there is significant variation in the model's ability to distinguish distinct patterns of respiration between patients, suggesting substantial differences in ventilation patterns from patient to patient.

Reverse Triggering

Below are the AUROC curves for reverse triggering of the model trained on all ventilation modes (Figure 23) and P-CMV mode (Figure 24).

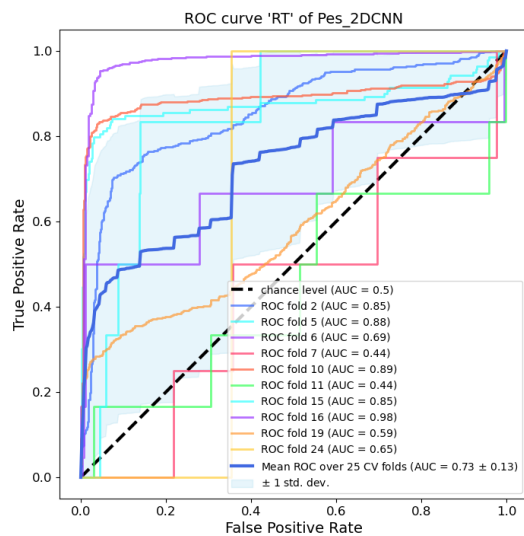


Figure 23. Performance RT, model = ALL

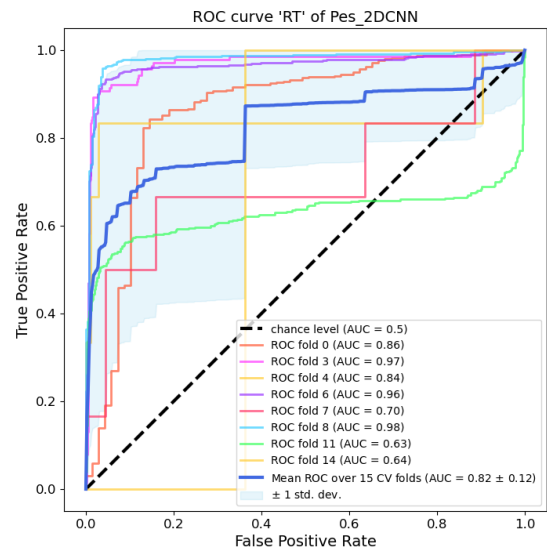


Figure 24. Performance RT, model = P-CMV

Table 4. Performance of the model for Reverse Triggering

Performance Reverse Triggering	ALL	P-CMV
AUC	0.73 (0.13)	0.82 (0.12)

The results indicate that reverse triggering of the P-CMV model performs better (AUROC = $0.82(0.12)$) compared to its performance across all ventilation modes (AUROC = $0.73(0.13)$). Reverse triggering, is known to be an asynchrony found mostly in controlled breaths. The AUROC curves of the other PVAS and for the artefacts can be seen in appendix C and D respectively.

Adjusting Scaling Factors for Improved Prediction

Based on current data, it seems that the algorithm struggles to accurately predict certain PVAs, due to limited data. Specifically, occurrences of IEE and delayed cycling are infrequent and often involve only a few breaths out of thousands in a single patient. This results in the training set for the model existing solely on data from these PVAs, collected from a single patient that provided an adequate number of examples for training. Scaling factors are introduced to address the limitation of the models' lack of exposure to data from multiple patients. The scaling factors also known as weight classes are used to improve the accuracy of recognising small classes. When using class weights, misclassifying smaller classes is penalised more heavily than larger classes. This emphasises the importance of correctly classifying the smaller classes, improving their recognition accuracy. Assigning higher weights to minority classes encourages the model to learn better representations for these classes, resulting in more balanced predictions overall.

The scaling factors modify the likelihood of a breath belonging to a specific class, originally set a threshold of 0.5. For classes with a smaller dataset, the probability is increased, favouring classification into those categories. The probability for the normal category stayed the same at 0.5.

Table 5. AUC values before and after adjusting the scaling factors

Ventilation modes: ALL	Scaling Factor	AUC	Scaling Factor	AUC
Auto triggering	0.5	0.69 (0.09)	0.7	0.89 (0.06)
Cough	0.5	0.84 (0.08)	0.5	0.84 (0.09)
DT	0.5	0.91 (0.19)	0.7	0.93 (0.18)
Delayed cycling	0.5	0.61 (0.04)	0.75	0.48 (0.02)
IEE	0.5	0.41 *(0.11)	0.7	0.76 (0.04)
Other artefacts	0.5	0.81 (0.35)	0.5	0.87(0.07)
Peristalsis	0.5	0.83 (0.23)	0.5	0.76 (0.24)
Premature cycling	0.5	0.67 (0.11)	0.7	0.89 (0.02)
RT	0.5	0.73 (0.13)	0.6	0.67 (0.01)
normal	0.5	0.84 (0.13)	0.5	0.90 (0.15)

AUC = Area Under the Curve

The selected scaling values are determined through trial and error. It is important to note that adjusting scaling factors can be beneficial for categories with limited data. This adjustment is particularly effective for premature cycling and IEE, but less so for delayed cycling. This suggest that the challenge with delayed cycling may not be solely attributed to the lack of data but could have other explanations.

Validation by Clinicians

As an additional evaluation, a clinician assessed the predictions of the model on new unseen data consisting of an hour of data from three different patients. The clinical validation showed that the ALL, PSV and PCV models display varied capabilities in identifying various PVAs. The PCV model's predictions revealed certain types of PVAs that were unique to PSV breathing and could not occur during the PCV modality, as illustrated in Figure 25. Upon close examination, it became apparent that while the breaths were correctly labelled, there were instances of PSV breathing in the PCV dataset. This challenge in distinguishing between PCV and PSV breaths is likely due to the ASV mode, which combines two distinct types of ventilation. To ensure proper training of the PCV model, breaths with labels of PVAs that matched PSV breathing were excluded from the dataset. The model was subsequently trained on five categories: normal, reverse triggering, cough, peristalsis, and other artifacts. The results showed a significant improvement in the predictions of the PCV model.

Figure 26 demonstrates that the model's predictions improved after eliminating breaths with PVA labels that only occur with PSV ventilation. These findings demonstrate the significance of choosing a distinct model for each ventilation mode, as certain PVAs are associated with particular modes of ventilation.

No statistics were performed on the clinical validation as it was only reviewed by one clinician and it was not accurately estimated per breath whether the predictions were correct or not. However, the number of correctly predicted breaths can be found in Appendix F.

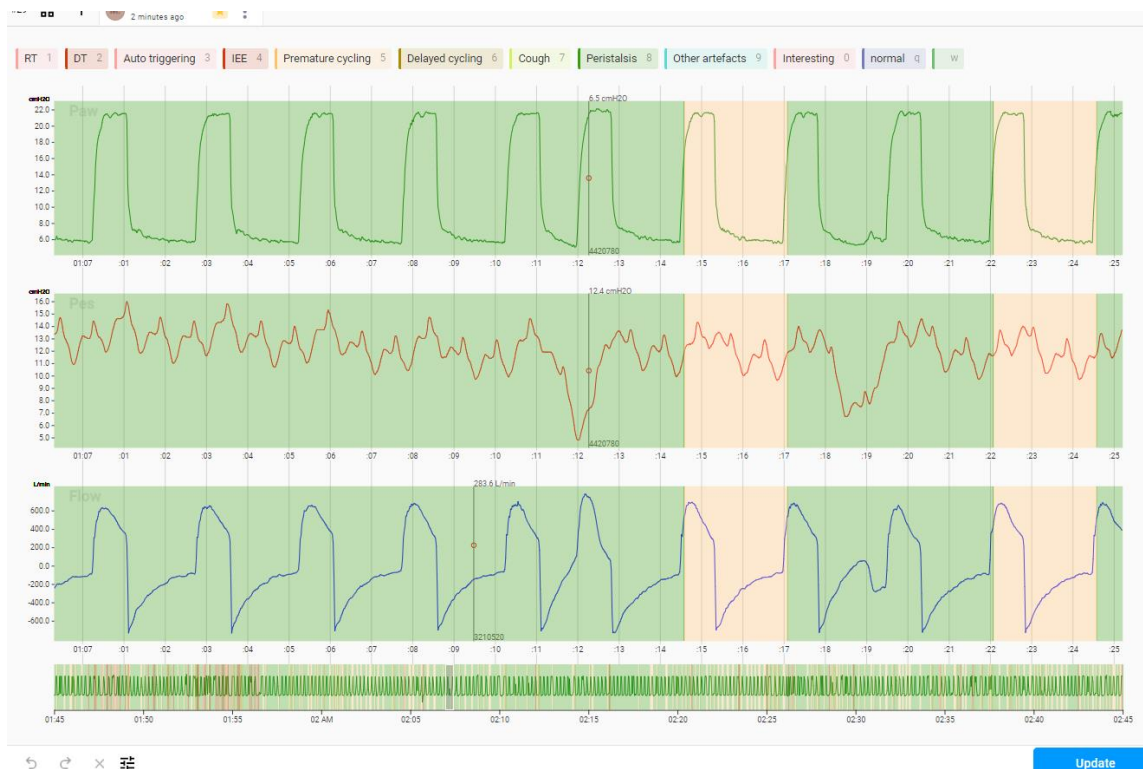


Figure 25. Predictions from the P-CMV model before removing the PVAs that occur only in PSV ventilation. Green = Peristalsis, light orange = premature cycling

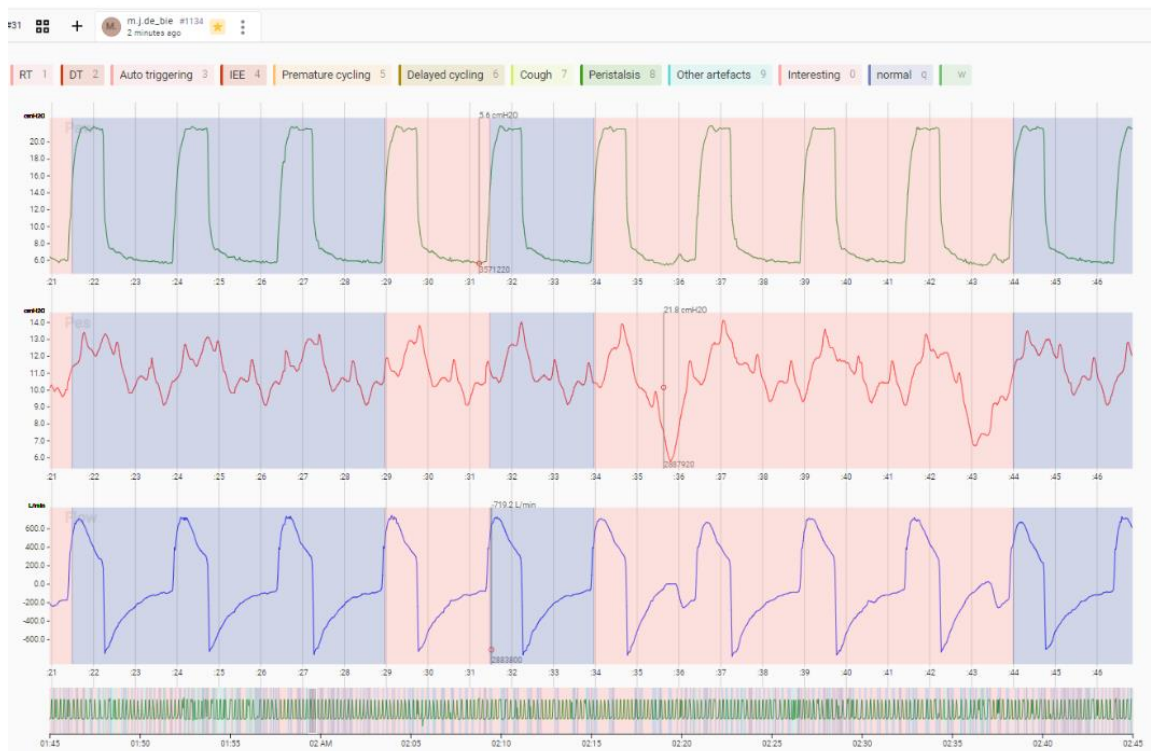


Figure 26. Predictions from the P-CMV model after removing the PVA types of the PSV ventilation. Purple = normal, pink = reverse triggering

Discussion

The findings of this study show that an CNN machine learning model is capable of differentiating between normal and abnormal (Patient-Ventilator asynchrony and artefacts) breaths. In addition, it is possible to create a different CNN model for each specific ventilation modality. The current model can distinguish between normal and abnormal respirations. However, it is difficult to predict specific patient-ventilator asynchronies and artefacts within the abnormalities.

This study examined the efficacy of an unsupervised machine learning algorithm to automatically pre-label regular breaths using the Mahalanobis distance. The threshold of 3.5 provides a 95% accuracy for detecting normal breaths compared to abnormal breaths.

Limitations

This study has multiple limitations which are discussed below.

Quality of the labelled data.

It is very important to feed a CNN with high quality data. The detection algorithm can only be as good as the data it is trained with. But there are some concerns about the quality.

First, the ventilation data was manually labelled by one of four clinicians with expertise in mechanical breathing. Preferably, every patient should be labelled by more than one clinician and discrepancies discussed to enhance the quality of labelling.

Second, there is insufficient data available about the inter- and intra-observer variability of labelling asynchrony. Several studies have looked at the sensitivity of clinicians in detecting PVA, showing that the sensitivity increases when clinicians have access to additional patient information such as the Pes. With an improvement of sensitivity from 53% to 66% (57). Conversely another study reported a sensitivity of 55% (9). These results show that not all present PVAs in the data are detected by clinicians and suggest that there may therefore be intra- and inter-observer variability in the detection of PVAs. Further research is needed to validate this intra- and interobserver variability.

Third, the labelling process has now been set up in such a way that one label can be placed in a single breath. However, in practice we find that sometimes there are several labels in 1 breath. The decision has now been made to use the most prominent PVA. An example of an breath with multiple labels can be seen in figure 27, where both delayed cycling and IEE are present in the breath, where IEE was chosen as the most prominent label. This manner of labelling leads to less uniform labelling and a loss of quality of the labelled data.

This machine learning model is based on ventilation data from 25 different patients. After visualising the data, it became clear that there is considerable variability in the breathing patterns among individuals. This makes training the CNN model difficult because there is a wide range of normal breathing, especially considering that future patients' breathing patterns will also vary. In addition, certain types of PVA are found to be particularly common in a few patients, namely delayed cycling, early cycling and ineffective triggering, so the model learns these PVAs specifically in the breathing patterns of these patients. It can be seen that when the model is validated, it has difficulty recognising these PVAs in patients other than those where they are common. This causes the model to overfit on certain PVAs.

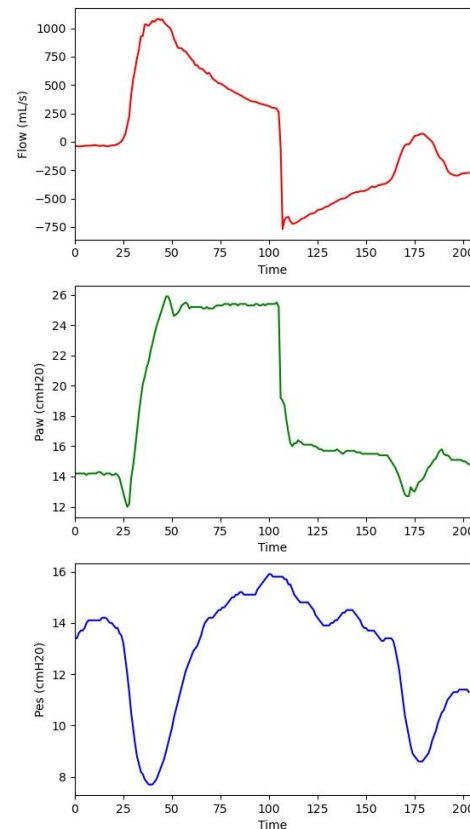


Figure 27. Example of an breath with two different PVAs.

Currently, inconsistencies in the accurate segmentation of breaths affect the effectiveness of the algorithm. This segmentation currently depends on data provided by the ventilator, which is a limitation as this dependence contributes to the occurrence of PVA. During this segmentation of data into individual breaths, instances of reverse trigger asynchronies are sometimes divided into two breaths (figure 28 A & B), hindering the model's learning process, while sometimes they are categorised as a single breath (figure 29). This discrepancy depends on whether the ventilator identifies it as a single breath. Therefore, the current method of breath segmentation is insufficient as it solely relies on machine properties.

Therefore, it is recommended that subsequent studies use the patient signal, specifically the esophageal pressure (Pes), as the basis for labelling and preprocessing data for input to the model. The machine signal can then be used to identify patient-ventilator asynchronies and weaknesses in the ventilation machine.

Furthermore, it has been observed that the data provided by the ventilator regarding the breath is not always accurate. For example, in Figure 30, a patient-triggered breath is classified by the ventilator as machine-triggered, with the result that it is included in the PCV group. This inconsistency in the input data is a challenge for the training of the model.

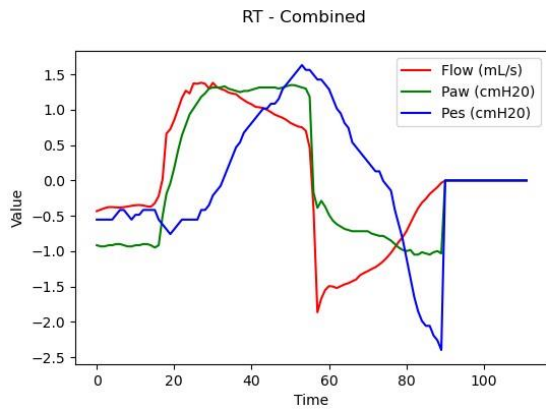


Figure 28 a. Example of reversed triggering splitted in two images (a and b) for one asynchrony

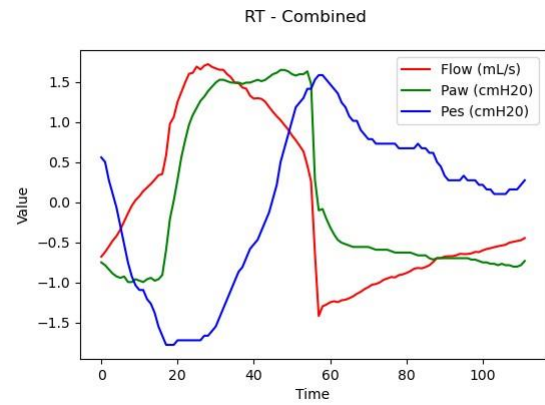


Figure 28 b. Example of reversed triggering splitted in two images (a and b) for one asynchrony

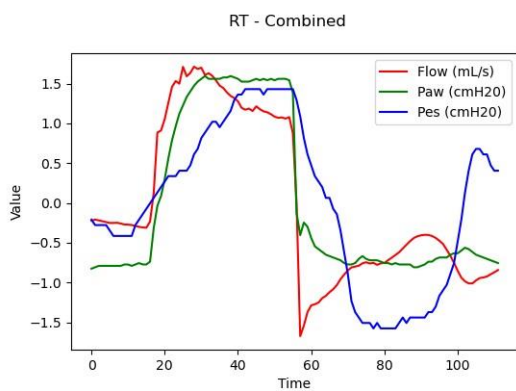


Figure 29. Example of an reverse triggering breath in one image captured

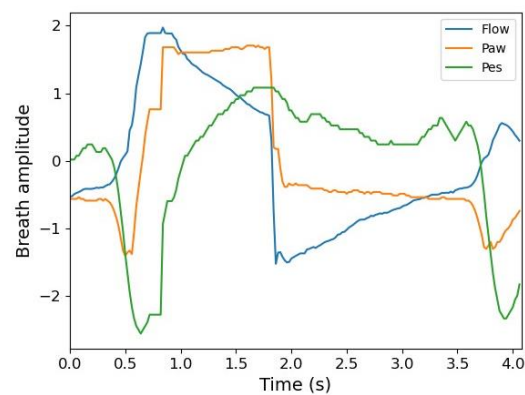


Figure 30. Example of an PSV breath registered by the machine as an PCV breath

Recommendations

A key recommendation for future research is to use a method that utilises the entire multivariate timeseries data, including the flow, Paw and Pes curves, instead of segmenting the breathing data into single breaths. This approach eliminates the dependence on individual breath segmentation, which has been identified as a problem in this study, since the PVAs largely consist of problems in triggering and start inspiration. A recent other study by Bakkes et al. (2023) which also developed an algorithm to automatically detect PVA used a U-net algorithm which is a subtype of a CNN algorithm developed just like the one in this study. The inputs to this algorithm are also segmented breaths (58). However there are possible alternatives these could include a Recurrent Neural Network (RNN), which is a deep-learning technique that can use timeseries data as input and also of variable length which is a major advantage, it is also known to be useful in many medical time-series classification problems (59). Two studies (Zhang et al. (2020) and van de Kamp et al. (2024)) have been conducted to automatically detect PVA using an RNN algorithm. The results of these studies demonstrate the potential of using an RNN network to automatically detect PVA (60, 61).

Another alternative is by using a rule-based algorithm. In particular, the combination of machine learning and a rule-based algorithm could offer significant advantages. This study demonstrates that a rule-based algorithm can distinguish between normal and abnormal respirations. This can be extended to other

asynchronies. For example, a distinction between premature cycling and delayed cycling and normal breaths can be made by examining the onset and termination of breaths and their alignment with the PES curve. If there is a difference of more than 100 milliseconds between these timing points, it can be identified as a PVA.

Currently there are no clear definitions of different PVAs. At this moment there are ambiguous definitions found in the existing literature, which prevent the definition of precise criteria and lead to confusion in classification (2, 6, 7). This lack of clarity makes it difficult to define and categorise different types of PVA, and complicates data annotation as there are no definitive guidelines. As a result, achieving model consistency can be difficult due to the variation in how different individuals interpret these definitions. To ensure agreement on PVA, it is important to review the ventilator graphs with a group of at least three people. If at least two people identify a particular asynchrony, it can be labelled as such. However, this process can be time-consuming.

Further perspectives

Anomaly detection in multivariate time-series data remains a challenging field that is currently under investigation. However, there are upcoming algorithms that now use time-series as input. The algorithm for automatic detection of PVA has promising future prospects, with plans to eventually implement it in a ventilator. To detect PVA at bedside, it is necessary to have the ability to detect it in real-time. Future research should focus on ensuring correct data input for the model, including proper labelling and segmentation. Further research is required to develop a reliable algorithm for real-time PVA detection. With accurately labelled and segmented data, it is possible to develop an effective algorithm for automatic PVA detection.

Conclusion

This study investigated the application of machine learning techniques to analyse ventilation data in critical care settings. Through a combination of unsupervised and supervised learning methods, we have explored the automation of the labelling process and the development of predictive models for identifying patient-ventilator asynchronies (PVAs). Our findings demonstrate the effectiveness of utilizing the Mahalanobis distance as an unsupervised algorithm to pre-annotate normal breaths, thereby streamlining the data labelling process. Furthermore, our supervised learning models, tailored to different ventilation modalities, exhibited promising performance in distinguishing between normal and abnormal breaths

References

1. Fialkow L, Farenzena M, Wawrzeniak IC, Brauner JS, Vieira SR, Vigo A, et al. Mechanical ventilation in patients in the intensive care unit of a general university hospital in southern Brazil: an epidemiological study. *Clinics (Sao Paulo)*. 2016;71(3):144-51.
2. Holanda MA, Vasconcelos RDS, Ferreira JC, Pinheiro BV. Patient-ventilator asynchrony. *J Bras Pneumol*. 2018;44(4):321-33.
3. Zhou Y, Holets SR, Li M, Cortes-Puentes GA, Meyer TJ, Hanson AC, et al. Etiology, incidence, and outcomes of patient-ventilator asynchrony in critically-ill patients undergoing invasive mechanical ventilation. *Sci Rep*. 2021;11(1):12390.
4. Epstein SK. How often does patient-ventilator asynchrony occur and what are the consequences? *Respir Care*. 2011;56(1):25-38.
5. Garofalo E, Bruni A, Pelaia C, Liparota L, Lombardo N, Longhini F, et al. Recognizing, quantifying and managing patient-ventilator asynchrony in invasive and noninvasive ventilation. *Expert Rev Respir Med*. 2018;12(7):557-67.
6. Bulleri E, Fusi C, Bambi S, Pisani L. Patient-ventilator asynchronies: types, outcomes and nursing detection skills. *Acta Biomed*. 2018;89(7-S):6-18.
7. Mirabella L, Cinnella G, Costa R, Cortegiani A, Tullo L, Rauseo M, et al. Patient-Ventilator Asynchronies: Clinical Implications and Practical Solutions. *Respir Care*. 2020;65(11):1751-66.
8. de Haro C, Ochagavia A, López-Aguilar J, Fernandez-Gonzalo S, Navarra-Ventura G, Magrans R, et al. Patient-ventilator asynchronies during mechanical ventilation: current knowledge and research priorities. *Intensive Care Med Exp*. 2019;7(Suppl 1):43.
9. Colombo D, Cammarota G, Alemani M, Carenzo L, Barra FL, Vaschetto R, et al. Efficacy of ventilator waveforms observation in detecting patient-ventilator asynchrony. *Crit Care Med*. 2011;39(11):2452-7.
10. Gilstrap D, MacIntyre N. Patient-ventilator interactions. Implications for clinical management. *Am J Respir Crit Care Med*. 2013;188(9):1058-68.
11. M.J. de Bie PJR, F. van der Velde, J.W. Schoones, J.W.M. Snoep, A.Schoe. The influence of patient-ventilator asynchrony on clinical outcomes in mechanically ventilated patients: A Systematic Review. 2023.
12. Sousa M, Magrans R, Hayashi FK, Blanch L, Kacmarek RM, Ferreira JC. Clusters of Double Triggering Impact Clinical Outcomes: Insights From the EPIdemiology of Patient-Ventilator aSYNChrony (EPISYNC) Cohort Study. *Crit Care Med*. 2021;49(9):1460-9.
13. Magrans R, Ferreira F, Sarlabous L, López-Aguilar J, Gomà G, Fernandez-Gonzalo S, et al. The Effect of Clusters of Double Triggering and Ineffective Efforts in Critically Ill Patients. *Crit Care Med*. 2022;50(7):e619-e29.
14. I. Ihaddouchen FvdV, D.M.J. Tax, A. Schoe. Automated detection of patient-ventilator asynchrony: A systematic review. 2022.
15. Dres M, Rittayamai N, Brochard L. Monitoring patient-ventilator asynchrony. *Curr Opin Crit Care*. 2016;22(3):246-53.
16. Mauri T, Yoshida T, Bellani G, Goligher EC, Carteaux G, Rittayamai N, et al. Esophageal and transpulmonary pressure in the clinical setting: meaning, usefulness and perspectives. *Intensive Care Med*. 2016;42(9):1360-73.
17. Akoumianaki E, Maggiore SM, Valenza F, Bellani G, Jubran A, Loring SH, et al. The application of esophageal pressure measurement in patients with respiratory failure. *Am J Respir Crit Care Med*. 2014;189(5):520-31.
18. Sidey-Gibbons JAM, Sidey-Gibbons CJ. Machine learning in medicine: a practical introduction. *BMC Med Res Methodol*. 2019;19(1):64.
19. Anwar SM, Majid M, Qayyum A, Awais M, Alnowami M, Khan MK. Medical Image Analysis using Convolutional Neural Networks: A Review. *J Med Syst*. 2018;42(11):226.
20. Medical H. Operator's manual. 2017.

21. Botta M, Tsonas AM, Sinnige JS, De Bie AJR, Bindels A, Ball L, et al. Effect of Automated Closed-loop ventilation versus conventional Ventilation on duration and quality of ventilation in critically ill patients (ACTIVE) - study protocol of a randomized clinical trial. *Trials*. 2022;23(1):348.
22. Bandeira M, Almeida A, Melo L, de Moura PH, Ribeiro Silva EO, Silva J, et al. Accuracy of Algorithms and Visual Inspection for Detection of Trigger Asynchrony in Critical Patients : A Systematic Review. *Crit Care Res Pract*. 2021;2021:6942497.
23. Shimatani T, Kyogoku M, Ito Y, Takeuchi M, Khemani RG. Fundamental concepts and the latest evidence for esophageal pressure monitoring. *J Intensive Care*. 2023;11(1):22.
24. Clinical Experts Group HM. How to set expiratory trigger sensitivity (ETS) 2018 [Available from: https://www.hamilton-medical.com/en_US/Resource-center/Article-page~knowledge-base~fd2e3a48-1f6a-4a90-bf3e-befbe7fad210~.html]
25. Sydow M, Thies K, Engel J, Golisch W, Buscher H, Zinserling J, et al. [Variation in inspiratory gas flow in pressure support ventilation. The effect on respiratory mechanics and respiratory work]. *Anaesthesist*. 1996;45(11):1051-8.
26. Gentile MA. Cycling of the mechanical ventilator breath. *Respir Care*. 2011;56(1):52-60.
27. Saavedra SN, Barisich PVS, Maldonado JBP, Lumini RB, Gómez-González A, Gallardo A. Asynchronies during invasive mechanical ventilation: narrative review and update. *Acute Crit Care*. 2022;37(4):491-501.
28. Haudebourg AF, Maraffi T, Tuffet S, Perier F, de Prost N, Razazi K, et al. Refractory ineffective triggering during pressure support ventilation: effect of proportional assist ventilation with load-adjustable gain factors. *Ann Intensive Care*. 2021;11(1):147.
29. Tecuci G. Artificial intelligence. *Wires Comput Stat*. 2012;4(2):168-80.
30. Janiesch C, Zschech P, Heinrich K. Machine learning and deep learning. *Electron Mark*. 2021;31(3):685-95.
31. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436-44.
32. Choudhary K, DeCost B, Chen C, Jain A, Tavazza F, Cohn R, et al. Recent advances and applications of deep learning methods in materials science. *Npj Comput Mater*. 2022;8(1).
33. Sakshi Indolia AKG, S.P. Mishra, Pooja Asopa Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach. *Procedia Computer Science*. 2018;132:679-88.
34. Sharma A, Lysenko A, Boroevich KA, Vans E, Tsunoda T. DeepFeature: feature selection in nonimage data using convolutional neural network. *Brief Bioinform*. 2021;22(6).
35. Yin L. A Summary of Neural Network Layers 2018 [Available from: <https://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e>].
36. Jeong J. The Most Intuitive and Easiest Guide for Convolutional Neural Network. 2019.
37. Weisstein EW. Convolution: MathWorld--A Wolfram Web Resource; [Available from: <https://mathworld.wolfram.com/Convolution.html>].
38. Mazet V. Basics of Image Processing - Convolution: Université de Strasbourg; 2020 [Available from: <https://vincmazet.github.io/bip/filtering/convolution.html>].
39. Prathap P. The Secret to Understanding CNNs: Convolution, Feature Maps, Pooling and Fully Connected Layers! : Medium; 2023 [Available from: <https://medium.com/@prajeeshprathap/the-secret-to-understanding-cnns-convolution-feature-maps-pooling-and-fully-connected-layers-97055431a847>].
40. Guifang Lin WS. Research on convolutional neural network based on improved Relu piecewise activation function. *Procedia Computer Science*. 2018;131:977-84.
41. Lichao Wu GP. On the Importance of Pooling Layer Tuning for Profiling Side-Channel Analysis. *Applied Cryptography and Network Security Workshops 2021*;12809.
42. Max Pooling [Available from: <https://paperswithcode.com/method/max-pooling>].
43. Al-Masri A. How Does Backpropagation in a Neural Network Work? [Available from: <https://builtin.com/machine-learning/backpropagation-neural-network>].
44. Dridi S. Unsupervised Learning - A Systematic Literature Review. 2021.

45. Eckhardt CM, Madjarova SJ, Williams RJ, Ollivier M, Karlsson J, Pareek A, et al. Unsupervised machine learning methods and emerging applications in healthcare. *Knee Surg Sport Tr A.* 2023;31(2):376-81.
46. Harshvardhan GM, Gourisaria MK, Pandey M, Rautaray SS. A comprehensive survey and analysis of generative models in machine learning. *Comput Sci Rev.* 2020;38.
47. Thudumu S, Branch P, Jin J, Singh J. A comprehensive survey of anomaly detection techniques for high dimensional big data. *J Big Data-Ger.* 2020;7(1).
48. Snoep JWMV-Q, F. van der, Schoe, A. Alaerts, T.M. Rietveld, P.J. . Beademing, oesophagusdrukmeting bij volwassene (Intensive Care) 2023.
49. Label Studio. In: HumanSignal, editor.: Open Source Data Labeling Platform.
50. Kenneth R. Castleman QW. Chapter Nine Object Classification. In: Press A, editor. *Microscope Image Processing (Second Edition).* 2 ed: Elsevier; 2022.
51. Chlis NK. Comparison of Statistical Methods for Genomic Signature Extraction: TECHNICAL UNIVERSITY OF CRETE; 2013.
52. Learn S. 3.1. Cross-validation: evaluating estimator performance [Available from: https://scikit-learn.org/stable/modules/cross_validation.html].
53. Parikh R, Mathai A, Parikh S, Chandra Sekhar G, Thomas R. Understanding and using sensitivity, specificity and predictive values. *Indian J Ophthalmol.* 2008;56(1):45-50.
54. Alba AC, Agoritsas T, Walsh M, Hanna S, Iorio A, Devereaux PJ, et al. Discrimination and Calibration of Clinical Prediction Models: Users' Guides to the Medical Literature. *JAMA.* 2017;318(14):1377-84.
55. Draelos R. MACHINE LEARNING

Measuring Performance: AUC (AUROC): Glassbox medicine; [Available from:

<https://glassboxmedicine.com/2019/02/23/measuring-performance-auc-auroc/>.

56. Mandrekar JN. Receiver operating characteristic curve in diagnostic test assessment. *J Thorac Oncol.* 2010;5(9):1315-6.
57. Silva DO, de Souza PN, de Araujo Sousa ML, Morais CCA, Ferreira JC, Holanda MA, et al. Impact on the ability of healthcare professionals to correctly identify patient-ventilator asynchronies of the simultaneous visualization of estimated muscle pressure curves on the ventilator display: a randomized study (P(mus) study). *Crit Care.* 2023;27(1):128.
58. Bakkes T, van Diepen A, De Bie A, Montenij L, Mojoli F, Bouwman A, et al. Automated detection and classification of patient-ventilator asynchrony by means of machine learning and simulated data. *Comput Methods Programs Biomed.* 2023;230:107333.
59. Li G, Jung JJ. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Inform Fusion.* 2023;91:93-102.
60. Kamp LV, Reinders J, Hunnekens B, Oomen T, Wouw NV. Automatic patient-ventilator asynchrony detection framework using objective asynchrony definitions. *Ifac J Syst Control.* 2024;27.
61. Zhang L, Mao K, Duan K, Fang S, Lu Y, Gong Q, et al. Detection of patient-ventilator asynchrony from mechanical ventilation waveforms using a two-layer long short-term memory neural network. *Comput Biol Med.* 2020;120:103721.

Appendix

A. Data Annotation Protocol

Data annotation protocol for patient-ventilator asynchrony

The following labels can be assigned to the data:

- Reverse triggering
- Auto triggering
- Ineffective effort during expiration (IEE)
- Premature cycling
- Delayed cycling
- Double triggering

- Cough
- Peristalsis
- Other artefacts

PAW_{ON} = the onset of airway pressure (beginning of ventilator pressurization)

PAW_{OFF} = the termination of airway pressure (end of insufflation)

PES_{ON} = the onset of esophageal pressure (beginning of inspiratory effort)

PES_{OFF} = the termination of esophageal pressure (end of inspiratory effort)

Criteria for annotation:

1. Normal breath (**no need to annotate these cycles**)
 - A. Mandatory or assisted breath
 - B. PES_{ON} and PAW_{ON} occur simultaneously, with a ± 100 ms error margin
2. Reverse trigger – three criteria:
 - A. Mandatory breath: PAW_{ON} does not start with a negative deflection
 - B. Presence of negative PES signal
 - C. PES_{ON} occurs >100 ms after PAW_{ON} but before PAW_{OFF}
3. Auto triggering:
 - A. Mandatory breath: PAW_{ON} does not start with a negative deflection
 - B. Absence of negative PES signal
4. Ineffective effort during expiration:
 - A. Mandatory or assisted breath
 - B. PES_{ON} occurs after PAW_{OFF} (i.e., during ventilator expiration)
5. Premature cycling:
 - A. Assisted breath: PAW_{ON} starts with a negative deflection
 - B. PES_{ON} and PAW_{ON} occur simultaneously, with a ± 100 ms error margin
 - C. PAW_{OFF} occurs >100 ms before PES_{OFF}
6. Delayed cycling:
 - A. Assisted breath: PAW_{ON} starts with a negative deflection

- B. PES_{ON} and PAW_{ON} occur simultaneously, with a ± 100 ms error margin
 C. PAW_{OFF} occurs >100 ms after PES_{OFF}
7. Double triggering:
 - A. First breath is an assisted breath that starts with a negative deflection
 - B. PES_{ON} of first effort and PAW_{ON} of first breath occur simultaneously, with a ± 100 ms error margin
 - C. PAW_{ON} of second breath occurs before PES_{OFF} of first effort
 8. Cough:
 - A. Sharp inhalation and exhalation spikes in the flow waveform
 - B. Presence of simultaneous disturbances in the PAW and PES signal
 9. Peristalsis
 - A. (Multiple) positive deflection(s) in the PES signal with a higher amplitude than average patient efforts
 - B. Absence (or minimal presence) of simultaneous disturbances in the PAW and flow signal
 10. Other artefacts: anything that is not a normal breath and does not meet the criteria outlined above

It is possible to assign more than one label to one breath, for example:

- Delayed cycling + IEE
- Double triggering + IEE

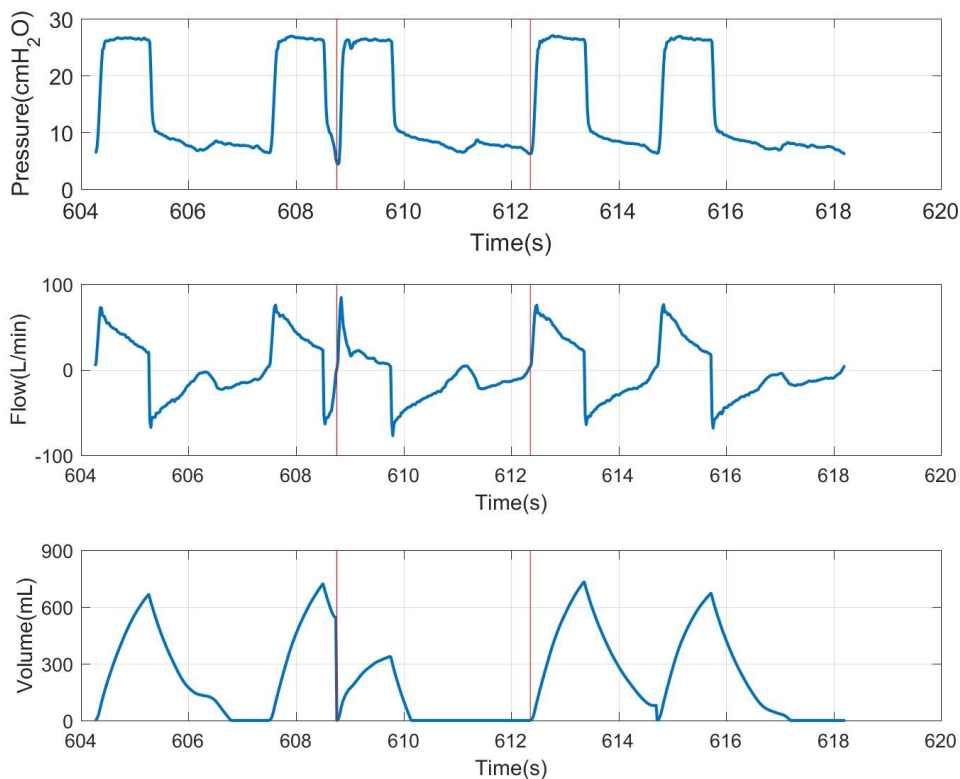


Figure 1. An example showing that a cycle could have two labels. Both cycles that form the double triggering event should be annotated as DT. Meanwhile, the second breath can also be annotated as IEE. DT: double triggering; IEE: ineffective effort during expiration.

B . PVA Types per Patient

Table 6. *Different types of patient-ventilator asynchrony per patient and their amounts*

Patient	Auto triggering	Cough	DT	Delayed Cycling	IEE	Other artefacts	Peristalsis	Premature cycling	RT	Normal
1		113	5	0	0	17	197	102	0	933
2	0	58	12	0	0	306	24	0	0	1220
3	0	7	6	10	0	32	83	8	503	583
4	0	57	0	0	0	24	12	0	0	924
5	0	16	0	0	0	131	21	0	0	7162
6	0	58	0	0	0	93	47	0	139	1002
7	0	0	0	0	1	31	12	2	6	1501
8	0	3	0	4	0	58	37	0	4	851
9	0	52	8	0	0	21	29	1	0	920
10	5	5	0	0	0	18	39	0	0	1604
11	0	0	0	0	0	5	9	0	541	645
12	0	3	0	581	205	8	6	0	6	43
13	0	1	0	28	4	77	0	1	0	797
14	121	0	0	0	0	29	51	125	0	625
15	2	43	15	0	0	20	197	1	0	1821
16	0	122	0	37	63	41	8	2	6	1361
17	0	0	0	0	0	2	13	0	912	272
18	0	0	0	0	0	7	2	0	0	1191
19	0	0	0	0	0	149	0	0	0	595
20	0	0	0	0	0	29	19	0	674	635
21	0	0	0	0	0	5	3	0	0	974
22	0	0	2	0	0	21	15	2	0	2145
23	0	2	0	0	0	10	161	0	0	1879
24	0	0	0	0	0	27	0	0	0	1671
25	0	0	0	0	0	26	0	0	1	1645
Total	128	540	48	660	273	1187	985	244	2792	32999

C. AUC performances from the ALL model

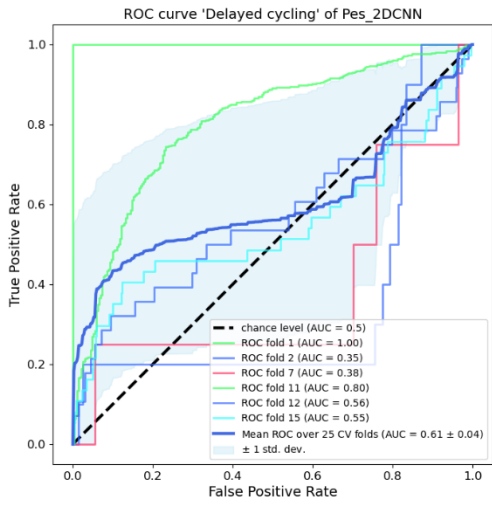


Figure 31. AUROC curve of delayed cycling

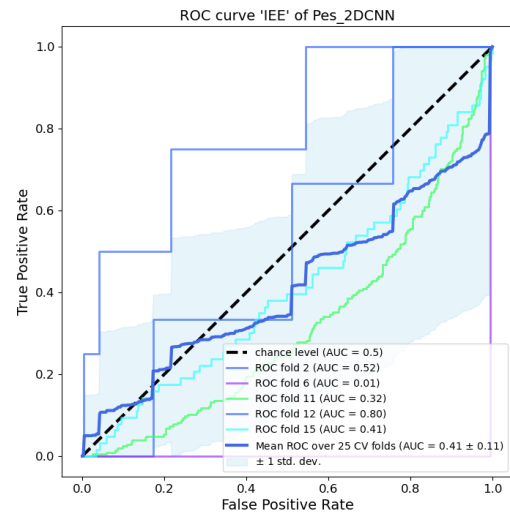


Figure 32. AUROC curve of Ineffective triggering

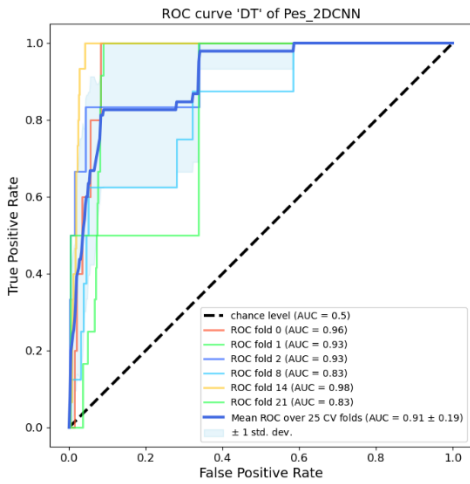


Figure 33. AUROC curve of double triggering

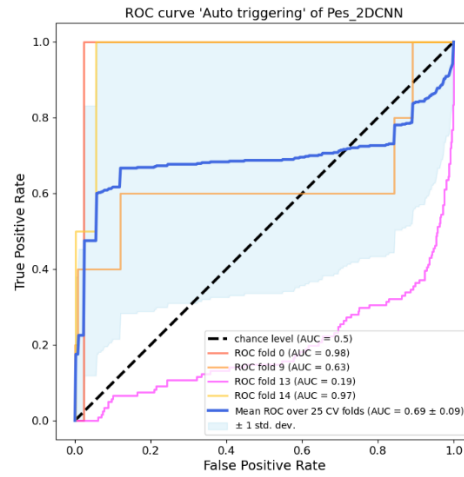


Figure 34. AUROC curve for auto triggering

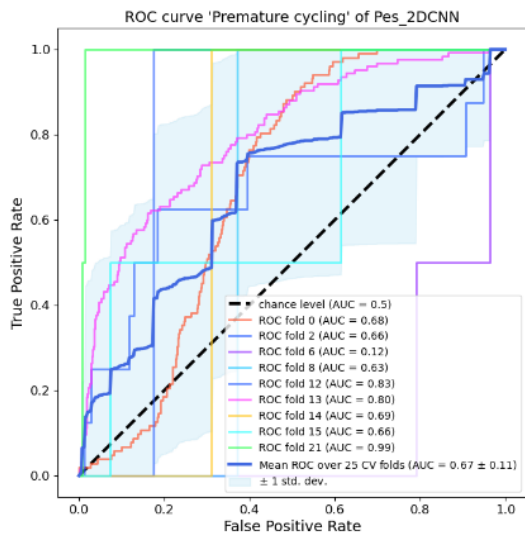


Figure 35. AUROC curve of premature cycling

D. Other Artefacts

Below are the AUROC curves for the artefacts: peristalsis, cough and other artefacts. It can be seen that the model can distinguish these from other breaths. An advantage of these artefacts is that they occur in every patient, giving the model a wide range of training data.

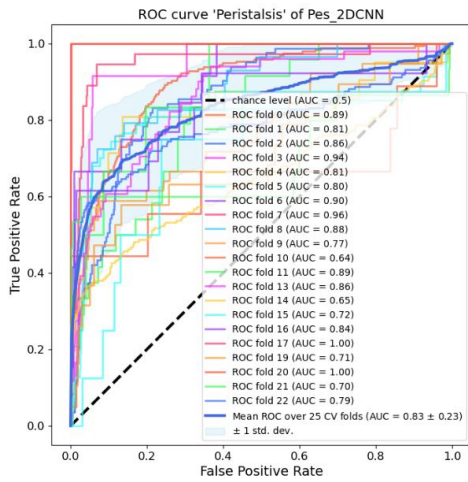


Figure 36. AUROC curve for peristalsis on the ALL model based on all patients

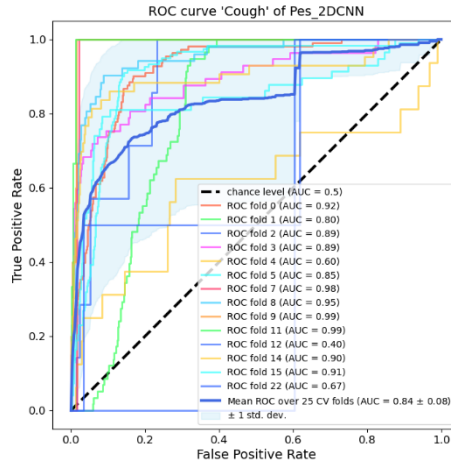


Figure 37. AUROC curve for cough based on the ALL model on all patients

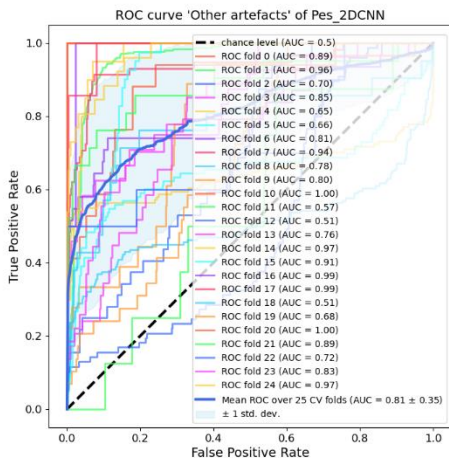


Figure 38. AUROC curve for other artefacts based on the ALL model on all patients

E. Performances of clinical validation by experts

Table 11. Clinical validation of the PSV model - This table presents the count of correct and incorrect detections for each ventilation mode, along with the total count for each category.

Patient-ventilator asynchrony	Correct	Incorrect – (correct: normal)	Total
autotriggering	0	9	9
Other artefacts	0	2	2
Normal	55	0	55
Peristalsis	0	10	10
Total	55	21	76

Table 12. Clinical validation of the PCV model - This table presents the count of correct and incorrect detections for each ventilation mode, along with the total count for each category.

Patient-ventilator asynchrony	Correct	Incorrect (correct: normal)	Incorrect (correct: peristalsis)	Total
Premature Cycling	0	6	0	6
Peristalsis	5	20	0	25
IEE	0	5	1	6
Reverse trigger	0	0	4	4
Total	5	31	5	41

Table 13. Clinical validation of the ALL model - This table presents the count of correct and incorrect detections for each ventilation mode, along with the total count for each category.

Patient-ventilator asynchrony	Correct	Incorrect – (correct: normal)	Incorrect – (correct: peristalsis)	Total
Normal	38	0	0	38
Reverse triggering	11	26	0	37
Peristalsis	2	3	0	5
Artefacts	2	7	0	9
Cough	2	29	1	32
Total	55	65	1	121