DETECTION OF PATIENT-VENTILATOR ASYNCHRONY BASED ON ESOPHAGEAL PRESSURE USING A CONVOLUTIONAL NEURAL NETWORK

MASTER THESIS
IMANE IHADDOUCHEN

TU Delft

Detection of patient-ventilator asynchrony based on esophageal pressure using a convolutional neural network

Imane Ihaddouchen
Student number: 4467167
March 2023

Thesis is 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 Biomedical Engineering, TU Delft
Intensive Care Unit, LUMC

Supervisor(s):
dr. A. Schoe
dr. D.M.J. Tax

Thesis committee members:
dr. A. Schoe (chair)
prof. dr. E. de Jonge
dr. D.M.J. Tax

Universiteit Leiden

TU Delft Delft University of Technology

Erasmus Universiteit Rotterdam

Preface

With this thesis, my time as a student at the TU Delft comes to an end. After 10 months of work at the Intensive Care Unit (ICU) of the LUMC, I am proud to present to you the results of my graduation project in which I developed a detection algorithm for patient-ventilator asynchrony (PVA).

Seven years ago I embarked on this journey, as part of only the second cohort of the bachelor program Clinical Technology. I started this journey without any idea where it would bring me. After completing my bachelor's degree in 2018, I started with the corresponding three-year Master in Technical Medicine. During my masters, I had the opportunity to intern at the Neonatal Intensive Care Unit and the ICU, where I discovered that my main interest lies in the intensive care environment. Its complex patient population, multidisciplinary team, and cutting-edge technology intrigued me, and it made me realize that there are numerous opportunities for technical physicians in this field.

I conducted my second-year ICU internship at the LUMC. On the department, Technical Medicine was well known, and their doors were always wide open to Technical Medicine students. I performed my first internship under the supervision of Bram Schoe, and his boundless enthusiasm for mechanical ventilation convinced me to return to this ICU and perform a graduation project on this topic. Bram, I would like to thank you for welcoming me to the ICU and trusting me to put the first steps in this project. You were the best mentor, and I really appreciated your positive attitude as well as your passion for technology and research. I am convinced that you and your research group are going to make big steps in the research field of mechanical ventilation.

Even though I am not among the best at programming, I am a woman who enjoys a good challenge. Therefore, I jumped into the deep end and chose to code a machine learning approach for the purpose of this thesis. I did not have to think twice when it came to finding a technical supervisor in the field of machine learning to guide me through this project. I remembered having a course from David Tax during my bachelors and his enthusiasm during the lectures had always stayed with me. David, thank you for your help and advice throughout the project. Our (bi-)weekly meetings in Delft always provided me with new insights in machine learning. As a beginner in the field of machine learning, I frequently asked questions, even about basic concepts. However, you were always patiently willing to create a beautiful painting on your whiteboard to ensure I understood the concepts you were discussing.

I would like to thank all the colleagues in the ICU who made my time there so enjoyable and who were always willing to share their knowledge with me. A special thanks to Evert de Jonge, the head of the department, for accepting this new profession of Technical Medicine in the medical field and for willing to be part of my graduation committee despite your busy schedule. I would also like to thank Petra Rietveld, Willem Snoep and Franciska van der Velde. My research on PVA would not have been possible without these experts in mechanical ventilation. Petra and Willem, thank you for always having your door open for all my questions, for your positivity, for annotating thousands of respiratory cycles, and for our endless discussions about PVA. You guys are a true asset to this department. Franciska, thank you for taking part in my literature review, annotating patient data, and providing insightful feedback.

Finally, I would like to express my appreciation to my family and friends for their unconditional support throughout my studies. Steven (and Tigri, my cat), thank you for always helping me to relieve my stress and for being there for me on my worst and best days. To my dear parents who made many sacrifices for us to have a better future than them, you taught me to reach for the stars, and I will never stop striving to make you proud. You are my biggest cheerleaders and I will be forever grateful for that.

I now bid farewell to this chapter of my life and I am very excited about what the future holds for me as a *Klinisch Technoloog*.

I. Ihaddouchen
The Hague, March 2023

Contents

List of abbreviations

2D	Two-dimensional
2DCNN	Two-dimensional convolutional neural network
ARDS	Acute respiratory distress syndrome
ASV	Adaptive support ventilation
AUROC	Area under the receiver operating characteristic
AutoPEEP	Intrinsic positive end-expiratory pressure
BN	Batch normalization
cmH2O	Centimeter of water
CNN	Convolutional neural network
Conv_2DCNN	Two-dimensional convolutional neural network with kernel size of 50 x 3 in first convolutional layer
Conv2D	Two-dimensional convolutional layer
CV	Cross-validation
DR	Dilation rate
DT	Double triggering
EAdi	Electrical activity of the diaphragm
ECG	Electrocardiogram
FC	Fully connected
FN	False negative
FP	False positive
ICU	Intensive Care Unit
IEE	Ineffective effort during expiration
Imb	Imbalanced dataset
Initial_2DCNN	Initial two-dimensional convolutional neural network prior to model adjustments
kg	kilograms
LUMC	Leiden University Medical Center
Max Pool	Max Pooling layer
ML	Machine learning
ms	millisecond
MV	Mechanical ventilation
NAVA	Neurally adjusted ventilation assist
Paw	Airway pressure
PCV	Pressure control ventilation
PEEP	Positive end expiratory pressure
Pes	Esophageal pressure
Pes_2DCNN	Two-dimensional convolutional neural network based on airway, flow-time and esophageal pressure
PF_2DCNN	Two-dimensional convolutional neural network based on airway and flow-time
Pmus	Inspiratory pressure generated by the respiratory muscles
Pool_2DCNN	Two-dimensional convolutional neural network with pool size of 2 x 1
PSV	Pressure support ventilation
PVA	Patient-ventilator asynchrony
RASS	Richmond Agitation Sedation Scale
ReLU	Rectified linear unit
ROC	Receiver operating characteristic
ROI	Region of interest
RR	Respiratory rate
RT	Reverse triggering
RUS	Random undersampling
SMOTE	Synthetic minority over-sampling technique
Std. dev.	Standard deviation
TN	True negative
VC-CMV	Volume-controlled continuous mandatory ventilation
V_T	Tidal volume

Abstract

Introduction: In intensive care units (ICU), the most significant life support technology for patients with acute respiratory failure is mechanical ventilation. A mismatch between ventilatory support and patient demand is referred to as patient-ventilator asynchrony (PVA), and it is associated with a series of adverse clinical outcomes. Although the use of a reference signal for patient effort is critical in recognition of PVA, existing detection algorithms are frequently solely based on the ventilator's airway pressure (Paw) and flow-time signals. The aim of this study was to develop an automated detection algorithm for PVA using the ventilator's Paw, flow-time and esophageal pressure (Pes) signals.

Methods: We proposed a two-dimensional convolutional neural network (2DCNN) to detect two types of PVA (reverse triggering (RT) and premature cycling) using a dataset of respiratory cycles recorded from 11 patients. Mechanical ventilation experts with access to the Pes signal annotated 12.337 respiratory cycles to create a gold standard dataset. Several techniques for a potential class imbalance problem, as well as several changes to the initial model architecture, were investigated. A leave-one-patient-out cross-validation technique was used to evaluate model performance. The proposed Pes-based 2DCNN (Pes_2DCNN) was compared to a similar model based solely on the ventilator Paw and flow-time signals (PF_2DCNN).

Results: The proposed Pes_2DCNN exhibited superior performance in detecting RT as compared to PF_2DCNN in terms of area under the receiver operating characteristic (AUROC) (0.80 ± 0.07 vs. 0.75 ± 0.13, respectively; $p < 0.01$). Furthermore, the results indicate that the class imbalance solutions did not improve the performance for detection of RT. For detection of premature cycling, Pes_2DCNN also outperformed PF_2DCNN in terms of AUROC (0.88 ± 0.09 vs. 0.71 ± 0.24, respectively; $p < 0.01$).

Conclusion: The findings of this study suggest the added value of the Pes signal in detection of RT and premature cycling. However, because this is a preliminary study, more research is required to further investigate the importance of the Pes signal in PVA detection.

1

Introduction

In intensive care units (ICU), invasive mechanical ventilation (MV) is the most significant life support technology for patients with acute respiratory failure. Mechanical ventilators precisely regulate the delivery of oxygen, pressure, and air flow to support patients' oxygenation, ventilation, and work of beathing. However, while they can be life-saving, they can also cause lung damage and substantial patient distress if patient effort and MV support are not well-matched. To provide comfortable ventilatory support to the patient, the interaction between the patient and the ventilator must be optimized. When either the initiation or termination of MV does not coincide with the neural timing of inspiration and expiration, or when the delivery of ventilatory support is insufficient to meet the patient's demand, it is referred to as *patient-ventilator asynchrony* (PVA). This phenomenon is associated with a series of adverse clinical outcomes, such as prolonged duration of mechanical ventilation (1, 2), extended stays in ICU (3), and even mortality (4).

PVA can be detected at the bedside by observing deviating patterns in the waveforms displayed on the ventilator. However, this is impractical because PVA can occur sporadically and detection is only possible when a physician is present at the bedside. Furthermore, the reported sensitivity of manual asynchrony detection based on visual analysis of ventilator waveforms is extremely low (16-28%) (5). These challenges prompted several attempts by researchers to develop computerized algorithms for automatic PVA detection. In our previous work, we reviewed existing detection algorithms for PVA, including their strengths and weaknesses (6). On the one hand, rule-based algorithms with heuristic rules and thresholds were proposed to distinguish a breath as PVA or non-PVA. On the other hand, machine learning (ML) models have been introduced to address the problem of PVA recognition.

Classification algorithms must be validated by comparing them to a gold standard. Various data annotation procedures for creating a gold standard dataset have been reported in literature. For example, Pan et al. (7) made use of a dataset annotated by five junior respiratory therapists who could make remarks on difficult cycles. Three senior respiratory therapists reviewed the annotations and were in charge of identifying the remarked difficult cycles. Gholami et al. (8) used a panel of five experts who made decisions by voting. Blanch et al. (9) had the data independently annotated by five experts, where disagreements were discussed before reaching a consensus. Surprisingly, most studies performed data annotation with lack of knowledge about the activity of the diaphragm. To our knowledge, only one study used a reference signal for patient effort during annotation (10). The use of a reference signal, such as electrical activity of the diaphragm (EAdi) or esophageal pressure (Pes), is critical to effectively confirm the presence of PVA (9, 11). As a result, existing PVA detection algorithms mimic human expertise instead of recognizing true PVA.

The Pes is a minimally invasive technique for monitoring transpulmonary pressure variations that originate in the diaphragm or any other inspiratory muscle. Because it decreases during inspiratory effort, it can be used as a signal to guide ventilation management. Pes recordings aid in the detection of PVA by directly comparing the onset and offset of patient effort in the esophageal pressure waveform to the onset and offset of inspiration in the airway pressure (Paw) and flow-time waveforms.

Detection algorithms for PVA are frequently solely based on the ventilator's Paw and flow-time signals. Despite the fact that the addition of Pes promises to aid in accurate recognition of PVA, no Pes-based detection algorithms exist for PVA. In order to prove the added value of using Pes in PVA detection, we developed a two-dimensional convolutional neural network (2DCNN) that detects PVA using the ventilator's Paw, flow-time and Pes signals. In addition, we compared its performance to that of a similar model based solely on the ventilator' Paw and flow-time waveforms.

2

Background

2.1 Technical background

Machine learning is a subfield of artificial intelligence in which machines learn or extract knowledge from the available data in order to make predictions or decisions. Machine learning combines statistical analysis techniques with computer science to develop algorithms capable of “statistical learning” (12). Unsupervised learning and supervised learning are the two types of learning techniques used in ML. Unsupervised learning algorithms search for patterns or clusters in the data without any input from the user (13). A supervised learning algorithm, on the other hand, is created by feeding it examples of a specific input and its corresponding output. The resulting algorithm is expected to be capable of predicting a unique output when exposed to new and previously unseen data (12).

Traditional supervised learning methods often rely on hand-crafted features. The selection and calculation of these features is a challenging and time-consuming task. Deep learning is a subset of machine learning in which a neural network can extract useful features from data automatically. CNNs are among the deep learning techniques that are actively used for medical image analysis. This includes application fields such as disease classification, abnormality detection and computer-aided diagnosis (14).

A CNN is composed of different layers, and the input image is passed through each layer in order to extract the features that are relevant for generating a classification output. **Figure S1** shows a visual representation of a typical CNN and its training process (**Supplementary materials 1**). The CNN is made up of three types of layers: convolutional layers, pooling layers, and fully connected (FC) layers. The convolutional layers extract the features by applying different kernels to the input tensor (subset array of values). A kernel is a M x M matrix that is slid across the input tensor to multiply its values with the superimposed input values. The sum of the products of each input tensor pixel and the kernel yields a single numerical value for the newly generated feature map. By applying multiple kernels to the data, multiple feature maps are generated. These outputs serve as the input for the subsequent CNN layer. A convolutional layer is usually followed by a pooling layer. The aim of this layer is to downsample the feature maps in order to reduce the network’s computational costs (15). Multiple convolution and pooling steps can be repeated, resulting in a CNN with many layers of data. These data are eventually transformed into a one-dimensional array, which is fed into the FC layers. As a result, the FC layers determine the relationship between the extracted features and the outcome.

The term ‘model architecture’ is used to refer to CNN building blocks such as the number, type, order, and shape of the layers. The model can start training after the model architecture has been determined and the data has been preprocessed. Training is the process by which the model learns the relationship between the input and the outcome. The input data consist of ground truth labels, which are typically assigned by experts during data annotation. To train the model, a subset of the dataset (the training set) is fed into it, along with the assigned labels. During training, the CNN learns which kernels to use in the convolutional layers and which weights to use in the FC layers in order to provide the best model performance. A loss function compares the predicted and actual outcomes under different kernels and weights. A high loss indicates that the model incorrectly predicts the outcome, whereas a low loss indicates that the model correctly predicts the outcome. Backpropagation is used to adjust the kernels and weights based on the loss value, with the aim to minimize the error between the output predictions and the ground truth labels (16).

The period during which the entire training set has been passed through the CNN once (both forward propagation and backpropagation) is referred to as an epoch. This is equivalent to an iteration in small datasets. In large datasets, the entire dataset may not be able to pass through the CNN in a single batch. In this case, the larger dataset is divided into smaller batches. Whenever a single batch undergoes forward- and backpropagation, this is called an iteration. Once all batches have been forward- and backpropagated, this is called an epoch (15).

During model development, nested cross-validation (CV) is typically used to optimize and train the model, as depicted in **Figure 1**. The inner loop of CV is used to optimize the hyperparameters of the model, such as the batch size and number of epochs. The outer loop of CV is used to evaluate the model's generalizability to new data. The inner loop is nested within the outer loop, hence the name "nested" CV. In the outer CV loop, the entire dataset is divided into a training-, and test set. The training set is used for the previously described training process, and the test set is used for evaluation of the final model. It is important that the test set contains data that the model has not seen during training. In every fold of the outer CV loop, the outer training set is used in the inner CV loop where the data is further split into an inner training set and a validation set. During the inner CV loop, different possible hyperparameter combinations are trained on the inner training set and evaluated on the validation set, resulting in a validation score for every inner CV fold. Validation scores are averaged over the inner CV folds to obtain a validation performance. The combination of hyperparameters with the highest validation score is returned to the outer CV loop. Here, these hyperparameters are used to train the model on the outer training set. When training on the outer training set is completed, the model is tested on the outer test set. This results in a test performance. A comparison of the validation performance and the test performance gives insight into the model's fit to the data. Models that perform well in the inner CV loop (i.e., high validation performance), but have poor accuracy when applied to the test data in the outer CV loop (i.e., low test performance), are overfitted (12). In this case, the model has learned statistical regularities specific to the training set. This means that instead of learning the relevant pattern, it memorizes the irrelevant noise and thus performs poorly on subsequent new data, e.g. the test data. If the model performs poorly on both the training- and test sets, then the model is underfit to the data. The goal is to achieve a high test performance that is comparable to the validation performance (16). The test performance is averaged over the outer CV folds to obtain the final performance of the model on unseen data.

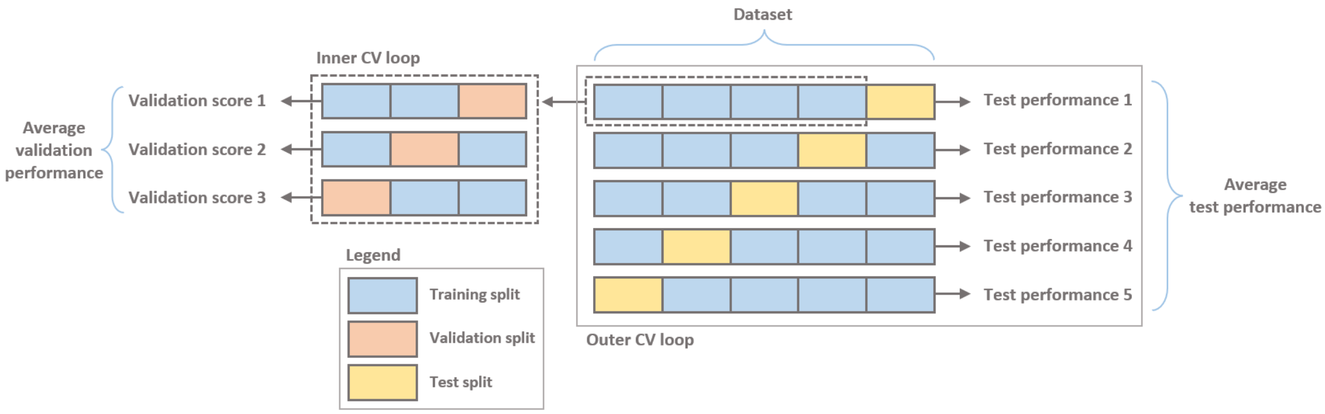


Figure 1 Schematic representation of nested cross-validation. This approach is typically used to optimize and train the model. The final performance of the model on unseen data is obtained by averaging the test performances over the outer cross-validation folds. CV: cross-validation.

2.2 Patient-ventilator asynchrony nomenclature and taxonomy

Research on patient-ventilator asynchrony lacks a standardized vocabulary and associated taxonomy. This complicates the communication among students and researchers, as well as comparison of study results. **Table 1** summarizes PVA taxonomy and the various names used in the literature, as well as providing graphic representations of the different types of PVA.

Asynchronies can be classified in a variety of ways. Several authors used a systematic approach to assess PVA based on the four stages of respiration: triggering, inspiration, cycling and expiration (2, 17). The trigger event is assessed in terms of synchrony with the start of patient inspiratory effort. It can happen *early* (before the start of the patient's inspiratory effort), on time (synchrony), or *late* (a clinically important delay). In addition, two other conditions are described that are not related to timing but rather to ventilator function: *auto triggering*, which occurs when a signal other than the inspiratory pressure generated by the respiratory muscles (Pmus) triggers an inspiration; and a *failed trigger*, which occurs when the Pmus fails to trigger inspiration.

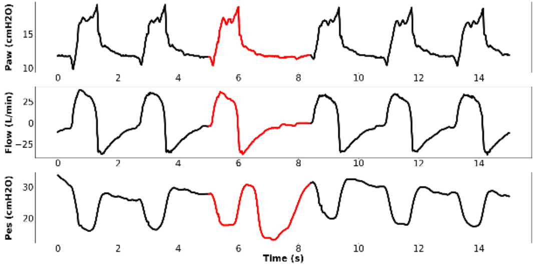
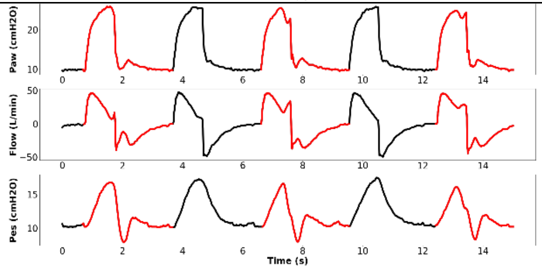
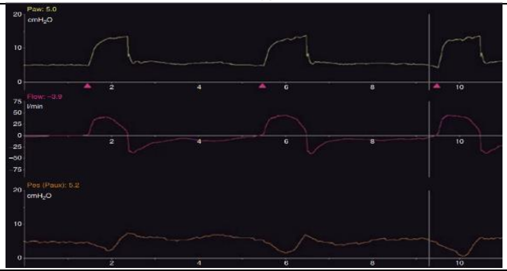
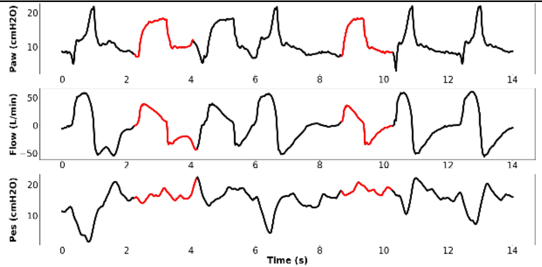
During the inspiratory phase, the patient-ventilation interaction is characterized by the relationship of work performed by the ventilator and the patient. *Flow asynchrony* can occur during inspiration when the ventilator fails to meet the patient's flow demand. This usually happens when the flow delivery is set too low or when the combination of tidal volume (V_T) and inspiratory time fails to provide adequate flow (18). The end of inspiration, cycling, is assessed in terms of synchrony with the end of the patient's inspiratory effort (i.e., Pmus). It can occur *early* (before the end of the patient's inspiratory effort), on time (synchrony), or *late* (a clinically important delay). During expiration, the patient-ventilator interaction is characterized by work, as normal expiration is passive. Patient expiratory work may be normal, as when exercising or coughing, but it may also indicate the presence of anxiety, acidosis or high resistive load (e.g., chronic obstructive pulmonary disease) (19).

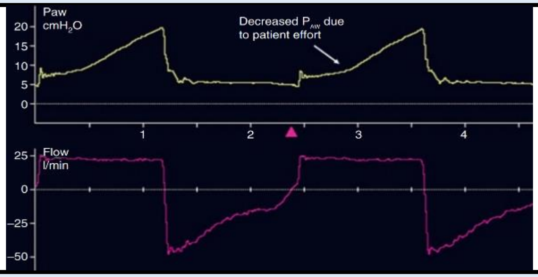
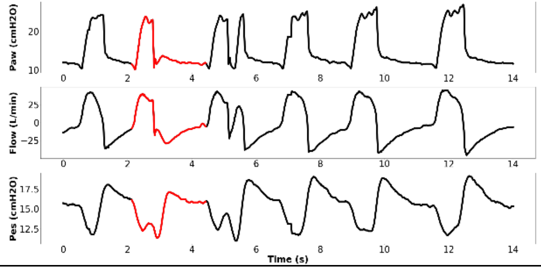
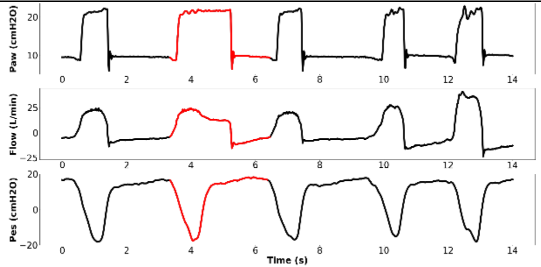
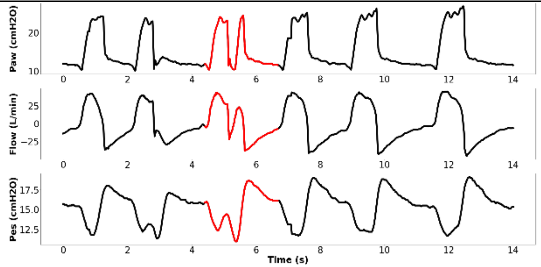
According to Esperanza et al. (20), categorizing asynchronies based on the phase of the respiratory cycle in which they occur does not correspond well with the clinical and pathophysiological mechanisms involved. Other authors agree that it is better to focus on the conditions that cause PVA in order to understand the underlying mechanisms and develop treatment strategies (21, 22).

Esperanza and coworkers focus on classification based on the appropriateness of ventilator assistance level (20). They divide asynchronies into two categories: *insufficient assistance* (patients with high respiratory drive) and *overassistance* (patients with low respiratory drive). Asynchronies caused by insufficient assistance include *flow starvation, short-cycling, double triggering* and *breath stacking*. Flow starvation is similar to *flow asynchrony* as described by De Wit (18) and *work shifting* as described by Chatburn et al. (19). Short-cycling corresponds to *premature cycling* as described by De Wit (18) and *early cycling* as described by Chatburn et al. (19). Esperanza et al. (20) consider double triggering and breath stacking to be the same phenomenon, referring to two complete inspirations separated by a very short expiratory time. However, some authors argue that breath stacking should not be used interchangeably with double triggering. According to them, the term should only be used to refer to the clinical consequence of double triggering when incomplete expiration between two breaths results in a higher than intended V_T (23). Overassistance includes *ineffective effort during expiration* (IEE) and *delayed* or *prolonged cycling*. Ineffective efforts, defined by Chatburn et al. (19) as *failed triggers*, are the most common type of asynchrony, affecting nearly 50% of mechanically ventilated patients (18, 24, 25). IEEs occur when the patient's attempt to initiate a breath does not reach the ventilator's trigger threshold. Delayed or prolonged cycling corresponds to *late cycling*, and it occurs when mechanical insufflation continues after neural inspiration has ended. Overassistance results in hyperventilation, hypocapnia, and respiratory alkalosis, which decreases respiratory drive and increases the likelihood of PVA in a vicious circle (26). The phenomenon of *reverse triggering* is discussed separately by Esperanza et al. (20). In reverse triggering, mechanical insufflation elicits a neural response, resulting in a ventilator-induced diaphragmatic contraction. According to Chatburn et al. (19), this phenomenon is referred to as *early triggering*.

It is clear that there is no consistent nomenclature or taxonomy for PVA. Training on ventilator waveforms is frequently based on simple pattern recognition (e.g., double triggering, ineffective triggering) and thus becomes an experience-based exercise rather than a systematic process. There is no widely accepted, formal, systematic method for reading ventilator waveforms, as there is for electrocardiograms (ECG). Existing automated detection algorithms for PVA are often validated by comparing them to the gold standard, which is the visual inspection of ventilator waveforms by an expert. However, because all experts in this field use different definitions of PVA, these algorithms are unreliable. To reliably detect PVA, a detection algorithm based on a reference signal of the patient effort, such as the Pes signal, is highly required.

Table 1 Taxonomy of patient-ventilator asynchrony (17-22, 27). PVA types are divided into trigger asynchronies, inspiration asynchronies, and termination asynchronies. The terminology of PVA types that we use in this study is presented in bold while the other names that appear in the literature are presented in italic. The graphic representations of late triggering and flow asynchrony (27) could not be plotted from our dataset as these types of PVA were not present in our study population.

PVA type	Description	Waveform characteristics	Graphic representation	Possible causes
Trigger asynchronies – during the beginning of inspiration				
Ineffective effort during expiration (IEE) • *Ineffective trigger* • *Failed trigger* • *Missed trigger* • *Wasted effort*	Patient effort not followed by a mechanical breath	Presence of a negative deflection in the Pes signal in the expiratory phase of a breath without triggering a new mechanical breath		• Inadequate trigger sensitivity • Overassistance • Sedation • Presence of AutoPEEP • Low respiratory drive
Reverse triggering (RT) • *Early trigger* • *Early inflation*	A machine-induced breath preceding patient effort	A controlled machinal breath followed by a negative deflection in the Pes signal		• Oversedation • Overdistention
Late triggering • *Delayed trigger* • *Late inflation*	The ventilator responding to patient effort after a clinically important delay (e.g., 100 ms (28))	Paw drops below baseline, positive deflection in flow-time or start negative deflection Pes signal > 100 ms before start mechanical breath	 Paw: 5.0 cmH₂O Flow: -3.9 l/min Pes (Paux): 5.2 cmH₂O	• Inadequate trigger sensitivity • Presence of AutoPEEP • Low respiratory drive
Auto triggering • *False trigger*	A nonpatient (e.g., non-Pmus) signal triggering a mechanical breath	A triggered breath without the presence of a negative deflection in the Pes signal		• Air leaks in the endotracheal tube cuff, ventilator circuit or chest tube • Flow oscillations (water of secretion in the circuit, cardiac oscillations)

PVA type	Description	Waveform characteristics	Graphic representation	Possible causes
Inspiration asynchronies – during the gas delivery				
Flow asynchrony • *Flow starvation* • *Work shifting* • *Flow limited* • *Insufficient flow* • *Inspiratory airflow dyssynchrony*	The delivered flow not meeting the patient's inspiratory flow demands	An upward concavity in Paw preceding the end of a mechanical breath		• Inadequate flow • Dyspnoea • Delirium/Pain
Termination asynchronies – during the end of inspiration				
Premature cycling • *Early cycling* • *Short cycling* • *Premature termination*	When inspiration ends before the end of patient effort	Observed termination of delivered breath while negative deflection in the Pes signal is still present		• Inadequate cycling criteria • Inadequate setting of ventilator inspiratory time
Delayed cycling • *Late cycling* • *Prolonged cycling* • *Delayed termination* • *Runaway phenomena*	When inspiration ends after the end of patient effort	Observed continuation of delivered breath while negative deflection in the Pes signal being no longer present		• Inadequate cycling criteria • Inadequate setting of ventilator inspiratory time
Double triggering (DT) • *Double cycling*	Two (or more) mechanical breaths are delivered during one single inspiratory effort	Two assisted breaths during one negative deflection of Pes signal		• Inadequate trigger sensitivity • Inadequate setting of ventilator inspiratory time • High respiratory drive

Pmus: inspiratory pressure generated by the respiratory muscles; Pes: esophageal pressure; Paw: airway pressure; AutoPEEP: intrinsic positive end-expiratory pressure.

3

Methods

3.1 Study population and data collection

The patients considered in this study are adults who were admitted to the ICU of the Leiden University Medical Center from December 2022 to March 2023. Only patients receiving invasive mechanical ventilation because of acute respiratory failure or with a ventilation duration of at least 48 hours are included. Patients are required to have an esophageal balloon catheter in order to measure the esophageal pressure (Pes) (Hamilton Medical AG, Bonaduz, Switzerland). Patients are only included if the patient or next of kin gives consent to the use of their data. In our study, 17 patients were eligible for inclusion, however, only 11 are included in the final dataset due to several reasons (**Figure S2, Supplementary materials 2**).

The Pes is recorded in conjunction with the ventilator airway pressure (Paw) and flow-time waveforms. All signals are recorded on a dedicated data acquisition system (Hamilton Medical AG). Only the recordings under pressure control ventilation (PCV), pressure support ventilation (PSV), and adaptive support ventilation (ASV) modes from the Hamilton-C6 ventilator (Hamilton Medical AG) are included. The baseline demographics, ICU admission diagnoses, and outcome of the mechanically ventilated patients enrolled in the study are collected. The study is approved by the local monitoring board Medisch Ethische Toetsingscommissie – Leiden Delft Den Haag (No. 2022-061).

3.2 Data screening

For data annotation purposes, the raw recordings were manually analyzed to identify a region of interest (ROI) where PVA was most prevalent. These target ROIs are essential for ensuring that the dataset contains sufficient PVA cycles. Given that PVA is a relatively rare event, identifying these ROIs across the patient recordings remains a difficult task.

One hour of data was selected for annotation for each patient, so that all patients contribute more or less equally to the dataset. This yielded 11 hours of data for annotation across all patients. The screening was performed by a technical medicine student who accepted professional education and training on MV and PVA recognition. The ROIs were selected based on visual inspection, with the aim of including parts of the recordings with as many different types of cycles as possible (normal breaths, PVA, and artefacts). The remainder of the recordings are excluded from the study.

3.3 Data annotation

Six types of PVA were considered; double triggering (DT), ineffective effort during expiration (IEE), reverse triggering (RT), auto triggering, premature cycling, and delayed cycling. Based on expert opinion and literature (11, 28), we created heuristic rules for detecting different types of PVA and implemented them in an annotation protocol (**Supplementary materials 3**). DT occurs when the patient's inspiratory effort exceeds the ventilator's inspiratory time, resulting in a second inspiration triggered by the same effort. IEE happens when a patient tries but fails to trigger a breath during the expiratory phase. RT occurs when a machine-induced breath precedes patient effort. Auto triggering is a mechanical breath being triggered without the presence of patient effort. Premature cycling occurs when the inspiration ends before the end of patient effort, while delayed cycling occurs when inspiration ends after the end of patient effort. The typical waveforms of these asynchronies are shown in **Table 1**.

The waveforms were manually annotated by four mechanical ventilation experts among the staff of the ICU. The tasks were assigned to the annotators at random, and annotation occurred independently. A self-developed software programmed in Python 3.9 with Label Studio 1.7 as the user interface library was used for annotation (see **Figure 2**). Continuous time-dependent ventilator waveforms (Paw, flow-time, V_T and Pes) were provided to the annotators. Each breath was manually annotated based on visual inspection and the annotation protocol. Next to the six asynchronies discussed above, there were three other labels that could be assigned to the data during annotation: cough, peristalsis, and other artefacts. Clinical artefacts such as cough and peristalsis have morphological similarities to common forms of PVA. Therefore, it was considered essential to explicitly identify and include these in the dataset in order to reduce the false positive detection rate for PVA classification. Furthermore, the annotation of these clinical artefacts is thought to be useful for potential future work. Normal breaths were defined as breaths not classified as PVA or artefact.

Data annotation was used to create a gold standard dataset of classified PVA observations for our supervised learning algorithm. The goal of our model is to achieve comparable classification performance as experts with access to the Pes signal, identifying PVA while discarding artefacts, without the labor-intensive visual inspection performed by experts.

Table 2 provides a comprehensive statistical overview of the breath types and numbers annotated in this study. Even after selecting ROIs with enriched PVA frequency, our dataset has a low frequency of PVA and clinical artefacts when compared to normal respiratory cycles. Because some PVA are too rare in our dataset, it was decided to exclude them from analysis. A minimum of 1% of the number of normal respiratory cycles was considered a requirement to be included in model training. As a result, only breaths labeled normal, reverse triggering, premature cycling, cough, peristalsis, or other artefacts were used for model development.

Table 2 The different breath types, their frequency in the dataset, and the number of patients who experienced them. Even after selecting ROIs with enriched PVA frequency, our dataset has a relatively low frequency of PVA and clinical artefacts as compared to normal respiratory cycles. Therefore, only the highlighted PVA types were selected to be included in model development.

Event Type	Number	Percentage (%)	Patients (n)
Normal	10.285	83.4	11
DT	19	0.2	4
IEE	4	0.03	2
RT	836	6.8	5
Auto triggering	6	0.05	2
Premature cycling	112	0.9	4
Delayed cycling	13	0.1	3
Cough	274	2.2	9
Peristalsis	453	3.7	11
Other artefacts	335	2.7	11
Total	12.337		

DT: double triggering; IEE: ineffective effort during expiration; RT: reverse triggering.

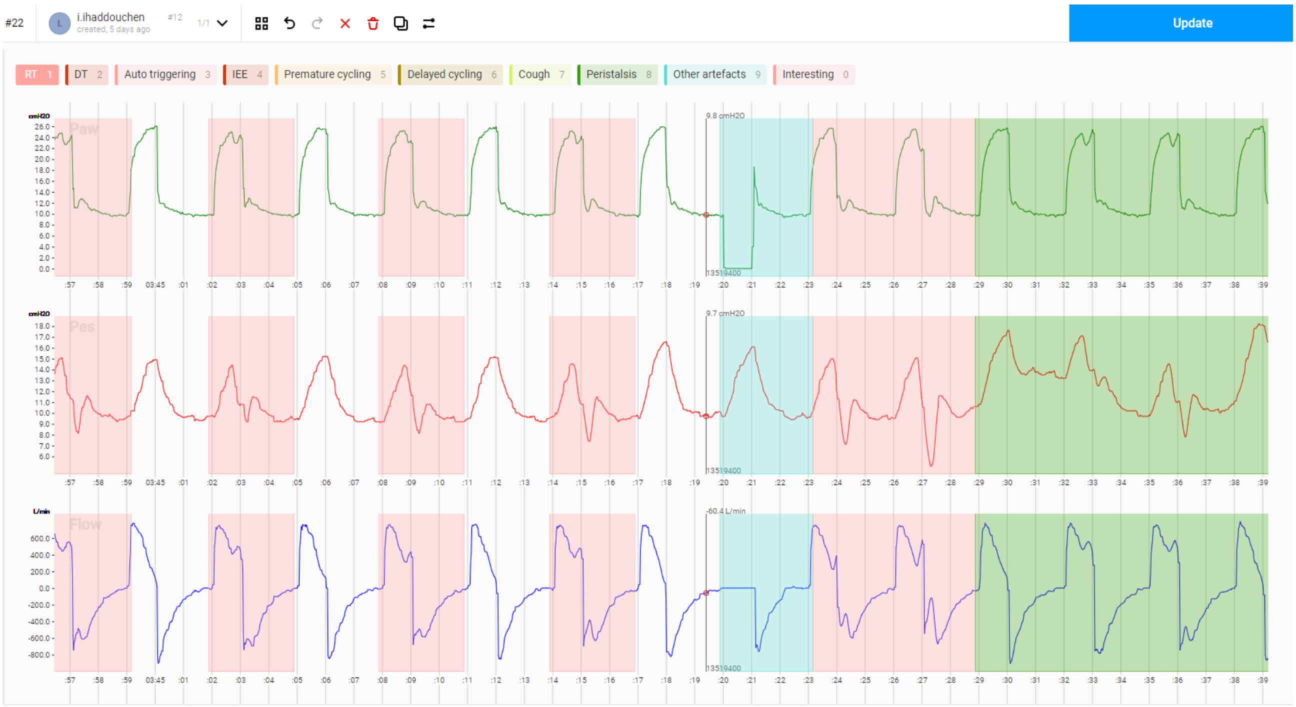


Figure 2 User interface of the annotation software.

3.4 Preprocessing

Before training the model, the raw continuous ventilator Paw, flow-time and Pes waveforms were preprocessed and some adjustments were made to the dataset.

3.4.1 Data transformation

In order to create appropriate inputs for a CNN, the ventilator waveforms are transformed. **Figure 3** depicts the data transformation steps in our study. First, the signals are segmented into individual respiratory cycles. Then, as CNNs can only accept a fixed size tensor as input, the Paw, flow-time and Pes of each segment are resampled to a uniform length of 300. In essence, this yields a 300 x 3 size image for each breath. Although the raw sample rate of the breath is altered by this resampling process, the characteristic shape of the asynchronous breath is still preserved for further analysis.

After resampling, each segment's amplitude is normalized using feature scaling in accordance with Equation (1), where X_i denotes the signal's amplitude value at the i_{th} sample point of the segment, X_{max} denotes the signal's maximum value inside the segment, and X_{min} denotes its minimum value.

$$X_i' = \frac{X_i - X_{min}}{X_{max} - X_{min}} \tag{1}$$

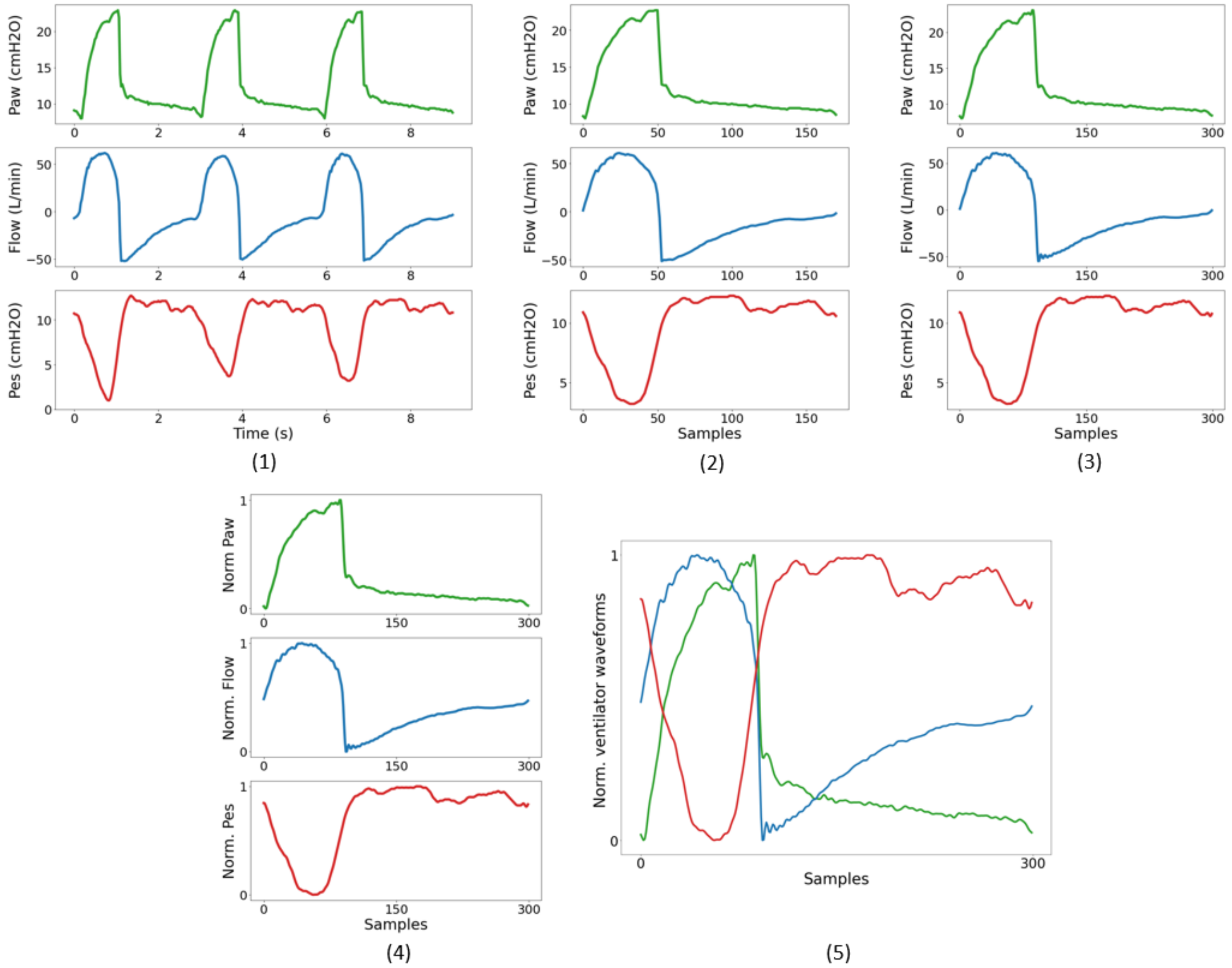


Figure 3 Schematic of the data transformation. **(1)** Ventilator waveforms prior to preprocessing. **(2)** Ventilator waveforms are segmented into individual respiratory cycles. **(3)** The duration of each segment is resampled to a uniform length of 300. **(4)** The amplitude of each segment is normalized. **(5)** The combination of all ventilator signals yields a 300 x 3 image that serves as the input tensor for the CNN.

3.4.2 Data cleaning

Following data transformation, some steps are taken to clean the dataset. First, all breaths with a missing Pes signal are removed from the dataset. Second, as discussed in **Section 3.3**, the minimum number of breaths per breath type in our dataset must be 1% of the number of normal respiratory cycles to be included in model training. As a result, all breaths with the labels DT, IEE, auto triggering, and delayed cycling are excluded from the analysis because they occurred too infrequently in the dataset to adequately train the algorithm to detect these types of PVA.

3.4.3 Class imbalance

Despite selecting ROIs for PVA and artefact enrichment in our dataset, the relatively low proportion of abnormal breath types resulted in a significant class imbalance problem (29). Because samples from the small, but important, classes can be overwhelmed by the majority class samples, unbalanced training sets can often be an obstacle to training accurate ML models (30). We experimented with various techniques to generate a balanced dataset for model training (see **Figure 4**). First, we experimented with the synthetic minority over-sampling technique (SMOTE) (31). SMOTE addresses the issue of class imbalance by generating synthetic samples of minority class observations, i.e. PVA and artefacts. SMOTE was used with a minority class to majority class observations ratio of 1:2. As a result, we maintained a constant number of normal breaths, say X, while upsampling the number of breaths in the other classes to $0.5X$. In this way, we attempted not to disrupt the relationship to the real world situation too much, where normal breaths are more common than PVA. Second, we used the random undersampling technique (RUS) to randomly remove samples from the normal breath subset (32). In this method we used the average number of breaths in all minority classes, say Y, and undersampled the number of normal breaths to $3Y$. To ensure that the majority class is larger than the largest minority class, we chose a factor 3 rather than a factor 2. Finally, we tried a combination of the two techniques in which we first oversampled all minority classes to 30% of the initial number of normal breaths, i.e. $0.3X$. The majority class observations (normal breaths) were then undersampled to twice as much, i.e. $0.6X$, to achieve the 1:2 minority class to majority class observations ratio. We compared the impact of these techniques on the performance of our model to determine which method works best for our data.

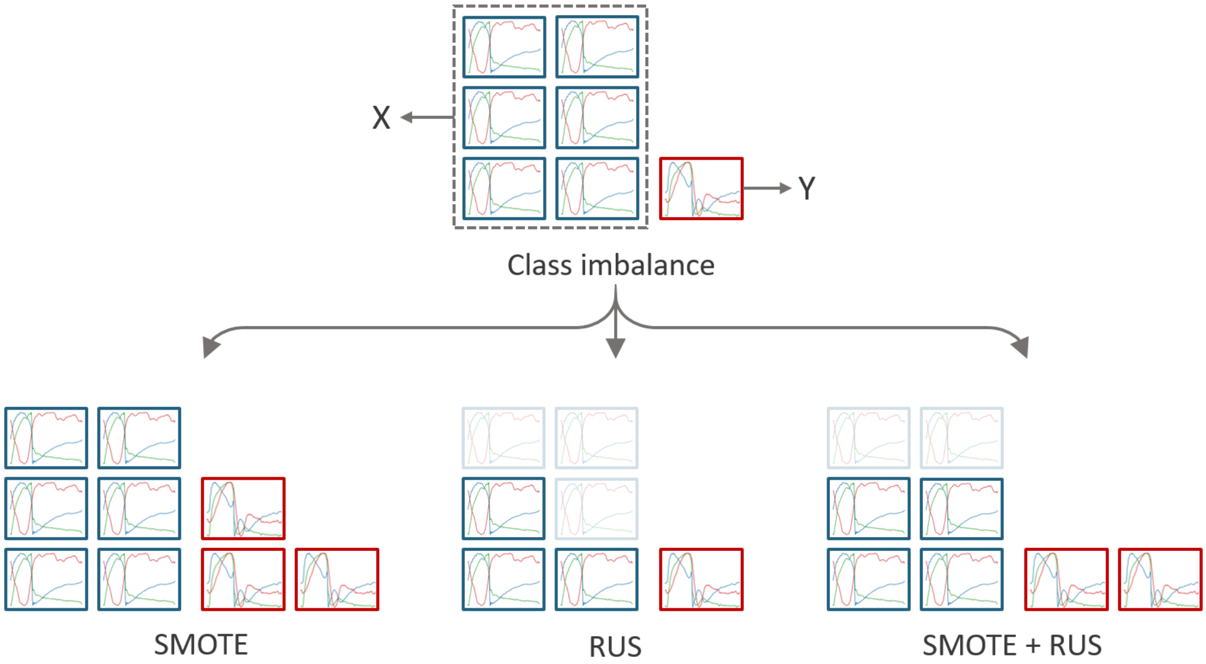


Figure 4 Class imbalance solutions examined in this study. X is the number of normal respiratory cycles (blue outlined graphs) in the original dataset and Y is the average number of PVA and artefacts (red outlined graphs) in the original dataset. SMOTE upsamples the number of samples in the minority classes (i.e., PVA and artefacts) with a minority class to majority class observations ratio of 1:2. RUS undersamples the number of majority class observations (normal respiratory cycles) with a minority class to majority class observations ratio of 1:3. A combination of SMOTE and RUS was used with a minority class to majority class observations ratio of 1:2. SMOTE: synthetic minority over-sampling technique; RUS: random undersampling technique.

3.5 The model architecture

The 2DCNN model architecture is illustrated in **Figure 5** and the detailed network configurations are shown in **Table 3**. The model consists of one convolutional branch with four convolutional blocks for extracting the features of the ventilator Paw and flow-time and Pes waveforms. Each block consists of a convolutional layer, a batch normalization (BN) layer and a max pooling layer. The decision was made to replace the traditional 2D convolution with dilated convolution. This method expands the kernel by inserting holes between the consecutive elements. This allows the same kernel to be applied to a larger portion of the input breath. The aim is to generate higher resolution feature maps, capturing information over a larger area with more context while using the same number of parameters (33).

During training, the distribution of each layer's inputs changes as the parameters of the previous layers change. This is referred to as internal covariance shift and is known to slow down the training (34). BN can stabilize the distribution of nonlinear inputs while the model trains. Therefore, we decided to add a BN layer after each convolutional layer but before the activation function.

The convolutional blocks are followed by two fully connected layers. The number of fully connected layers was determined empirically. Initially, only one fully connected layer was used. However, adding another fully connected layer improved the performance. This might be due to the model's ability to learn a more complex and flexible decision boundary to distinguish PVA from non-PVA breaths with two fully connected layers. However, as shown in **Table 3**, adding another fully connected before the final fully connected layer greatly increased the number of trainable parameters, increasing the risk of overfitting. As a result, it was decided to add a dropout layer after the first fully connected layer and before the output layer. During training, this layer randomly drops units (along with their connections) from the neural network. This prevents units from excessive co-adaptation. Dropout has been shown to significantly reduce overfitting and outperform other regularization techniques (35, 36). The output of the last fully connected layer is fed into a 6-way softmax activation function, which generates a probability distribution over the six possible class labels. The final output is a vector containing the estimated probabilities of each class label, given the input breath (37).

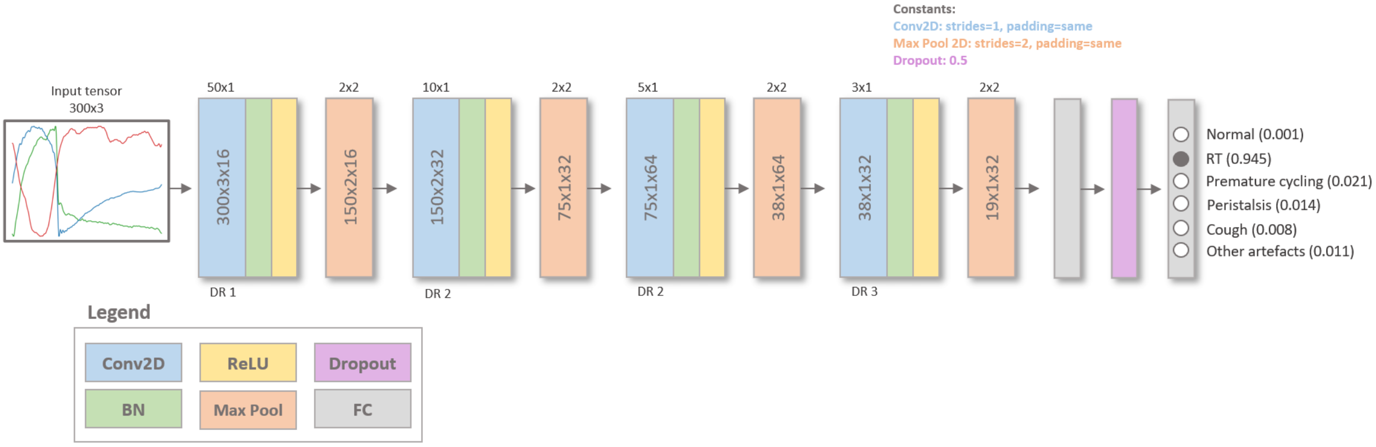


Figure 5 The proposed 2DCNN's model architecture, along with the procedure for generating a probability distribution over the six possible class labels given the input breath. In this visual representation, the model has generated the highest probability for reverse triggering (0.945). Conv2D: two-dimensional convolutional layer; ReLU: Rectified linear unit; BN: batch normalization; Max Pool: max pooling layer; FC: fully connected; DR: dilation rate; RT: reverse triggering.

Table 3 Layer details and parameters used for the proposed 2DCNN 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

ReLU: Rectified linear unit

To lessen information loss at the edges of the input feature maps during convolution, we used padding in each convolutional layer of our initial model. Padding is the technique of adding additional rows and columns of zeros to the input feature map before performing convolution, in order to maintain the spatial dimensions of the input and output feature maps.

We experimented with a few changes to our initial model for optimization purposes. First, we wanted to experiment with the kernel size in the convolutional layers. As such, the effect of changing the first dimension of the kernel size of the kernels in the first convolutional layers on the performance was assessed. We started out with a small kernel size and increased it after evaluating the model's performance. The following kernel sizes were tested: 5 x 1, 10 x 1, 20 x 1, and 50 x 1. We found that changing the kernel size in the first dimension had little effect on both the performance and the number of trainable parameters. As a result, we chose the kernel size of 50 x 1 and established this model as our initial model (Initial_2DCNN).

In comparison to the first dimension, the second dimension of our input tensor is relatively small (3 vs. 300). The previously discussed kernels with size A x 1, with A being 5, 10, 20 and 50, are able to move both horizontally and vertically over the 300 x 3 input breath. Using a kernel that only moves horizontally over the input tensor highly reduces the number of training parameters, and consequently, the computational cost of the model. To determine whether the use of such a kernel would maintain the initial model's performance, we changed the kernel size in the first convolutional layer to 50 x 3. Furthermore, we stopped using padding in each convolutional layer in order to prevent zeros from being added around the input feature map. In this way, we can ensure that the kernels in the convolutional layers can only move horizontally over the input feature maps. We named this model Conv_2DCNN and provided its detailed network configurations in **Table S1** (**Supplementary materials 4**).

In addition to the kernel size in the convolutional layers, we wanted to experiment with the pool size in the 2D Max Pooling layers. Therefore, we changed the pool size from 2 x 2 to 2 x 1. The aim of the pooling layers, as discussed in **Section 2.1**, is to downsample the feature maps in order to reduce the network's computational costs. The pool size specifies which dimensions of the input data are reduced. In our initial model, we used a pool size of 2 x 2 to reduce the data in both dimensions. This means that in each pooling layer, the amount of data in both the first and second dimensions is divided by two. As a result, the output shape in the second dimension decreases from 3 to 1 across the layers. However, the second dimension of our data contains the ventilator waveforms, which are expected to contain the most important features for the CNN to learn. Therefore, we were interested in the performance with a pool size of 2 x 1 in which only the first dimension of the feature maps is reduced in each pooling layer. This means that after each pooling layer, the shape of the output in the second dimension remains 3. We named this model Pool_2DCNN and the detailed network configurations of this model are presented in **Table S2** (**Supplementary materials 4**).

3.6 Performance evaluation

The model we have developed can be described by Equation (2):

$$y_{ik} = f(x_i) \tag{2}$$

Let f be the model that is trained on N training samples, x_i , for $i = 1,2,3, \ldots, N$. Each sample is a segmented breath which is a two-dimensional matrix of size 300 x 3, indicating 300 timepoints and 3 ventilator waveforms (Paw, flow-time and Pes). The corresponding breath labels, y_i , for $i = 1,2,3, \ldots, N,$ are one-hot encoded labels, denoted by y_{ik}, for $k = 0,1, \ldots, k-1$, where k is the total number of classes. In our study, $k = 6$ because we are classifying six different breath labels.

In order to train and evaluate the proposed model f, the data was divided into two parts: a training set and a test set. Several calculations are repeated during model training with the aim of minimizing the expected value ($E_{(x,y)}$) of a loss function $L(f(x), y)$, resulting in the optimization of model function f to f^*, as shown in Equation (3):

$$f^* = arg\ min_f E_{(y,x)} L(f(x), y) \tag{3}$$

This optimized model function f^* is used in the testing phase to predict the labels of data that were not used during training. In this study, we used categorical cross-entropy as the loss function. When the validation loss did not decrease for 5 consecutive epochs, the training was stopped, and the model was saved as the best one.

For model training and testing, we used leave-one-patient-out cross-validation to divide patients into a training and testing cohort. In this way, a single patient's observations cannot be in both training and test sets, which may introduce bias and lead to poor generalizability to subsequent patients. This bias can be caused by intra-patient waveform similarities resulting from static ventilation settings and other patient-specific physiologic factors.

To assess the classification metrics, we used 11 folds, corresponding to the number of patients in our dataset. Thus, in each fold 10 patients were used for training and 1 patient was used for testing until all patients were evaluated. This strategy is illustrated in **Figure 6**. In each iteration, the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) outcomes were determined to evaluate several performance metrics. It was decided to evaluate the sensitivity and specificity as these are two established methods for validating clinical alarm algorithms (38). In addition, we assessed the accuracy as this is a simple and intuitive metric that is easy to interpret. However, because we are dealing with imbalanced data, we also decided to evaluate the F1 score as this metric emphasizes the detection of the positive class (PVA) over the detection of the negative class (normal breaths). To determine the average model performance, these metrics are computed in each fold according to Equation (4) to (7), and then averaged over all folds.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

$$F1 = \frac{2\,TP}{2\,TP + FP + FN} \tag{7}$$

These metrics use a fixed threshold to either classify an observation as positive or negative. This threshold is often set by default at 0.5. In order to evaluate the models' performance across different classification thresholds, the ROC curves of the models were plotted in each fold of the cross-validation loop. The area under the ROC (AUROC) curve was subsequently averaged across all folds.

After evaluating different solutions to the class imbalance problem and experimenting with changes to the model architecture, we chose the model with the best performance per PVA type (i.e., RT and premature cycling) to conduct the final performance evaluation. In our final performance evaluation, we compared the performance of our 2DCNN based on the Paw, flow-time and Pes waveforms of the ventilator (Pes_2DCNN) with a 2DCNN solely based on the Paw and flow-time waveforms (PF_2DCNN). The PF_2DCNN was trained using the same dataset and evaluated using the same protocol as with our Pes_2DCNN. To allow for comparison, the batch size and number of epochs were set to be the same for both models.

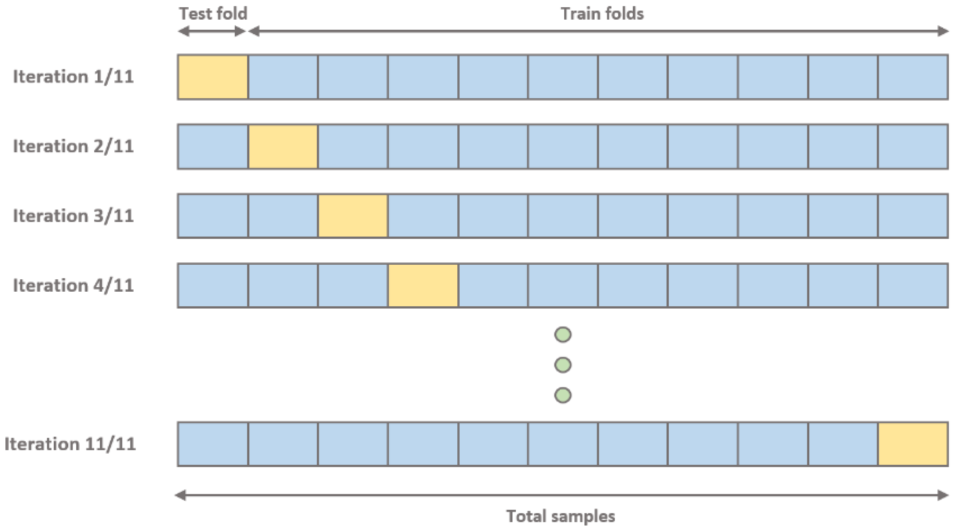


Figure 6 Leave-one-patient-out cross validation. In this study, 11 folds were used according to the number of included patients.

3.7 Statistical analysis and software

Preprocessing and development of the models was performed in Python 3.9 using the following packages: NumPy 1.21.5, Pandas 1.3.5, SciPy 1.9.1, Matplotlib 3.5.2, Scikit-learn 1.0.2, TensorFlow 2.11.0 and Keras 2.11.0.

Patient characteristics and performance metrics are reported as mean ± standard deviation (std. dev.). The performance of the Pes-based 2DCNN (Pes_2DCNN) and the 2DCNN solely based on the ventilator's airway and flow-time signals (PF_2DCNN) was compared using DeLong test (39, 40). A two-sided p-value of 0.05 was considered statistically significant.

4

Results

4.1 Patient characteristics

Of the 17 patients who met the inclusion criteria, 11 were enrolled in the study, while the remaining six were excluded for various reasons (**Figure S2**, **Supplementary materials 2**). **Table 4** shows the demographics, clinical information on ICU stay, and conditions that led to the initiation of mechanical ventilation of the 11 patients enrolled in this study. Patients were 55.8 ± 11.8 years old on average, with 5 (45%) being female. Mechanical ventilation was initiated because of a respiratory disease in three (27%) patients, a cardiac disease in three (27%) patients, sepsis in two (18%) patients, an operation in two patients (18%), and acute liver failure in one (9%) patient. The average length of ICU stay was 27.4 ± 20.9 days, and the mean duration of mechanical ventilation was 23.9 ± 16.6 days. During data collection, patients were deeply sedated (low RASS value).

Table 4 Patient demographics and clinical data.

Patient	Age (years)	Gender	Weight (kg)	Reason for MV	Days in ICU	Days on MV	RASS
1	40	Female	59	Cardiac disease	17	17	- 2
2	47	Male	90	Postoperative	67	50	- 4
3	75	Female	95	Sepsis	28	28	- 4
4	61	Male	93	Postoperative	19	15	- 4
5	65	Female	100	Sepsis	56	44	- 4
6	57	Female	87	Respiratory disease	7[a]	7[a]	- 4
7	55	Male	62	Respiratory disease	7	6	- 4
8	44	Male	35	Cardiac disease	50	46	- 4
9	39	Female	63	Acute liver failure	8	8	- 4
10	58	Male	128	Cardiac disease	36[b]	36[b]	- 5
11	73	Male	100	Respiratory disease	6[a]	6[a]	- 5
Mean ± std. dev.	55.8 ± 11.8		82.9 ± 24.6		27.4 ± 20.9	23.9 ± 16.6	- 4 ± 0.7

MV: mechanical ventilation; ICU: intensive care unit; RASS: Richmond Agitation Sedation Scale; std. dev.: standard deviation. *a*: This patient was transferred to another hospital's ICU during admission. As a result, the stated days in ICU and days on MV only cover admission to LUMC's ICU.

b: This patient was still admitted to the ICU at the end of this study. As a result, the stated days in ICU and days on MV are not the definite durations.

Table 5 summarizes the ventilator settings and the patients' respiratory mechanics during data collection. The ventilator mode was PSV in seven (64%) patients, PCV in two (18%) patients, and ASV in two (18%) patients.

Table 5 Ventilator settings and respiratory mechanics of the patients during data collection.

Patient	MV mode	Inspiratory pressure (cmH2O)	Applied PEEP (cmH2O)	V_T (ml)	RR (breaths/min)
1	PSV	14	11	402	24
2	PSV	12	10	470	27
3	ASV	11	12	397	21
4	PSV	8	8	557	17
5	PSV	16	8	395	30
6	ASV	20	14	458	21
7	PCV	17	8	299	26
8	PSV	11	8	514	19
9	PSV	8	5	573	17
10	PSV	10	8	456	28
11	PCV	16	10	459	20
Mean ± std. dev.		13 ± 3.7	9.3 ± 2.3	452.7 ± 75.1	22.7 ± 4.3

MV: mechanical ventilation; PEEP: positive end expiratory pressure; V_T: tidal volume; RR: respiratory rate; PSV: pressure support ventilation; ASV: adaptive support ventilation; PCV: pressure control ventilation; std. dev.: standard deviation.

4.2 Performance evaluation

4.2.1 Approach selection for final model evaluation

Figure 7 illustrates the performance comparison among Initial_2DCNN, Conv_2DCNN, and Pool_2DCNN for RT detection using the original class imbalanced dataset. Additionally, the figure shows the comparison of these models using the three different solutions for the class imbalance problem, i.e. SMOTE, RUS, and a combination of SMOTE and RUS. Whereas the accuracy and specificity appear to be acceptable, the sensitivity and F1 score are extremely low. However, these metrics only reflect performance based on a default threshold of 0.5 for differentiating between positive and negative class observations. The area under the ROC (AUROC) curve provides a more comprehensive representation of the model's ability to differentiate between classes as the performance of the models is analyzed for all possible threshold values (41). When evaluating the AUROC figure, it is important to note that the applied class imbalance solutions have little to no effect on the models' performance when compared to using the imbalanced dataset. When the class imbalance solutions were used, the performance of Initial_2DCNN and Conv_2DCNN even deteriorated.

We found a slight advantage in AUROC for Conv_2DCNN on the class imbalanced dataset over the other approaches (0.80 ± 0.07). Consequently, it was decided to use this approach for the final performance comparison of our Pes-based 2DCNN (Pes_2DCNN) and the 2DCNN solely based on the ventilator's Paw and flow-time waveforms (PF_2DCNN) for the detection of RT.

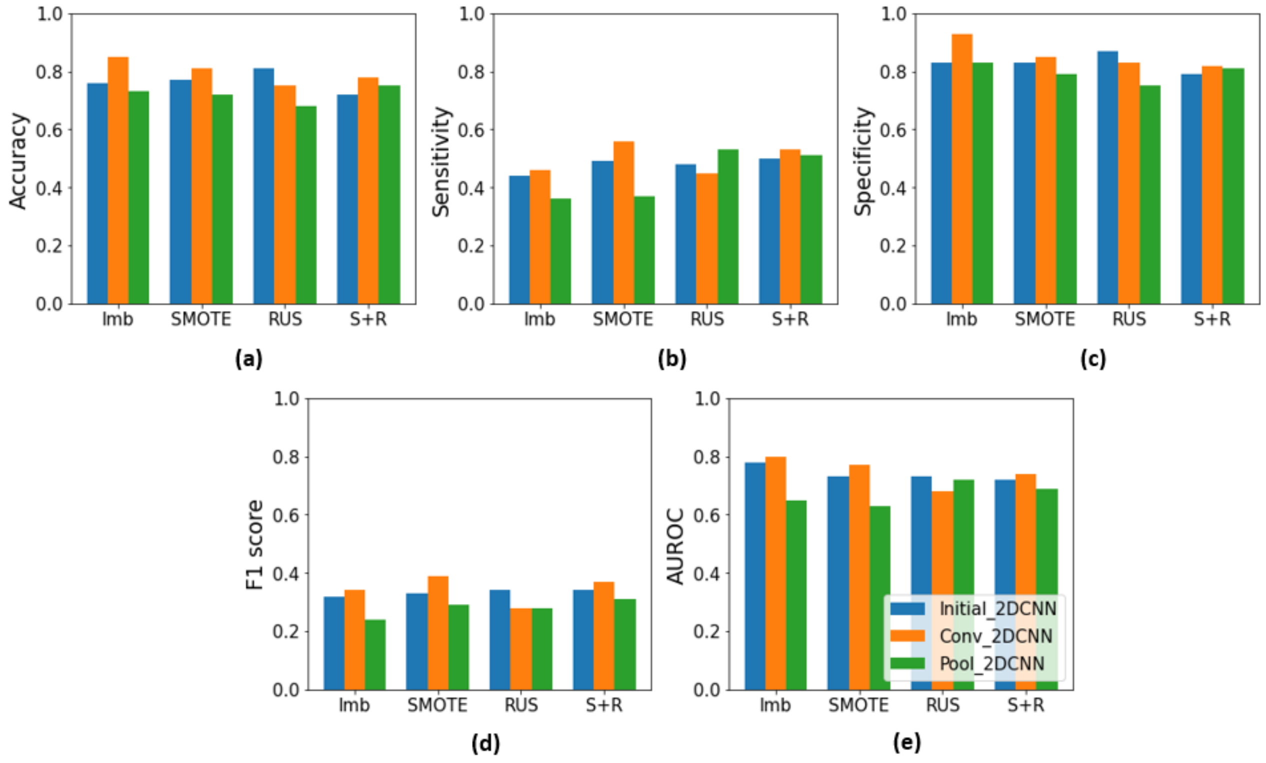


Figure 7 Performance comparison among Initial_2DCNN, Conv_2DCNN, and Pool_2DCNN for RT detection using the original class imbalanced dataset and the three different solutions for the class imbalance problem: SMOTE, RUS, and a combination of SMOTE and RUS. **(a)** Accuracy, **(b)** Sensitivity, **(c)** Specificity, **(d)** F1-score, **(e)** Area under the ROC curve. Initial_2DCNN: the initial 2DCNN prior to model adjustments; Conv_2DCNN: 2DCNN with kernel size of 50 x 3 in first convolutional layer; Pool_2DCNN: 2DCNN with pool size of 2 x 1; Imb: imbalanced dataset; SMOTE: synthetic minority over-sampling technique; RUS: random undersampling technique; S+R: combination of SMOTE and RUS; AUROC: area under the receiver operating characteristic.

Figure 8 compares the performance of Initial_2DCNN, Conv_2DCNN, and Pool_2DCNN in detecting premature cycling using the original class imbalanced dataset and the three different class imbalance solutions. For this type of asynchrony, Initial_2CNN based on a combination of SMOTE and RUS resulted in the highest AUROC (0.88 ± 0.09). As a result, we decided to apply this approach to the final performance comparison of Pes_2DCNN and PF_2DCNN for the detection of premature cycling. Performance results under Initial_2DCNN, Conv_2DCNN, and Pool_2DCNN for the detection of all PVA and non-PVA breaths are presented in **Table S3**, **Table S4**, and **Table S5** of **Supplementary materials 5**.

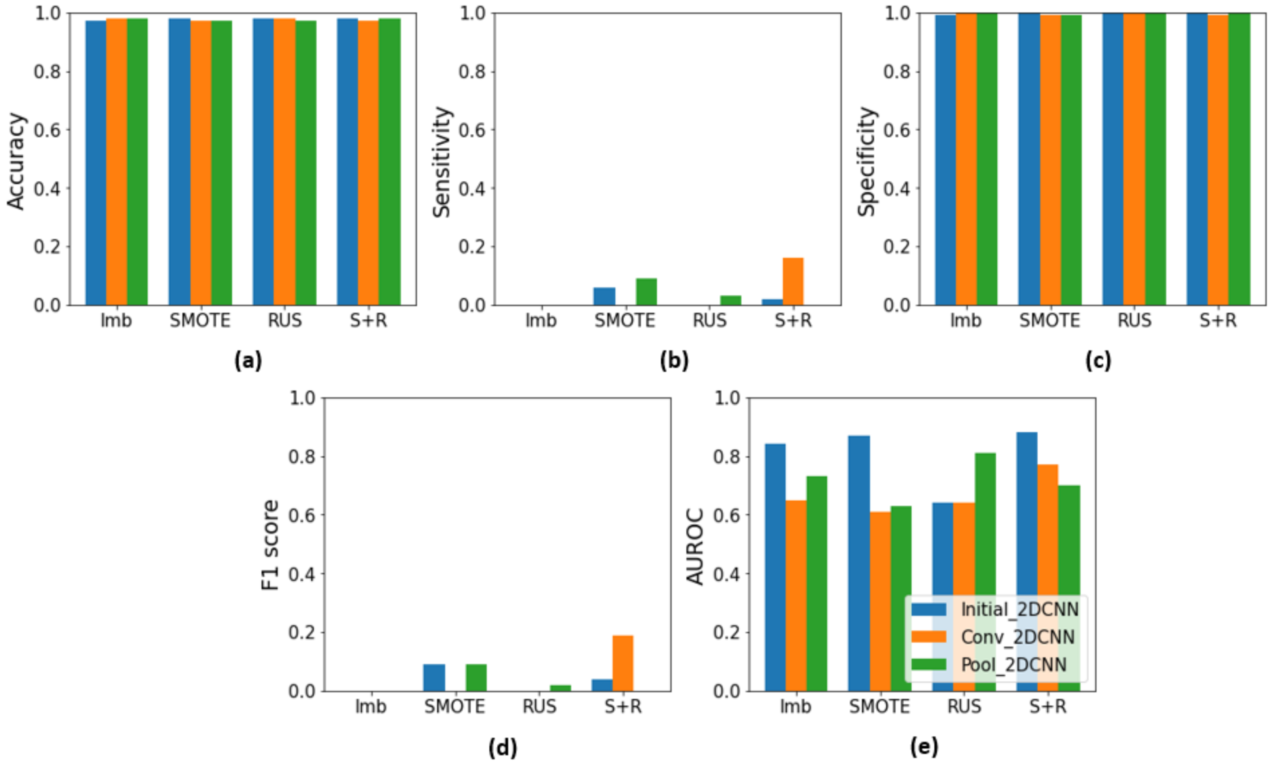


Figure 8 Performance comparison among Initial_2DCNN, Conv_2DCNN, and Pool_2DCNN for premature cycling detection using the original class imbalanced dataset and the three different solutions for the class imbalance problem: SMOTE, RUS, and a combination of SMOTE and RUS. **(a)** Accuracy, **(b)** Sensitivity, **(c)** Specificity, **(d)** F1-score, **(e)** Area under the ROC curve. Initial_2DCNN: the initial 2DCNN prior to model adjustments; Conv_2DCNN: 2DCNN with kernel size of 50 x 3 in first convolutional layer; Pool_2DCNN: 2DCNN with pool size of 2 x 1; Imb: imbalanced dataset; SMOTE: synthetic minority over-sampling technique; RUS: random undersampling technique; S+R: combination of SMOTE and RUS; AUROC: area under the receiver operating characteristic.

4.2.2 Final model evaluation

Because the best performance results for RT detection were achieved with Conv_2DCNN on the original imbalanced dataset, it was decided to use this approach to compare the performance of Pes_2DCNN and PF_2DCNN in detecting RT. **Table 6** displays the classification results of Pes_2DCNN and PF_2DCNN in detecting all breath types. The results for detection of RT are highlighted. In terms of AUROC, the proposed Pes_2DCNN model performed slightly better in detecting RT than the PF_2DCNN (0.80 ± 0.07 vs. 0.75 ± 0.13, respectively; $p < 0.01$).

Table 6 Performance comparison of Pes_2DCNN and PF_2DCNN in detecting all breath types. These results are achieved with the Conv_2DCNN model architecture on the original class imbalanced dataset. Performance metrics for detection of RT are highlighted.

Model	Type of breath	Accuracy	Sensitivity	Specificity	F1 score	AUROC
Pes_2DCNN	Normal	0.80 ± 0.21	0.85 ± 0.24	0.56 ± 0.12	0.84 ± 0.21	0.83 ± 0.18
	RT	0.85 ± 0.07	0.46 ± 0.14	0.93 ± 0.05	0.34 ± 0.27	0.80 ± 0.07
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.65 ± 0.19
	Cough	0.95 ± 0.04	0.21 ± 0.14	0.98 ± 0.03	0.20 ± 0.15	0.86 ± 0.10
	Peristalsis	0.89 ± 0.20	0.40 ± 0.18	0.91 ± 0.20	0.32 ± 0.19	0.82 ± 0.16
	Other artefacts	0.96 ± 0.05	0.17 ± 0.28	0.98 ± 0.03	0.17 ± 0.29	0.80 ± 0.15
PF_2DCNN	Normal	0.83 ± 0.16	0.90 ± 0.21	0.40 ± 0.20	0.87 ± 0.17	0.78 ± 0.10
	RT	0.84 ± 0.08	0.44 ± 0.13	0.92 ± 0.10	0.34 ± 0.28	0.75 ± 0.13
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.83 ± 0.14
	Cough	0.96 ± 0.03	0.19 ± 0.19	0.99 ± 0.02	0.22 ± 0.18	0.88 ± 0.11
	Peristalsis	0.96 ± 0.04	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.83 ± 0.14
	Other artefacts	0.96 ± 0.05	0.16 ± 0.13	0.98 ± 0.02	0.14 ± 0.14	0.55 ± 0.13

Pes_2DCNN: 2DCNN based on Paw, flow-time and Pes; PF_2DCNN: 2DCNN solely based on Paw and flow-time; RT: reverse triggering; AUROC: area under the receiver operating characteristic.

Figure 9 depicts the ROC curves for Pes_2DCNN and PF_2DCNN, as well as the corresponding AUC values for the detection of RT. It can be seen that, in addition to a better performance in detecting RT, Pes_2DCNN has lower variability in performance across different subsets of the data. ROC curves for the detection of RT for all cross-validation folds are presented in **Figure S3** of **Supplementary materials 6**.

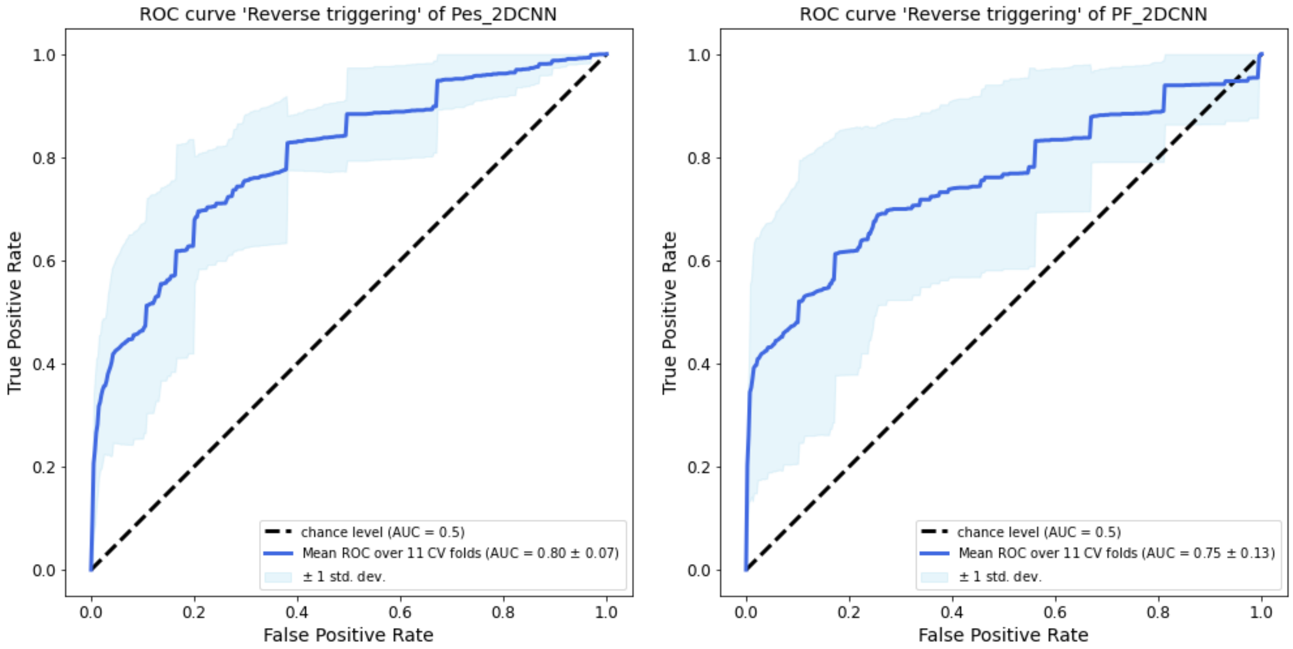


Figure 9 ROC curves for the detection of reverse triggering for both Pes_2DCNN (left) and PF_2DCNN (right) using the Conv_2DCNN model based on the original class imbalanced dataset. Difference in AUC between the two models is statistically significant ($p < 0.01$). ROC: receiver operating characteristic; Pes_2DCNN: 2DCNN based on Paw, flow-time and Pes; PF_2DCNN: 2DCNN solely based on Paw and flow; AUC: area under the curve; std. dev.: standard deviation.

The application of Initial_2DCNN on our data, after being modified by a combination of SMOTE and RUS, yielded the best performance for detection of premature cycling. As a result, we decided to use this approach for the comparison of Pes_2DCNN and PF_2DCNN in detecting premature cycling. **Table 7** shows the classification results of Pes_2DCNN and PF_2DCNN in detecting all types of breaths. The results for premature cycling detection are highlighted. The results indicate that, in terms of AUROC, the proposed Pes_2DCNN model outperformed PF_2DCNN in detecting premature cycling (0.88 ± 0.09 vs. 0.71 ± 0.24, respectively; p < 0.01).

Table 7 Performance comparison of Pes_2DCNN and PF_2DCNN in detecting all breath types. These results are achieved with the Initial_2DCNN model architecture using a combination of SMOTE and RUS as a solution for the class imbalance problem in the dataset. Performance metrics for detection of premature cycling are highlighted.

Model	Type of breath	Accuracy	Sensitivity	Specificity	F1 score	AUROC
Pes_2DCNN	Normal	0.73 ± 0.20	0.71 ± 0.26	0.77 ± 0.13	0.77 ± 0.19	0.85 ± 0.10
	RT	0.72 ± 0.18	0.50 ± 0.17	0.79 ± 0.23	0.34 ± 0.27	0.72 ± 0.20
	Premature cycling	0.98 ± 0.03	0.02 ± 0.04	1.00 ± 0.00	0.04 ± 0.07	0.88 ± 0.09
	Cough	0.93 ± 0.07	0.41 ± 0.24	0.95 ± 0.07	0.24 ± 0.15	0.83 ± 0.11
	Peristalsis	0.98 ± 0.03	0.50 ± 0.21	0.93 ± 0.07	0.30 ± 0.17	0.82 ± 0.08
	Other artefacts	0.91 ± 0.13	0.25 ± 0.18	0.93 ± 0.13	0.18 ± 0.17	0.66 ± 0.17
PF_2DCNN	Normal	0.71 ± 0.21	0.71 ± 0.28	0.55 ± 0.23	0.75 ± 0.22	0.70 ± 0.11
	RT	0.80 ± 0.14	0.27 ± 0.25	0.87 ± 0.17	0.20 ± 0.22	0.67 ± 0.17
	Premature cycling	0.97 ± 0.04	0.08 ± 0.09	0.99 ± 0.02	0.10 ± 0.10	0.71 ± 0.24
	Cough	0.94 ± 0.05	0.29 ± 0.19	0.96 ± 0.04	0.22 ± 0.13	0.79 ± 0.14
	Peristalsis	0.89 ± 0.12	0.09 ± 0.13	0.91 ± 0.12	0.04 ± 0.07	0.52 ± 0.07
	Other artefacts	0.90 ± 0.14	0.31 ± 0.29	0.91 ± 0.14	0.23 ± 0.26	0.63 ± 0.20

Pes_2DCNN: 2DCNN based on Paw, flow-time and Pes; PF_2DCNN: 2DCNN solely based on Paw and flow-time; RT: reverse triggering; AUROC: area under the receiver operating characteristic.

Figure 10 shows the ROC curves for Pes_2DCNN and PF_2DCNN with the corresponding AUC values for the detection of premature cycling. It is observed that removing the Pes signal compromised the model's performance for premature cycling detection. Furthermore, the results suggest that the 2DCNN based solely on ventilator pressure and flow-time signals is more sensitive to data subset selection and thus less stable than the 2DCNN based on the ventilator pressure, flow-time and Pes signals. ROC curves for the detection of premature cycling for all cross-validation folds are presented in **Figure S4** of **Supplementary materials 6**.

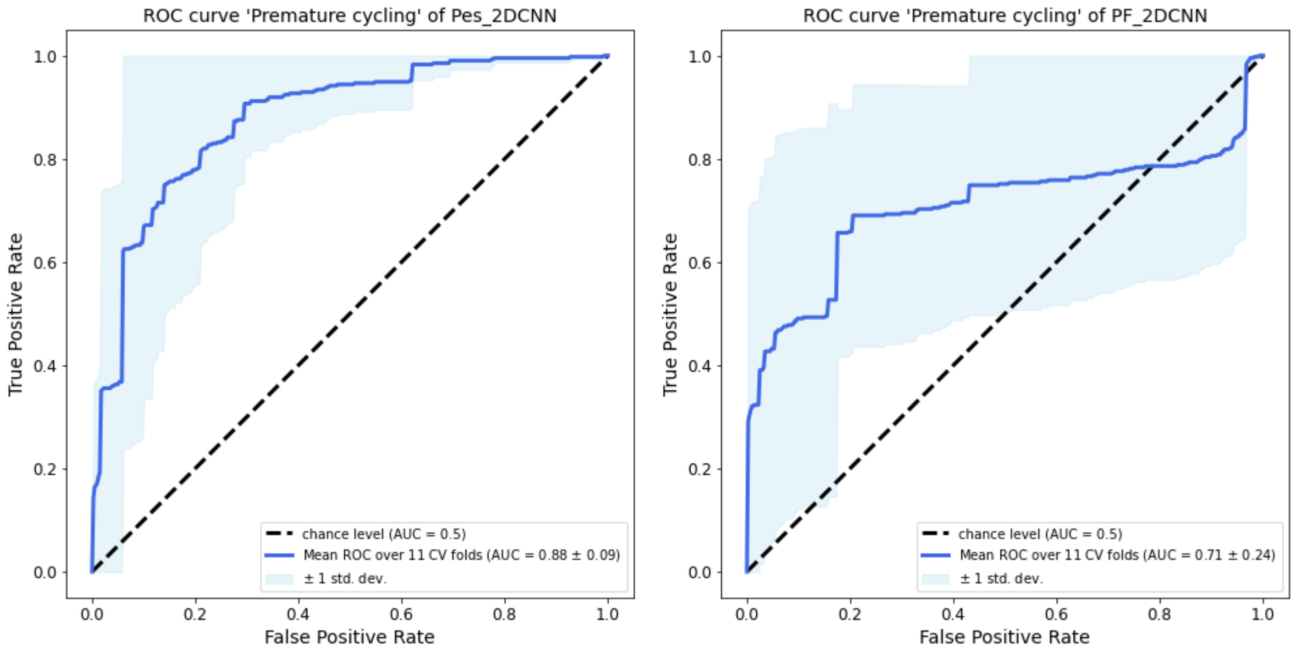


Figure 9 ROC curves for the detection of premature cycling for both Pes_2DCNN (left) and PF_2DCNN (right) using the Initial_2DCNN model on data modified by a combination of SMOTE and RUS. Difference in AUC between the two models is statistically significant (p < 0.01). ROC: receiver operating characteristic; Pes_2DCNN: 2DCNN based on Paw, flow-time and Pes; PF_2DCNN: 2DCNN solely based on Paw and flow-time; AUC: area under the curve; std. dev.: standard deviation.

5

Discussion

The aim of this study was to demonstrate the added value of using the Pes signal in PVA detection by developing a 2DCNN based on the ventilator's Paw, flow-time and Pes signals (Pes_2DCNN) that detects two types of PVA, namely reverse triggering and premature cycling. Several class imbalance solutions and model adjustments were considered to select the approach that results in the best performance for Pes_2DCNN. Subsequently, the performance of the final Pes_2DCNN was compared to that of a similar model based solely on the ventilator's Paw and flow-time signals (PF_2DCNN). For reverse triggering, the difference in performance between Pes_2DCNN and PF_2DCNN was trivial, but statistically significant (0.80 ± 0.07 vs. 0.75 ± 0.13, respectively; $p < 0.01$). For premature cycling, the difference in performance between Pes_2DCNN and PF_2DCNN was more prominent (0.88 ± 0.09 vs. 0.71 ± 0.24, respectively; $p < 0.01$). Despite the fact that this is a proof-of-concept study, these results suggest the importance of using the Pes signal in detection of reverse triggering and premature cycling.

Patient-ventilator asynchrony is the mismatch between the patient's respiratory demand and the ventilator's support. During spontaneous MV, patients interact with the ventilator and the ventilator responds according to a set of rules. PSV is a mode that is frequently used in ICU patients who are recovering from critical illness and are thus less sedated. As a result, some PVA is to be expected in these patients. Controlled ventilation modes, such as PCV, are preferred during the acute phase of the disease to improve oxygenation without causing lung damage, and to reduce the patient's work of breathing (42). Patients are usually deeply sedated, yet patient-ventilator interaction is still to be expected. Akoumianaki et al. (43) reported presence of RT in eight of eight patients with acute respiratory distress syndrome (ARDS) during deep sedation. The authors in (44) reported on the incidence of RT in a larger group of heavily sedated, mechanically ventilated patients with a broader range of admission diagnoses. They found that 44% of the patients had at least 10% of their breaths showing RT. This suggests that this is a very common phenomenon in deeply sedated, mechanically ventilated ICU patients.

As RT was only recently discovered, our understanding of its clinical meaning in mechanically ventilated patients is limited. Thus, few clinicians are familiar with it and trained to recognize it (20). An accurate automated detection algorithm will aid in proper understanding and management of RT. Rodriguez et al. (45) presented a rule-based framework for detecting RT using the ventilator's Paw and flow-time signals. The algorithm computes the time between the ventilator waveforms' local minima and maxima. Asynchrony is established if these durations exceed predefined thresholds. This approach, however, has some limitations. First, as the authors point out, this algorithm may misclassify RT that occurs during a long inspiratory pause. Furthermore, the algorithm was created and proven useful only in volume-controlled continuous mandatory ventilation (VC-CMV) with constant flow (square shaped flow-time signal). As a result, it cannot be used in other modes of ventilation such as PCV and PSV. Finally, the algorithm was developed and tested using a homogeneous patient population of only ARDS patients.

Mellado et al. (44) proposed a rule-based method for detecting RT as well. They developed an algorithm that consists of four simple criteria based on the ventilator's Paw and EAdi signals. The sensitivity and specificity were 0.84 and 0.93, respectively. However, they assessed EAdi with a catheter that filters the signal rather than presenting the raw signal. This may cause the onset of EAdi to be delayed. In addition, EAdi is only available in the neurally adjusted ventilation assist (NAVA) mode by Servo (Maquet, Sweden) (20).

One significant difference between our study and the previously discussed studies is that we use a machine learning approach to detect RT rather than a rule-based approach. Rule-based algorithms have a limited ability to handle complex patterns as they rely on predefined rules and criteria to detect PVA. They may be incapable of detecting more complex patterns or subtle changes in the patient's ventilator waveforms that may indicate RT. Furthermore, they are sensitive to noise and outliers in the data. Another significant difference is that we use the Pes signal in the detection of RT. Multiple experts in this field agree that the only way to reliably recognize RT is by using the esophageal pressure measurement to detect muscle contraction after the start of mechanical insufflation (9, 20, 46). This is consistent with our findings, which suggest the added value of the Pes signal in the detection of RT.

The results of this study also suggest the added value of using the Pes signal in the detection of premature cycling. Attempts to develop detection algorithms for this type of PVA have previously been made. Gholami et al. (8) presented a machine learning framework for detecting cycling asynchronies through the use of random forests. Pan et al. (7) proposed a 1DCNN for the detection of premature cycling and delayed cycling. Both algorithms, however, are based on the ventilator's Paw and flow-time signals. Premature cycling is typically recognized by an early reversal of flow accompanied by a decreased airway pressure in the expiratory phase. However, if you only consider the airway pressure and flow-time signal, an IEE event also fits this description. These ambiguity problems were not encountered in these studies because they proposed binary classification models. However, in order to develop multiclass classification algorithms for multiple types of PVA, the Pes signal must be included in order to accurately differentiate between premature cycling and IEE. Furthermore, when the proposed model in this study has access to the Pes signal in addition to the Paw and flow-time waveforms, it performs better in detecting premature cycling than when it only has access to the Paw and flow-time waveforms.

There are several limitations to this study that we wish to highlight. First, CNNs are required to be trained with a sufficient amount of data to learn the complex relationships between input features and output labels, and to improve generalization. As this is a single-center study, the generalizability of the proposed model can be questioned. Different types of patient care in other centers may affect the model's performance. Furthermore, we only included data from 11 patients. Therefore, the number of breaths used for model training may be insufficient to cover all possible shapes and aspects of asynchronous breaths encountered by patients. We attempted to mitigate this problem by selecting ROIs prior to annotation where PVA is more prevalent than it would be normally. Nonetheless, this emphasizes the fact that our dataset does not adequately represent the true prevalence of PVA in the ICU population, which also casts doubt on our model's generalizability.

Another limitation of this study is that we did not perform external validation of the model to assess the generalization ability of our model. This also reflects the preliminary nature of this investigation. However, we should note that the leave-one-patient-out cross-validation approach we employed to overcome overfitting is well established in the machine learning literature (47). Specifically, we made sure that a single patient's observations could not be mixed in both training and test sets to avoid bias introduced by intra-patient waveform similarities and, as a result, poor generalizability to subsequent patients.

Third, because CNNs are more commonly used to process 2D spatial data such as images (48), we decided to develop a 2DCNN. Therefore, we had to segment the ventilator waveforms into individual breaths that serve as 2D images for the CNN to use as input. However, in this way we did not take into account the temporal information in the ventilator waveforms. This means that the 2DCNN may not be able to detect important patterns or trends in the data that span across time. Further refinement is required to account for the temporal dynamics of ventilator waveforms in the detection of PVA.

A fourth limitation of this study is that, despite the selection of ROIs prior to annotation, only two types of PVA occurred with sufficient frequency in the study population to be included in model development. As a result, our model is not able to detect other clinically relevant asynchronies, such as double triggering and ineffective efforts during expiration (4, 25). Furthermore, flow asynchrony was not considered from the beginning of the study. This is because the only mandatory ventilation type that is used in the LUMC is pressure control, whereas flow asynchrony occurs more frequently in the volume control condition. Future research should include additional medical centers to increase the diversity of the dataset and, as a result, extend the model's detection capability to all possible types of PVA.

Finally, the annotated dataset was established by independent clinicians with access to the esophageal pressure as a reference signal of patient effort. Due to time constraints, each patient's data was annotated by only one clinician. Despite the use of an annotation protocol and the clinicians in this study having extensive experience in the interpretation of ventilator waveforms, errors in the annotated dataset may still exist. One possible explanation is that the annotation process did not provide a complete picture of the patients. Providing clinicians with not only the ventilator waveforms, but also the ventilator settings and measurements, patient demographics, and vital signs, could lead to a more comprehensive understanding of the problematic cycles during annotation. Nonetheless, future research should include assessment of inter-rater agreement. If the inter-rater agreement appears to be poor, more reliable annotation could be obtained by involving multiple clinicians in the annotation of every single patient's data and accepting only breaths that are labeled identically by the majority of the clinicians.

There are numerous important and unanswered questions regarding PVA and its impact on clinical outcomes. While the association between PVA and poor patient outcome has been recognized, causality has yet to be proven (1, 3, 4, 18). It is possible that PVA simply reflects more severe lung injury, and that the underlying lung injury, rather than the PVA, is the cause of poorer outcomes. Once a causal relationship is proven, the question rises which underlying mechanisms contribute to this relationship. It could be the result of increased sedation in response to the clinician's detection of PVA, the result of respiratory muscle fatigue due to excess work of breathing, or the result of excessive tidal volumes due to double triggering or reverse triggering.

It is also unknown whether the relationship between PVA and poor outcomes applies to all types of PVA. More research in ICU populations is needed to answer these questions, with the aim to ultimately better understand the concept of PVA. To conduct this research, accurate automated detection algorithms for all types of PVA must be developed. This type of detection algorithms is typically trained on "ground truth" labels generated by human experts. However, there is no widely accepted, formal, systematic method for reading ventilator waveforms, as there is for ECGs. Most healthcare professionals and researchers in this field have relied on self-study, experience, and learning from mentors (17). Nonetheless, visual inspection of ventilator waveforms by experts is frequently used as the gold standard in the validation of detection algorithms for PVA (8, 46, 49-51). The questions remains how this can be considered the gold standard when every expert employs a different taxonomy. Agreement on the definitions for various types of PVA is the foundation of a systematic method for detecting asynchronies. Therefore, establishing these definitions with a reference signal for patient effort is crucial for development of these algorithms. The development of an automated detection algorithm based on a reference signal of patient effort, as demonstrated in this study, is a promising approach for accurately detecting PVA and gaining a better understanding of this phenomenon.

6

Conclusion

In conclusion, a 2DCNN was developed for the detection of two types of PVA, reverse triggering and premature cycling, based on the ventilator’s Paw, flow-time and Pes signals. Additionally, the performance of this detection algorithm was compared to that of a similar model based solely on the ventilator’s Paw and flow-time signals. The Pes-based 2DCNN showed better performance in detecting RT and premature cycling as compared to the Paw and flow-based model. However, because this is a preliminary study, more research is required to further investigate the added value of the Pes signal in PVA detection.

References

1. Chao DC, Scheinhorn DJ, Stearn-Hassenpflug M. Patient-ventilator trigger asynchrony in prolonged mechanical ventilation. Chest. 1997;112(6):1592-9.
2. de Wit M, Miller KB, Green DA, Ostman HE, Gennings C, Epstein SK. Ineffective triggering predicts increased duration of mechanical ventilation. Crit Care Med. 2009;37(10):2740-5.
3. Tobin MJ, Jubran A, Laghi F. Patient-ventilator interaction. Am J Respir Crit Care Med. 2001;163(5):1059-63.
4. Blanch L, Villagra A, Sales B, Montanya J, Lucangelo U, Luján M, et al. Asynchronies during mechanical ventilation are associated with mortality. Intensive Care Med. 2015;41(4):633-41.
5. 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.
6. Ihaddouchen I, van der Velde F, Tax DMJ, Schoe A. Automated detection of patient-ventilator asynchrony: A systematic review. 2022.
7. Pan Q, Zhang L, Jia M, Pan J, Gong Q, Lu Y, et al. An interpretable 1D convolutional neural network for detecting patient-ventilator asynchrony in mechanical ventilation. Comput Methods Programs Biomed. 2021;204:106057.
8. Gholami B, Phan TS, Haddad WM, Cason A, Mullis J, Price L, et al. Replicating human expertise of mechanical ventilation waveform analysis in detecting patient-ventilator cycling asynchrony using machine learning. Comput Biol Med. 2018;97:137-44.
9. Blanch L, Sales B, Montanya J, Lucangelo U, Garcia-Esquirol O, Villagra A, et al. Validation of the Better Care® system to detect ineffective efforts during expiration in mechanically ventilated patients: a pilot study. Intensive Care Med. 2012;38(5):772-80.
10. Phan TS, Costa R, Haddad WM, Mullis JC, Price LT, Cason AD, et al. Validation of an automated system for detecting ineffective triggering asynchronies during mechanical ventilation: a retrospective study. J Clin Monit Comput. 2020;34(6):1233-7.
11. Mojoli F, Pozzi M, Orlando A, Bianchi IM, Arisi E, Iotti GA, et al. Timing of inspiratory muscle activity detected from airway pressure and flow during pressure support ventilation: the waveform method. Critical Care. 2022;26(1):32.
12. Gutierrez G. Artificial Intelligence in the Intensive Care Unit. Crit Care. 2020;24(1):101.
13. Sidey-Gibbons JAM, Sidey-Gibbons CJ. Machine learning in medicine: a practical introduction. BMC Med Res Methodol. 2019;19(1):64.
14. 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.
15. Currie G, Hawk KE, Rohren E, Vial A, Klein R. Machine Learning and Deep Learning in Medical Imaging: Intelligent Imaging. J Med Imaging Radiat Sci. 2019;50(4):477-87.
16. Yamashita R, Nishio M, Do RKG, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Imaging. 2018;9(4):611-29.
17. Mireles-Cabodevila E, Siuba MT, Chatburn RL. A Taxonomy for Patient-Ventilator Interactions and a Method to Read Ventilator Waveforms. Respir Care. 2021.
18. de Wit M. Monitoring of patient-ventilator interaction at the bedside. Respir Care. 2011;56(1):61-72.
19. Chatburn RL, Mireles-Cabodevila E. 2019 Year in Review: Patient-Ventilator Synchrony. Respir Care. 2020;65(4):558-72.
20. Esperanza JA, Sarlabous L, de Haro C, Magrans R, Lopez-Aguilar J, Blanch L. Monitoring Asynchrony During Invasive Mechanical Ventilation. Respir Care. 2020;65(6):847-69.
21. Vaporidi K, Akoumianaki E, Telias I, Goligher EC, Brochard L, Georgopoulos D. Respiratory Drive in Critically Ill Patients. Pathophysiology and Clinical Implications. Am J Respir Crit Care Med. 2020;201(1):20-32.
22. Pham T, Telias I, Piraino T, Yoshida T, Brochard LJ. Asynchrony Consequences and Management. Crit Care Clin. 2018;34(3):325-41.
23. Beitler JR, Sands SA, Loring SH, Owens RL, Malhotra A, Spragg RG, et al. Quantifying unintended exposure to high tidal volumes from breath stacking dyssynchrony in ARDS: the BREATHE criteria. Intensive Care Med. 2016;42(9):1427-36.
24. Tassaux D, Gainnier M, Battisti A, Jolliet P. Impact of expiratory trigger setting on delayed cycling and inspiratory muscle workload. Am J Respir Crit Care Med. 2005;172(10):1283-9.

25. Thille AW, Rodriguez P, Cabello B, Lellouche F, Brochard L. Patient-ventilator asynchrony during assisted mechanical ventilation. Intensive Care Med. 2006;32(10):1515-22.

26. Gilstrap D, Davies J. Patient-Ventilator Interactions. Clin Chest Med. 2016;37(4):669-81.

27. Arnal J-M. Monitoring Mechanical Ventilation Using Ventilator Waveforms2018.

28. Mireles-Cabodevila E, Chatburn RL. Work of breathing in adaptive pressure control continuous mandatory ventilation. Respir Care. 2009;54(11):1467-72.

29. Wang S, Minku LL, Yao X. A Systematic Study of Online Class Imbalance Learning With Concept Drift. IEEE Trans Neural Netw Learn Syst. 2018;29(10):4802-21.

30. Japkowicz N, editor The Class Imbalance Problem: Significance and Strategies2000.

31. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. J Artif Int Res. 2002;16(1):321–57.

32. Prusa J, Khoshgoftaar TM, Dittman DJ, Napolitano A, editors. Using Random Undersampling to Alleviate Class Imbalance on Tweet Sentiment Data. 2015 IEEE International Conference on Information Reuse and Integration; 2015 13-15 Aug. 2015.

33. Liu S, Huang D, Wang Y. Receptive Field Block Net for Accurate and Fast Object Detection. arXiv. 2017.

34. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv. 2015.

35. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014;15:1929-58.

36. Hinton G, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov R. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint. 2012;arXiv.

37. Duan K, Keerthi SS, Chu W, Shevade SK, Poo AN. Multi-category classification by soft-max combination of binary classifiers. 2003.

38. Imhoff M, Kuhls S. Alarm algorithms in critical care monitoring. Anesth Analg. 2006;102(5):1525-37.

39. DeLong ER, DeLong DM, Clarke-Pearson DL. Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. Biometrics. 1988;44(3):837-45.

40. Sun X, Xu W. Fast Implementation of DeLong's Algorithm for Comparing the Areas Under Correlated Receiver Operating Characteristic Curves. IEEE Signal Processing Letters. 2014;21(11):1389-93.

41. de Figueiredo M, Cordella CBY, Jouan-Rimbaud Bouveresse D, Archer X, Bégué J-M, Rutledge DN. A variable selection method for multiclass classification problems using two-class ROC analysis. Chemometrics and Intelligent Laboratory Systems. 2018;177:35-46.

42. Rodriguez P, Dojat M, Brochard L. Mechanical ventilation: Changing concepts. Indian Journal of Critical Care Medicine. 2005;9.

43. Akoumianaki E, Lyazidi A, Rey N, Matamis D, Perez-Martinez N, Giraud R, et al. Mechanical ventilation-induced reverse-triggered breaths: a frequently unrecognized form of neuromechanical coupling. Chest. 2013;143(4):927-38.

44. Mellado Artigas R, Damiani LF, Piraino T, Pham T, Chen L, Rauseo M, et al. Reverse Triggering Dyssynchrony 24 h after Initiation of Mechanical Ventilation. Anesthesiology. 2021;134(5):760-9.

45. Rodriguez PO, Tiribelli N, Gogniat E, Plotnikow GA, Fredes S, Fernandez Ceballos I, et al. Automatic detection of reverse-triggering related asynchronies during mechanical ventilation in ARDS patients using flow and pressure signals. J Clin Monit Comput. 2020;34(6):1239-46.

46. 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.

47. James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning. New York: Springer; 2013.

48. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data. 2021;8(1):53.

49. Pan Q, Jia M, Liu Q, Zhang L, Pan J, Lu F, et al. Identifying Patient-Ventilator Asynchrony on a Small Dataset Using Image-Based Transfer Learning. Sensors (Basel). 2021;21(12).

50. Sottile PD, Albers D, Higgins C, McKeehan J, Moss MM. The Association Between Ventilator Dyssynchrony, Delivered Tidal Volume, and Sedation Using a Novel Automated Ventilator Dyssynchrony Detection Algorithm. Crit Care Med. 2018;46(2):e151-e7.

51. Rehm GB, Han J, Kuhn BT, Delplanque JP, Anderson NR, Adams JY, et al. Creation of a Robust and Generalizable Machine Learning Classifier for Patient Ventilator Asynchrony. Methods Inf Med. 2018;57(4):208-19.

1. Typical CNN and its training process

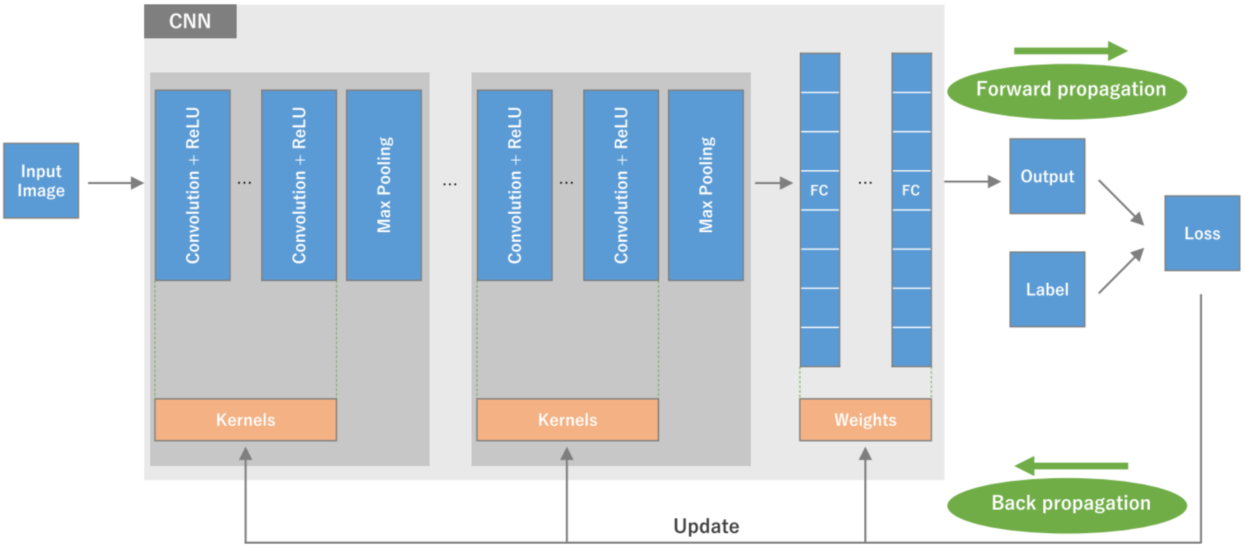


Figure S1 A visual representation of a typical convolutional neural network (CNN) architecture and its training process (16). A CNN consists of different layers: convolutional layers, pooling layers (e.g., max pooling), and fully connected (FC) layers. An input image is passed through these layers during the training process. Through forward propagation, the loss of the model under specific kernels and weights is computed by comparing the output of the model with the ground truth label. Subsequently, the model's learnable parameters, i.e., kernels and weights, are adjusted via backpropagation with the aim of minimizing the loss function.

2. Patient inclusion

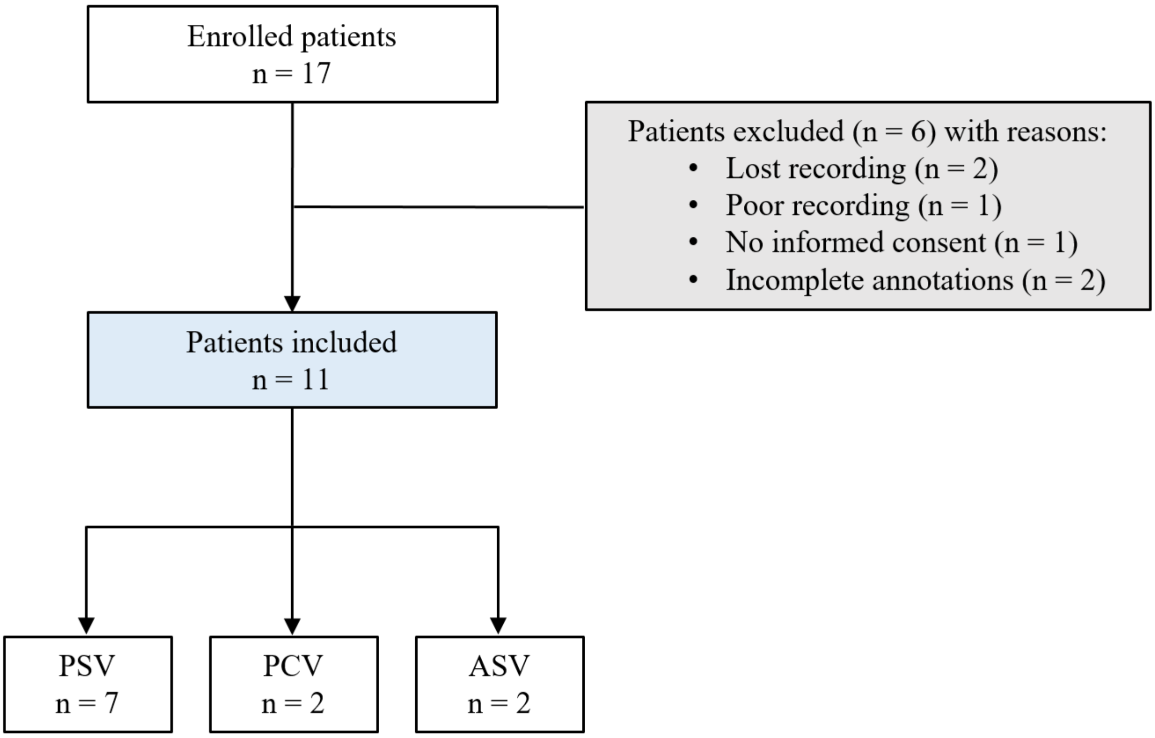


Figure S2 Flowchart of patient inclusion.

3. 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 (28)
2. 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 (28)
   C. PAW_{ON} of <u>second</u> breath occurs before PES_{OFF} of <u>first</u> effort
3. Ineffective effort during expiration:
   A. Mandatory or assisted breath
   B. PES_{ON} occurs after PAW_{OFF} (i.e., during ventilator expiration)
4. 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} (28)
5. Auto triggering:
   A. Mandatory breath: PAW_{ON} does not start with a negative deflection
   B. Absence of negative PES signal
6. 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 (28)
   C. PAW_{OFF} occurs < 100 ms before PES_{OFF} (28)
7. 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 (28)
   C. PAW_{OFF} occurs > 100 ms after PES_{OFF} (28)
8. Cough:
   A. Sharp inhalation and exhalation spikes in the flow-time 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-time signal
10. Other artefacts: anything that is not a normal breath and does not meet the criteria outlined above

4. Model architecture of alternative models

Table S1 Layer details and parameters used in the 2DCNN with adjusted kernel size of 50 x 3 in the first 2D convolutional layer and padding set to “valid” in all convolutional layers (Conv_2DCNN).

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	251 x 1 x 16	50 x 3	16	1	2416
2	Batch Normalization	-	-	251 x 1 x 16	-	-	-	1004
3	2D Max Pooling	-	-	126 x 1 x 16	2 x 2	-	2	0
4	2D Convolution	2	ReLU	108 x 1 x 32	10 x 1	32	1	5152
5	Batch Normalization	-	-	108 x 1 x 32	-	-	-	432
6	2D Max Pooling	-	-	54 x 1 x 32	2 x 2	-	2	0
7	2D Convolution	2	ReLU	46 x 1 x 64	5 x 1	64	1	10304
8	Batch Normalization	-	-	46 x 1 x 64	-	-	-	184
9	2D Max Pooling	-	-	23 x 1 x 64	2 x 2	-	2	0
10	2D Convolution	3	ReLU	17 x 1 x 32	3 x 1	32	1	6176
11	Batch Normalization	-	-	17 x 1 x 32	-	-	-	68
12	2D Max Pooling	-	-	9 x 1 x 32	2 x 2	-	2	0
13	Fully connected	-	ReLU	256	-	-	-	73984
14	Fully connected	-	Softmax	6	-	-	-	1542

ReLU: Rectified linear unit

Table S2 Layer details and parameters used in the 2DCNN with adjusted pool size of 2 x 1 in all 2D Max Pooling layers (Pool_2DCNN).

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 3 x 16	2 x 1	-	2	0
4	2D Convolution	2	ReLU	150 x 3 x 32	10 x 1	32	1	5152
5	Batch Normalization	-	-	150 x 3 x 32	-	-	-	600
6	2D Max Pooling	-	-	75 x 3 x 32	2 x 1	-	2	0
7	2D Convolution	2	ReLU	75 x 3 x 64	5 x 1	64	1	10304
8	Batch Normalization	-	-	75 x 3 x 64	-	-	-	300
9	2D Max Pooling	-	-	38 x 3 x 64	2 x 1	-	2	0
10	2D Convolution	3	ReLU	38 x 3 x 32	3 x 1	32	1	6176
11	Batch Normalization	-	-	38 x 3 x 32	-	-	-	152
12	2D Max Pooling	-	-	19 x 3 x 32	2 x 1	-	2	0
13	Fully connected	-	ReLU	256	-	-	-	467200
14	Fully connected	-	Softmax	6	-	-	-	1542

ReLU: Rectified linear unit

5. Complete performance results of the investigated models

Table S3 Performance of Initial_2DCNN for all breath types.

	Type of breath	Accuracy	Sensitivity	Specificity	F1 score	AUROC
Class imbalance	Normal	0.82 ± 0.19	0.85 ± 0.24	0.60 ± 0.17	0.85 ± 0.21	0.86 ± 0.13
	RT	0.76 ± 0.21	0.44 ± 0.12	0.83 ± 0.24	0.32 ± 0.25	0.78 ± 0.12
	Premature cycling	0.97 ± 0.03	0.00 ± 0.00	0.99 ± 0.01	0.00 ± 0.00	0.84 ± 0.18
	Cough	0.96 ± 0.03	0.22 ± 0.14	0.98 ± 0.02	0.22 ± 0.17	0.86 ± 0.10
	Peristalsis	0.93 ± 0.07	0.42 ± 0.13	0.95 ± 0.06	0.35 ± 0.20	0.85 ± 0.13
	Other artefacts	0.97 ± 0.04	0.14 ± 0.14	0.99 ± 0.00	0.14 ± 0.13	0.87 ± 0.09
SMOTE	Normal	0.70 ± 0.19	0.70 ± 0.25	0.76 ± 0.12	0.76 ± 0.19	0.83 ± 0.11
	RT	0.77 ± 0.13	0.49 ± 0.16	0.83 ± 0.17	0.33 ± 0.30	0.73 ± 0.15
	Premature cycling	0.98 ± 0.03	0.06 ± 0.11	1.00 ± 0.00	0.09 ± 0.16	0.87 ± 0.10
	Cough	0.90 ± 0.10	0.32 ± 0.20	0.92 ± 0.10	0.21 ± 0.15	0.76 ± 0.11
	Peristalsis	0.89 ± 0.16	0.45 ± 0.22	0.91 ± 0.16	0.28 ± 0.17	0.76 ± 0.13
	Other artefacts	0.91 ± 0.12	0.21 ± 0.18	0.93 ± 0.12	0.19 ± 0.20	0.66 ± 0.19
RUS	Normal	0.63 ± 0.20	0.61 ± 0.26	0.85 ± 0.15	0.69 ± 0.23	0.86 ± 0.09
	RT	0.81 ± 0.09	0.48 ± 0.16	0.87 ± 0.09	0.34 ± 0.27	0.73 ± 0.19
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.64 ± 0.35
	Cough	0.82 ± 0.18	0.41 ± 0.23	0.84 ± 0.19	0.19 ± 0.17	0.77 ± 0.23
	Peristalsis	0.85 ± 0.12	0.55 ± 0.16	0.87 ± 0.13	0.29 ± 0.23	0.82 ± 0.11
	Other artefacts	0.94 ± 0.08	0.30 ± 0.27	0.96 ± 0.08	0.20 ± 0.17	0.72 ± 0.21
SMOTE + RUS	Normal	0.73 ± 0.20	0.71 ± 0.26	0.77 ± 0.13	0.77 ± 0.19	0.85 ± 0.10
	RT	0.72 ± 0.18	0.50 ± 0.17	0.79 ± 0.23	0.34 ± 0.27	0.72 ± 0.20
	Premature cycling	0.98 ± 0.03	0.02 ± 0.04	1.00 ± 0.00	0.04 ± 0.07	0.88 ± 0.09
	Cough	0.93 ± 0.07	0.41 ± 0.24	0.95 ± 0.07	0.24 ± 0.15	0.83 ± 0.11
	Peristalsis	0.98 ± 0.03	0.50 ± 0.21	0.93 ± 0.07	0.30 ± 0.17	0.82 ± 0.08
	Other artefacts	0.91 ±0.13	0.25 ± 0.18	0.93 ± 0.13	0.18 ± 0.17	0.66 ± 0.17

SMOTE: synthetic minority over-sampling technique; RUS: random undersampling technique; AUROC: area under the receiver operating characteristic; RT: reverse triggering.

Table S4 Performance of Conv_2DCNN for all breath types.

	Type of breath	Accuracy	Sensitivity	Specificity	F1 score	AUROC
Class imbalance	Normal	0.80 ± 0.21	0.85 ± 0.24	0.56 ± 0.12	0.84 ± 0.21	0.83 ± 0.18
	RT	0.85 ± 0.07	0.46 ± 0.14	0.93 ± 0.05	0.34 ± 0.27	0.80 ± 0.07
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.65 ± 0.19
	Cough	0.95 ± 0.04	0.21 ± 0.14	0.98 ± 0.03	0.20 ± 0.15	0.86 ± 0.10
	Peristalsis	0.89 ± 0.20	0.40 ± 0.18	0.91 ± 0.20	0.32 ± 0.19	0.82 ± 0.16
	Other artefacts	0.96 ± 0.05	0.17 ± 0.28	0.98 ± 0.03	0.17 ± 0.29	0.80 ± 0.15
SMOTE	Normal	0.66 ± 0.15	0.63 ± 0.20	0.80 ± 0.09	0.73 ± 0.15	0.81 ± 0.11
	RT	0.81 ± 0.10	0.56 ± 0.19	0.85 ± 0.11	0.39 ± 0.33	0.77 ± 0.12
	Premature cycling	0.97 ± 0.04	0.00 ± 0.00	0.99 ± 0.01	0.00 ± 0.00	0.61 ± 0.06
	Cough	0.94 ± 0.05	0.33 ± 0.18	0.96 ± 0.04	0.25 ± 0.19	0.66 ± 0.19
	Peristalsis	0.84 ± 0.14	0.45 ± 0.16	0.85 ± 0.15	0.25 ± 0.17	0.74 ± 0.13
	Other artefacts	0.87 ± 0.13	0.37 ± 0.28	0.89 ± 0.13	0.18 ± 0.22	0.67 ± 0.13
RUS	Normal	0.71 ± 0.21	0.69 ± 0.29	0.82 ± 0.15	0.74 ± 0.26	0.84 ± 0.13
	RT	0.75 ± 0.19	0.45 ± 0.27	0.83 ± 0.24	0.28 ± 0.25	0.68 ± 0.18
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.01	0.00 ± 0.00	0.64 ± 0.24
	Cough	0.93 ± 0.05	0.33 ± 0.19	0.95 ± 0.04	0.19 ± 0.12	0.84 ± 0.10
	Peristalsis	0.89 ± 0.15	0.55 ± 0.16	0.90 ± 0.15	0.35 ± 0.19	0.82 ± 0.14
	Other artefacts	0.89 ± 0.13	0.31 ± 0.22	0.91 ± 0.14	0.17 ± 0.24	0.75 ± 0.14
SMOTE + RUS	Normal	0.62 ± 0.24	0.61 ± 0.29	0.82 ± 0.09	0.68 ± 0.24	0.81 ± 0.10
	RT	0.78 ± 0.15	0.53 ± 0.21	0.82 ± 0.19	0.37 ± 0.31	0.74 ± 0.19
	Premature cycling	0.97 ± 0.03	0.16 ± 0.20	0.99 ± 0.01	0.19 ± 0.27	0.77 ± 0.13
	Cough	0.94 ± 0.04	0.37 ± 0.21	0.95 ± 0.03	0.24 ± 0.18	0.73 ± 0.21
	Peristalsis	0.89 ± 0.08	0.49 ± 0.27	0.91 ± 0.08	0.22 ± 0.20	0.77 ± 0.14
	Other artefacts	0.81 ± 0.22	0.20 ± 0.13	0.83 ± 0.23	0.08 ± 0.09	0.55 ± 0.19

Table S5 Performance of Pool_2DCNN for all breath types.

	Type of breath	Accuracy	Sensitivity	Specificity	F1 score	AUROC
Class imbalance	Normal	0.80 ± 0.19	0.86 ± 0.25	0.54 ± 0.18	0.84 ± 0.21	0.85 ± 0.16
	RT	0.73 ± 0.20	0.36 ± 0.06	0.83 ± 0.23	0.24 ± 0.22	0.65 ± 0.22
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.73 ± 0.13
	Cough	0.95 ± 0.04	0.36 ± 0.27	0.97 ± 0.04	0.23 ± 0.12	0.91 ± 0.07
	Peristalsis	0.96 ± 0.03	0.39 ± 0.23	0.98 ± 0.02	0.37 ± 0.18	0.87 ± 0.09
	Other artefacts	0.94 ± 0.09	0.21 ± 0.27	0.96 ± 0.09	0.20 ± 0.27	0.80 ± 0.16
SMOTE	Normal	0.65 ± 0.23	0.66 ± 0.27	0.75 ± 0.15	0.72 ± 0.21	0.80 ± 0.13
	RT	0.72 ± 0.18	0.37 ± 0.25	0.79 ± 0.23	0.29 ± 0.28	0.63 ± 0.23
	Premature cycling	0.97 ± 0.03	0.09 ± 0.10	0.99 ± 0.01	0.09 ± 0.13	0.63 ± 0.17
	Cough	0.94 ± 0.03	0.37 ± 0.26	0.97 ± 0.03	0.21 ± 0.16	0.86 ± 0.08
	Peristalsis	0.91 ± 0.08	0.36 ± 0.20	0.92 ± 0.08	0.22 ± 0.18	0.78 ± 0.12
	Other artefacts	0.85 ± 0.19	0.31 ± 0.29	0.87 ± 0.20	0.16 ± 0.24	0.67 ± 0.17
RUS	Normal	0.67 ± 0.24	0.62 ± 0.32	0.81 ± 0.15	0.68 ± 0.31	0.79 ± 0.16
	RT	0.68 ± 0.23	0.53 ± 0.19	0.75 ± 0.30	0.28 ± 0.25	0.72 ± 0.20
	Premature cycling	0.97 ± 0.03	0.03 ± 0.05	1.00 ± 0.01	0.02 ± 0.03	0.81 ± 0.21
	Cough	0.94 ± 0.06	0.25 ± 0.16	0.96 ± 0.07	0.21 ± 0.18	0.87 ± 0.06
	Peristalsis	0.89 ± 0.13	0.55 ± 0.26	0.91 ± 0.14	0.31 ± 0.19	0.83 ± 0.09
	Other artefacts	0.86 ± 0.20	0.33 ± 0.30	0.88 ± 0.20	0.15 ± 0.14	0.72 ± 0.22
SMOTE + RUS	Normal	0.74 ± 0.20	0.75 ± 0.27	0.77 ± 0.14	0.78 ± 0.24	0.81 ± 0.15
	RT	0.75 ± 0.22	0.51 ± 0.23	0.81 ± 0.28	0.31 ± 0.26	0.69 ± 0.18
	Premature cycling	0.98 ± 0.03	0.00 ± 0.00	1.00 ± 0.00	0.00 ± 0.00	0.70 ± 0.17
	Cough	0.95 ± 0.03	0.37 ± 0.25	0.97 ± 0.01	0.21 ± 0.13	0.80 ± 0.15
	Peristalsis	0.95 ± 0.03	0.44 ± 0.23	0.97 ± 0.03	0.34 ± 0.19	0.81 ± 0.08
	Other artefacts	0.86 ± 0.18	0.34 ± 0.27	0.88 ± 0.19	0.17 ± 0.19	0.70 ± 0.16

SMOTE: synthetic minority over-sampling technique; RUS: random undersampling technique; AUROC: area under the receiver operating characteristic; RT: reverse triggering.

6. ROC curves for all cross-validation folds

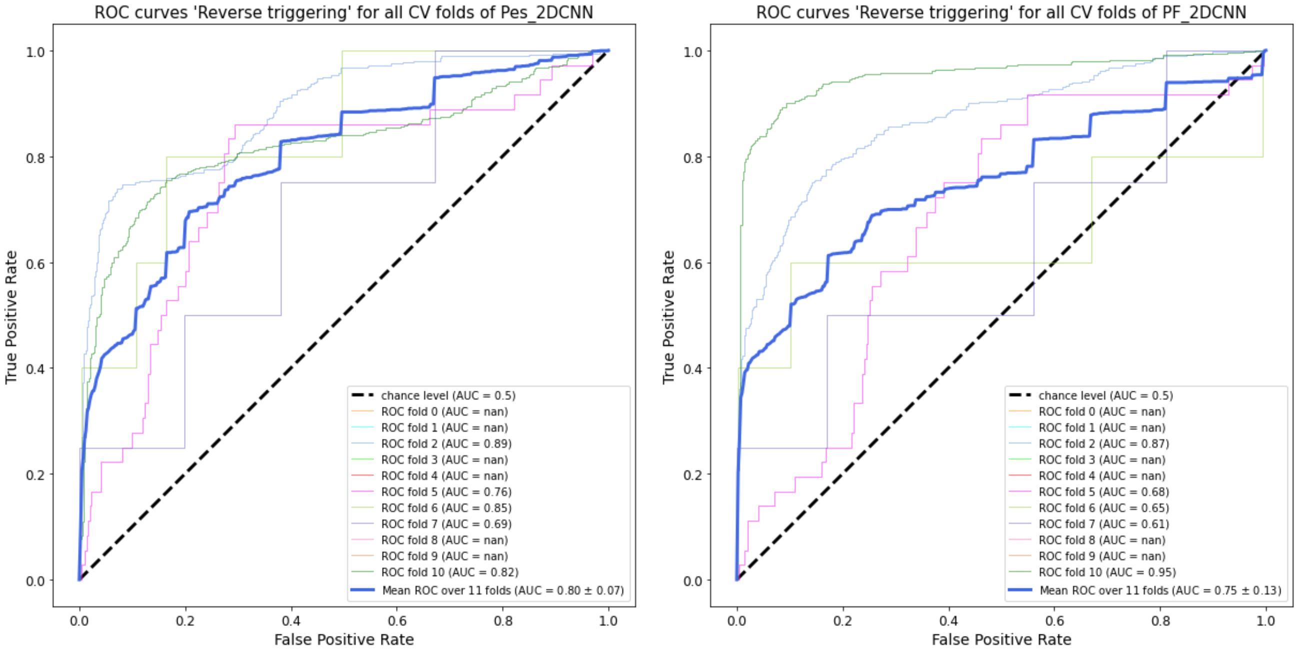


Figure S3 ROC curves for the detection of reverse triggering of both Pes_2DCNN (left) and PF_2DCNN (right) for all cross-validation folds and mean ROC over 11 folds. Note that folds with an AUC of nan indicate that the test patient in that fold did not show any reverse triggering breaths. ROC: receiver operating characteristic; Pes_2DCNN: 2DCNN based on Paw, flow-time and Pes; PF_2DCNN: 2DCNN solely based on Paw and flow; CV: cross-validation; AUC: area under the curve.

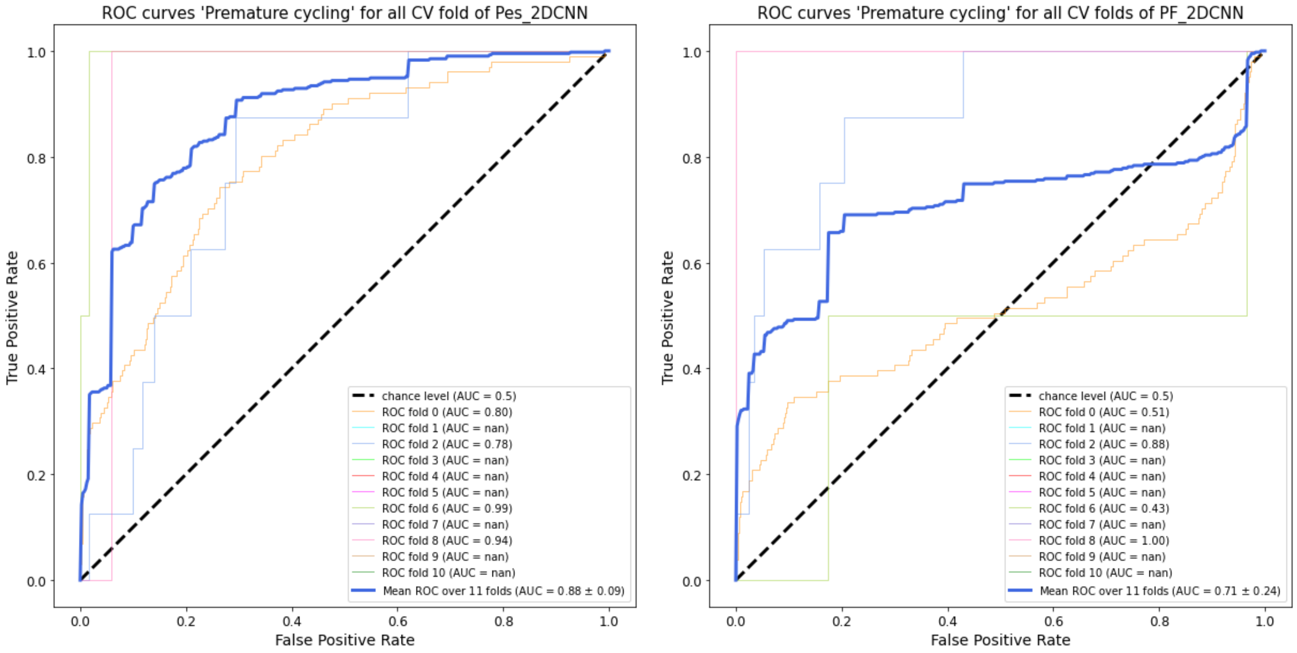


Figure S4 ROC curves for the detection of premature cycling of both Pes_2DCNN (left) and PF_2DCNN (right) for all cross-validation folds and mean ROC over 11 folds. Note that folds with an AUC of nan indicate that the test patient in that fold did not show any premature cycling breaths. ROC: receiver operating characteristic; Pes_2DCNN: 2DCNN based on Paw, flow-time and Pes; PF_2DCNN: 2DCNN solely based on Paw and flow; CV: cross-validation; AUC: area under the curve.