# Time-Series Out-of-Distribution Data Detection in Mechanical Ventilation

van de Kamp, L.; Hunnekens, B.; Oomen, T.; van de Wouw, N.

**DOI**
10.1109/OJCSYS.2025.3585427

**Publication date**
2025

**Document Version**
Final published version

**Published in**
IEEE Open Journal of Control Systems

**Citation (APA)**
van de Kamp, L., Hunnekens, B., Oomen, T., & van de Wouw, N. (2025). Time-Series Out-of-Distribution Data Detection in Mechanical Ventilation. *IEEE Open Journal of Control Systems*, *4*, 236-249. https://doi.org/10.1109/OJCSYS.2025.3585427

IEEE Open Journal of
Control Systems

Received 28 February 2025; revised 18 June 2025; accepted 23 June 2025. Date of publication 2 July 2025;
date of current version 29 August 2025. Recommended by Senior Editor S. Bhasin.

*Digital Object Identifier 10.1109/OJCSYS.2025.3585427*

# Time-Series Out-of-Distribution Data Detection in Mechanical Ventilation

L. VAN DE KAMP [1,2], B. HUNNEKENS [1] (Member, IEEE), T. OOMEN [2,3] (Senior Member, IEEE), AND N. VAN DE WOUW [2] (Fellow, IEEE)

[1] Demcon Life Sciences and Health Eindhoven, 5683 CR Best, The Netherlands
[2] Department of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands
[3] Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands

CORRESPONDING AUTHOR: L. VAN DE KAMP (e-mail: l.g.j.v.d.kamp@tue.nl)

This work involved human subjects or animals in its research. Approval of all ethical and experimental procedures and protocols was granted by Ethics Committee of Fondazione IRCCS Policlinico San Matteo under Application No. 41223, and written informed consent to participation was obtained from all patients. The approval of the ethical and experimental procedures and protocols of the Maasstad data was granted as a non-WMO study by the local Ethics Committee (Maasstad Hospital and Medical Research Ethics Committees United (MEC-U)).

**ABSTRACT** Safe deployment of neural networks to classify time series in safety-critical applications relies on the ability of the classifier to detect data that does not originate from the same distribution as the training data. The aim of this paper is to propose a framework for detecting whether time-series data is sampled from a different distribution than the training data, known as the problem of *out-of-distribution* (OOD) detection. We propose a novel distance-based OOD method for time-series data using a hierarchical clustering method together with dynamic time-warping to measure the difference between a new data instance and the training set. The method is evaluated in the context of mechanical ventilation, a safety critical application, using both simulated and clinical datasets. Results of the mechanical ventilation use case demonstrate that the proposed approach effectively detects out-of-distribution data and improves classification performance in diverse settings.

**INDEX TERMS** Mechanical ventilation, out-of-distribution detection, safety in machine learning, time-series analysis.

## I. INTRODUCTION

In recent years, many real-world applications have incorporated neural networks to classify time-series data and thereby improve their performance. In safety-critical applications, such as mechanical ventilation in healthcare, it is important that these neural networks are robust, quantify uncertainty, and detect data that is not only from a different realization, but is from a different distribution, i.e., a stocastic process, than the training data, also called *out-of-distribution* (OOD) data, see Fig. 1. If this detection is accurately performed, it can be ensured that the neural network is only deployed on data for which it can reliable perform classification. Without OOD detection, neural networks tend to predict these out-of-distribution data wrongly with high-confidence, which might lead to catastrophic consequences [1]. At the same time, out-of-distribution data is common in applications where there is a limited training set, which is often the case for time-series data. Therefore, it is crucial to detect the OOD data, ignore the output of the classification algorithm and give control over classification back to the user in this situation. OOD detection is conceptually related to fault detection. In

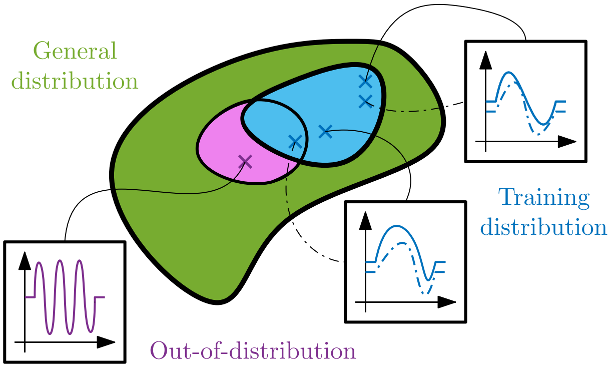

**FIGURE 1.** Illustration of out-of-distribution data for time-series data.

fault-detection of dynamical systems, the goal is to detect faulty behavior from healthy system behavior or to diagnose different faulty behaviors. Furthermore, fault detection is a crucial part of fault-tolerant control design.

Recently, substantial research has been conducted on the topic of OOD detection in the image domain [2], [3], [4]. However, OOD detection in time series pose different challenges due to their unique characteristics: time series contain temporal relations while images contain spatial relations; values of a time series are continuous while pixel values in images are from the categorical set $\{0, 1, \ldots, 255\}$. Therefore, image OOD detection techniques, including the approaches in [5], are not directly applicable for OOD detection of time series.

In this paper, we focus on OOD detection for time-series classification networks. The decision on OOD must be taken solely based on the raw time-series data, i.e., on the input of the classification network. This is different from prior works as we do not rely on labels or high-level features extracted by the classification network. Consequently, the method is characterized as an unsupervised OOD detection approach. It is emphasized that the decision should not be taken based on the (intermediate) output of the classification network, because neural networks are insensitive to the difference between ID and OOD samples. Furthermore, a stand-alone OOD detector can be used as an extension to pre-trained networks. Additionally, the decision criteria for OOD must be interpretable since this helps users of safety-critical applications to gain trust more easily in the classification and OOD detection. Lastly, a computationally efficient method that requires little data storage improves the chance of it being added to existing applications. Thus, the developed method must satisfy the following requirements:

1) Only the new time-series input data and the training set (without labels) are available for OOD detection,
2) The OOD decision criteria must be user-friendly,
3) The method must be computationally efficient without requiring much storage capacity.

Two relevant methods for OOD detection of time series are available in the literature. In [6], two deep generative models are designed to find a seasonal ratio score, which works by dividing a time-series signal into its seasonal patterns and random noise. It then compares new data to the expected patterns. If data deviates substantially it is classified as OOD. In [7], time-series data are converted into features and latent distributions to apply already existing OOD techniques from the image domain. Both methods achieve superior results to their respective benchmarks, but conflict with requirements (i) and (ii) because users of the classification (outside the field of machine learning) have difficulty to interpret OOD decision criteria.

A method that satisfies the requirements is distance-based OOD detection, see [5]. This method is an intuitive manner to detect OOD examples, if an example is far away in terms of a well-defined distance measure from the training set, then it is OOD. However, a distance-based metric for time-series OOD detection is challenging because small misalignments between time series typically result in a large distance. Note that a distance-based metric on the latent features extracted from the classifier is not applicable as it conflicts with requirement (i). Furthermore, it is not computationally tractable to compute the distance between a new example and the entire training set due to the size of this set.

Training set size can be reduced through clustering techniques. The combination of OOD detection and clustering is previously researched in [8] and [9], where both studies demonstrate the potential of distance-based methods applied to learned feature representations for detecting OOD instances in image data. However, these methods are not applicable to time-series data and do not align with the constraint specified in requirement (i).

Although several methods have been presented to detect OOD samples, at present none satisfies (i)-(ii)-(iii). The aim of this paper is to develop a method that solves these challenges by reducing the training set size via clustering and compare a new time-series sample with the reduced training set by means of a time-invariant distance measure. Hence, the main contribution of this paper is a design methodology for a distance-based time-series OOD detector.

Note that the main contribution can be directly extended to the domain of fault detection in dynamical systems. Faulty behavior can potentially be distinguished from healthy behavior using the presented distance-based method. Furthermore, the reliability of data-driven fault detectors can be confirmed through outofdistribution (OOD) detection. These detectors remain trustworthy only when the new timeseries data generated by the system closely match the data on which the detector was trained. The OOD detection approach introduced in this thesis makes such validation feasible.

An additional contribution is the demonstration of the effectiveness of this methodology on a case-study concerning the real-life safety-critical control application of mechanical ventilation. In mechanical ventilation, a supervisory control strategy is employed to reduce patient ventilator asynchrony (PVA), which is of crucial importance to reduce the length of hospital stays and mortality rates [10]. To enable such a strategy, it is necessary to detect PVA using a classification network. However, these classifiers are typically trained on limited data sets, highlighting the need for an OOD detector to ensure safe deployment of the supervisory control system.

The outline is as follows. In Section II, the problem setting of OOD is described. In Section III, the novel methodology for OOD detection is presented. Thereafter, in Section IV, the application of mechanical ventilation is introduced in the context of OOD detection. In Section V, results of the mechanical ventilation case-study are shown. Finally, in Section VI, the conclusions and recommendations are presented.

## II. PROBLEM SETTING OF OOD DATA DETECTION

The definition of out-of-distribution data detection and the problem setting is first introduced through an example in

Section II-A. Thereafter, the general definition and problem setting are presented in Section II-B.

### A. OOD DATA DETECTION EXAMPLE

Let a time series $x(t) \in \mathbb{R}$, with time $t$, be generated by the system

$$x(t) = g(\theta, t) = \sin(2\pi\theta(1)t + \theta(2)), \tag{1}$$

where $\theta$ is the parameter vector $\theta = [\theta(1),\ \ \theta(2)] \in \mathbb{R}^2$. Each time series has a label $y \in \{0, 1\}$. An instance (the time-series signal) is classified as 'high-frequency', i.e., it has label $y = 1$, if $\theta_1 > 3$ Hz. The parameter vector $\theta$ is a realization of the random variable $\Theta$, where

$$\Theta \sim \mathcal{P}(\Lambda) = \mathcal{U}_{[0,10]} \times \mathcal{U}_{[-\pi,\pi)} \tag{2}$$

with $\mathcal{P}(\Lambda)$ a probability distribution, $\Lambda = \{0, 10, -\pi, \pi\}$ the shape parameters of the distribution and $\mathcal{U}_{[a,b]}$ a continuous uniform distribution on the interval $[a, b]$.

Suppose a training set contains data for the parameter set $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_m\}$, resulting in $m$ different realizations of the random variable $\Theta$, such that $\theta_i(1) \in [0, 5]$, and $\theta_i(2) \in [0, \frac{1}{2}\pi]$ $\forall i \in \{1, \ldots, m\}$. With these $m$ realizations, the general (or prior) distribution $p(\Theta) = \mathcal{P}(\Lambda)$ is conditioned as the conditional (posterior) distribution

$$p(\Theta|\boldsymbol{\theta}) \approx \mathcal{U}_{[0,5]} \times \mathcal{U}_{[0,\frac{1}{2}\pi]}. \tag{3}$$

The classifier $f_{\text{NN}}$ is trained for a dataset that corresponds to the posterior distribution $p(\Theta|\boldsymbol{\theta})$, while ideally we want to train it for a dataset that covers the whole input space, i.e., $\mathcal{P}(\Lambda)$. A new instance $x_n$ with unknown label $y_n$ arrives generated with $\theta_n = [8, 0]$, which is unknown to the classifier. The question becomes whether $x_n = g(\theta_n)$ is likely from the conditional distribution $p(\Theta|\boldsymbol{\theta})$, which should only be determined based on information on $x_n$ and $\boldsymbol{x}$ (since clearly $y_n$ is not available). If this is not the case then the instance is classified as out-of-distribution.

The problem setting is defined explicitly as: how to design a methodology to decide, only on the basis of $x_n$ and $\boldsymbol{x}$, whether $x_n$ is likely drawn from the distribution in (3), and thus is in-distribution, or not, and thus is out-of-distribution.

### B. PROBLEM DEFINITION

More specifically to the application at hand, consider a system $g(\theta, t)$, generating time series $x(t) = g(\theta, t)$ characterized by the parameter vector $\theta \in \mathbb{R}^r$ with $r$ the number of parameters in $\theta$. This system generates a multivariate time series $x(t) \in \mathbb{R}^b$, where $t$ is the time and $b$ the dimension of the multivariate time series. Each sequence has a variable length $P$ and is uniformly sampled at rate $f_s$, yielding a discrete time series $x \in \mathcal{X} = \mathbb{R}^{b\times P}$. This time series contains implicit information about the corresponding categorical label $y \in \mathcal{Y} = \mathbb{N}_{\geq 0}$.

Further analysis of the time series, e.g., by a human expert, makes the label $y$ explicit. This is generally time-consuming and challenging. Therefore, a classifier $f_{\text{NN}} : \mathcal{X} \rightarrow \mathcal{Y}$ is designed that automatically detects the label $y$ from the input $x$. In this paper, it is assumed that a classifier is trained using the training set $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$, where $\tilde{\boldsymbol{x}} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m\}$ and $\tilde{\boldsymbol{y}} = \{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m\}$. These training instances are generated with $m$ underlying parameter vectors $\tilde{\boldsymbol{\theta}} = \{\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_m\}$. Each parameter vector $\tilde{\theta}_i$ is sampled from the random variable $\Theta \sim \mathcal{P}(\Lambda)$ where $\Lambda$ are the shape parameters of the probability distribution $\mathcal{P}$. The general distribution $p(\Theta) = \mathcal{P}(\Lambda)$ is visualized in Fig. 2 by the green ellipse.

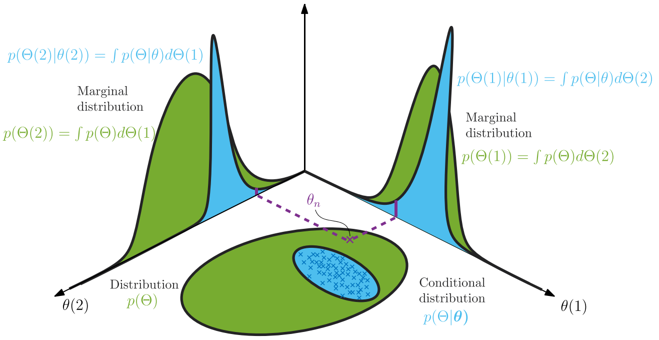
$p(\Theta(2)|\theta(2)) = \int p(\Theta|\theta)d\Theta(1)$
Marginal distribution
$p(\Theta(2)) = \int p(\Theta)d\Theta(1)$
$p(\Theta(1)|\theta(1)) = \int p(\Theta|\theta)d\Theta(2)$
Marginal distribution
$p(\Theta(1)) = \int p(\Theta)d\Theta(2)$
$\theta_n$
Distribution $p(\Theta)$
Conditional distribution $p(\Theta|\boldsymbol{\theta})$
$\theta(2)$
$\theta(1)$

**FIGURE 2. Simplified schematic representation of the out-of-distribution definition for a process with two parameters $\Theta(1)$ and $\Theta(2)$. A sample $(x_n, y_n)$ generated by the parameter vector $\theta_n$ is out-of-distribution if it is unlikely that $\theta_n$ belongs to the conditional probability $p(\Theta|\theta)$.**

Ideally, a classifier $f_{\text{NN}} : \mathcal{X} \rightarrow \mathcal{Y}$ is trained on a set $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ with a set $\tilde{\boldsymbol{\theta}}$ that entirely captures the general distribution $p(\Theta)$. In practice, only a finite number of realizations are available through measurements, making it impossible to span the entire distribution $p(\Theta)$. Instead the classifier is trained on the observed set $(\boldsymbol{x} = g(\boldsymbol{\theta}), \boldsymbol{y})$, where $\boldsymbol{\theta}$ is the set of parameter vectors visualized by the blue crosses in Fig. 2. The observed data conditions the general distribution as $p(\Theta|\boldsymbol{\theta})$, which is represented by the blue ellipse in Fig. 2. This conditional distribution reflects the probability density of $\Theta$ constrained to the set of observed data $\boldsymbol{\theta}$. From a Bayesian perspective, the prior distribution $p(\Theta)$ is updated with the likelihood $p(\boldsymbol{\theta}|\Theta)$ via Bayes' rule

$$p(\Theta|\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}|\Theta)p(\Theta)}{p(\boldsymbol{\theta})}, \tag{4}$$

where $p(\boldsymbol{\theta}) = \int p(\boldsymbol{\theta}|\Theta)p(\Theta)d\Theta$ ensures normalization, to find the posterior distribution. The likelihood $p(\boldsymbol{\theta}|\Theta) := \prod_{i=1}^{m} p(\theta_i|\Theta)$ (under the assumption that each realization $\theta_i$ is independent) describes how probable the measured $\theta_i$ is for a given $\Theta$. The conditional distribution represents the coverage of the parameter space due to the fact that the finite number of realizations $\boldsymbol{\theta}$ represent only a subset of all possible realization from $p(\Theta)$.

Now, we measure a new instance $x_n \in \mathbb{R}^b$ with unknown label $y_n$, generated by an unknown parameter vector $\theta_n$. Our goal is to assess whether it is likely that $\theta_n$, the purple cross in Fig. 2, is a realization from the conditional distribution $p(\Theta|\boldsymbol{\theta})$ or not. If $p(\theta_n|\boldsymbol{\theta})$ is large (to be defined more precisely later) then the instance $x_n$ is called in-distribution (ID) otherwise the instance $x_n$ is called out-of-distribution (OOD). Note that the definition of OOD on distribution level cannot be used in the

OOD detection method and that only the measurements $\{\boldsymbol{x}, \boldsymbol{y}\}$ are available. Furthermore, it is worth noting that the problem definition can be generalized to non-parametric distributions, making the definition applicable to a broader class of models.

The problem setting is defined explicitly as: how to design a methodology to decide, only on the basis of $x_n$ and $\boldsymbol{x}$, whether $x_n$ is likely drawn from the posterior distribution $p(\Theta|\boldsymbol{\theta})$, and thus is in-distribution, or not, and thus is out-of-distribution.

## III. OOD METHODOLOGY

In this section, the methodology of the distance-based OOD time-series detector is presented. Firstly, the methodology is concisely presented in a stepwise manner in Section III-A. Thereafter, each step of the methodolgy is explained in more detail. In Section III-B, the time-invariant distance measure (Dynamic-Time Warping (DTW)) is introduced that is used to properly align two time series. Subsequently, the training set reduction method (Agglomerative Hierarchical Clustering (AHC)) is introduced in Section III-C. Lastly, In Section III-D, a method for the OOD distance threshold computation is presented.

### A. DISTANCE-BASED OUT OF DISTRIBUTION DATA DETECTION

Given that $x_n$ is measured and that $\theta_n$ remains (partly) unknown, we aim to evaluate whether $\theta_n$ belongs to the conditional distribution $p(\Theta|\boldsymbol{\theta})$ solely based on the knowledge of $x_n$ and the training data $(\boldsymbol{x}, \boldsymbol{y})$. Note that parameter set $\boldsymbol{\theta}$ of the training set is considered to be unknown. This evaluation is done using a distance-based detector $h(.)$, which is defined as follows:

$$h(d_n, d^*) = \begin{cases} 1 & \text{if } d_n \geq d^* \quad \text{(OOD)}, \\ 0 & \text{if } d_n < d^* \quad \text{(ID)}. \end{cases} \tag{5}$$

Where $d^*$ is the OOD threshold and $d_n$ the distance between the new example $x_n$ and the most similar instances from the training set $\boldsymbol{x}$, i.e.,

$$d_n = \min_{j \in \{1,\ldots,m\}} d_{\text{DTW}}(x_n, x_j). \tag{6}$$

The distance function $d_{\text{DTW}}$ aligns two signals in time via a non-linear mapping and computes the Euclidean distance between two dynamically time-warped signals [11].

Computing the DTW distance $d_{\text{DTW}}$ between a new example $x_n$ and the entire training set is computational demanding with the number of operations of the order $\mathcal{O}(mT^2)$, where $T$ is the length of the time series. If the size of the training set ($m$) is large, then computing $d_{\text{DTW}}$ becomes intractable. Therefore, we propose a clustering method to reduce the training set size. Clustering enables us to select only the most representative samples $\boldsymbol{x}^* = \{x_1^*, \ldots, x_K^*\}$ from the training set, where $K$ is the number of clusters. More specifically, the clustering method that is used, is Agglomerative hierarchical clustering [12]. In a distance-based method, an OOD threshold $d^*$ in (5) is necessary to evaluate the samples. This threshold is computed based on the distance between the training set $\boldsymbol{x}$ and the most representative samples $\boldsymbol{x}^*$.

Let us now make the proposed approach for OOD explicit in a step-wise approach. The following steps are conducted (and shown in Fig. 3) to design the OOD distance-based detector $h(.)$ in (5):

1) Compute the dynamic time-warping (DTW) distance between all instances from the training set $\boldsymbol{x} = \{x_1, \ldots, x_m\}$, i.e., $\mathcal{D}(i, j) = d_{\text{DTW}}(x_i, x_j)\ \forall i, j \in \{1, 2, \ldots, m\}$, where $m$ is the number of instances in the training set (see Subsection III-B for the definition of $d_{\text{DTW}}$), see also Fig. 3.1.
2) Cluster the training set $\boldsymbol{x}$ into $K$ clusters using the dynamic time-warping distances $\mathcal{D}$ and agglomerative hierarchical clustering with complete linkage (see Subsection III-C for details). This leads to the clusters $\boldsymbol{c} = \{c_1, \ldots, c_K\}$, where each cluster $c_i$ contains one or multiple instances from $\boldsymbol{x}$, see also Fig. 3.2.
3) Compute the most representative instance in each cluster. The most representative instance is the instance with the smallest maximum distance to the other instances, i.e.,

$$x_a^* = \text{argmin}_{u \in c_a} \max_{v \in c_a}(d_{\text{DTW}}(u, v)), \tag{7}$$

where $(u, v)$ are instances within cluster $c_a$ with $a \in \{1, 2, \ldots, K\}$. This results in the set of most representative instances $\boldsymbol{x}^* = \{x_1^*, \ldots, x_K^*\}$, where $K < m$, see the bold crosses in Fig. 3.3.
4) Compute the OOD threshold $d^*$:
   a) Define $\underline{\boldsymbol{x}} := \boldsymbol{x} \backslash \boldsymbol{x}^* = \{\underline{x}_1, \ldots, \underline{x}_Z\}$, where $(\backslash)$ is the set difference operator and $Z = m - K$.
   b) Compute the smallest distance $\boldsymbol{d}_{\text{td}}$ between all samples in $\underline{\boldsymbol{x}}$ and the representative instances $\boldsymbol{x}^*$ using

$$\boldsymbol{d}_{\text{td}}(i) = \min_{a \in \{1,\ldots,K\}} d_{\text{DTW}}(\underline{x}_i, x_a^*) \qquad \forall i \in \{1, \ldots, Z\}. \tag{8}$$

   c) Fit a probability density function on $\boldsymbol{d}_{\text{dt}}$ using maximum likelihood and determine the 99.5% confidence bound, which is defined as the OOD threshold $d^*$ (see Subsection III-D for more details), see also Fig. 3.4.
5) Compute distance

$$d_n := \min_{a \in \{1,\ldots,K\}} d_{\text{DTW}}(x_n, x_a^*) \tag{9}$$

which is the minimal distance between the new instance $x_n$ and the most representative instances $x^* = \{x_1^*, \ldots, x_K^*\}$, see Fig. 3.5 and 3.6.
6) The OOD detector determines whether an instance is ID or OOD with (5).

The methodology is a design procedure for a distance-based out-of-distribution data detector. In the upcoming sections, certain aspects of the methodology are explained in more detail.

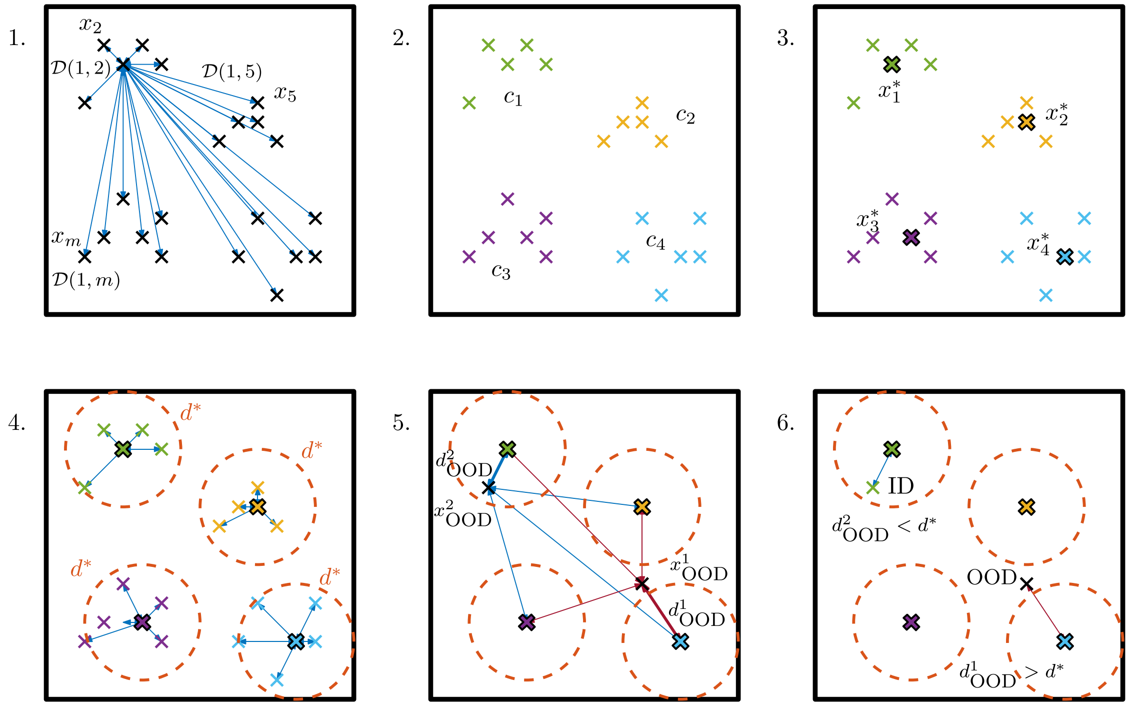

**FIGURE 3.** Schematic representation of the OOD detector design methodology.

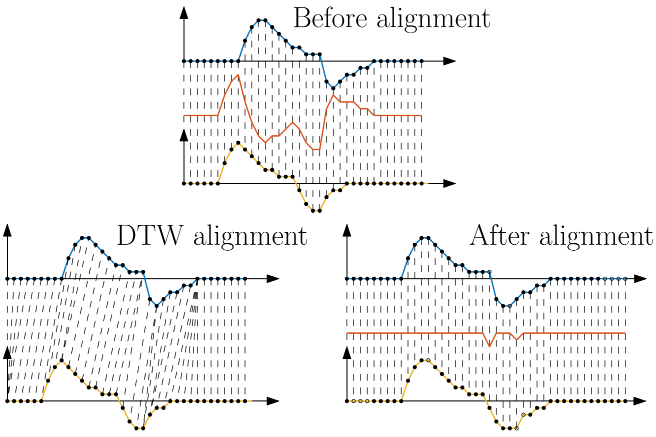

**FIGURE 4.** Schematic representation of the nonlinear temporal warping effort of DTW with respect to the Euclidean distance. On the top, the unwarped signals (▬) and (▬) aligned by the Euclidean distance measure are shown together with the sample-wise error (▬). In the bottom left, the unwarped signals aligned by DTW are shown. On the bottom right, the dynamically time-warped signals are shown, where the open dots represent the repeated data points. The Euclidean distance of the warped signals is much smaller compared to the Euclidean distance of the original signals.

### B. DYNAMIC-TIME WARPING

Dynamic time-warping (DTW) [11] finds the similarity between two different time series by allowing non-linear transformations on both signals, see Fig. 4. Two time-series signals are temporally aligned by local repetition of points in both signals (the open circles in the bottom right plot of Fig. 4). Hereby, the Euclidean distance between the two signals is minimized.

Normally, it is challenging to compare time series based on the Euclidean distance because small timing misalignments result in large distances; however, this is solved with DTW as shown in Fig. 4. Two time-series signals with similar shape but a small timing mismatch are shown. Computing the Euclidean distance results in a large error as shown by the red line in the top plot of Fig. 4. In the bottom left plot of Fig. 4, DTW finds a new alignment, which results in the bottom right plot. Using the Euclidean distance on the warped signals results in a much smaller distance (red line in the bottom right), confirming that the signals are indeed similar. The advantage of DTW over correlation methods is that the distance between two time-series signals is determined after a non-linear transformation (in time) that aligns time-series signals based on their shape. If the signals are not (non-linearly) transformed in such a manner, then the signals do not appear to be similar in distance-based metrics.

In this paper, we deal with dependent multivariate time series, where a change in one signal affects all other signals. This dependency implies that all signals follow the same temporal structure. Therefore, aligning one signal is sufficient to synchronize the entire time series. Hereby, eliminating $b-1$ computations and as a result reducing the computation time significantly. If the channels within a multi-variate time series are coupled but drift independently, then a full multivariate DTW is necessary. A single signal of the time series is defined as $z_i := x_i^\kappa$, where $\kappa \in \{1, 2, \ldots, b\}$ is the $\kappa$-th signal of $x_i$. The DTW-distance $d_{\text{DTW}}$ is defined as the distance between

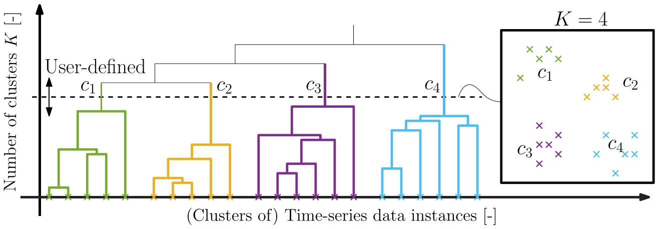
$K = 4$
Number of clusters $K$ [-]
User-defined
$c_1$
$c_2$
$c_3$
$c_4$
(Clusters of) Time-series data instances [-]

**FIGURE 5.** Illustration of the agglomerative hierarchical clustering method, where the user-defined amount of clusters is set to $K = 4$. Each cross is a time-series data instance.

time-warped signals with the optimal warping path $\pi^*$:

$$d_{\mathrm{DTW}}(x_i, x_j) : = c_{\pi^*}(z_i, z_j) = \min_{\pi \in \mathcal{A}} c_\pi(z_i, z_j), \qquad (10)$$

where $c_\pi(.)$ is a local cost function, $\mathcal{A}$ is the set of admissible warping paths $\pi$. In Appendix A, further details regarding the cost function, warping paths, and implementation of DTW are given.

For the clustering, which is described next, the DTW distance between every instance in the training set $\boldsymbol{x}$ needs to be computed, i.e., $\mathcal{D}(i, j) = d_{\mathrm{DTW}}(x_i, x_j)\ \forall i, j \in \{1, 2, \ldots, m\}$.

### C. AGGLOMERATIVE HIERARCHICAL CLUSTERING WITH DTW

As stated in Section III-A, computing the DTW distance between a new instance $x_n$ and the entire training set is intractable if the training set is large due to the computational burden of computing DTW. Therefore, it is necessary to reduce the size of the training set for the OOD distance comparison; hereby the computation time is reduced from $\mathcal{O}(mT^2)$ to $\mathcal{O}(KT^2)$, with $K$ the number of clusters.

By applying clustering, the training set is systematically partioned divided into groups (clusters), where the data instances are similar. Selecting a representative instance from each cluster (to be used for the OOD detection) enables a substantial reduction in the training set size while preserving the diversity of the shapes in the training set. Clustering is preferred over random selection because time-series data often exhibit class imbalance, i.e., certain shapes are overrepresented in the training set. Statistically, random selection tends to favor the most common shapes while neglecting the under-represented shapes or classes. In contrast, the cluster-based approach in this paper ensures that the reduced training set is of limited size and contains all shape variations.

In this paper, an Agglomerative Hierarchical Clustering (AHC) method is used that progressively merges clusters based on a similarity measure. It does not require an iterative optimization procedure to converge; hence, this algorithm always finds the best solution invariant of the initial conditions. In AHC, every instance is considered a cluster at the start, see the crosses on the horizontal axis in Fig. 5. Progressively, instances are merged into clusters using the DTW similarity measure until the user-defined number of clusters is achieved (e.g., $K = 4$ in Fig. 5). The distance between the clusters can be defined in multiple ways and is referred to as the linkage method [12].

The distance between clusters is measured using the complete linkage distance measure. Note that the triangle inequality does not hold when using the DTW distance as measure. The distance between clusters is measured using the complete linkage distance measure. This linkage method is chosen because it can ensure that every instance in a cluster is at most some threshold DTW distance apart without relying on the triangle inequality. Other linkage methods (e.g., single linkage or centroid-based linkage methods) rely on the triangle inequality to guarantee that every instance is close to the other instances in a cluster; therefore, these linkage methods cannot be used. The complete linkage distance measure is defined as the distance between the least similar instances within two clusters, i.e.,

$$\sigma(c_r, c_s) = \max_{u \in c_r,\, v \in c_s} d_{\mathrm{DTW}}(u, v), \qquad (11)$$

where $c_r$ and $c_s$ are the two selected clusters. Data instances (or clusters) are merged if their complete linkage distance $\sigma$ is smallest among all possible pairs of data instances (or clusters). In Appendix B, the AHC procedure with complete linkage is presented. After iteratively solving the AHC procedure, we obtain a set of clusters which contain one or multiple instances from the training set, e.g., $c_1 = \{x_2, x_{100}, x_{519}\}$ and $c_K = \{x_1\}$.

After the clustering, it is possible to find the most representative instance in each cluster and thereby reducing the size of the training set massively. The most representative instance within a cluster is the instance with the minimal maximum distance to the other instances within that cluster, i.e.,

$$x_a^* = \mathrm{argmin}_{u \in c_a} \max_{v \in c_a}(d_{\mathrm{DTW}}(u, v)), \qquad (12)$$

where $(u, v)$ are instances within cluster $c_a$ with $a \in \{1, \ldots, K\}$. The size of the training set is reduced from $m$ to $K$ instances using the above described method, which decreases the amount of evaluations that need to be done for the computation of $d_n$ to

$$d_n = \min_{j \in \{1, \ldots, K\}} d_{\mathrm{DTW}}(x_n, x_j^*). \qquad (13)$$

A lower value for $K$ results in a larger reduction; however, if $K$ is too low, valuable time series information is lost. The value for $K$ is application-specific and should be chosen at least as large as the number of classes for the classification task. Note that AHC computes the clusters for all values of $K$. This property enables computationally efficient experimentation with different values of $K$ to support its tuning.

### D. OOD THRESHOLD COMPUTATION

The threshold for OOD detection is determined based solely on the training set. By definition the training samples belong to the conditional distribution $p(\Theta|\boldsymbol{\theta})$ and are defined as in-distribution (ID). Subsequently, the OOD threshold is found by computing the maximum distance between all training points and the representative instances (see Fig. 3.4).

First, we define a set of all training samples without the representative instances $\underline{\boldsymbol{x}} := \boldsymbol{x}\backslash\boldsymbol{x}^* = \{\underline{x}_1, \dots, \underline{x}_Z\}$, where $(\backslash)$ is the set difference operator. The smallest distance of all instances in $\underline{\boldsymbol{x}}$ to the representative instances $\boldsymbol{x}^*$ is computed using the DTW distance and stored in $\boldsymbol{d}_{\mathrm{td}}$ as

$$\boldsymbol{d}_{\mathrm{td}}(i) = \min_{j\in\{1,\dots,K\}} d_{\mathrm{DTW}}(\underline{x}_i, x_j^*) \quad \forall i \in \{1, \dots, Z\}. \tag{14}$$

The distances in $\boldsymbol{d}_{\mathrm{td}}$ are used to determine the threshold for OOD detection. By definition, all training samples should be in-distribution, because these samples condition the distribution. Therefore, it is logical to set the OOD distance to the largest distance in $\boldsymbol{d}_{\mathrm{td}}$. However, there are always some outliers and noise in the training data; therefore, we set the OOD threshold such that the probability is 99.5% that the instances from the training set are in-distribution. Note that this threshold is user-defined and application-specific. To find the threshold, we first need to fit a distribution to $\boldsymbol{d}_{\mathrm{td}}$ and, subsequently, determine the 99.5% confidence interval.

Let $\Gamma(\gamma)$ be the distribution with probability density function $f(\boldsymbol{d}_{\mathrm{td}}|\gamma)$ that is the best fit for $\boldsymbol{d}_{\mathrm{td}}$, where $\gamma$ are the shape parameters of the distribution. To estimate the shape parameters, the maximum likelihood $\mathcal{L}(\gamma|\boldsymbol{d}_{\mathrm{td}}) = f(\boldsymbol{d}_{\mathrm{td}}|\gamma)$ is optimized via

$$\hat{\gamma} = \mathrm{argmax}_{\gamma} \log \mathcal{L}(\gamma|\boldsymbol{d}_{\mathrm{td}}). \tag{15}$$

As a result of this optimization, the shape parameters $\hat{\gamma}$ of the probability density function $f(\boldsymbol{d}_{\mathrm{td}}|\hat{\gamma})$ are estimated. Subsequently, the 99.5% probability threshold is found with

$$d^* := d_{99.5\%} = F^{-1}(0.995, \hat{\gamma}), \tag{16}$$

where $F^{-1}(.)$ is the inverse of the cumulative density function $F(.) = \int_{-\infty}^{(.)} f(\tau|\hat{\gamma})d\tau$. This threshold should be computed for each training set and each set might be exhibiting another distribution. In the case-study, in Section V, the gamma distribution is the best fit for this distribution.

## IV. OOD DETECTION IN MECHANICAL VENTILATION

An example of a real-world application in a safety-critical environment is the detection of patient-ventilator asynchrony (PVA) in mechanical ventilation [13]. Patient-ventilator asynchrony is a mismatch in demand between patient and ventilator which is associated with longer hospital stays and increased mortality [10], [14]. Automatic detection of PVA is enabled by neural networks [15], [16]. These networks are trained on real patient data. Gathering this data is a time-consuming, privacy-sensitive, and challenging process. As a result, only limited datasets are available that contain data from within one medical facility. Therefore, training a neural network that is able to detect PVA in all different patient ventilator combinations is practically infeasible.

Therefore, detection of out-of-distribution data is of crucial importance because the neural network always comes across instances it has not seen during training when deployed in the medical field. Showing OOD data to the trained network results in wrong predictions, possibly leading to wrong patient treatments or false alarms, which results in patient discomfort and longer hospital stays [10]. Therefore, it is crucial that these OOD examples are detected and provided to the clinician for human inspection.

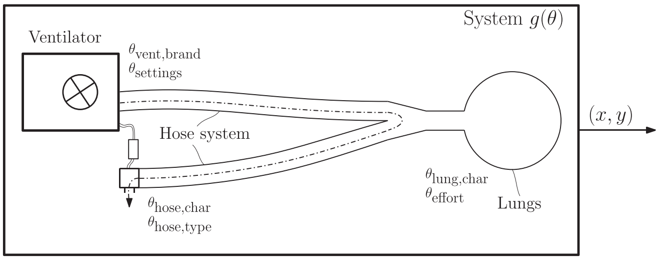
System $g(\theta)$
Ventilator
$\theta_{\mathrm{vent,brand}}$
$\theta_{\mathrm{settings}}$
Hose system
$(x, y)$
$\theta_{\mathrm{lung,char}}$
$\theta_{\mathrm{effort}}$
Lungs
$\theta_{\mathrm{hose,char}}$
$\theta_{\mathrm{hose,type}}$

**FIGURE 6.** Schematic illustration of the patient-ventilator hose system. Each sub-component contains multiple parameters $\theta_i$ used to characterize that particular part of the system.

Below, a description of the patient, ventilator, and hose system is given and subsequently connected to the mathematical description of OOD as presented in Section II. Consider the system shown in Fig. 6 that generates a patient-ventilator breath based on three main components: a ventilator, a patient, and a hose system. The system is characterized by the parameter vector $\theta$ which is composed of three sub-vectors, each describing a part of the system:

$$\theta = \left[\theta_{\mathrm{patient}}, \theta_{\mathrm{ventilator}}, \theta_{\mathrm{hose}}\right]. \tag{17}$$

The patient parameters are denoted by

$$\theta_{\mathrm{patient}} = \left[\theta_{\mathrm{lung,char}}, \theta_{\mathrm{effort}}\right], \tag{18}$$

where $\theta_{\mathrm{lung,char}}$ are the lung characteristics and $\theta_{\mathrm{effort}}$ describes the patient breathing effort. The ventilator parameters are denoted by

$$\theta_{\mathrm{ventilator}} = \left[\theta_{\mathrm{brand}}, \theta_{\mathrm{settings}}\right]. \tag{19}$$

Where $\theta_{\mathrm{brand}}$ specifies the ventilator brand and $\theta_{\mathrm{settings}}$ represent the ventilator settings. The hose parameters are denoted by

$$\theta_{\mathrm{hose}} = \left[\theta_{\mathrm{type}}, \theta_{\mathrm{hose,char}}\right], \tag{20}$$

where $\theta_{\mathrm{type}}$ is the type of hose set-up and $\theta_{\mathrm{hose,char}}$ are the hose characteristics. Eventually, we have the parameter vector $\theta = [\theta(1), \dots, \theta(l)]$ which contains $l$ parameters that characterize the total patient-ventilator system.

Each realization of the parameter vector $\theta$ defines a configuration of the system and generates a breath $x(t) \in \mathcal{X} = \mathbb{R}^2$, through the generator $x(t) = g(\theta, t)$, for all $t \in \{1, .., T\}$ with $T$ the total signal length. The signals measured each breath are the pressure at the patients mouthpiece $p_{\mathrm{aw}}$ and patient flow $Q_{\mathrm{pat}}$, i.e., $x(t) = [p_{\mathrm{aw}}(t), Q_{\mathrm{pat}}(t)]^{\top}$. In Fig. 7, two different realizations of $x(t)$ for two different $\theta$ are given, where all

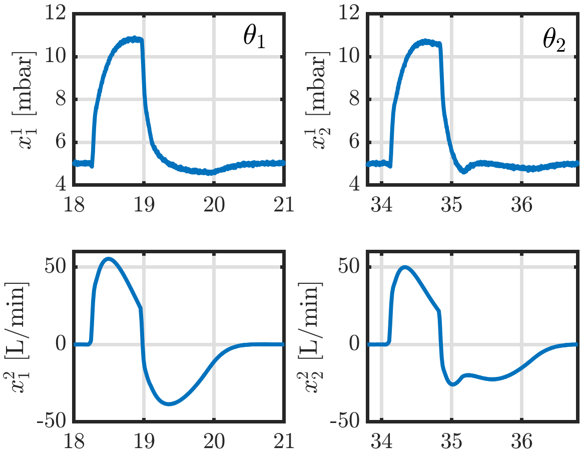

**FIGURE 7.** An example of two different breaths with different configurations of the parameter vectors $\theta_1$ and $\theta_2$. Only the patient breathing effort parameter $\theta_{\text{effort}}$ is different between $\theta_1$ and $\theta_2$, while all other parameters are equal. This difference in breathing effort already leads to a different flow (see $x_1^2$ and $x_2^2$) shape and possibly a different label $\boldsymbol{y}$. The superscript in $x_1^i$ represents that the $i$-th signal of $x_1$ is visualized.

parameters within $\theta$ are equal except for $\theta_{\text{effort}}$. This difference leads already to different breaths as can be seen in Fig. 7.

The breath (information in $x$) contains implicit information about the associated breath label $y \in \mathcal{Y} = \{1, 2, 3\}$ representing the asynchrony type. This information is only available after a clinical expert inspects $x$ thereby making the label $y$ explicit. Hereby, it is assumed that two identical $x$'s have the same asynchrony type $y$ and two different asynchrony types have different $x$'s. A set of $m$ breaths with associated labels, denoted by $(\boldsymbol{x}, \boldsymbol{y})$, is generated by processing $m$ different realizations of $\theta$ through $g(.)$, such that

$$\boldsymbol{x} = \{x_1, \ldots, x_m\}, \boldsymbol{y} = \{y_1, \ldots, y_m\}, \text{ and } \boldsymbol{\theta} = \{\theta_1, \ldots, \theta_m\}. \tag{21}$$

The generator $g(.)$ is typically modelled as a dynamical process [17]. The dataset $(\boldsymbol{x}, \boldsymbol{y})$ is used to train a classification network $f_{\text{NN}} : \mathcal{X} \rightarrow \mathcal{Y}$ that maps an input breath $x$ to its asynchrony type $y$. Hereby, automating the process of the visual inspection of asynchrony by a clinical expert [16].

The set of parameter vectors $\boldsymbol{\theta}$ contains $m$ realizations of the random variable $\Theta$ that is governed by the distribution $p(\Theta)$. The distribution $p(\Theta)$ reflects all possible configurations of all patient, ventilator, and hose systems over the entire world. The classifier $f_{\text{NN}}$ is trained on a limited amount of realizations; hence, only a subset of the distribution $p(\Theta)$ is observed. As visualized in Fig. 2, where the training set (blue crosses) conditions the general distribution $p(\Theta)$ as $p(\Theta|\boldsymbol{\theta})$. Thus, the classifier is trained on the conditional distribution $p(\Theta|\boldsymbol{\theta})$.

In a new scenario, we measure a breath $x_n$ that contains implicit knowledge about the unknown breath label $y_n$. The aim is to assess whether $(x_n, y_n)$ could plausibly have been produced by the generator $g(\theta_n)$ where $\theta_n$ is a realization that belongs to the conditional distribution $p(\Theta|\boldsymbol{\theta})$, as visualized by Fig. 2. A realization $\theta_n$ that is unlikely to belong to the conditional distribution $p(\Theta|\boldsymbol{\theta})$ is defined as an out-of-distribution realization, and thus $x_n$ is defined as an out-of-distribution breath.

In a clinical situation, $\theta_n$ is unknown, making it impossible to test directly if $\theta_n$ belongs to $p(\Theta|\boldsymbol{\theta})$ via $p(\theta_n|\boldsymbol{\theta})$. Instead, we want to determine if the breath is out-of-distribution based on the distance between $x_n$ and $(\boldsymbol{x}, \boldsymbol{y})$, using the method as described in Section III. In the next section, a case-study is presented where the OOD detection method is validated with the mechanical ventilation case-study.

## V. CASE-STUDY: OOD DETECTION IN MECHANICAL VENTILATION

In this section, the results of the mechanical ventilation case-study are presented. In Section V-A, the results of a simulation case-study are shown. Here, a situation is investigated where a slight change in ventilator settings results in a shift in asynchrony labels leading to OOD instances. In Section V-B, the results of clinical case-study are presented. Mechanical ventilation data from two hospitals are used in the analysis to see if the method is able to detect whether data from a different medical facility is detected as OOD.

### A. SEVERE ASYNCHRONY TYPE USE-CASE (SYNTHETIC DATA)

In this section, the results of the use-case with synthetic data are presented. First, the details of the use-case are introduced and thereafter results of the methodology as explained in Section III are shown.

### A. DESCRIPTION USE-CASE

The first use-case contains synthetic patient data. This data is generated with the system as described in Section IV. To motivate the choice for this use-case, the data included in the training set is defined first. Thereafter, the potential OOD data is introduced.

For the synthetic training set, a set of parameter vectors $\boldsymbol{\theta}_{\text{syn}} = \{\theta_{\text{syn},1}, \ldots, \theta_{\text{syn},m}\}$ is designed; thereby, the distribution $p(\Theta)$ is conditioned as $p(\Theta|\boldsymbol{\theta}_{\text{syn}})$. The set of parameter vectors $\boldsymbol{\theta}_{\text{syn}}$ contains variations in $\theta_{\text{patient}}$ and $\theta_{\text{settings}}$ while the ventilator brand and hose system are equal for all configurations. With the set of parameter vectors $\boldsymbol{\theta}_{\text{syn}}$, the set $\boldsymbol{x}_{\text{syn}}$ is generated that contains three different (a)synchrony labels $\boldsymbol{y}_{\text{syn}} \in \{1, 2, 3\}$, respectively, normal inspiration, early cycling, and delayed cycling (see [16] for the definition of these asynchronies). The potential OOD instances $\boldsymbol{x}_{n,\text{syn}}$ are generated by changing the patient effort and ventilator settings of one realization of the parameter vector $\theta_{\text{syn}}$, leading to a shift in asynchrony types (adding a double triggering asynchrony $y_{n,\text{syn}} = 4$), such that $\boldsymbol{y}_{n,\text{syn}} \in \{1, 2, 3, 4\}$ which is different from the training set.

The double-trigger asynchrony is not included in the training set, hence, it is impossible for the PVA classifier to classify this instance correctly. Therefore, the OOD detector needs

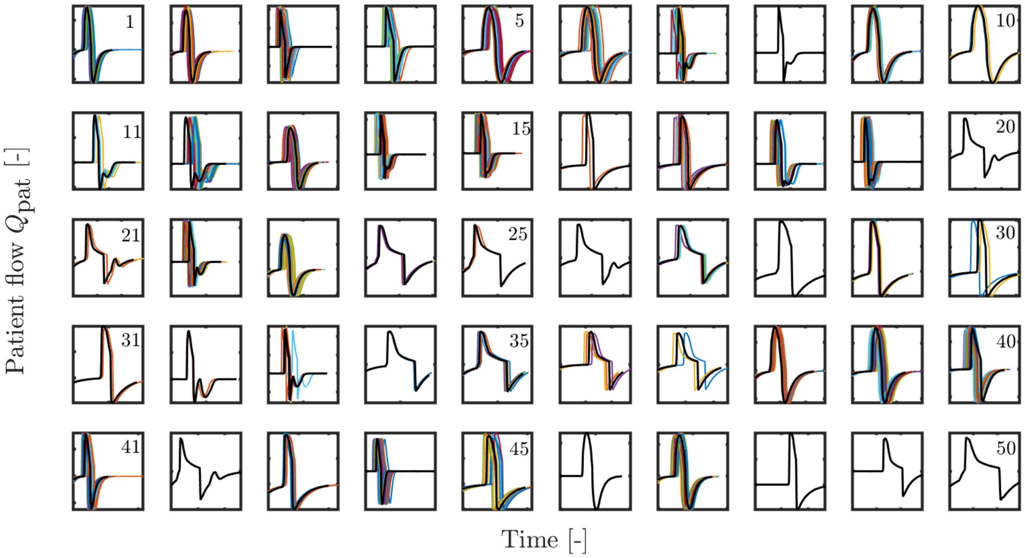

**FIGURE 8.** Division of the synthetic training set over 50 different clusters with the most representative breath $x_a^*$ displayed as the bold black line (▬). The clusters are numbered from 1 to 50 from the top left to the bottom right.

to detect these potential OOD breathing instances. The hypothesis is that double-triggers should be easily detectable because the waveforms in $\boldsymbol{x}_{n,\text{syn}}$ labeled as double-triggers are rather different from the waveforms in $\boldsymbol{x}_{\text{syn}}$, meaning that $d_{\text{DTW}}(x_{n,\text{syn}}, \boldsymbol{x}_{\text{syn}})$ is large.

### A. RESULTS SYNTHETIC DATA

In this result section, we showcase the methodology as explained in Section III and analyse the performance of the OOD detector based on the accuracy of the PVA classifier before and after removing the detected OOD samples. In Fig. 8, the training data $\boldsymbol{x}_{\text{syn}}$ is clustered into 50 clusters using the DTW-distance. The most representative breath of each cluster $x_a^*$ is indicated by the black line. It is shown that agglomerative hierarchical clustering based on the DTW distance results in clusters where the breaths in each cluster have similar shapes. It can also be seen that different clusters still have the same shape, so the amount of clusters $K$ can potentially be further optimized. For this use-case, we are not interested in the optimal cluster number, but only in the proof of principle.

After finding the clusters with their representative breaths, the OOD threshold is determined based on the distance between the training samples and the representative breaths $\boldsymbol{d}_{\text{dt,syn}}$. Subsequently, the minimal distances $\boldsymbol{d}_{n,\text{syn}}$ between the set of new instances $\boldsymbol{x}_{n,\text{syn}}$ and representative breaths $\boldsymbol{x}^*$ are calculated. The results are shown in Fig. 9. A breath of $\boldsymbol{x}_n$ is defined as OOD if the distance $d_{\text{OOD}}$ is larger than the OOD threshold $d^*$. The OOD treshold is determined by fitting a Gamma distribution on $\boldsymbol{d}_{\text{dt,syn}}$ (because all distances are larger than zero) and finding the 99.5% confidence interval. In Fig. 9, every breath above the red dotted line, which is the OOD threshold $d^*$, is OOD. The figure shows some illustrative examples of the flow waveforms of breaths in $\boldsymbol{x}_{n,\text{syn}}$ (in blue) and the most representative breath (in black) of the cluster they belong to. This result confirms the hypothesis that

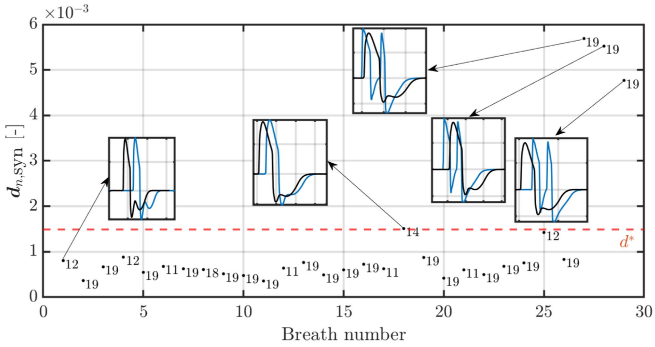

**FIGURE 9.** OOD distance $d_{n,\text{syn}}$ between the new synthetic samples $x_{n,\text{syn}}$ and their most representative cluster $x_a^*$. The red dotted line is the OOD threshold $d^*$. The numbers at each breath are the cluster number they are closest to. It can be seen that all double trigger asynchronies (breath number 27,28,29), the ones where the flow peaks two times, are all detected as out-of-distribution.

double-trigger asynchronies ($y_{n,\text{syn}} = 4$) are different from all the other breaths, and as a result have a large distance $d_{n,\text{syn}}$ between the $\boldsymbol{x}_{n,\text{syn}}$ instances and the most representative breaths $\boldsymbol{x}^*_{\text{syn}}$.

The performance of the PVA classifier before and after filtering of the OOD samples is shown in Fig. 10. The OOD detector is able to filter out all the double-trigger breaths. This improves the accuracy of the classifiction because all double-triggers are all wrongly classified, i.e., not on the diagonal of the confusion matrix in Fig. 10. Especially, the double-trigger that is classified as a normal breath by the classifier (a false negative) results in a lot of discomfort for the patient if no appropriate actions are taken.

By detecting the breath as OOD and sending this information to a clinician, a more accurate analysis can be conducted by the clinician to assess whether the breath is asynchronous or not. If the breath is considered to be asynchrony, the

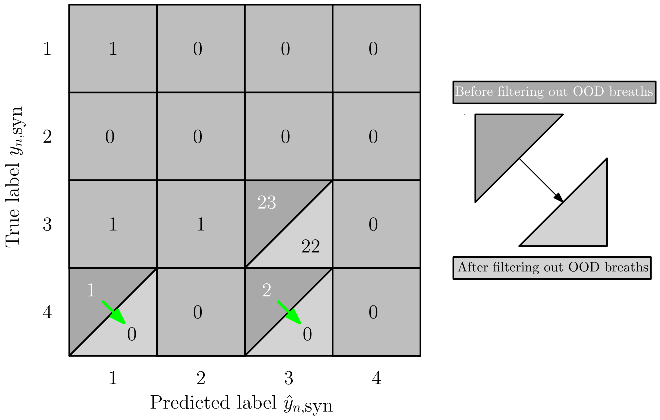

**FIGURE 10.** Confusion matrix of PVA classifier for the entire $\boldsymbol{x}_{\boldsymbol{n},\text{syn}}$ set before and after filtering of the OOD samples. The grey squares remain unchanged after filtering, while the dark and light grey triangles show the changes. The labels in the bottom row are all flagged as OOD breaths by the OOD detector.

clinician can act by changing the ventilator settings if needed.

## B. DIFFERENT HOSPITAL DATA USE-CASE (CLINICAL DATA)

In this section, the results of the clinical data use-case are presented. First, the details of the use-cases are introduced and thereafter results of the methodology as explained in Section III are shown.

### B. DESCRIPTION USE-CASE

The second use-case contains clinical patient data from two separate hospitals. Data from the Fondazione I.R.C.C.S. Policlinco San Matteo (Pavia, Italy) [15] is used as training data and data from Maasstad hospital (Rotterdam, the Netherlands) is used as potential OOD data. In contrast to the synthetic use-case, little information is available regarding the parameter vectors $\boldsymbol{\theta}_{\text{clin}}$ and $\boldsymbol{\theta}_{n,\text{clin}}$. The only available information is that data from both facilities are gathered from different adult patients on different ventilator brands with different hose systems. The training data conditions the distribution of all patient, hose, ventilator combinations $p(\Theta)$ as $p(\Theta|\boldsymbol{\theta}_{\text{clin}})$. Now, we want to know whether it is likely that the set of parameter vectors $\boldsymbol{\theta}_{n,\text{clin}}$ belongs to the conditional distribution $p(\Theta|\boldsymbol{\theta}_{\text{clin}})$.

The training data contains three different (a)synchrony types $\boldsymbol{y}_{\text{clin}} \in \{1, 2, 3\}$, normal breathing, early cycling, and premature cycling, respectively. The potential OOD data from Maasstad contains also other asynchronies which are combined into one type leading to $\boldsymbol{y}_{n,\text{clin}} \in \{1, 2, 3, 4\}$. Based on the results from the synthetic use-case, it is expected that the OOD detector finds all breaths in $\boldsymbol{x}_{n,\text{clin}}$ that have another asynchrony type compared to $\boldsymbol{y}_{\text{clin}}$.

### B. RESULTS CLINICAL DATA

In this result section, we showcase the methodology as explained in Section III and analyse the performance of the OOD detector based on the accuracy of the PVA classifier before and after removing the detected OOD samples. In Fig. 11, the training data $\boldsymbol{x}_{\text{clin}}$ is clustered into 50 clusters using the DTW-distance. The most representative breath of each cluster $x_a^*$ is indicated by the black line. It is shown that agglomerative hierarchical clustering based on the DTW distance results in clusters where the breaths in each cluster have similar shapes and there are not many clusters that have the same shape. Hence, the user-defined amount of clusters is chosen correctly. Furthermore, it can be observed that some clusters contain breaths that are atypical due to a malfunction of the ventilator (e.g., clusters 30, 35, and 49). In the future, it would be better to exclude these breaths before clustering.

After finding the clusters with their representative breaths, the OOD threshold is determined and the minimal distances $\boldsymbol{d}_{n,\text{clin}}$ between the set of new instances $\boldsymbol{x}_{n,\text{clin}}$ and representative breaths $\boldsymbol{x}^*$ are calculated. In Fig. 12, $\boldsymbol{d}_{\text{td,clin}}$ is shown for the clinical set. The distribution that is the best fit for this data is a gamma distribution, because it support inputs between $[0, \infty)$ and it is flexible in its shape. The estimated probability density function becomes $f(\boldsymbol{d}_{\text{td,clin}}|\hat{\gamma})$, which is displayed by the red line in Fig. 12. Note that this line is scaled to match the bar chart. To find the threshold for OOD instances we use the 99.5% probability threshold, i.e.,

$$d^* := d_{99.5\%} = F^{-1}(0.995|\hat{\gamma}), \tag{22}$$

which gives the threshold $d^*$ as indicated by the red dashed line in Fig. 13.

The distance computation between $\boldsymbol{x}_{n,\text{clin}}$ are shown in Fig. 13. A breath of $\boldsymbol{x}_{n,\text{clin}}$ is defined as OOD if the distance $\boldsymbol{d}_n$ is larger than the OOD threshold $d^*$. In Fig. 13, this is every breath above the red dotted line. The figure shows some illustrative examples of the flow waveforms of breaths in $\boldsymbol{x}_n$ (in blue) and breath of the most representative breath (in black) of the cluster they belong to. In general, it is shown that most breaths are close to the OOD threshold, which is an indication that other ventilators indeed produce other types of waveforms. However, not all breaths are classified as OOD, meaning that data from another hospital is not necessarily OOD by definition. Furthermore, it is shown that the double-trigger asynchronies (breath number 8 and 13) are clearly detected as OOD, similar to the synthetic data case because they belong to an asynchrony type not present in the training set $\boldsymbol{y}_{\text{clin}}$.

The performance of the PVA classifier before and after filtering of the OOD samples is shown in Fig. 14. The OOD detector is able to filter out all the asynchrony types that are not available during training. This is shown by the bottom row of Fig. 14. Furthermore, we see that the PVA classifier produces 5 false positives (the third column in Fig. 14 excluding the diagonal element) before detection of the OOD samples. The OOD samples are filtered out after the OOD data detection, leading to 0 false positives. In that sense, the OOD detector improves the detection performance of the PVA classifier for this use-case. The OOD detector also excludes correctly classified breaths (i.e., the instances on the diagonal

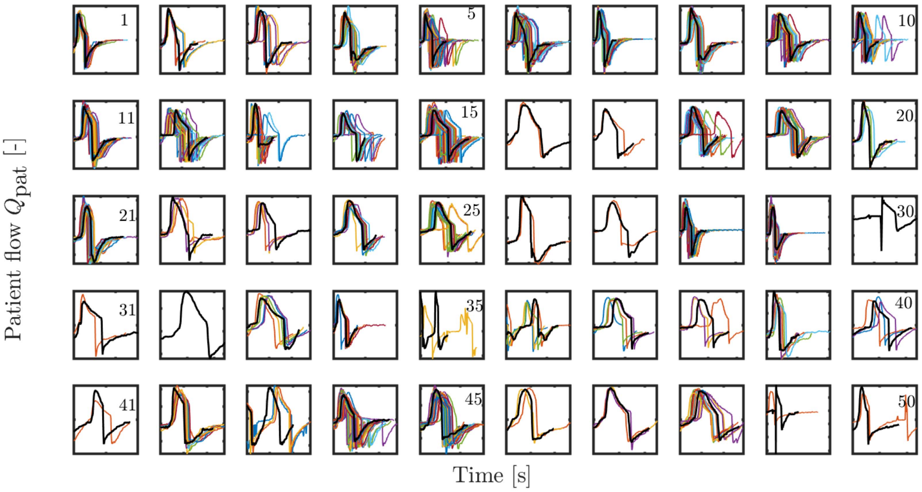

**FIGURE 11.** Division of the clinical training set over 50 different clusters with the most representative breath $x_a^*$ displayed as the bold black line (▬). The clusters are numbered from 1 to 50 from the top left to the bottom right.

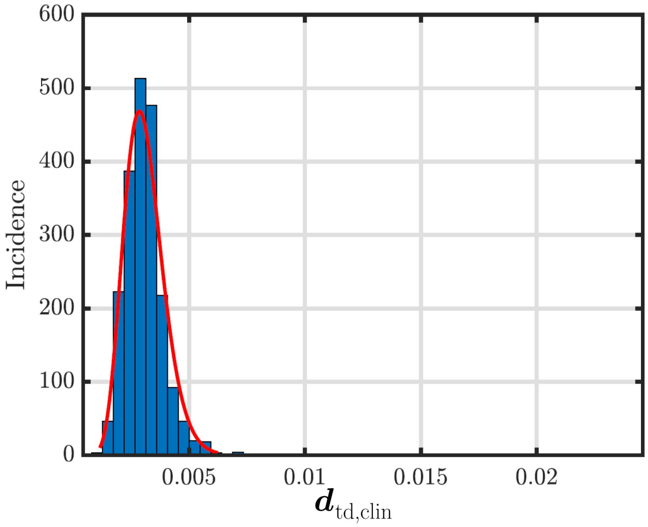

**FIGURE 12.** Distribution of distance between representative breaths and training data $d_{\text{td,clin}}$.

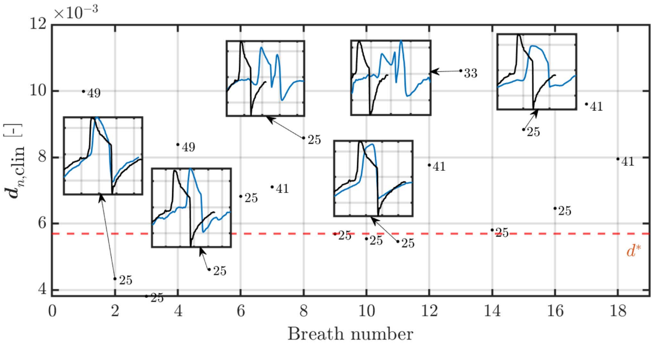

**FIGURE 13.** OOD distance $d_{n,\text{clin}}$ between the new clinical instances $x_{n,\text{clin}}$ and their most representative breath $x_a^*$ from cluster $a$, where $a \in [1, K]$. The red dotted line is the OOD threshold $d^*$. The numbers at each breath are the cluster number they are closest to. Most instances are detected as OOD (e.g., the double triggers in breath number 8 and 13), while very similar breaths (e.g., breath number 2 and 5) are defined as in-distribution.

in Fig. 14) from the set. Two possible explanations exist for this: either the OOD detector filtered some in-distribution (ID) samples, or the classification network correctly classified the OOD samples (by chance). However, with the available clinical data, it is not possible to determine the exact cause.

## VI. CONCLUSION AND RECOMMENDATIONS

In this paper, a novel distance-based method for Out-Of-Distribution (OOD) data detection of time series for classification networks is presented. A data sample is detected as in-distribution or out-of-distribution based on the Dynamic-Time Warping (DTW) distance between the data sample and the data from the training set. To efficiently determine the distance between the data sample and the entire training set, the size of the training set is reduced by means of Agglomerative Hierarchical Clustering (AHC). If a data sample is detected as OOD, it is excluded from the data that is going through the classification network. Hence, the classification network produces more meaningful results. The OOD data samples can be inspected afterwards by the user of the safety-critical application.

The effectiveness of the distance based method for OOD detection is tested on the use-case of Patient-Ventilator Asynchrony (PVA) detection in mechanical ventilation. In this use-case, we found that asynchrony classes not present in the training set are always flagged by the designed OOD detector. Furthermore, it is concluded that data from another hospital with respect to the training data is not out-of-distribution by definition. This is important for training PVA classifiers since this means that it is not necessary to gather data from every individual hospital to train a robust PVA classifier.

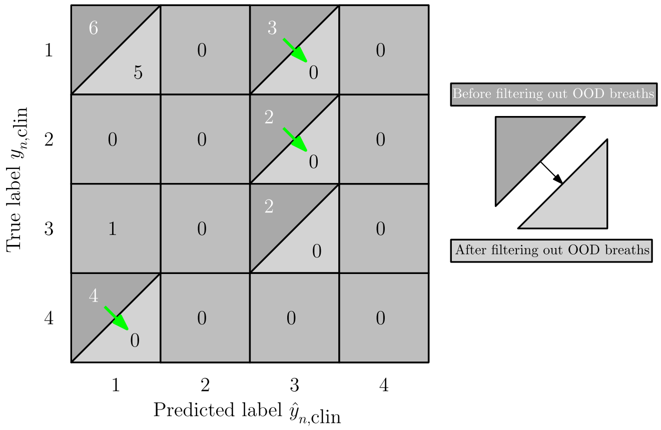

**FIGURE 14.** Confusion matrix of PVA classifier for the entire $x_{n,\mathrm{clin}}$ set before and after filtering of the OOD samples. The grey squares remain unchanged after filtering, while the dark and light grey triangles show the changes. The labels in the bottom row are all flagged as OOD breaths by the OOD detector together with some predictions in the third column.

For future research, clustering with DTW will be exploited even further. In this paper, classic DTW is used based on a univariate time series; however, for future research using multivariate dynamic time-warping would also be interesting because this would make the alignment invariant of disturbances. Besides that, employment of this OOD detection in a real-world environment is valuable for future advancements. Besides detection of OOD, this method also shows potential for active learning. In particular, if instances are detected as OOD they can be stored and eventually be added to the training set. The last potential use-case for the clustering is in the data selection for labeling. The reduction method selects the most representative data of a large set that could be given to expert to annotate only a subset of a large data set that contains the most informative data. Additionally, the presented method can be applied to other applications where classification based on time series is conducted. Lastly, a quantitative comparison with other OOD detection methods, such as generative models or approaches based on latent feature distances, would be beneficial to benchmark the performance of the OOD detection method presented in this paper. However, to do this, further advances in such approaches are necessary to make them applicable to time-series data.

## APPENDIX

### A. DYNAMIC TIME-WARPING

Dynamic time-warping is described as follows. The discrete measured time-series signals are defined as $z_i := (z_i(1), z_i(2), \ldots, z_i(W))$ of length $W \in \mathbb{N}$ and $z_j := (z_j(1), z_j(2), \ldots, z_j(Q))$ of length $Q \in \mathbb{N}$. To compare the samples $z_i(w), z_j(q) \in \mathbb{R}$ for $w \in \{1, 2, \ldots, W\}$ and $q \in \{1, 2, \ldots, Q\}$ we define a local distance measure

$$c : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_{\geq 0}. \tag{23}$$

Typically, $c(z_i(w), z_j(q))$ is small if samples are similar to each other, and otherwise, $c(z_i(w), z_j(q))$ is large. In this case, the local cost is defined as

$$c(z_i(w), z_j(q)) = \|z_i(w) - z_j(q)\|_2. \tag{24}$$

Evaluating the cost for each element pair in the sequence of $z_i$ and $z_j$, one obtains the cost matrix $C \in \mathbb{R}_{\geq 0}^{W \times Q}$, defined by $C(w, q) := c(z_i(w), z_j(q))$. Then the goal is to find an alignment path based on $C$ with minimal overall cost.

The next definition formalizes the notion of such an alignment path.

*Definition 1:* The $(W, Q)$-warping path is a sequence $\pi := (\pi(1), \ldots, \pi(L))$ with $\pi(l) = (w_l, q_l) \in \{1, 2, \ldots, W\} \times \{1, 2, \ldots, Q\}$ for $l \in \{1, 2, \ldots, L\}$, where $L$ at least $\max(W, Q)$, satisfying the following three conditions:

1) *Boundary conditions:* $\pi(1) = (1, 1)$ and $\pi(L) = (W, Q)$
2) *Monotonicity conditions:* $w_1 \leq w_2 \leq \ldots \leq w_L$ and $q_1 \leq q_2 \leq \ldots \leq q_L$
3) *step-size condition:* $\pi_{l+1} - \pi_l \in \{(1, 0), (0, 1), (1, 1)\}$ for $l \in \{1, 2, \ldots, L-1\}$

The $(W, Q)$-warping path $\pi = (\pi(1), \ldots, \pi(L))$ defines an alignment between two sequences $z_i = (z_i(1), \ldots, z_i(W))$ and $z_j = (z_j(1), \ldots, z_j(Q))$ by assigning the element $z_i(w_l)$ to the element $z_j(q_l)$. The boundary condition enforces that the first elements (and last elements) of both sequences are aligned with each other. The monotonicity and step-size conditions ensure that no samples are omitted during alignment and no repetitions occur.

The total cost of a warping path $\pi$ between $z_i$ and $z_j$ with respect to the local distance measure $c$ is defined as:

$$c_\pi(z_i, z_j) := \sum_{l=1}^{L} c(z_i(w_l), z_j(q_l))). \tag{25}$$

Furthermore, the optimal warping path $\pi^*$ has minimal total distance among the set of all possible warping paths $\mathcal{A}$. The DTW distance $d_{\mathrm{DTW}}$ is defined as the total distance of the optimal warping path $\pi^*$:

$$\begin{aligned} d_{\mathrm{DTW}}(x_i, x_j) :&= c_{\pi^*}(z_i, z_j) \\ &= \min_{\pi \in \mathcal{A}} c_\pi(z_i, z_j), \end{aligned} \tag{26}$$

where $\mathcal{A}$ is the admissible set of warping paths. The DTW distance as defined in (26) is the distance between two multivariate time-series signals based on the warping of a selected univariate time series using $z_i := x_i^\kappa$.

### B. AGGLOMERATIVE HIERARCHICAL CLUSTERING

The algorithm of the agglomerative hierarchical clustering method with the complete linkage method is shown below.

**Algorithm 1:** Agglomerative Hierarchical Clustering With Complete Linkage.

```
Inputs:
    The amount of clusters $K$
    All instances in the training set $\boldsymbol{x}$
Output:
    Clusters $\boldsymbol{c} := \{c_1, \ldots, c_K\}$
Initialization:
    $R_0 = \{c_i = x_i | i = 1, 2, \ldots, m\}$
    $t = 1$
while $t < m - K$ do
    $\arg \min_{c_r, c_s \in R_{t-1}} \sigma(c_r, c_s)$ with
     $\sigma(c_r, c_s) := \max_{u \in c_r,\, v \in c_s} d_{\mathrm{DTW}}(u, v)$
    define $c_G := c_r \cup c_s$
    $R_t = (R_{t-1} - \{c_r, c_s\}) \cup c_G$
    $t = t + 1$
end
$\boldsymbol{c} = R_t$
```

## ACKNOWLEDGMENT

The authors wish to thank Francesco Mojoli and Tom Bakkes for giving access to the dataset of 15 patients from the Fondazione I.R.C.C.S. Policlinco San Matteo (reference number 41223). Furthermore, the authors wish to thank Dolf Weller and Corstiaan den Uil for giving access to the dataset of 20 patients from the Maasstad Hospital Rotterdam.

## REFERENCES

[1] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*. 2015, pp. 427–436.

[2] K. Lee, K. Lee, H. Lee, and J. Shin, “A simple unified framework for detecting out-of-distribution samples and adversarial attacks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7167–7177.

[3] W. Liu, J. D. Owens, and X. Wang, “Energy-based out-of-distribution detection,” in *Proc. 34th Int. Conf. Neural Inf. Process. Syst.*, 2020, pp. 21464–21475.

[4] S. Liang, Y. Li, and R. Srikant, “Enhancing the reliability of out-of-distribution image detection in neural networks,” in *Proc. 6th Int. Conf. Learn. Representations*, 2018, pp. 1–15.

[5] J. Yang, K. Zhou, Y. Li, and Z. Liu, “Generalized out-of-distribution detection: A survey,” *Int. J. Comput. Vis.*, vol. 132, no. 12, pp. 5635–5662, 2024.

[6] T. Belkhouja, Y. Yan, and J. R. Doppa, “Out-of-distribution detection in time-series domain: A novel seasonal ratio scoring approach,” *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 1, pp. 1–24, 2023.

[7] W. Lu et al., “Diversify: A general framework for time series out-of-distribution detection and generalization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 6, pp. 4534–4550, Jun. 2024.

[8] V. Sehwag, M. Chiang, and P. Mittal, “SSD: A unified framework for self-supervised outlier detection,” in *Proc. ICLR 2021-9th Int. Conf. Learn. Representations*, 2021, pp. 1–17.

[9] P. Sinhamahapatra, R. Koner, K. Roscher, and S. Günnemann, “Is it all a cluster game? - Exploring Out-of-Distribution Detection based on Clustering in the Embedding Space,” in *Proc. CEUR Workshop Proc.*, vol. 3087, 2022.

[10] L. Blanch et al., “Asynchronies during mechanical ventilation are associated with mortality,” *Intensive Care Med.*, vol. 41, no. 4, pp. 633–641, 2015.

[11] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–50, Feb. 1978.

[12] X. Ran, Y. Xi, Y. Lu, X. Wang, and Z. Lu, *Comprehensive Survey on Hierarchical Clustering Algorithms and the Recent Developments*, vol. 56. Berlin, Germany: Springer the Netherlands, 2023.

[13] M. A. Holanda, R. d. S. Vasconcelos, J. C. Ferreira, and B. V. Pinheiro, “Patient-ventilator asynchrony,” *Jornal Brasileiro de Pneumologia*, vol. 44, pp. 321–333, Jul. 2018.

[14] S. K. Epstein, “How often does patient-ventilator asynchrony occur and what are the consequences?,” *Respir. Care*, vol. 56, no. 1, pp. 25–35, 2011.

[15] T. Bakkes et al., “Automated detection and classification of patient–ventilator asynchrony by means of machine learning and simulated data,” *Comput. Methods Programs Biomed.*, vol. 230, 2023, Art. no. 107333.

[16] L. van de Kamp, J. Reinders, B. Hunnekens, T. Oomen, and N. van de Wouw, “Automatic patient-ventilator asynchrony detection framework using objective asynchrony definitions,” *IFAC J. Syst. Control*, vol. 27, 2024, Art. no. 100236.

[17] A. van Diepen et al., “A model-based approach to generating annotated pressure support waveforms,” *J. Clin. Monit. Comput.*, vol. 36, no. 6, pp. 1739–1752, Mar. 2021.

**L. VAN DE KAMP** received the B.Sc. degree in mechanical engineering and the M.Sc. degree in mechanical engineering (cum laude), in 2019 and 2021, respectively, from the Eindhoven University of Technology, Eindhoven, the Netherlands, where he is currently working toward the Ph.D. degree in mechanical engineering with the Dynamics and Control group. His M.Sc. graduation work on “Patient-ventilator asynchrony detection and classification within mechanical ventilation” was awarded the M.Sc. thesis award (Mechanical Engineering).

His research interests include system identification, interpretable data-driven and machine learning techniques, and control, with application to mechanical ventilation.

**B. HUNNEKENS** (Member, IEEE) received the B.Sc. and M.Sc degrees (cum laude) in mechanical engineering from the Eindhoven University of Technology, Eindhoven, the Netherlands, in 2008 and 2011, respectively, and the Ph.D. degree in mechanical engineering for his thesis “Performance optimization of hybrid controllers for linear motion systems” in 2015. In 2016, he was the recipient of the DISC “Best Thesis Award.” He is currently with Demcon as a System Engineer. His research interests include nonlinear control, performance, high-tech systems, medical systems, and mechanical ventilation.

**T. OOMEN** (Senior Member, IEEE) received the M.Sc. degree (cum laude) and Ph.D. degree from the Eindhoven University of Technology, Eindhoven, the Netherlands.

He is currently a Full Professor with the Department of Mechanical Engineering of the Eindhoven University of Technology. He is also a part-time Full Professor with the Delft University of Technology. He held visiting positions at KTH, Stockholm, Sweden, and with The University of Newcastle, Australia. He was the recipient of the 7th Grand Nagamori Award, the Corus Young Talent Graduation Award, the IFAC 2019 TC 4.2 Mechatronics Young Research Award, 2015 IEEE Transactions on Control Systems Technology Outstanding Paper Award, 2017 IFAC Mechatronics Best Paper Award, 2019 IEEJ Journal of Industry Applications Best Paper Award, and the recipient of a Veni and Vidi personal grant. He is currently a Senior Editor of IEEE Control Systems Letters (L-CSS) and Associate Editor for *IFAC Mechatronics*, and he was on the Editorial Boards of the IEEE Control Systems Letters (L-CSS) and IEEE Transactions on Control Systems Technology. He has also been vice-chair for IFAC TC 4.2 and a member of the Eindhoven Young Academy of Engineering. His research interests include the field of data-driven modeling, learning, and control, with applications in precision mechatronics.

**N. VAN DE WOUW** (Fellow, IEEE) received the M.Sc. (Hons.) degree and Ph.D. degree in mechanical engineering from the Eindhoven University of Technology, Eindhoven, the Netherlands, in 1994 and 1999, respectively.

He was with Philips Applied Technologies, the Netherlands, in 2000 and with the Netherlands Organisation for Applied Scientific Research, the Netherlands, in 2001. He has been a Visiting Professor with the University of California Santa Barbara, CA, USA, in 2006 and 2007, with the University of Melbourne, Australia, in 2009 and 2010 and with the University of Minnesota, Minneapolis, MN, USA, in 2012 and 2013. He was a (part-time) Full Professor with the Delft University of Technology, Delft, the Netherlands, from 2015 to 2019. He was also an Adjunct Full Professor with the University of Minnesota, USA, from 2014 to 2021. He is currently a Full Professor with the Mechanical Engineering Department of the Eindhoven University of Technology. He has authored or coauthored the books *Uniform Output Regulation of Nonlinear Systems: A Convergent Dynamics Approach* with A.V. Pavlov and H. Nijmeijer (Birkhauser, 2005) and *Stability and Convergence of Mechanical Systems with Unilateral Constraints* with R.I. Leine (Springer-Verlag, 2008).

In 2015, he was the recipient of the IEEE Control Systems Technology Award "for the development and application of variable-gain control techniques for high performance motion systems." He has contributed to hybrid, data-based and networked control.