POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Nonparametric Statistics

Ventilator Pressure Prediction

Authors: Manuel Bressan, Leonardo Perelli, Sebastiano Rossi, Daniela Zanotti

Professors: Francesca Ieva, Simone Vantini, Andrea Cappozzo, Matteo Fontana

Academic year: 2021-2022

1. Introduction

Our project stems from a Kaggle Competition organized by Google Brain [4] where the dataset is composed by numerous time series of breaths produced using a ventilator connected to an artificial lung. The objective of this project is to predict the airway pressure in the respiratory circuit during the inspiratory phase of the breath, given the time series of control inputs and other auxiliary variables.

2. Background

What do doctors do when a patient has trouble breathing?
They use a ventilator to pump oxygen into a sedated patient's lungs via a tube in the windpipe. But mechanical ventilation is a clinician-intensive procedure, a limitation that was prominently on display during the early days of the COVID-19 pandemic. At the same time, developing new methods for controlling mechanical ventilators is prohibitively expensive, even before reaching clinical trials: this is why high-quality simulators could reduce this barrier.
The ventilators we deal with are machines composed by valves which control inspiratory and expiratory flows, and a pressure sensor monitors airway pressure.
In particular, the diagram below (Figure 1) illustrates the setup, with the two control inputs highlighted in light blue and the state variable (airway pressure) to predict in blue.
The first control input is a continuous variable from 0 to 100 representing the percentage of the inspiratory solenoid valve that is open to let air into the lung (i.e., 0 is completely closed and no air is let in and 100 is completely open).
The second control input is a binary variable representing whether the expiratory valve is open (1) or closed (0) to let air out.

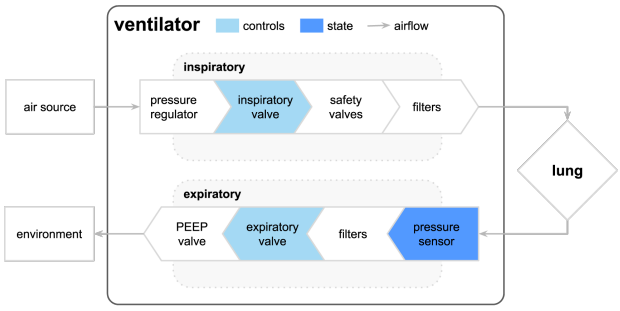

Figure 1: Ventilator setup

We are given numerous time series of breaths and we want to predict the airway pressure in the respiratory circuit during the inspiratory phase of the breath, given the time series of control inputs.

3. Dataset

The dataset is composed by around 70000 breaths characterized by the following covariates:

- **id** - globally-unique time step identifier across an entire file (80 per breath);
- **breath_id** - globally-unique identifier for breaths (around 70000);
- **R** - lung attribute indicating how restricted the airway is (in cmH2O/L/S).
  Physically, this is the change in pressure per change in flow (air volume per time). Intuitively, one can imagine blowing up a balloon through a straw. We can change R by changing the diameter of the straw, with higher R being harder to blow.
- **C** - lung attribute indicating how compliant the lung is (in mL/cmH2O).
  Physically, this is the change in volume per change in pressure. Intuitively, one can imagine the same balloon example. We can change C by changing the thickness of the balloon's latex, with higher C having thinner latex and easier to blow.
- **time_step** - the actual time stamp.
- **u_in** - the control input for the inspiratory solenoid valve. Ranges from 0 to 100.
- **u_out** - the control input for the exploratory solenoid valve. Either 0 or 1.

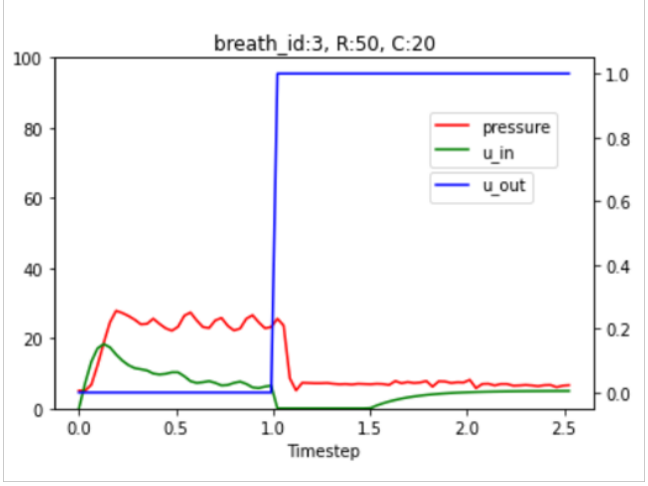

Figure 2: Example of a single breath

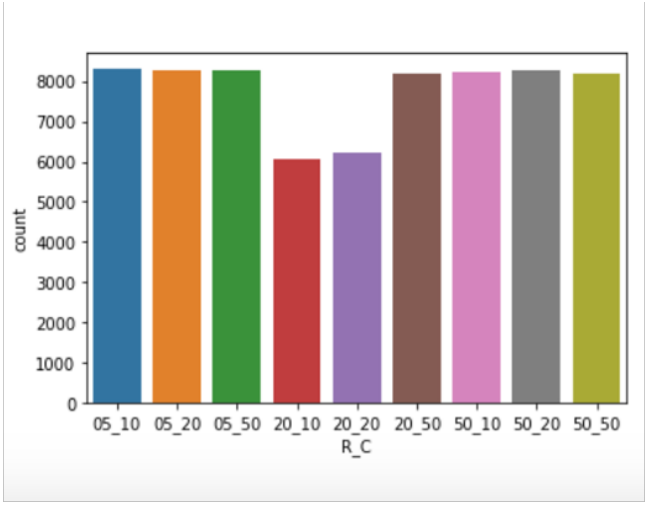

Figure 3: Distribution of different configurations

Each breath is characterized by 80 time stamps, of which only the first 30 represent the inspiratory phase, which is of our interest. Moreover both u_in and pressure go to zero in the expiratory phase: as a consequence we reduce the dimension of our dataset, considering only the first 30 instances of each breath.

4. Approach

4.1. Feature engineering

In order to obtain a good prediction of the pressure we need to take into account various aspects of the breath, and since the dataset contains only a few attributes, we created new covariates.

- **R_C** - factor variable indicating levels of R and C;
- **tot_u_in** - cumulative sum of u_in up to time t;
- **last_u_in** - last value of the u_in of each breath;
- **first_u_in** - first value of the u_in of each breath;
- **max_u_in** - maximum value of the u_in of each breath;
- **u_in_diff1** - difference between two consecutive values of u_in;
- **u_in_diff2** - difference between the current value of u_in and the value of u_in 2 time steps before;
- **u_in_diff3** - difference between the current value of u_in and the value of u_in 3 time steps before;
- **u_in_diff4** - difference between the cur-

rent value of u_in and the value of u_in 4 time steps before;
- **u_in_diff5** - difference between the current value of u_in and the value of u_in 5 time steps before;
- **u_in_diff_max** - difference between the current value of u_in and the maximum value within the breath;
- **n_change_sign_u_in** - number of times the u_in of a breath changes direction;
- **area** - area below the curve of u_in between two consecutive time step;
- **u_in_shift1** - u_in shifted of 1 position;
- **u_in_shift2** - u_in shifted of 2 positions;
- **u_in_shift3** - u_in shifted of 3 positions;
- **u_in_shift4** - u_in shifted of 4 positions;
- **u_in_shift5** - u_in shifted of 5 positions;

As we will see, most of this variables will turn out to be significant.

4.2. Clustering

Before starting our analysis, we split our dataset in training and test set: the latter one is created sampling the 20% of the breaths. We do this in order to check the quality of our predictions at the end of the work.

To explore more the data and capture some pattern, we apply functional clustering on u_in. The group to which each breath belongs could also be of interest to understand the shape of the pressure.
Using the package *fdakma* studied in the Applied Statistics course is not an option since it is very slow when dealing with large dataset, so we rely on the *fdakmapp* package [1], a more efficient implementation of the same algorithm. The function divides our data in k subgroups, searching for the best k clusters that minimize the within-cluster variability, while assigning each of the data to the nearest cluster.

4.2.1. Creation of the clusters

The first thing to select is the similarity method, i.e. specify how we want to compare our curves. We want to give importance to the different shapes of our breaths rather than their magnitude, for this reason we use Pearson similarity, that is the cosine of the angle between two functions:

$$\frac{||f-g||_2^2}{||f||_2^2||g||_2^2}$$

The second parameter to select is the warping method, indicating the allowed transformation for the abscissa. We set it equal to 'dilation', meaning that the abscissa can only be multiplied by a coefficient, and shifting is not allowed: x_final = dilation*x. Indeed in our study both amplitude variation and phase variation are important; for example, we do not only care of having a peak at the beginning of the breath, we want to know if it happens at 0.1 s or at 0.3 s.

In Figure 4 we show 4 breaths, the blue curves are clustered together even if they have different magnitude, since they have the same shape and the peaks are in almost the same time stamp. On the other hand the green curves are grouped in another cluster because their peaks are shifted on the right.

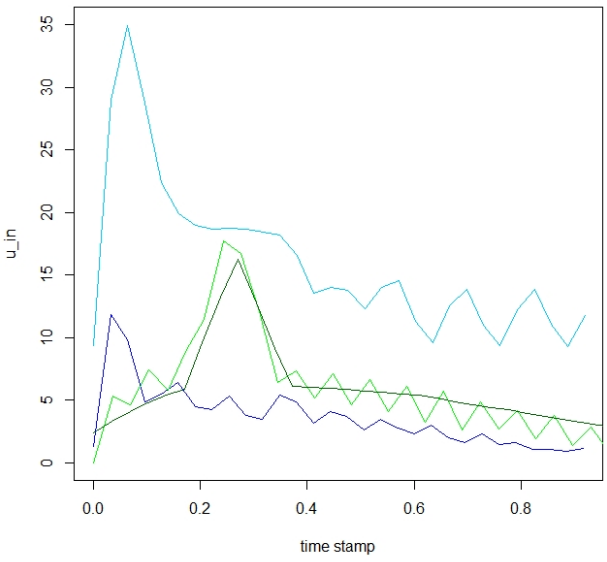

Figure 4: Example clustering

The last important parameter to set is the number of clusters. Simply exploring our data we are not able of making an idea about what can be the right number of groups, so we start by making a silhouette analysis.

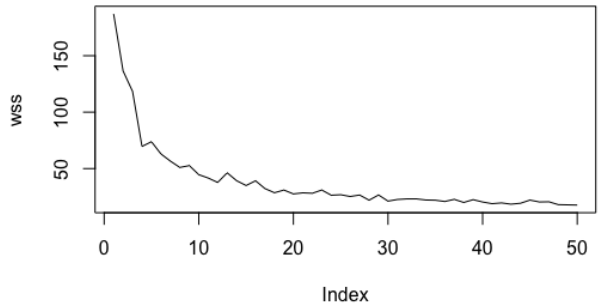

Figure 5: WSS plot

Looking at Figure 5, the right number of clusters is not so clear, the elbow seems to be around 10∼15. However since our goal is not the clustering per se, we do not focus on finding the perfect amount of groups but on extracting as much patterns as possible from the data. For this reason we set the number of clusters equal to 50: even if the number is high, many clusters present different shapes and the most similar ones will be aggregated in the next steps, moreover almost all the groups contain a high number of breaths.

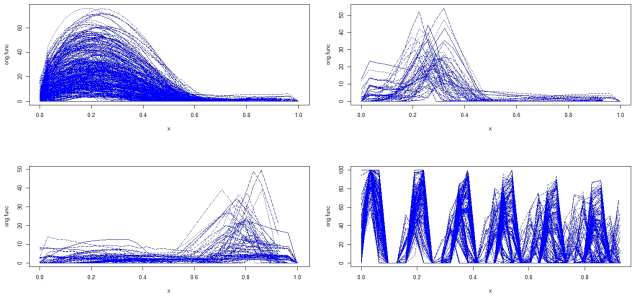

Figure 6: Example of good clusters

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1691	730	1331	126	872	1688	419	1386	284	1547	1914	588	2184	829	2803	568	1907
18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
2088	804	1032	525	1831	1556	2125	529	876	357	936	1932	595	1348	302	1061	2799
35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	
2555	1510	69	108	1400	870	1070	1132	1490	332	1065	559	2410	673	323	3231	

Figure 7: Numerosity of the clusters

4.2.2. Aggregation

Performing a visual analysis of the clusters it seems that some of them have very similar shapes, but given the high number of groups generated this do not surprise us; an example is shown in (Figure 8).

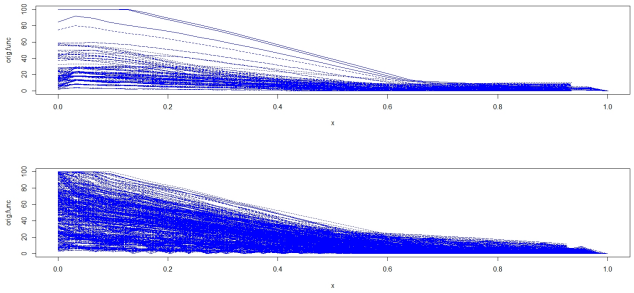

Figure 8: Example of similar clusters

To check if redundant groups exist, we do Interval Wise Testing, that is an inferential procedure for functional data able to select the intervals of the domain imputable of rejecting a functional null hypothesis. [7]
For each interval I=(t_1, t_2) p^I is defined as the p-value of the functional test:

$$H_0 : \mathbf{Y}_1{}^I = \mathbf{Y}_2{}^I \; vs. \; H_1 : \mathbf{Y}_1{}^I \neq \mathbf{Y}_2{}^I$$

with $\mathbf{Y}_i$ restricted to I.
For any t in T we define the IWT-adjusted p-value function as:

$$\tilde{p}(t) = \sup_{I \ni t} p^I$$

We apply Interval Wise Testing to all the possible pairs of clusters and search for the ones with adjusted p-value > 0.05 for more than 70% of the time. In Figure 9 and Figure 10 there are example of groups that should be merged and groups that are significantly different with the plot of the respectively p-values.

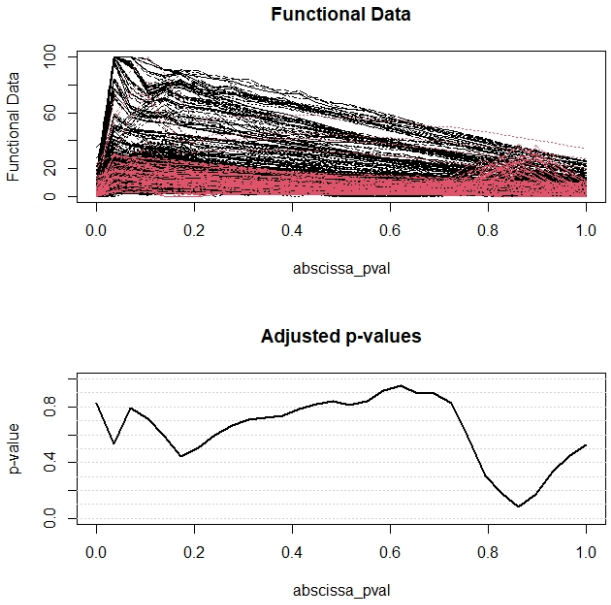

Figure 9: Example of similar clusters

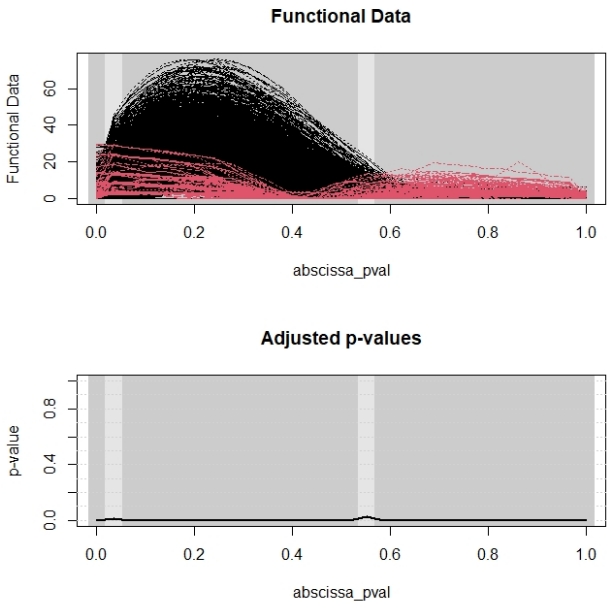

Figure 10: Example of different clusters

The tests return 8 pairs of cluster for which H_0 cannot be rejected. At this point all this groups are not simply merged, instead the following steps are iteratively repeated:

- compare the two clusters with greater number of time stamp in which p-value > 0.05 repeating the Interval Wise test;
- merge them if there is evidence to do it.

At the end of this merging procedure we remain with 46 clusters.

4.2.3. Medians

The cluster to which a breath belongs will be a crucial information for the prediction of the pressure, so a way of assigning new observation to a group is needed. For this purpose we characterize each cluster with his sample median based on a depth measure and label the new data with the class corresponding to the most similar median with respect to the Pearson similarity. However, in order to prevent problems during the prediction, we check only the medians corresponding to groups in which the value of R_C of the new breath is present, indeed it may happen that the data has a new level of the variables.
The medians are the deepest element of the sample and they are computed using Modified Band Depth:

$$MBD_{n,J}(x) = \sum_{j=2}^{J} MBD_n^{(j)}(x) =$$

$$= \sum_{j=2}^{J} \binom{n}{j}^{-1} \sum_{1 \le i_1 < i_2 < \ldots < i_j \le n} \bar{\lambda}\{E(x, x_{i_1}, \ldots, x_{i_j})\}$$

where

$$E(x, x_{i_1}, .., x_{i_j}) = \{t \in I, m \le x(t) \le M\}$$

$$m = \min_{r=i_1,..,i_j} x_r(t) \qquad M = \max_{r=i_1,..,i_j} x_r(t)$$

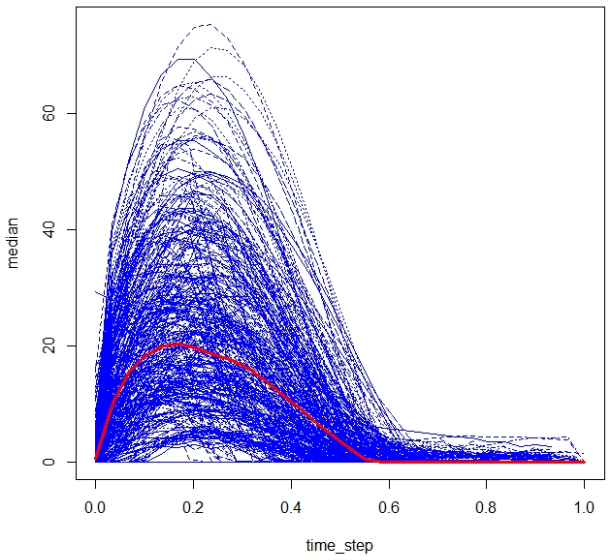

Figure 11: Example of median curve

4.3. Models

The main goal of our project is to provide a prediction of the pressure, thus it is necessary to build a predictive model. To do this the clustering structure created before is exploited.

4.3.1. First attempt: GAMM

At first we work with GAMM, using the clusters only as covariates.
Generalized Additive Models deal with functions of the form

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \epsilon_i$$

and allow nonlinear functions of each predictor while mantaining additivity.
Mixed-effects GAM are a specific type of GAM involving two components:

- fixed effects are variables whose levels are defined and that affect the whole dataset;
- random effects account for subject-specific effects.

Since the measurements of the same breath are correlated, we account for such correlation including random effects in our predictor. We build two models which differ for the random effects considered and that respectively are:

- breath's id;
- breath's id and cluster.

Both models are built with the following fixed effects:

- **time_step**;
- **u_in**;
- **tot_u_in**;
- all the **u_in_diff**
- all the **u_in_shift**

Regarding the fixed effects' part the GAMMs reach $R^2 = 0.764$, it the same for both since they rely on the same covariates. Comparing them using degrees of freedom, AIC and BIC seems that the second model is slightly better. (Figure 12)

	df	AIC	BIC
m2$mer	24	58578.42	58754.16
m1$mer	23	58594.21	58762.64

Figure 12: Comparison of GAMMs

Analyzing the residuals of the model (Figure 15), one can clearly see that the gaussianity assumption is violated and the residuals present a strange pattern.

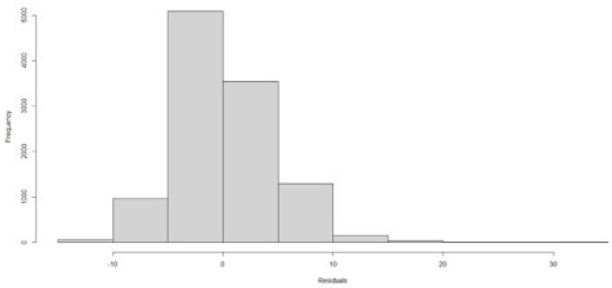

Figure 13: Barplot of GAMM's residuals

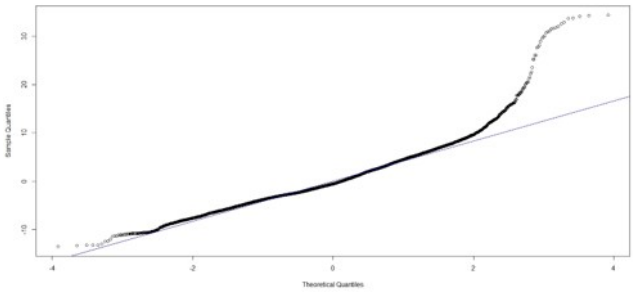

Figure 14: QQplot of GAMM's residuals

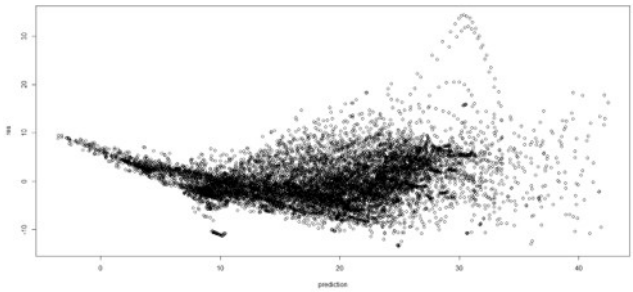

Figure 15: Scatterplot of GAMM's residuals

However this is not the only problem of this approach, the models are trained using only a sample of ∼600 breaths and trying to increase the number of breaths requires a lot of computational resources, not manageable both from a time and a memory point of view.

4.3.2. Second attempt: GAM

Given the difficulties encountered, we change approach and build a GAM for each cluster,

obtaining 46 models.

The covariates that turn out to be significant in our models are:

- **time_step**
- smoothed **u_in**
- factor **R_C**
- smoothed **tot_u_in**
- smoothed **u_in_diff1**
- smoothed **u_in_diff2**
- smoothed **u_in_diff3**
- smoothed **u_in_diff4**
- smoothed **u_in_diff5**
- smoothed **area**
- smoothed **u_in_shift1**
- smoothed **u_in_shift2**
- smoothed **u_in_shift3**
- smoothed **u_in_shift4**
- smoothed **u_in_shift5**

The models perform well: the R^2 ranges from 0.723 to 0.979, with a weighted mean of 0.849. Unfortunately, we still have some problems with the residuals, since they are not gaussian distributed, but they don't present anymore a strange pattern.
Consider one of the models as example. GAMs let us analyze the effect of each covariate on the response, holding all the other covariates fixed: in Figure 16 - 21 is shown the effect of some variables on the pressure.

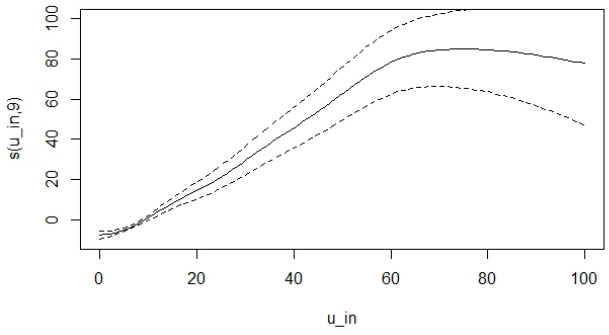

Figure 16: Effect of u_in on pressure

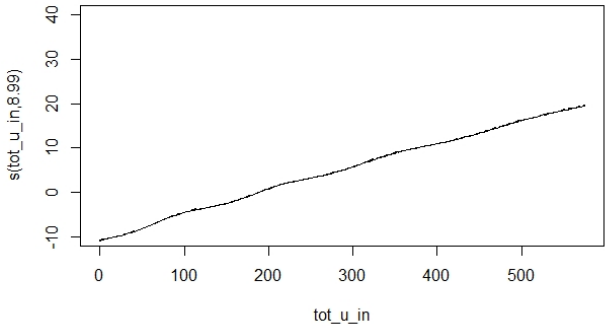

Figure 17: Effect of tot_u_in on pressure

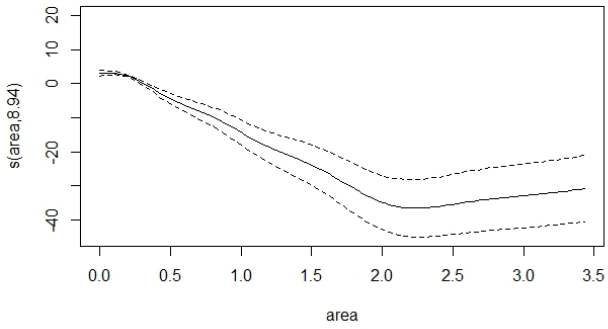

Figure 18: Effect of area on pressure

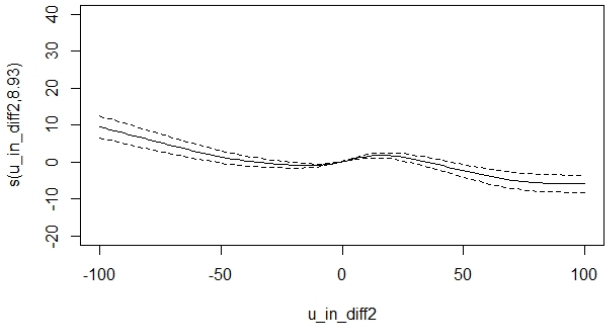

Figure 19: Effect of u_in_diff2 on pressure

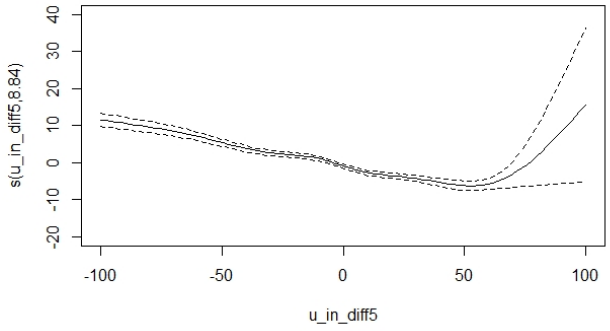

Figure 20: Effect of u_in_diff5 on pressure

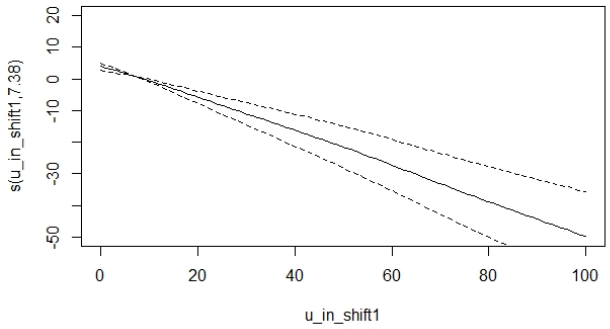

Figure 21: Effect of u_in_shift1 on pressure

We plot also the residuals (Figure 22 and 23). As we can see they are not particularly good and could be improved in an eventual future development of the project.

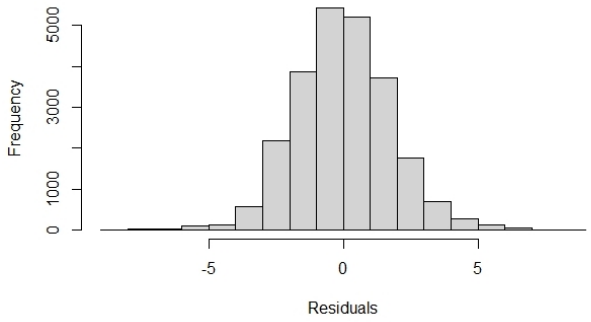

Figure 22: Barplot of a model's residuals

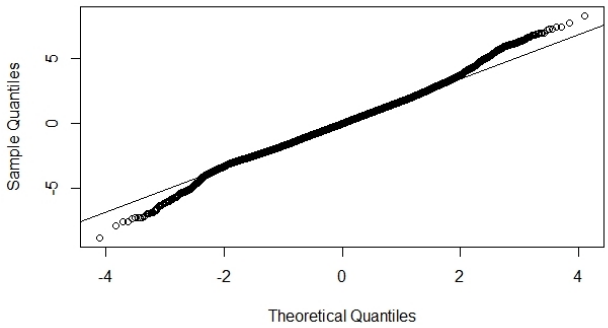

Figure 23: QQplot of a model's residuals

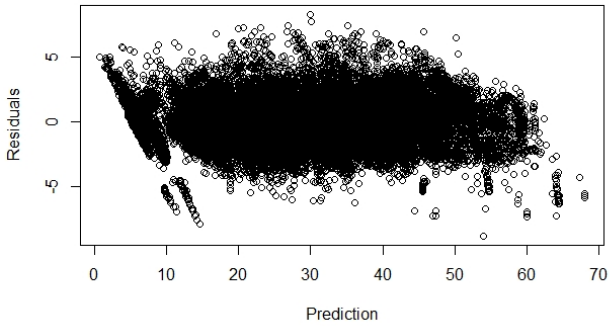

Figure 24: Scatterplot of a model's residuals

4.4. Conformal prediction

At this point of our work, with all the models generated, there is only one last thing to do, the goal of the project: predict the pressure. However we do not want to confine ourselves to the simple prediction, we search also for a region in which the curves will fall with a certain confidence.

This can be obtained with Conformal Prediction, a nonparametric approach that generates prediction sets under the only assumption of exchangeable regression pairs. Given a set of *i.i.d.* random functions $\mathbf{Y}_1, ..., \mathbf{Y}_n \sim P$ and an independent random function $\mathbf{Y}_{n+1} \sim P$, a valid prediction set $C_{n,1-\alpha} := C_{n,1-\alpha}(\mathbf{Y}_1, ..., \mathbf{Y}_n)$ for $\mathbf{Y}_{n+1}$ is such that:

$$\mathbb{P}(\mathbf{Y}_{n+1} \in C_{n,1-\alpha}) \geq 1 - \alpha$$

for any significance level $\alpha \in (0, 1)$, with $\mathbb{P}$ the probability corresponding to the product measure induced by P.

In particular we will apply Split Conformal: given data $y_1, ..., y_n$, $\{1,...,n\}$ is randomly

divided into two sets I_1, I_2, the training set is defined as $\{y_h : h \in I_1\}$ while the calibration set is $\{y_h : h \in I_2\}$.

Following the approach presented by [5] we adopt the same nonconformity measure and modulation function. We want prediction bands that adapt their width according to the local variability of functional data, so the nonconformity measure used is:

$$A(\{y_h : h \in \mathcal{I}_1\}, y) = \sup_{t \in \mathcal{T}} \left| \frac{y(t) - g_{\mathcal{I}_1}(t)}{s_{\mathcal{I}_1}(t)} \right|$$

and the nonconformity scores are:

$$R_j^s := \sup_{t \in \mathcal{T}} \left| \frac{y_j(t) - g_{\mathcal{I}_1}(t)}{s_{\mathcal{I}_1}(t)} \right|$$

$$R_{n+1}^s := \sup_{t \in \mathcal{T}} \left| \frac{y(t) - g_{\mathcal{I}_1}(t)}{s_{\mathcal{I}_1}(t)} \right|$$

with $j \in \mathcal{I}_2$ and $s_{\mathcal{I}_1} := s(\{y_h : h \in \mathcal{I}_1\})$.
The split conformal prediction band induced by the nonconformity measure is:

$$C_{n,1-\alpha}^s := \{y \in \mathcal{Y}(\mathcal{T}) : y(t) \in [g_{\mathcal{I}_1}(t) \pm k^s s_{\mathcal{I}_1}(t)] \forall t \in \mathcal{T}\}$$

with k^s the $\lceil (l+1)(1-\alpha) \rceil$th smallest value in $\{R_h^s : h \in \mathcal{I}_2\}$.
In our application $\alpha = 0.1$, $g_{\mathcal{I}_1}(t)$ is defined as the prediction obtained by the GAM and the modulation function is

$$s_{\mathcal{I}_1}(t) = \frac{\max_{j \in \mathcal{H}_2} \left| y_j(t) - g_{\mathcal{I}_1}(t) \right|}{\int_{\mathcal{T}} \max_{j \in \mathcal{H}_2} \left| y_j(t) - g_{\mathcal{I}_1}(t) \right| dt}$$

with

$$\mathcal{H}_2 := \{j \in \mathcal{I}_2 : \sup_{t \in \mathcal{T}} \left| y_j(t) - g_{\mathcal{I}_1}(t) \right| \leq k\}$$

.
In Figure 25 - 28 are reported a couple of examples of the results obtained, the black line is the predicted pressure, the red one is the true pressure and it has been plotted to show that lays inside the band.

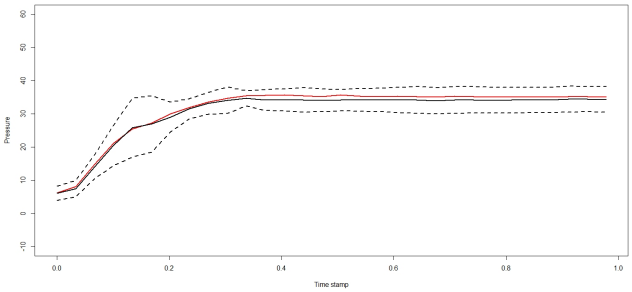

Figure 25: Prediction of a new breath

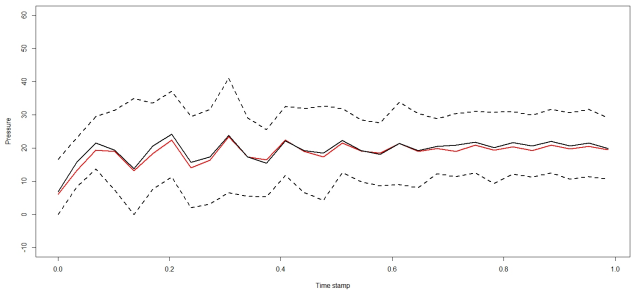

Figure 26: Prediction of a new breath

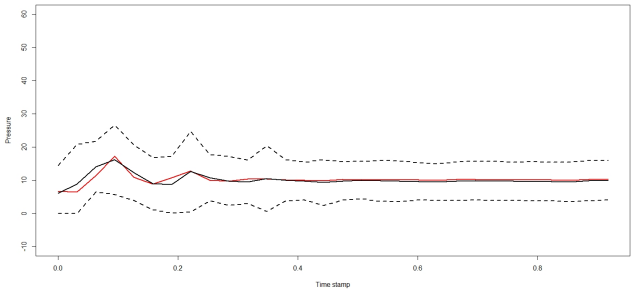

Figure 27: Prediction of a new breath

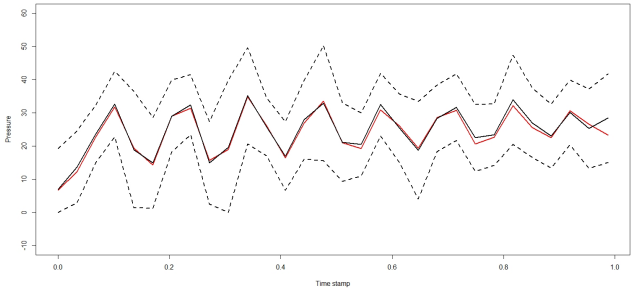

Figure 28: Prediction of a new breath

5. Conclusions

This project is finalized at predicting the airway pressure in the respiratory circuit during the inspiratory phase of a breath given the lung's characteristics and some control variables. In this report we explained the procedure that leads to obtain such predictions, making us able to simulate the behaviour of a lung under different conditions.
What we are doing is simulating a ventilator connected to a sedated patient's lung, and so that can generalize across lungs with varying characteristics. This would allow for new ways of controlling ventilators to be developed without having a high entry price and without them

being tested inside a clinical trial. In such way, when a new procedure is developed and need to be tested, our simulator can be used: given new breath and lung's characteristics, the machine will have to extract all the needed feature from the control variables and identify at which cluster the breath belongs, comparing the u_in to the medians of the group and choosing the most similar one. Then the pressure will simply be predicted using the model corresponding to the selected cluster and will represent the behaviour of the lung. In such way it can be understood if the ventilator works well or has dangerous side effects on the lungs. Moreover, the conformal prediction bands provide additional information of the effects on the lungs, if some critical values that are dangerous for the patient rely in the bands, the ventilator should be adjusted.
The project can be further developed, for example another way of identifying a cluster and consequently of assigning a new breath to a group can be found, moreover one can focus on the residuals of GAM, trying to make them gaussian distributed.
We also believe that in many cases the approach presented in the top solutions in Kaggle is inherently wrong since they reverse engineered the data presented to reconstruct how they were produced. This led to almost perfect solutions that unfortunately missed the focal point of the challenge, that is to embody also the lung characteristics in the model.
Nonetheless we are satisfied with the results obtained, since the predictions are quite good as we obtain an $R^2 = 0.867$ on the validation set.

References

[1] Zito A. Fdakmapp: Optimization of the k-mean alignment algorithm.

[2] Don Hedeker. Introduction to mixed models for longitudinal continuous data.

[3] Osio G. C.; Jardim E.; Minto C.; Scott F. ; Patterson K. Model based cfp indicators, f/fmsy and ssb. Technical report, European Commission - Joint Research Centre, 2015.

[4] Kaggle. Google brain - ventilator pressure prediction.

[5] Diquigiovanni J.; Fontana M.; Vantini S. The importance of being a band: Finite-sample exact distribution-free prediction sets for functional data. Technical report, 12 April 2021. arXiv:2102.06746v2.

[6] Diquigiovanni J.; Fontana M.; Vantini S. Conformal prediction bands for multivariate functional data. Technical report, 3 June 2021. arXiv:2106.01792v1.

[7] Pini A.; Vantini S. Interval-wise testing for functional data. Technical report, MOX– Department of Mathematics, Politecnico di Milano.