

Spirometry-based airways disease simulation and recognition using Machine Learning approaches

Riccardo Di Dio, André Galligo, Angelos Mantzaflaris, Benjamin Mauroy

▶ **To cite this version:**

Riccardo Di Dio, André Galligo, Angelos Mantzaflaris, Benjamin Mauroy. Spirometry-based airways disease simulation and recognition using Machine Learning approaches. LION 2021 - 15th Learning and Intelligent Optimization Conference, Jun 2021, Athens, Greece. pp.98-112, 10.1007/978-3-030-92121-7_8 . hal-03326950v2

HAL Id: hal-03326950

https://inria.hal.science/hal-03326950v2

Submitted on 5 Nov 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Spirometry-based airways disease simulation and recognition using Machine Learning approaches

Riccardo Di Dio[1,2], André Galligo[1,2], Angelos Mantzaflaris[1,2], and Benjamin Mauroy[2]

[1]Université Côte d'Azur, Inria, France
[2]Université Côte d'Azur, CNRS, LJAD, VADER Center, France

Abstract. The purpose of this study is to provide means to physicians for automated and fast recognition of airways diseases. In this work, we mainly focus on measures that can be easily recorded using a spirometer. The signals used in this framework are simulated using the linear bi-compartment model of the lungs. This allows us to simulate ventilation under the hypothesis of ventilation at rest (tidal breathing). By changing the resistive and elastic parameters, data samples are realized simulating healthy, fibrosis and asthma breathing. On this synthetic data, different machine learning models are tested and their performance is assessed. All but the Naive bias classifier show accuracy of at least 99%. This represents a proof of concept that Machine Learning can accurately differentiate diseases based on manufactured spirometry data. This paves the way for further developments on the topic, notably testing the model on real data.

Keywords: Lung disease · Machine Learning · Mathematical modeling

1 Introduction

Having a fast and reliable diagnosis is a key step for starting the right treatment on time; towards this goal, Machine Learning (ML) techniques constitute potential tools for providing more information to physicians in multiple areas of medicine. More specifically, as far as respiratory medicine is concerned, there is a recent blooming of publications regarding the investigations of Artificial Intelligence (AI), yet the majority of them refers to computer vision on thoracic X-Rays or MRI [8]. However, for lung diseases using Pulmonary Function Tests (PFTs) recent studies have only scratched the surface of their full potential, by coupling spirometry data with CT scans for investigating Chronic Obstructive Polmunary Diseases COPD on large datasets like COPDGene [2]. In our study, only normal ventilation is used allowing diagnosis also for children. Our aim is to provide a first proof-of-concept and provide the first positive results that could lead to fast, accurate and automated diagnosis of these diseases, similarly e.g. to Cystic Fibrosis (CF) where Sweat chloride test is a central asset [7].

Normally, ML models are trained and tested on data and labels are provided by medical doctors. However, the originality of this study consists in using mathematical equations to simulate the ventilation following the directives of IEEE

[9], then this data will be used to train the ML models. The obtained volume flows respect the expectations for both healthy and not healthy subjects. Using a synthetic model with a low number of parameters allows us to have everything under control.

During this study, the lungs are modeled as elastic balloons sealed in the chest wall and the airways are modeled as rigid pipes, this allows to play with few parameters to simulate healthy and not healthy subjects and create synthetic volumetric data of tidal breathing. This data is then split and used to train and test different ML models for diagnosis. The accuracies reached during the study are very high, however, the choice of the parameters and the restriction of synthetic data allowed for promising results. Further tests are needed with real data to validate the accuracy reached. Nevertheless, this study points out that not every classifier is suited for this task.

A brief introduction to human lungs and its physiology is given in 1.1, then section 1.2 shows how ventilation has been modeled. In section 2.1 is shown how the dataset has been realized, section 2.3 reflects the training of the models on the dataset and finally in section 3 and 4 the results are exposed and discussed.

1.1 Lung ventilation

The respiratory system can be split in two different areas, the bronchial tree (also referred as *conducting zone*) and the acini, the *respiratory zone*. The very beginning of the bronchial tree is composed of the trachea which is directly connected to the larynx, the mouth and the nose. The trachea can be seen as the *root* of our tree, which then ramifies into two bronchi that will split again and again until about 23 divisions [22]. During each division, the dimensions of the children are smaller compared to the parent, according to Weibel's model, the reduction factor between each split is around 0.79 [6, 11, 18, 12]. After the very first ramification, the two bronchi leads to the left and the right lung. Inside the lungs, the ventilation takes place. The lungs are inflated and deflated thanks to the respiratory muscles. Their role is to transport the air deep enough in the lung so that the gas exchanges between air and blood could occur efficiently.

Some diseases affect the physiological behavior of the bronchial tree and of the lungs. In asthma, a general shrinking of the bronchi happens and the patient feels a lack of breath due to the increased total resistance of the bronchial tree. In mechanical terms, the patient will need a greater muscular effort in order to provide adequate pressure for restoring a normal flow within the lungs.

In cystic fibrosis, there is an accumulation of mucus within the bronchial tree that will impact the capacity of the lungs to inflate and deflate, hence its rigidity. Normally, it is harder to breath for patients with cystic fibrosis because of the increased rigidity of their lungs.

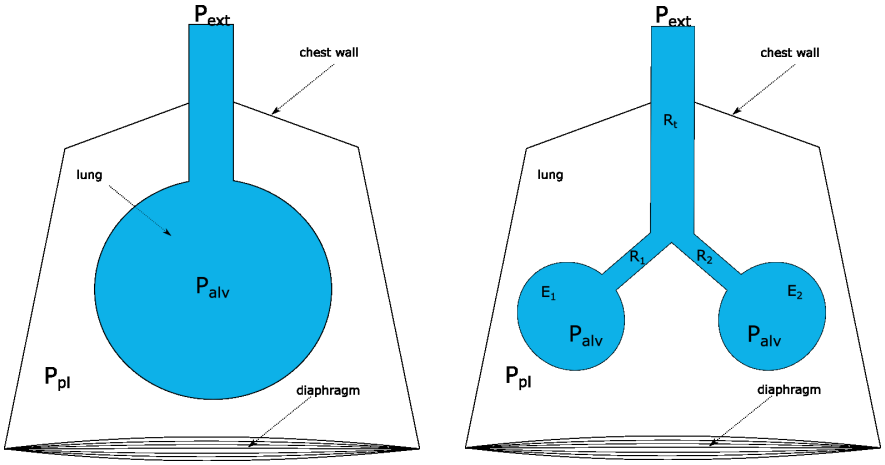
The lungs mechanical properties can be used to build a mathematical model that can mimic the respiratory system.

1.2 Mathematical modeling

The more tractable model to mimic the lung mechanics and ventilation is to mimic separately its resistive and elastic parts. We represent the resistive tree using a rigid tube with given length l and radius r. The resistance R of such a tube can be calculated by using Poiseuille's law that depends on the air viscosity μ_{air}[13].

$$R = \frac{8\mu_{\text{air}} l}{\pi r^4} \tag{1}$$

The elastic part of the lung can be mimicked with an elastic balloon with elastance property E. Figure 1a is a representation of such a model. However, for this study, the model used is slightly more sophisticated to get a better representation of the distribution of the ventilation, see figure 1b. The profile of the pressure used to mimic the muscular action is taken from L. Hao et al. [9] and represents a standard for tidal breathing. It is necessary to highlight that the hypotheses of linearity used in this model are respected in the regime of tidal breathing [17].



(a) Mono-compartment model (b) Bi-compartment model

Fig. 1: (a) Mono-compartment model of the lung. The Bronchial tree is collapsed in the tube having the total resistance R and the alveoli are mimicked by balloons characterized by their elastance E. (b) Parallel bi-compartment model. This model better respects the anatomy of the respiratory system.

The fundamental equation that links the resistance R and the elastance E to the Pressure P, Volume V and its derivative in time $\dot{V}$ can be easily derived from the Mono-compartment model:

$$P_{ext} - P_{alv} = \Delta P(t) = R\dot{V}(t) \tag{2}$$

$$P_{alv} - P_{pl} = P_{el}(t) = EV(t) \tag{3}$$

P_{el} represents the pressure drop between the acini and the pleural space, this depends on the elastance of the compartment. ΔP is the air pressure drop between the airways opening and the acini and it takes into account the resistance of the airways and the parenchyma. The total pressure drop of the model is the sum of the two contributions:

$$P(t) = P_{el} + \Delta P(t) \tag{4}$$

This equation holds true regardless of whether $P(t)$ is applied at airways' opening or at the outside of the elastic compartment [17].

In the parallel model, figure 1b, there are two governing equations, one for each compartment, respectively of volume V_1 and V_2:

$$\begin{cases} P(t) = E_1 V_1(t) + (R_1 + R_t)\dot{V}_1(t) + R_t\dot{V}_2(t) \\ P(t) = E_2 V_2(t) + (R_2 + R_t)\dot{V}_2(t) + R_t\dot{V}_1(t) \end{cases} \tag{5}$$

where R_1 and R_2 refers to the resistances of each bronchi, R_t is the resistance of the trachea and E_1 and E_2 are the elastances of the left and right lung, respectively, see figure 1b. These are the parameters of the model. $V_1(t)$ and $V_2(t)$ are the volumes associated to each lung and $P(t)$ is the muscular pressure that drives the lung ventilation.

Let us take the derivative of each equation:

$$\begin{cases} \dot{P}(t) = E_1\dot{V}_1(t) + (R_1 + R_t)\ddot{V}_1(t) + R_t\ddot{V}_2(t) \\ \dot{P}(t) = E_2\dot{V}_2(t) + (R_2 + R_t)\ddot{V}_2(t) + R_t\ddot{V}_1(t) \end{cases} \tag{6}$$

We substitute $\ddot{V}_2$ from eq (6)a into eq (6)b and replace $\dot{V}_2$ with its expression derived in eq (5)a. The equation for compartment 1 alone is:

$$\begin{aligned} R_2\dot{P}(t) + E_2P(t) = &\big[R_1R_2 + R_t(R_1 + R_2)\big]\ddot{V}_1(t) + \\ &+ \big[(R_2 + R_t)E_1 + (R_1 + R_t)E_2\big]\dot{V}_1(t) + E_1E_2V_1(t) \end{aligned} \tag{7}$$

Because the model is symmetric, the equation for compartment 2 is the same as (7) with inverted indexes 1 and 2. Then remembering that $V(t) = V_1(t) + V_2(t)$ the referral equation for the bi-compartment parallel model is:

$$\begin{aligned} (R_1 + R_2)\dot{P}(t) + (E_1 + E_2)P(t) = &\big[R_1R_2 + R_t(R_1 + R_2)\big]\ddot{V}(t) + \\ &+ \big[(R_2 + R_t)E_1 + (R_1 + R_t)E_2\big]\dot{V}(t) + \\ &+ E_1E_2V(t) \end{aligned} \tag{8}$$

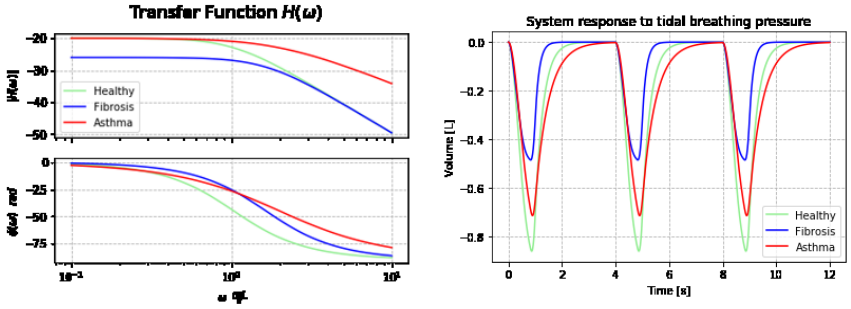
2 Methods

2.1 Creation of the dataset

It is possible to mimic the behavior of healthy subjects by setting physiological values of $R_{eq} = R_t + \frac{R_1 R_2}{R_1 + R_2}$ and $E_{eq} = \frac{E_1 E_2}{E_1 + E_2}$. In the literature, they are set to: $R_{eq} = 3\ cmH_2O/L/s$ and $E_{eq} = 10\ cmH_2O/L$ [9]. In this work, we mimic cystic fibrosis by doubling the healthy elastance (doubling the rigidity of the balloons): $E_{eq} = 20\ cmH_2O/L$, and asthmatic subjects by setting $R_{eq} = 5\ cmH_2O/L/s$. It is possible to follow the characteristic approach of electrical analysis [1] in which complex differential equations are studied in the frequency domain through the Laplace transform. The Laplace transform of eq (8) is:

$$H(s) = \frac{s(R_1 + R_2) + (E_1 + E_2)}{s^2\Big[R_1R_2 + R_t(R_1 + R_2)\Big] + s\Big[(R_2 + R_t)E_1 + (R_1 + R_t)E_2\Big] + E_1E_2} \tag{9}$$

Figure 2a shows the module and phase of the transfer function of the system in the cases of healthy, fibrosis and asthma, while Figure 2b shows the responses of each system to physiological $P(t)$.

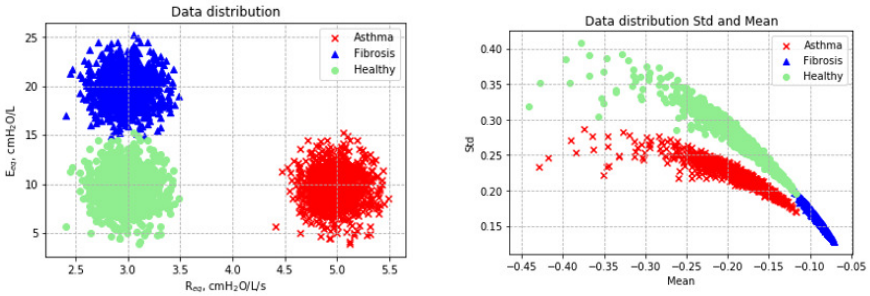


(a) Transfer Function

(b) System Response

Fig. 2: (a) Three different transfer functions, in the subplot above there is the module of the transfer function: $|H(\omega)|$ and below the phase: $\phi(\omega)$. Increasing the rigidity affects the response of the system at lower frequencies whereas increasing the total resistance affects higher frequencies. Tidal breathing happens at around 0.25 Hz being in the middle of the cutting frequence of $H(\omega)$. Consequently the output of the Volume changes. Figure (b) represents the output of the system (Volumetric signal) for one sample for each class.

Gaussian noise with mean $\mu = 0$ and standard deviation $\sigma = 0.5$ for the R_{eq} parameter and $\sigma = 5$ for the E_{eq} parameter, is added to R_{eq} and E_{eq} to mimic physiological diversity among different subjects as showed in figure 3a, there are 1000 samples for each class.



(a) Parameter distribution

(b) Tidal breathing simulation.

Fig. 3: (a) Synthetic data distribution, the three different clusters are well visible in this space. (b) Mean and Std features taken from volumetric signals of tidal breathing. These signals are the output of the model as explained in equation (8)

2.2 Training Machine Learning algorithms

Before talking about the training, a short summary for each classifier used is reported. The implementation has been done using Python and the open-source library scikit-learn.

Naive Bayes Naive Bayes classifier is a classifier that naively apply the Bayes theorem. A classifier is a function f that take an example $\boldsymbol{x} = (x_1, x_2, ..., x_n)$ where x_i is the i^{th} feature and transform it in a class y. According to Bayes theorem the probability P of an example $\boldsymbol{x}$ being class y is:

$$P(y \mid \boldsymbol{x}) = \frac{P(\boldsymbol{x} \mid y)P(y)}{P(\boldsymbol{x})} \tag{10}$$

Assuming that all the attributes are independent, the likelihood is:

$$P(\boldsymbol{x} \mid y) = P(x_1, x_2, ..., x_n \mid y) = \prod_{i=1}^{n} P(x_i \mid y), \tag{11}$$

hence, rewriting the posterior probability:

$$P(y \mid \boldsymbol{x}) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(\boldsymbol{x})} \tag{12}$$

because $P(\boldsymbol{x})$ is constant with regards of y, equation 12 can be rewritten and used to define the Naive Bayes (NB) classifier:

$$P(y \mid \boldsymbol{x}) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y)$$
$$\Downarrow \tag{13}$$
$$\hat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i \mid y),$$

Albeit the hypothesis of independence among attributes is never respected in real world, this classifier still has very good performance. Indeed, it has been observed that its classification accuracy is not determined by the dependencies but rather by the distribution of dependencies among all attributes over classes [19, 5].

Logistic Regression Despite its name, this is actually a classification algorithm. This is a linear classifier used normally for binary classification even though it can be extended to multiclass through different techniques like OvR (One versus Rest) or multinomial [20]. In our work, the *newton-cg* solver has been used together with *multinomial* multiclass. In this configuration, the ℓ_2 regularization is used and the solver learns a true multinomial logistic regression model using the cross-entropy loss function [14]. Using these settings allows the estimated probabilities to be better calibrated than the default "one-vs-rest" setting, as suggested in the official documentation of scikit-learn [15].

When multinomial multiclass is used, the posterior probabilities are given by a softmax transformation of linear functions of the feature variables [14]:

$$P(y_k|\boldsymbol{x}) = \frac{e^{\boldsymbol{w}_k^T \boldsymbol{x}}}{\sum_j e^{w_j^T \boldsymbol{x}}} \tag{14}$$

where $\boldsymbol{w} \in \mathbb{R}^n$ is the vector of trainable weights, $\boldsymbol{x}$ is the feature vector and y is the class label. Using 1-of-K encoding scheme, it is possible to define a matrix $\boldsymbol{T}$ composed by n rows (being N the total number of features for each class) and k columns (being K the total number of classes) [14]. In our case N=2 and K=3. Each vector $\boldsymbol{t_n}$ will have one in the position of its class and zeros all over the rest. In this scenario, the *cross-entropy* loss function to minimize for the multinomial classification regularized with ℓ_2 is:

$$\min_{\boldsymbol{w}} \left(\frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} - \sum_{n=1}^{N} \sum_{k=1}^{K} t_{nk} \ln(\hat{y}_{nk}) \right) \tag{15}$$

being $\hat{y}_{nk} = P(y_k|\boldsymbol{x_n})$.

Perceptron For linear separable datasets, Perceptron can achieve perfect performances because it guarantees to find a solution, hence the learning rate η is

not essential and by default is set to 1.0 in scikit-learn. In our implementation, the loss function is the number of mislabelled samples and it is not regularized. The weights of the model are updated on mistakes as follows:

$$w_{j+1} = w_j + \eta\big(y^{(i)} - \hat{y}^{(i)}\big)x_j^{(i)} \tag{16}$$

where i is the sample, j is the feature, y is the target and $\hat{y}$ its respective prediction. [20]. The weights are updated with Stochastic Gradient Descent (SGD) optimizer, meaning that the gradient of the loss is estimated for each sample at a time and the model is updated along the way [16].

Support-Vector Machines This is one of the most robust supervised ML algorithms, it is used for both regression and classification problems and it can be used in a non-linear fashion thanks to kernel tricks [10]. In SVMs, we used the Radial Basis Function (rbf) kernel:

$$K(X, X') = e^{\gamma||X-X'||^2} \tag{17}$$

When implementing this function, there are 2 parameters required:

- γ, which is the term in the expression of the rbf and it is the coefficient of multiplication for the euclidean distance
- C, which is the error Cost, this is not directly related to the kernel function, instead it is the penalty associated with misclassified instances.

Setting these parameters together is important for achieving good results. In our case, a vast selection of pairs reports similar results, as shown in figure 4.

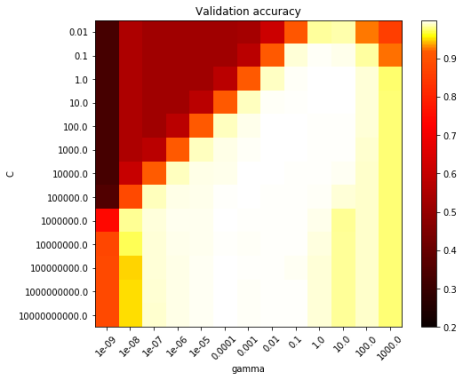


Fig. 4: Heat map for setting the best pair of γ and C in rbf kernel function for SVMs, brightest colors correspond to highest validation accuracy.

Random Forest Random Forest (RF) is an ensemble of decision tree estimators in which each estimator classifier has been used with 100 trees and Gini impurity as criterion. The class prediction is performed by averaging the probabilistic prediction of each estimator instead of using the voting system as implemented in its original publication [3]. Random Forests follows the exhaustive search approach for the construction of each tree, where the main steps are listed in algorithm 1 [4].

Algorithm 1 Pseudocode for tree construction - **Exhaustive search**

1: Start at the root node
2: **for** each X **do**
3: Find the set S that minimizes the sum of the node impurities in the two child nodes and choose the split $S^* \in X^*$ that gives the minimum overall X and S
4: **if** Stopping criterion is reached **then**
5: Exit
6: **else**
7: Apply step 2 to each child node in turn

2.3 Training

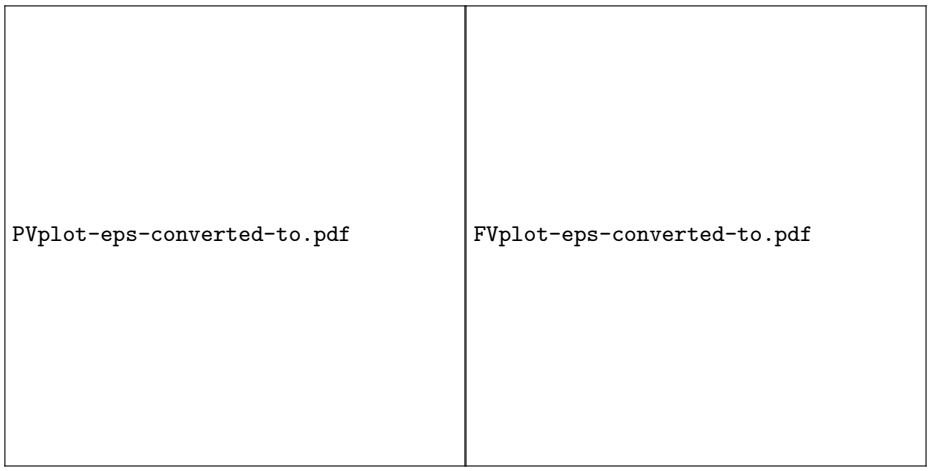
Once the dataset is ready, simulations are performed and significant statistical features are extracted from the signals (examples of output signals of the system are shown in Figure 2b). Here, for facilitating the graphical representation, two features are extracted: mean and standard deviation. These features have enough information to correctly differentiate among the three classes. Before training each of the previous models, standard scaling has been fit on the training set and applied on both training and test sets. The dataset has been split randomly by keeping the size of the training set at 80% of the total dataset. Hence performances have been evaluated on 800 samples after having checked the correct balance among the classes. The classifiers previously reported have been trained and tested and their relative decision plots can be seen in figure 7.

3 Results

3.1 Lung model

The parallel model used is a good representation of the lung when detailed geometrical characteristics are not important to model. Working with this model allows to control the resistance and elastance of the respiratory system, allowing the simulation of certain diseases. However, it is important to ensure that the results given by our model are coherent with reality. Because of this, the output signals of volumes and flows have been observed and visually compared with

real signals, see figure 2b. The flow Φ has been calculated as $\Phi(t) = \partial V(t)/\partial t$. Pressure-Volume plots and Flow-Volume plots have been evaluated for each class, see figure 5. In typical Pressure-Volume plots a decreasing of compliance is manifested as a shift on the right of the loop. However the model that we are using is pressure driven. In a pressure driven model, when the compliance decreases, the pressure control yields less and less volume for the same pressure level, causing a lowering of the loop curve. Asthmatic subjects on the other hand are simulated as having same compliance as healthy subjects but greater resistance, and because of this, their flow is lower than the others in a pressure driven model as visible in Figure 5b.

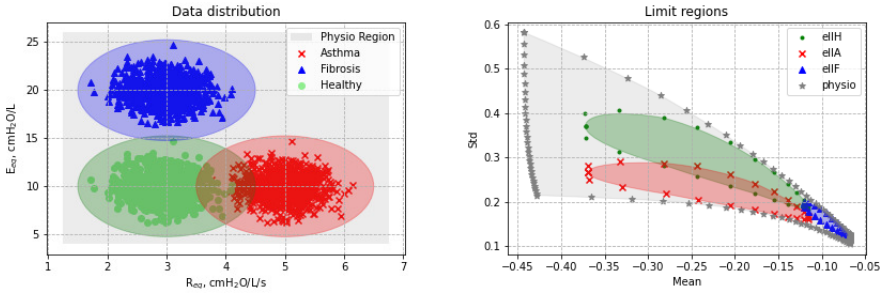


(a) Pressure-Volume loop (b) Flow-Volume loop

Fig. 5

The model performs a transformation $(R_{eq}, E_{eq}) \in \mathbb{R}^2 \to \mathbb{S}$, with $\mathbb{S} \subseteq \mathbb{R}$, which is the signal space, in our case the signal is $V(t) \in \mathbb{S}$. By extracting features from the signal space, we pass to $\mathbb{R}^N$, where N is the number of features extracted. Here, we have extracted two features, therefore we have a mapping $\mathbb{R}^2 : (R_{eq}, E_{eq}) \to \mathbb{R}^2 : (\mu, \sigma)$ where μ and σ are the extracted features, respectively the mean and the standard deviation of the signal. The image of this mapping is particularly useful to set limits in the prediction of the AI. Indeed, for measurements that are not contained in the image, and thus inconsistent, it makes sense to have a separate treatment that will alert when a measurement must be discarded and replaced by a new one because it is not physiological. To achieve this, a physiological region has been defined (gray area in Figure 6a), the boundary of the rectangle region (the physiological set) is passed to the system and the output path is then patched again to form a polygon of acceptable measurements (gray area in figure 6b). Out of this patch of acceptable mea-

surements, the data will not be passed to the AI for prediction and a message of "wrong acquisition" will be displayed. In Figure 6, the data distribution is



(a) (b)

Fig. 6: Data distribution and visual representation of each set in the (R_{eq}, E_{eq}) space (a) and in the (μ, σ) space (b)

obtained with standard deviation $\sigma(R_{eq}) = 1\ cmH_2O/L/s$ and $\sigma(E_{eq}) = 3.5\ cmH_2O/L$ to allow for a better spreading and overlapping. Moreover, because we want the elliptic patch of each class to be comprehensive of almost all its possible samples, we fix the width and the height of the ellipse to be three times the respective standard deviation.

3.2 Machine Learning results

Multiple Machine Learning algorithms have been tested against synthetic data retrievable using spirometers. Their performances are shown in figure 7 in terms of accuracy and decision boundaries. In this section, there are some considerations regarding the different classifiers used. By looking at the data distribution in figure 3b, we observe that a linear separation will not be perfect but still good. Multinomial logistic regression and Perceptron have been used and compared for such a linear separation. Their performances are then compared with other commonly used non-linear classifiers; the first to be tried has been Naive Bayes because of its interpretability and ease of use, however its poor performance led us to try more sophisticated models like SVM with rbf kernel and RF. These latter classifiers lead to great performance, even if they have local errors close to the decision region boundaries. Nevertheless, these regions are characterized by spurious areas located on both bottom corners in the case of SVM (figure 7 SVM subplot) and the right-up region in the case of RF.

– **Naive Bayes**: It is possible to see how this algorithm is not suited for this kind of multi-class classification. Indeed, this classifier is usually used for binary classification (it is used a lot for spam detection). In this particular

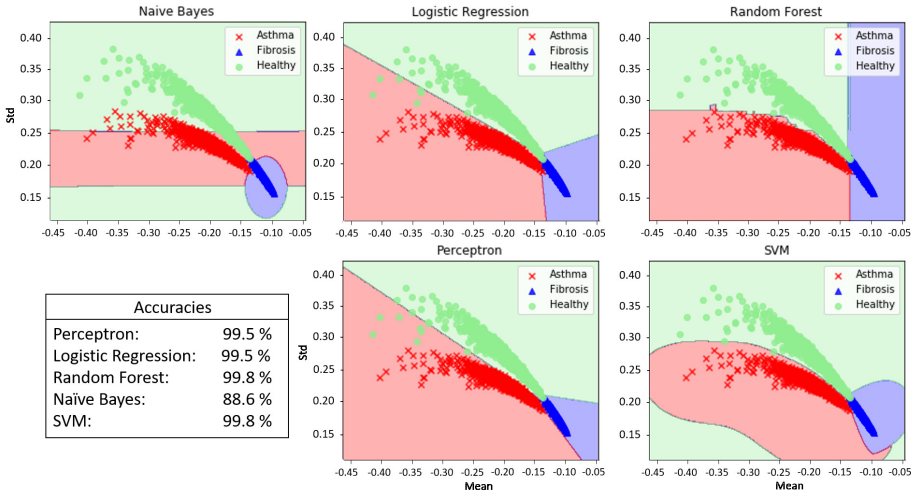


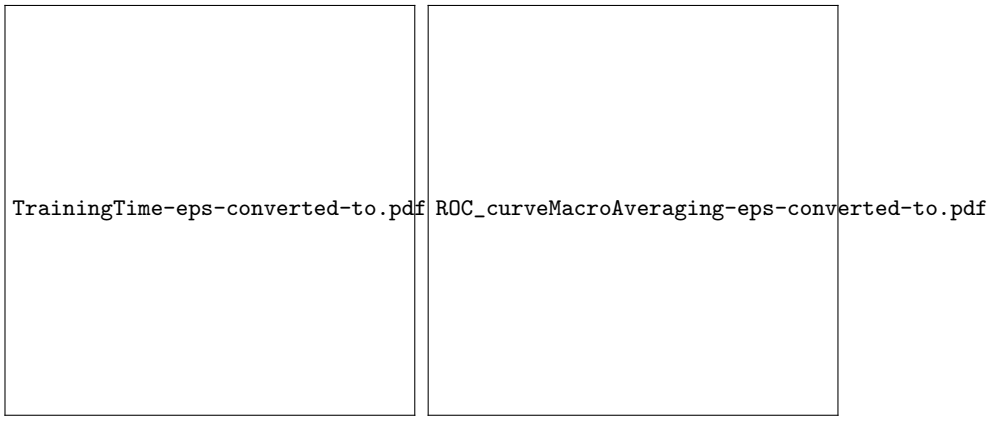
Fig. 7: Decision regions and accuracy of the implemented classifiers. The regions should be shortened according to the limitations indicated in figure 6b.

dataset, Naive Bayes fails in detecting a border between healthy and asthmatic subjects and the found boarder is not significant compared with other classifiers.

- **Logistic Regression**: This classifier is one of the most used in biomedical applications, both for its easy comprehension and its great performances when the classes are linearly separable. In this case, classes are not linearly separable. Nevertheless, this keeps a good level of generalization without renouncing to ease of usage, comprehension and good performances.
- **Random Forest**: RF is probably one of the most powerful classifiers. It is used also in biomedical applications for its good performances and its resistance to overfitting. It is an ensemble method where multiple decision trees (DTs) are singularly trained. Finally the average of the predictions of all the estimators will be used to make the decision of the RF classifier. DTs are used in medicine because of their clarity in the decision, however with RF there is a loss of this explainability in the decision, caused by an enhancing of complexity due to the ensemble.
- **Perceptron**: This is a linear model as long as only one layer is provided. It is powerful and performs very well in this particular situation. It can be useful to increase the deepness of its structure once there are lots of features and their relationships are not of easy interpretation. Also, in contrast with logistic regression it can be easily used for non-linearity.
- **Support Vector Machine**: This model is widely used in a lot of applications. In this work, the Radial Basic Function kernel has been used. Its non-linear nature allows to follow better the separation between healthy and asthma. In this dataset, this is the most performant model to distinguish

these two classes. However, like the Naive Bayes it creates a green area below the red and blue zone that are incorrect.

In contrast with Deep Learning, ML models are normally faster to train. Figure 8a shows the differences of training times among the used classifiers. Even if the timings are very small, it is interesting to see, for instance, how fast the Perceptron is compared to Random Forest (RF). This is an interesting property for large datasets. Figure 8b shows the Receiver Operating Characteristic (ROC) curves and the respective calculated Area Under the Curve (AUC) for each classifier. ROC curves have been adapted for multiclass using the macro averaging technique. As expected, SVM and RF outperforms the other classifiers, however, using this metric is possible to observe the difference between Logistic Regression and Perceptron. The origin of this difference is probably on the separation between asthma and fibrosis as observable in figure 7.



(a) (b)

Fig. 8: (a) Different timings of training for each classifiers. Trainings are performed using the CPU runtime on Colab. (b) Macro averaging of Receiver Operating Characteristic curve for each curve, zoom on the upper left part. RF and SVM are overlapped and their value is fixed on 1.

It is worth to mention that we carried out supplementary simulations using data more spread than in figure 3a, which resulted in even more overlaps between the different classes (see figure 6). In these cases, the decision regions of each classifier stay very similar to those shown in figure 7 with a resulting lower accuracy due to the overlapping.

4 Conclusions and outlook

As a brief recap, in this work the following has been done:

A second order ODE mathematical model of the lung has been used to generate synthetic data of asthma, cystic fibrosis and healthy subjects. This data has been used to train Machine Learning models. The models have been evaluated on different synthetic data sampled from the same distribution of the training set. A solution has been proposed for non physiological measurements, see subsection 3.1. Finally, differences among the classifiers have been studied in terms of accuracy, ROC curves and training timings.

We elaborated on the potential use of modern ML techniques to diagnose diseases of the human respiratory system. The direct conclusion of this work is the ability of ML algorithms to distinguish among linear separable clusters in the $\mathbb{R}^2$ (R_{eq}, E_{eq}) space, also in the non-linear feature space of $f(g(R_{eq}, E_{eq}))$ where $g : \mathbb{R}^2 \to \mathbb{S}$ is the parallel lung model and $f : \mathbb{S} \to \mathbb{R}^N$ is the feature extraction with N being the number of features.

One limitation of our work is the training and testing based entirely on simulated data and the utilization of the same pressure profile for all the classes. After the present proof-of-concept, future work will include training on simulated data and testing on acquired real data. Moreover, the usage of depth camera will be investigated to extract tidal breathing patterns [21]. Furthermore, the pressure was assumed to be uniform throughout the lungs and there was no difference in the application of the pressure among the three considered cases. This condition is in principle not respected because some patients can increase their muscle effort in order to keep a satisfactory ventilation. However, it is possible to assist the patients and train them to follow a specific pattern while breathing in the spirometer.

To conclude, the different ML models presented are proven in principle reliable, therefore they could provide the physicians with real-time help for the diagnosis decision.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Curie grant agreement No 847581 and is co-funded by the Région SUD Provence-Alpes-Côte d'Azur and IDEX UCA JEDI.



References

[1] Otis AB et al. *Mechanical factors in distribution of pulmonary ventilation.* J Appl Physiol, 1956, pp. 427–43.

[2] Sandeep Bodduluri et al. *Deep neural network analyses of spirometry for structural phenotyping of chronic obstructive pulmonary disease.* en. Publisher: American Society for Clinical Investigation. July 2020. DOI: 10.1172/jci.insight.132781. URL: https://insight.jci.org/articles/view/132781/pdf.

[3] Leo Breiman. "Random Forests". en. In: *Machine Learning* 45.1 (Oct. 2001), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[4] Riccardo Di Dio. "Analyzing movement patterns to facilitate the titration of medications in late stage Parkinson's disease". it. laurea. Politecnico di Torino, July 2019. URL: https://webthesis.biblio.polito.it/11370/.

[5] Pedro Domingos and Michael Pazzani. "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss". en. In: *Machine Learning* 29.2 (Nov. 1997), pp. 103–130. ISSN: 1573-0565. DOI: 10.1023/A:1007413511361. URL: https://doi.org/10.1023/A:1007413511361.

[6] Weibel ER. *Geometry and Dimensions of Airways of Conductive and Transitory Zones. Morphometry of the Human Lung.* Springer Berlin Heidelberg, pp. 110–135.

[7] Philip M. Farrell et al. "Diagnosis of Cystic Fibrosis: Consensus Guidelines from the Cystic Fibrosis Foundation". eng. In: *The Journal of Pediatrics* 181S (Feb. 2017), S4–S15.e1. ISSN: 1097-6833. DOI: 10.1016/j.jpeds.2016.09.064.

[8] Sherif Gonem et al. "Applications of artificial intelligence and machine learning in respiratory medicine". In: *Thorax* 75.8 (2020), pp. 695–701. ISSN: 0040-6376. DOI: 10.1136/thoraxjnl-2020-214556. URL: https://thorax.bmj.com/content/75/8/695.

[9] Liming Hao et al. "Dynamic Characteristics of a Mechanical Ventilation System With Spontaneous Breathing". In: *IEEE Access* 7 (2019). Conference Name: IEEE Access, pp. 172847–172859. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2955075.

[10] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. "Kernel methods in machine learning". In: *The Annals of Statistics* 36.3 (June 2008). arXiv: math/0701907. ISSN: 0090-5364. URL: http://arxiv.org/abs/math/0701907.

[11] K Horsfield and G Cumming. "Morphology of the bronchial tree in man." In: *Journal of Applied Physiology* 24.3 (1968). PMID: 5640724, pp. 373–383. DOI: 10.1152/jappl.1968.24.3.373. URL: https://doi.org/10.1152/jappl.1968.24.3.373.

[12] B. Mauroy et al. "An optimal bronchial tree may be dangerous". In: *Nature* 427.6975 (Feb. 2004), pp. 633–636. ISSN: 1476-4687. DOI: 10.1038/nature02287. URL: https://doi.org/10.1038/nature02287.

[13] J. Pfitzner. "Poiseuille and his law". en. In: *Anaesthesia* 31.2 (1976), pp. 273–275. ISSN: 1365-2044. DOI: 10.1111/j.1365-2044.1976.tb11804.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2044.1976.tb11804.x.

[14] Heiko Schulz. *Pattern Recognition and Machine Learning*. 2011. Chap. 4.3.4. ISBN: 9780387310732.

[15] *Scikit learn documentation - Logistic regression*. URL: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.

[16] *Scikit learn documentation - Perceptron*. URL: https://scikit-learn.org/stable/modules/linear_model.html#perceptron.

[17] Bates T. and Jaspm H. *Lung Mechanics, an inverse modeling approach*. Cambridge University Press, 2009.

[18] Merryn H. Tawhai et al. "CT-based geometry analysis and finite element models of the human and ovine bronchial tree". In: *Journal of Applied Physiology* 97.6 (Dec. 2004). Publisher: American Physiological Society, pp. 2310–2321. ISSN: 8750-7587. DOI: 10.1152/japplphysiol.00520.2004. URL: https://journals.physiology.org/doi/full/10.1152/japplphysiol.00520.2004.

[19] "The Optimality of Naive Bayes". In: *AA* 1.2 (2004), p. 6.

[20] Raschka Sebastian NAD Mirjalili Vahid. *Python Machine Learning*. Packt, 2018. Chap. 3, pp. 59 –61.

[21] Yunlu Wang et al. "Abnormal respiratory patterns classifier may contribute to large-scale screening of people infected with COVID-19 in an accurate and unobtrusive manner". In: *arXiv:2002.05534 [cs, eess]* (Dec. 2020). arXiv: 2002.05534. URL: http://arxiv.org/abs/2002.05534.

[22] Ewald R. Weibel. *The pathway for oxygen*. 1984. ISBN: 9780674657908.