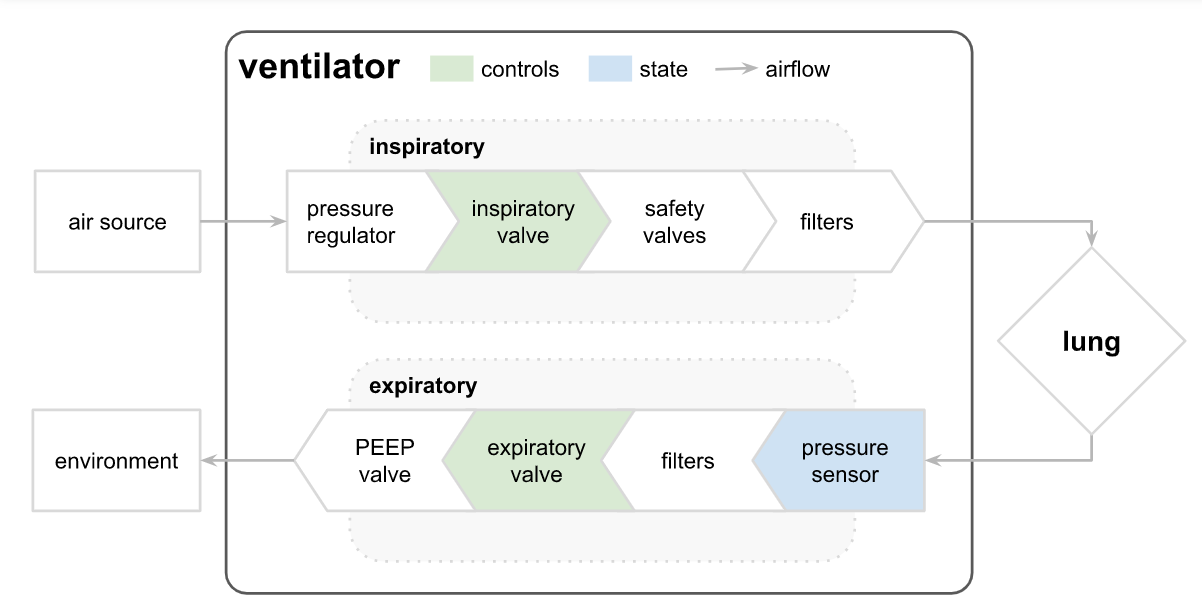
**Kaggle Project 9**

**Ventilator Pressure Prediction**

**Prepared by: Dwaipayan Mukherjee (2211569 – AIA AIML)**

About the Project:

The ventilator data used in this competition was produced using a modified open-source ventilator connected to an artificial bellows test lung via a respiratory circuit. The diagram below illustrates the setup, with the two control inputs highlighted in green 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 the inspiratory solenoid valve 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 exploratory valve is open (1) or closed (0) to let air out.



In this competition, participants are given numerous time series of breaths and will learn to predict the airway pressure in the respiratory circuit during the breath, given the time series of control inputs.

Steps I have followed in this project:

1. Imports:

I have done the necessary imports, that is, of numPy, Pandas, random, tensorflow and os. Then I have loaded the train and test data as Pandas dataframes.

1. I have grouped the train dataset by the **breath\_id** to form the final dataset.
2. I have created some new features from the available ones.
3. I have filled the null values with zeros.
4. Then I have separated the features and the target columns to fit the model.
5. I have reshaped the features and the target column to fit a CNN model.
6. I have created a Recurrent Convolutional neural network model with deep learning, with the following layers:
7. Convolutional 1D with activation function as ReLu.
8. MaxPooling 1D
9. Batch Normalization
10. Convolutional 1D with activation function as ReLu.
11. MaxPooling 1D
12. Batch Normalization
13. Bidirectional LSTM with return sequences as True
14. Bidirectional LSTM with return sequences as True
15. Global Average Pooling 1D
16. A dropout of 0.5
17. Dense layer

Adam optimizer has been used with learning\_rate as 0.001 and loss as Mean Absolute error.

1. The model is trained on the features and target with validation split as 0.2, epochs as 350, batch size as 512, and callback as EarlyStopping with patience as 260, monitor as validation loss, mode as minimum and restore best weights equal True.
2. Upon evaluation the validation loss comes as 0.36005.
3. I have predicted the pressure values using this model.
4. I have drawn the relationship between predicted and actual pressure for a series of ranges.
5. Next, I have plotted a scatterplot of the predicted pressure values along with the actual pressure values.
6. Lastly, I have created columns for the actual and predicted pressure values and the difference between them.