# Respiratory Rate Estimation with Accelerometer and Gyrometer Recordings

## I. Introduction

The measurement of respiration rate[1] is critical to the diagnosis and monitoring of a wide range of breath disorders as well as being useful to broader parameters of a patients condition. In recent years, accelerometers have been increasingly used to derive respiration rate. As the number of patients monitored via different wearable sensors has increased, there is a demanding need for accurate estimation of respiratory rate from these sensors in a robust manner. It is also useful to observe breathing rate patterns and their relation with respect to the kind of activities performed. This project presents a method to predict the breathing rate of a person from accelerometer and gyroscope readings and is designed for prediction with respect to day-to-day activities. Through this paper, we highlight the use of different feature selection and learning techniques that will help us in the prediction of a precise respiration rate.

## II. Data

In this paper a tri-axial accelerometer for respiratory monitoring is used because it is a promising approach to achieve comfortable, low cost, continuous and ambulatory monitoring. The accelerometer and gyroscope readings, as recorded from a wristband sensor, is used for monitoring the respiratory and heart rate over a span of 5 days. We also consider body temperature values from a chest strap sensor. In addition to this, we have a collection of time domain features extracted for IMU which include mean, variance, skewness, kurtosis, correlation between the axes (xy, yz, and xz); and frequency domain features including mean power spectral density and peak power spectral density. The time windows are in the order of 3 seconds for feature extraction. Raw values of heart rate will be used for prediction as well.

### A. Data Pre-processing

Normalization is a scaling or a pre-processing stage of any given data. All the features are thus re-scaled so that they would have the properties of a standard normal distribution. Normalization takes an important role in the field of soft computing for the manipulation of the data so as to accordingly scale up or scale down the range of data before it is used in the further processes. In the method mentioned below, we scale each feature to a given range.

The transformation is given by:

$$X_{std} = \frac{(X - X_{min}(axis{=}0))}{(X_{max}(axis{=}0) - X_{min}(axis{=}0))}$$

$$X_{scaled} = X_{std} * (max - min) + min$$

where min, max = feature range.

This estimator scales and translates each feature individually such that it is in the given range on the training set, i.e. between zero and one. This transformation is often used as an alternative to zero mean, unit variance scaling.

## III. Feature Selection

To improve prediction accuracy and reduce processing time on the data, appropriate feature selection[7] procedures were chosen to reduce dimensionality. Often input features that need to be trained are highly correlated[7], i.e. they carry similar information. Highly correlated data columns do not add much new information to the existing pool of features. Using only one from these correlated columns can help in reducing computations without decreasing much information. As a basic step in the feature selection process, highly correlated features will be removed with a set threshold.

Number of features after removing highly correlated features: 41

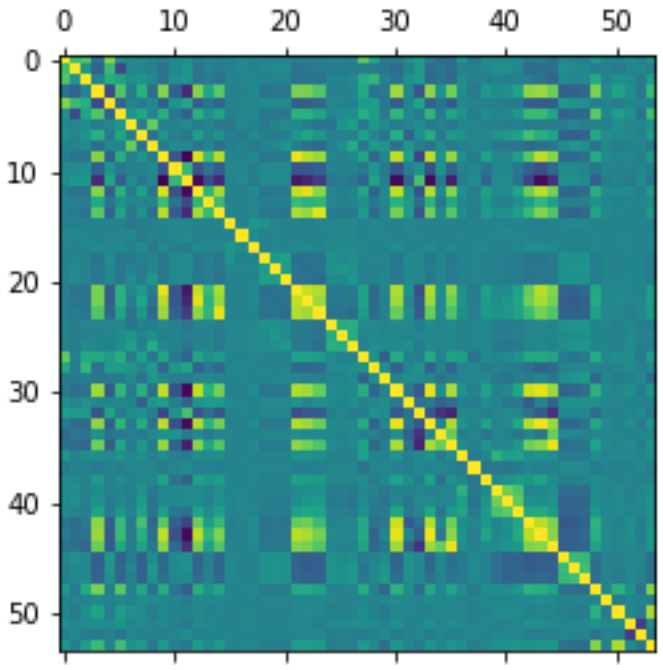


Fig. 1. Correlation graph of features

Figure 1 shown above depicts a correlation graph that gives an indication of how related features are with respect to each other. If two features are highly correlated, removing one would not expect to affect result significantly and hence

to reduce dimensionality we can remove the highly correlated features as can be seen in the figure.

**PCA:** Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. This is being used as a work around for preserving high dimensionality and significantly reducing computation power.

The PCA graph, Fig. 2, plotted below shows the change in variance with respect to the different no of features taken into consideration. We observe that after a point, the variance does not change much even when the number of features is increased. Therefore we can conclude that our minimum necessary requirement for low variance can be obtained by taking into account around 14 features.

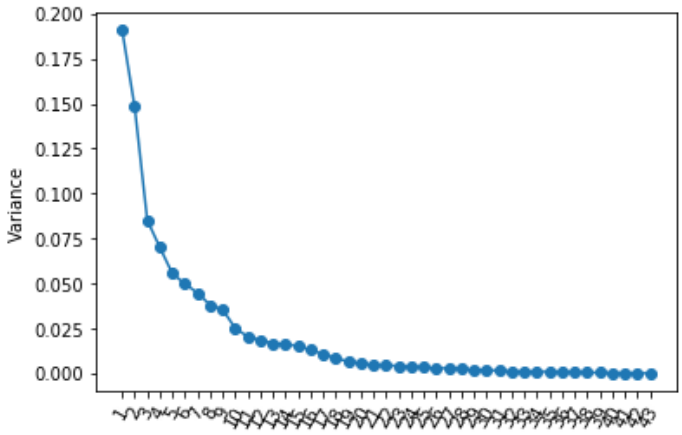


Fig. 2. PCA variance

**Random Forests:** Random forests[1] are a non-parametric method that build an ensemble model of decision trees from random subsets of features and bagged samples of the training data. Every node in a decision tree is a condition on a single feature, designed to split the dataset so that similar response values end up in the same set. The reason for using a tree-based strategy is to naturally capture and improve the purity of the node.

Using Random Forest[6], we were able to arrange all the features in their decreasing order of importance. In Fig. 3 we plot error by training our model using different number of features in their order of importance. The plot clearly shows that after a particular value, it is futile to consider more features as it does not decrease our error value further.

## IV. Model Selection

As the data has a continuous target, variations of regression models will be implemented and compared. The baseline method for prediction would be a linear regression model with the extracted features. In addition to the baseline method, we used penalized and other advanced version of regression models for better accuracy and performance.

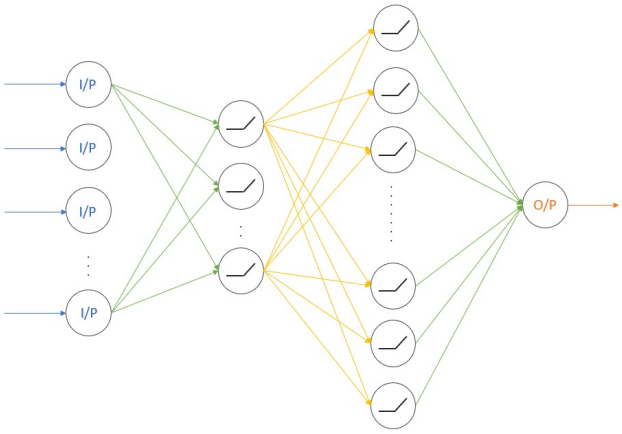


Fig. 3. Neural Network

### A. Neural Networks:

A multi-layer perceptron is a feedforward artificial neural network [3] that uses a nonlinear activation function and back-propagation[5] for training. We implemented a fully connected neural network, Fig. 4, a multi-layer perceptron as a regression model. The network consists of 50 nodes in its first layer and 100 nodes in its second layer. The advantage of using a neural network was the ability to have multiple hidden layers which would improve the evaluation compared to simple regression models. L2 regularization was used to avoid overfitting by penalizing weights with large magnitudes. We tried different activation functions such as tanh and rectified linear unit(relu). We modified the learning rate as adaptive and constant and tried the adam and stochastic gradient descent solver. On evaluating with various parameters, noticeable results were obtained as shown below.

We iterated through various parameters based on the error rate and $R^2$ score concluding at the model parameters that gave us the best result for MSE and $R^2$ score.

| Learning Rate | Solver | Activation | Error |
|---|---|---|---|
| adaptive | adam | tanh | 3.19 |
| constant | adam | relu | 3.32 |
| constant | adam | tanh | 3.21 |
| adaptive | sgd | tanh | 3.24 |
| constant | sgd | relu | 3.25 |

### B. XGBoost:

As a solution for dealing with high dimensional data we propose to use a gradient boosting method for faster computations and lower modelling time. XGBoost[2] is an implementation of gradient boosting machine and a more regularized model formalization. It is a combination of stochastic gradient descent, gradient boosting and regularization.

We iterated through various parameters based on the error rate and $R^2$ score concluding at the model parameters that gave us the best result for MSE and $R^2$ score.

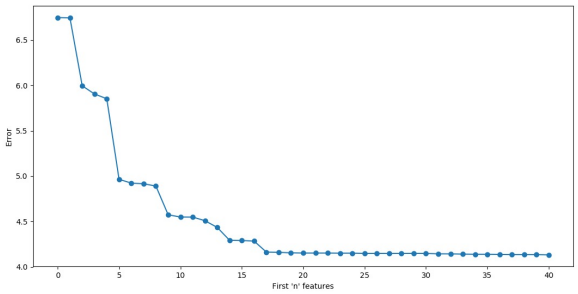


Fig. 4. Error vs different No. of features using Random Forest feature selection followed by Neural Network

| Learning Rate | Max Depth | Min Weight | RMSE | $R^2$ |
|---|---|---|---|---|
| 0.0 | 4 | 4 | 17.06 | -5.25 |
| 0.03 | 9 | 1 | 2.57 | 0.86 |
| 0.03 | 4 | 2 | 3.51 | 0.74 |
| 0.01 | 9 | 5 | 6.82 | 0.001 |
| 0.01 | 5 | 3 | 7.10 | -0.08 |

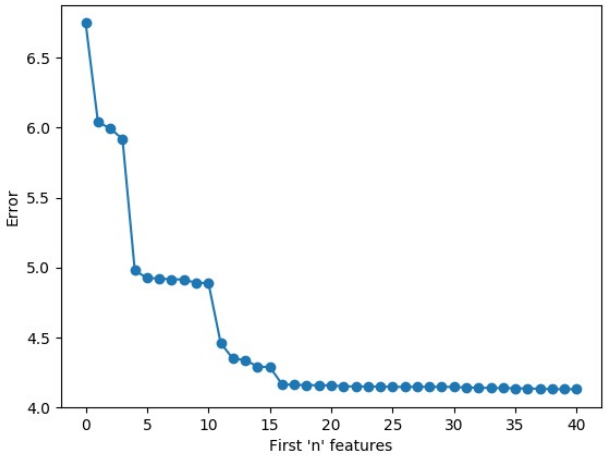


Fig. 5. Error vs different No. of features using Random Forest feature selection followed by XGBoost

As can be seen from the table, the best result was achieved at a minimum RMSE of 2.57 or a Mean Squared Error of 6.6.

## V. Evaluation Metrics

To avoid suffering from improper fitting of data, cross-validation measures need to be implemented. This involves a hold out of the available data as a test set in multiple iterations. We used k-fold cross-validation (k=10) to evaluate the results and ensure to build a model that supports the best fit of data. The error metric used was the Mean Square Error (MSE).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(c_i - \bar{c}_i)^2$$

The correlation coefficient formula specifies how strong of a linear relationship there is between two variables. R-Squared is the square of the correlation coefficient. The correlation coefficient can be calculated as:

$$\rho(X,Y) = \frac{(X,Y)}{\sqrt{\mathbf{Var}(X)\mathbf{Var}(Y)}}$$

## VI. Softwares

The primary language of development will be Python and the libraries we would refer for the implementation are SciPy, Pandas, Numpy, XGBoost, Scikit-Learn and Statsmodels.

- PCA - sklearn.decomposition.PCA
- Random Forest Regressor - sklearn.ensemble
- Linear Regression - sklearn.linear_model
- MSE, $R^2$ - sklearn.metrics.mean_squared_error, r2_score
- MLP Regressor - sklearn.neural_network.MLPRegressor
- MinMaxScaler - sklearn.preprocessing
- XGBoost - xgboost

## VII. Conclusion

As we have high dimensional data which we used to implement the methodology decided, suitable efforts were taken to implement the various data pre-processing mechanisms and extract the appropriate features for building the model. Based on the mechanism we developed, we concluded that the best result was obtained using Pearson correlation method to remove highly correlated features and then implementing Extreme Gradient boosting method on the feature set. An MSE of 6.6 and a $R^2$ value of 0.85 was obtained.

## References

[1] Breiman, Leo. *Random forests*, Machine learning 45.1 (2001): 5-32

[2] Chen, Tianqi, Guestrin, Carlos. *XGBoost: A Scalable Tree Boosting System*, Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016

[3] Specht, Donald *A General Regression Neural Network* IEEE Transactions on Neural Networks, 1991

[4] Fekr, Atena Roshan, et al. *A medical cloud-based platform for respiration rate measurement and hierarchical classification of breath disorders* Sensors 14.6 (2014): 11204-11224

[5] SITTIG, D. E, and ORe, J. A. *'A parallel implementation of the backwaxd error propagation neural network training algorithm: experiments in event identification'*, Comput. Biomed. Res., 25, pp. 547-561

[6] Robin Genuer, Jean-Michel Poggi, Christine Tuleau-Malot. *Variable selection using Random Forests* Pattern Recognition Letters, Elsevier, 2010, 31 (14), pp.2225-2236.

[7] Yu, Lei, Liu, Huan *Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution*Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003