Dry vs Wet Cough Automatic Classification using the COUGHVID Dataset

Jostein Leirgulen, Mathias Nuris-Souquet, Clémentine Lévy-Fidel
Supervised by Lara Orlandic and Tomas Teijeiro from the Embedded Systems Lab (ESL)
EPFL, Switzerland

Abstract—**Machine learning provides valuable tools for health care, such as helping detecting respiratory infections by analyzing audio recordings of coughs. The audio recordings present patterns that indicates whether the cough is dry or wet, which is useful to reveal infections like Covid-19. The database COUGHVID created by the Embedded Systems Lab has managed to gather a large collection of 20'000 coughs recordings from both healthy or Covid-19 ill patients, from which 10% have been assessed by medical experts to determine whether or not they were related to a Covid-19 condition. We present a model able to classify audio recordings between dry or wet. The model has been trained on audio features from the recordings labeled by the experts and reaches 0.66 testing accuracy. Results show that including metadata like the age, gender or respiratory symptoms determined by the experts could help improve the classification.**

I. Introduction

Cough is a defensive physiological reflex stemming from signals from the larynx and the lower respiratory tract to the brain [1]. The reflex results in a sudden expiration of air aiming at rejecting noxious materials from the lungs, accompanied by a recognizable sound [2]. It is the main symptom of many infections that affect the respiratory tract, and is the main reason that complies people to reach for medication [3]. Recently, Covid-19 has become the most worrying cause of respiratory conditions, with dry cough being a symptom for the infection in 67.7% of cases [4]. With the sanitary urgency caused by the spreading of the Covid-19 worldwide, being able to determine quickly whether a patient is suffering from respiratory conditions has become of highest importance – especially if automated as it would enable a valuable time saving for medical experts as well as diagnoses when no experts are available.

The presence or not of secretions in the airways influences the sound produced when coughing. The specific sound of a cough can be separated into three phases, corresponding to the glottis opening, the expiration of the airflow and the closing of glottis [5]. The particular sequence and characteristics of these phases allow classifying the cough into wet or dry, and might reveal an underlying condition [6], often a pulmonary disease ranging from acute pertussis to asthma or Covid-19. Usually, the diagnosis has to be performed clinically since the differences are subtle – opinions even diverge among physicians [7].

To improve the human diagnosis, coughs sounds can be unraveled into several patterns like the number of bursts, their length, their intensity and their spacing which can then be analyzed statistically [2]. To gather more information, they can also get converted into digital signals, which enables spectral analysis. Some algorithms already exist that are trained to classify cough sounds between dry and wet [7][8][5]. However, their training data sets were not extensive enough, or the methods were based on acoustic characteristics, hence not adaptable or robust enough.

We present a model for dry or wet classification based on COUGHVID, an extensive database created by the Embedded System Lab (ESL) at EPFL containing 20'000 coughs recordings [4]. Out of this number, 2'000 recordings were annotated by medical experts who assessed whether the cough was dry or wet. Rather than to train the model on raw audio recordings, we work on the features of the signals also pre-processed by ESL. They have provided us with the data of 1'659 subjects from COUGHVID, to which were added two data sets where the audio recordings have been segmented via two different methods.

The model we designed compares the predictions between the three data sets. We tested different classification methods: logistic regression, k-nearest neighbors, support-vector-machine, linear discriminant analysis, Gaussian naive bayes, decision tree, random forest and eXtreme Gradient Boosting (XGBoost). As XGBoost performed significantly better, we kept the other classifiers as baseline models for the analysis of the features used.

II. Data manipulation

A. Database description

The data sets we worked on consists of samples from 1'659 subjects from the COUGHVID database. Each sample contains information in the time and frequency domains from the audio signals, augmented with meta-data of the subject. In case the latter haven't been provided, they are replaced by mean values.

Examples of features about the time and spectral analysis:

- Spectral roll-off
- Power spectral density
- Spectral bandwidth
- Mel Frequency Cepstral Coefficients (MFCC)
- EEPD
- ...

Meta-data: age, gender, pre-existing respiratory conditions, specified symptoms

Additionally, each sample contains a feature indicating which expert labeled the cough and a feature indicating the

expert's label. This makes up to 78 features.
There are three version of the data set at stake: one for which each audio recording was segmented into sequences of successive bursts (coarse segmentation) thus making up to 3'440 samples, one for which each burst was separated independently (fine segmentation) and one for which no segmentation was applied.

B. Exploratory data analysis and feature engineering

We investigated the different features of the data set to potentially increase the classification accuracy of our selected model. This includes:

1) Handling categorical values: Some of the features presented categorical values like the expert choice, respiratory condition, or again gender. They were turned into dummy variables.

2) Thorough indexing: Although the segmented data sets give the possibility to handle more data, they also allow several samples to stem from the same subject, thus introducing interdependency between them. To avoid correlation bias when splitting the data between train and test data sets, we first observe that the majority of subjects have around 4 subsamples with fine segmentation and around 2 sub-samples with coarse segmentation, and for both cases the number of sub-samples are very closely distributed to the respective mean number of sub-samples. We created a hierarchical index differentiating the subject and its respective sub-samples and made sure that splitting the data sets between train and test was done on the index related to the subject, rather than on the sub-sample. That way, if a (sub) sample is found in one data set, all samples from the same subject are also found in this data set.

3) Data transformation: We also investigated the distributions of each feature and observed that while most noncategorical features were Gaussian distributed, some were showing lognormal distributions (like the power spectral density features). We tried applying power transformation methods proposed by the Scikit-learn library ([9]) such as Box-Cox. Unfortunately, this didn't lead to any improvement for our model so we decided not to use it in further analysis.

4) Feature selection: Feature selection is useful to train the model on the features that are the most relevant to predict the labels. Selecting the features that meet this criterion make the model faster and more effective. A first analysis was performed to evaluate correlation between the features. The cross-correlation matrix can be found in Fig. 8. For the most part the dataset is only weakly correlated, or not at all. Even if there are a very few notable exceptions such as 'RMS Power' and 'Crest Factor', we decided not to remove any correlated (positively or negatively) features.

In addition, variance threshold was performed to remove low variance features, which did help the prediction. The best accuracy was performed using threshold t=0.1 thanks to the *VarianceThreshold* function (from Scikit-Learn [9]). The analysis removed such features like 'Zero Crossing Rate', Spectral Flatness' and 'Spectral Slope', passing the numbers of features from 71 to 56.

5) Over-sampling: 27% of the entire dataset consists of wet coughs (class 1), with the remaining samples being dry coughs. This class imbalance leads some models (such as a very basic XGBoost) to almost always label unseen data as the majority class. In order to avoid this we randomly sample the minority class with replacement until class balance is achieved. This lead to an improvement in the AUC score.

III. Classification models

A. Baseline classifiers

State-of-the-art models were implemented with Scikit-Learn ([9]) pipelines handling a standard scaler and the method at stake. The hyper parameters of the pipelines were then optimized by cross-validated randomized search. Group K-fold splitting was used as cross-validation strategy with the number of splits set at 5.

The models that were used were Decision Tree, Gaussian Naive Bayesian, Support Vector Machines, K-Nearest Neighors, Logistic Regression, Random forest and Linear Discriminant Analysis.

B. eXtreme Gradient Boosting for classification

EXtreme Gradient Boosting (XGBoost) is a library prividing powerful tools for machine learning [10]. XGBoost outperforms the competing candidates in the hyper optimization using RandomizedGridCV, and thus lends itself as the optimal choice for fine tuning. Furthermore, the best model is obtained using all features, including metadata, without transformation such as standardization. Bayesian optimization is implemented using the python API and the Hyperopt library ([11]). This implementation works by defining an objective function to be minimized, thus the best score is output with a negative value. As before, GroupKFold is passed to the cross validation function in order to avoid correlation bias.

IV. Results

A. Classification with baseline models

We get the accuracies (AUC Scores [%]) displayed in Fig. 1 for the different algorithms: Decision Tree, Gaussian Naive Bayesian, Support Vector Machines, K-Nearest Neighors, Logistic Regression, Random forest, Linear Discriminant Analysis and eXtreme Gradient Boosting (each time using the best algorithm hyperparameter tunning, running on test set).

B. Impact of expert bias

Next, we studied the impact of the expert bias on our running algorithms. We performed eXtreme Gradient Boosting on the fine segmented data set, with one expert at a time and then combining the samples from Experts 1 and 3, which get the highest accuracy score. It turned out that the model performed better when removing the samples from expert 2 (which gets the lowest accuracy score when considered alone).

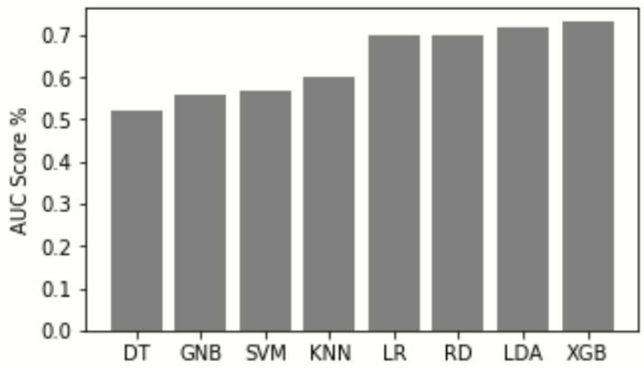

Fig. 1: AUC scores [%] computed with different models on raw data.

Expert	All	1	2	3	1 & 3
AUC score %	0.67	0.54	0.50	0.59	0.72

Fig. 2: Impact of experts bias on XGBoost perform, computing AUC scores %.

C. Impact of metadata

Next, we studied the impact of the metadata provided by the users. This definitely leads to improvements in the Accuracy of our model going up with 3% compared without metadata. Indeed, comparing each models with single metadata : 'Age', 'Gender', 'Resp. cond', and 'Sympt' with the model based without any metadata : 'None', we always get a higher or equal AUC score. This leads the final model to be based on every metadata feature which gets the highest score of 0.72% accuracy. Then we decided to keep metadata features in our training data set.

/	None	Age	Gender	Resp. cond.	Sympt.
Score	0.69	0.71	0.69	0.71	0.70

Fig. 3: Impact of metadata, using Experts, Respiratory Condition and Symptoms features as dummy variables, and computing AUC scores %.

D. Classification with XGBoost using Bayesian Optimization

We get the highest AUC score up to 0.746% with eXtreme Gradient Boosting Classifier using the following parameters:

- colsample-bytree: 0.5732957584868299
- gamma: 3.2697672617295446,
- learning_rate: 0.22855810864114873,
- max_delta_step: 10.0,
- max_depth: 40.0,
- min_child_weight: 19.0,
- n_estimators: 50.0,
- reg_alpha: 52.0,
- reg_lambda: 0.11594407730658135

We get the following results when applying the model to the different segmentation methods. We observe that fine segmentation has a much higher score with 0.746% accuracy.

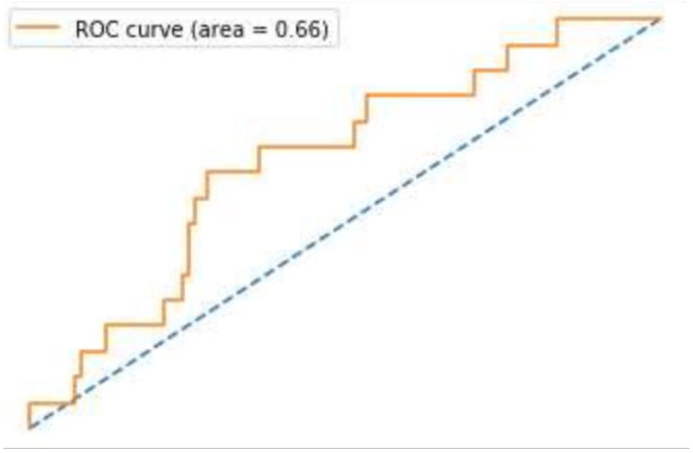

Fig. 4: ROC on unseen data

However, when testing on the unseen dataset provided by ESL ([4]), the AUC value drops to 0.66, as can be seen in 4.

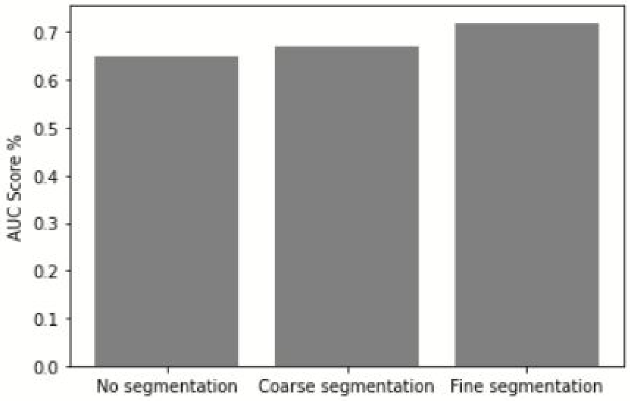

Fig. 5: AUC scores [%] computed on the different segmentations, performed with eXtreme Gradient Boosting.

V. Explanation of the result with SHAP values

A useful tool to determine the role of the features at explaining the model's output is SHAP values [12]. It shows the contribution of features at increasing and decreasing the prediction if considered separately. The SHAP values of the XGBoost model trained on the fine segmented data set after optimization are summarized in 6:

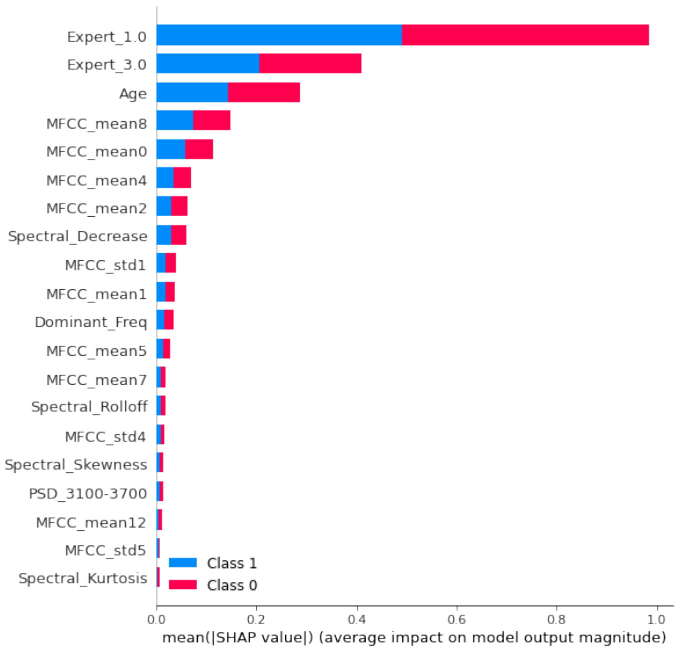

Fig. 6: SHAP values of the optimized XGBoost model

It suggests that expert 1 contributes more than expert 3. Apart from the expert feature that should be considered carefully, we can observe that some other features like age or signal characteristics such as MFCC should be investigated with more caution. We show in Fig. 7 that greater age tends to lower the prediction that ranges from 0 to 1, with 0 being a dry cough.

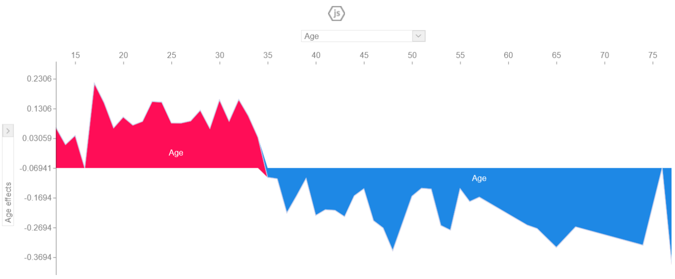

Fig. 7: SHAP values of the age effects

VI. Discussion and Conclusions

The aim of the this project has been to classify between wet and dry coughs, based on a dataset labeled by domain experts, and evaluating the performance of a variety of both traditional and novel machine learning implementations. All in all 8 models have been implemented, namely Decision Tree, Gaussian Naive Bayesian, Support Vector Machines, K-Nearest Neighbors, Logistic Regression, Random forest, Linear Discriminant Analysis and eXtreme Gradient Boosting.

Implementing feature preprocessing, such as standardization and power transform, leads to improved performance in some classical approaches but has little-to-no effect on the models based decision trees. Additionally, the decision tree models outperform the classical models substantially, with eXtreme Gradient Boosting displaying best performance overall. Finally, eXtreme Gradient Boosting is finely tuned using Bayesian Optimization, which is shows a CV score of 0.749. However, when tested on the final test set the best score obtained was 0.66.

The impact of more training data has a substantial effect on the CV-score. This is apparent when splitting into a training and validation set with a ratio of 0.2 prior to cross validating, which shows a difference in the final score of roughly 0.05. Additionally, despite the fact that the data was labeled by domain experts, improvements were possible by removing data labeled by one expert. The possibility of mislabeled data could be affecting the performance of the models.

For future work we suggest providing more testing data, and perhaps implementing some form of consensus decision on the final label in order to avoid miss-labeling. Additionally, Deep learning could possibly provide a better classification model to this problem.

References

[1] J. Widdicombe, “Neurophysiology of the cough reflex,” *European Respiratory Journal*, vol. 8, no. 7, pp. 1193–1202, 1995.

[2] J. Korpáš, J. Sadloňová, and M. Vrabec, “Analysis of the cough sound: an overview,” *Pulmonary pharmacology*, vol. 9, no. 5-6, pp. 261–268, 1996.

[3] R. S. Irwin, L.-P. Boulet, M. M. Cloutier, R. Fuller, P. M. Gold, V. Hoffstein, A. J. Ing, F. D. McCool, P. O’Byrne, R. H. Poe, *et al.*, “Managing cough as a defense mechanism and as a symptom: a consensus panel report of the american college of chest physicians,” *Chest*, vol. 114, no. 2, pp. 133S–181S, 1998.

[4] L. Orlandic, T. Teijeiro, and D. Atienza, “The coughvid crowdsourcing dataset: A corpus for the study of large-scale cough analysis algorithms,” *arXiv preprint arXiv:2009.11644*, 2020.

[5] A. Murata, Y. Taniguchi, Y. Hashimoto, Y. Kaneko, Y. Takasaki, and S. Kudoh, “Discrimination of productive and non-productive cough by sound analysis,” *Internal Medicine*, vol. 37, no. 9, pp. 732–735, 1998.

[6] K. F. Chung, D. Bolser, P. Davenport, G. Fontana, A. Morice, and J. Widdicombe, “Semantics and types of cough,” *Pulmonary pharmacology & therapeutics*, vol. 22, no. 2, pp. 139–142, 2009.

[7] V. Swarnkar, U. R. Abeyratne, A. B. Chang, Y. A. Amrulloh, A. Setyati, and R. Triasih, “Automatic identification of wet and dry cough in pediatric patients with respiratory diseases,” *Annals of biomedical engineering*, vol. 41, no. 5, pp. 1016–1028, 2013.

[8] H. Chatrzarrin, A. Arcelus, R. Goubran, and F. Knoefel, “Feature extraction for the differentiation of dry and wet cough sounds,” in *2011 IEEE International Symposium on Medical Measurements and Applications*, pp. 162–166, IEEE, 2011.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[10] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[11] J. Bergstra, D. Yamins, and D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures,” in *International conference on machine learning*, pp. 115–123, PMLR, 2013.

[12] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 4765–4774, Curran Associates, Inc., 2017.

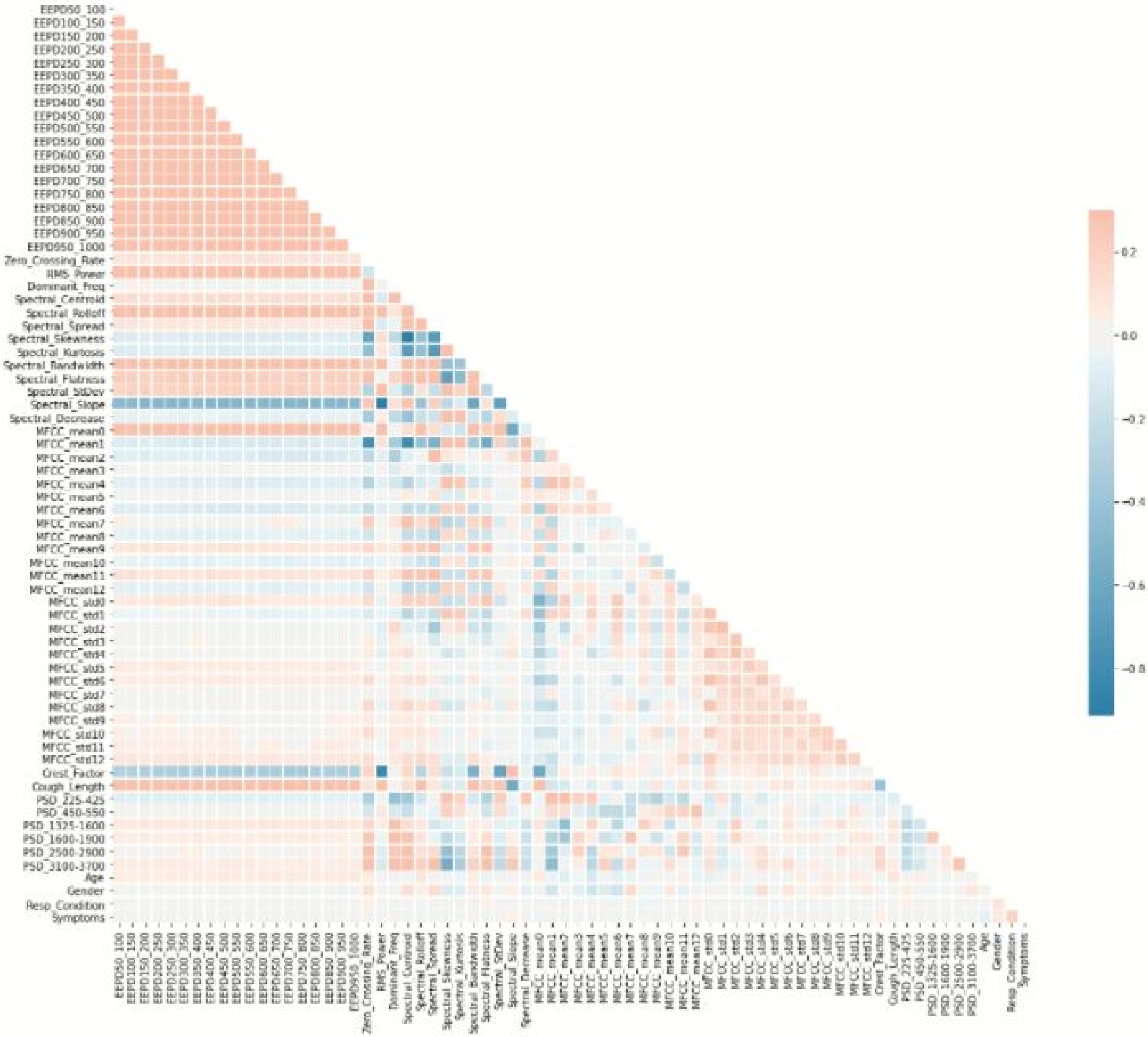

Fig. 8: Correlation matrix analysis on raw data.