Evaluation of Machine Learning Algorithms for Neuropsychiatric Systemic Lupus Erythematosus classification using resting-state fMRI connectivity data

Vasileios Vittis, Nicholas John Simos

Abstract

In this study we explored the robustness of different machine learning algorithms for the classification of NPSLE patients and healthy control subjects using resting-state fMRI connectivity indices. Neuropsychiatric systemic lupus erythematosus, which is a systemic autoimmune inflammatory disorder, involves a wide range of focal and diffuse central and peripheral nervous system symptoms. Machine Learning applications on clinical data may enhance the existing workflow for NPSLE classification as there is no established method of applying neuroimaging data to the diagnosis of NPSLE. We evaluated different ML algorithms: Support Vector Machines (SVM), K-Nearest-Neighbor (KNN), Naïve Bayesian and Random Forest in order to achieve this goal. Feature selection or reduction methods were applied prior to the classification process. The Connectivity Matrix used consisted of pairwise regional functional associations of the fMRI signals within each of three predetermined brain networks in 31 NPSLE patients and 31 healthy control subjects. Optimal, cross-validated classification accuracy was obtained using Random Forest (77%) followed by SVM and NB methods (74%). Results demonstrate the potential for the future implementation of similar methods in the diagnosis of NPSLE.

Keywords

NPSLE, resting state fMRI, functional connectivity, Support Vector Machine, Random Forest, K Nearest Neighbor, Naïve Bayes, Principal Component Analysis, Select from Model, Sequential Feature Selection, Pearson Correlation

Introduction

In recent years Machine Learning techniques have often been used for a wide range of classification tasks using resting-state functional connectivity (FC) fMRI data (Saccà et al. 2018; True et al. 2014).

Resting-state fMRI has received notable attention lately for its great potential of providing biomarkers useful in disease diagnosis. It is a non-invasive examination method based on the blood-oxygen-level-dependent signals. What differentiates resting-state fMRI from other common fMRI modalities is that the subjects are not performing any specific task during the examination, mental or otherwise.

FC can be considered a transformation of the original recorded fMRI time series, quantified by simple metrics such as correlation, covariance and mutual information of time series computed between different brain regions. In the data used in this study a set of brain networks were selected a priori and the FC matrices produced contain values corresponding to the functional connections of all the regions of interest (ROI's) within these networks. Methods using ML algorithms based on FC data have shown promising results in the diagnosis of other diseases such as Multiple Sclerosis (MS) (Saccà et al. 2018). However, research conducted using such algorithms on lupus and specifically NPSLE is almost nonexistent.

Systemic lupus erythematosus (SLE) is a chronic, multisystem, autoimmune disease. Depending on the type of manifestations included and the method used for evaluation, the frequency of neuropsychiatric (NP) manifestations varies widely across studies. Because of the lack of standardization techniques as well as NP manifestations often being due to secondary causes, the differentiation between SLE and NPSLE may be challenging. Over the last decades, several protocols have attempted to overcome this issue (Vivaldo et al. 2018). In the present study we aim to provide a method for the effective classification of NPSLE patients from healthy control subjects.

Support Vector Machines (SVM) (Cortes et al. 1995), K-Nearest-Neighbor (KNN) (Peterson 2009), Naïve Bayesian (Murphy 2006) and Random Forest (Breiman 2001) are some of the most robust and frequently used ML algorithms for classification tasks. The classification we aim to perform is a binary classification problem of which all the proposed methods are capable of. Furthermore, SVM has proven an effective classifier in similar rs-fMRI connectivity data sets (Amiraly et al. 2019; Khazaee et al. 2016).

Feature Reduction and Selection techniques were used in order to reduce the dimensions of the original feature vector, namely Pearson correlation coefficient ranking, sequential feature selection (mlxtend), SelectFromModel (scikit-learn) and Principal Component Analysis. For each algorithm both supervised and unsupervised reduction/selection techniques from the above were evaluated while seeking the best classification performance.

Methods

MRI acquisition

Resting state fMRI data was recorded during the period 2015-2016 in the MRI Unit, University Hospital of Heraklion. Brain MRI examinations were performed on a clinical, upgraded 1.5T Siemens Vision/Sonata scanner (Erlangen, Germany) with powerful gradients (Gradient strength: 45 mT/m, Gradient slew rate: 200 mT/m/ms) and a standard four channel head array coil (minimum voxel dimensions: 70 μm X 70 μm X 300 μm).

Resting-state functional MRI (RS-fMRI) was derived from a T2*-weighted, fat-saturated 2D-FID-EPI sequence with repetition time (TR) 3500 ms, echo time (TE) 50 ms, field of view (FOV) 192 x 192 x 108 (x, y, z), and acquisition voxel size 3 x 3 x 3 mm. Whole brain scans consist of 36 transverse slices with 3.0-mm slice thickness and no interslice gap. Each BOLD time series consists of 80 dynamic volumes. FC matrices were extracted using CONN (CONN functional connectivity toolbox) for three key brain networks and the networks were selected manually according to the suggestions of the healthcare professionals. Namely, Default Mode Network (containing 6 ROI's), Frontoparietal (14 ROI's) and Sensorimotor (3 ROI's). This resulted in a cumulative matrix of 109 FC values for each patient. These will eventually become the features used in the Machine Learning algorithms. The 109 values of the final matrix represent the connectivity values (or weights) between each of the ROI's inside the three networks selected.

The data in this study was acquired during the period 2015-2016 from 31 patients suffering from Neuropsychiatric systemic lupus erythematosus (NPSLE) and 31 healthy control subjects. The hospital review board approved the study and the procedure was thoroughly explained to all patients and volunteers, who signed informed consent before undergoing MRI. The data were provided in fully anonymized form by the Director of the MRI Unit, Dr. Efrosyni Papadaki.

Feature Selection Methods

The dimensionality of the data required considerable reduction in order for the classification algorithms to perform optimally (Guyon et al. 2003). The curse of dimensionality was an issue in our feature set as the original features were 109 with only 31(NPSLE) + 31(HC) subjects (samples). Optimally, the number of features used should be smaller than the number of subjects. This problem was also obvious in the preliminary testing conducted without the application of a selection/reduction method where the classification accuracies using all 109 features were very poor.

The methods tested for supervised feature selection were Pearson correlation coefficient ranking, sequential feature selection (mlxtend)/SelectFromModel (scikit-learn), and PCA. Feature selection using Pearson entails the computation of the Pearson correlation coefficient for each feature across all samples and labels of the samples. The produced array contains the Pearson correlation coefficient for each one of the 109 features with the dependent variable (clinical group).

Sequential forward selection (mlxtend), similar to SelectFromModel (scikit-learn) and Sequential feature selection (Matlab)) is a recursive feature selection algorithm that modifies the number of features in each run until the termination criteria are met. One of the termination criteria is the preselected number of features to select. These selection methods utilize a classification function such as SVC (sklearn.svm.LinearSVC) or RF and are used to evaluate classification performance prior to the final classification. In each run of the selection algorithm the features selected are rearranged (or re-selected) based upon preliminary classification results. The classification score represents a feature importance measure used by the algorithm. All labeled/supervised selection methods were computed for the train data only without using information from the test data in order to avoid overfitting the model. For instance, in the case of Leave-one-Out cross validation the feature selection is done inside the for sequence for each changing train data set. By performing selection in this manner, biasing the classification model is avoided.

The risk of overfitting is not present for unsupervised methods such as PCA. For such methods, the transformation was done initially before the cross-validation process on the entire dataset. PCA aims at mapping the data to a lower-dimensional space. This is accomplished by selecting the “best” eigenvectors of the covariance matrix of the original features. “Best” meaning the eigenvectors corresponding to the highest eigenvalues. The selected eigenvalues are called Principal Components. They are the projections of the original data with the highest variance. In this manner, PCA identifies the most significant patterns in the data so as a result a reduction in dimensionality is possible without significant loss of information. The MATLAB PCA algorithm was applied on the whole dataset using the built-in implementation, the same results were also achieved by manually computing the covariance matrix and sorting the eigenvectors by their respective eigenvalues (in MATLAB). In the results section we present classification results using different numbers of principal components used in the classification process.

The reduced feature set derived from each of the above methods were used as inputs in the classification algorithms discussed below. All selection/reduction methods were tested with each classifier as each classifier often shows different results depending on the feature transformation method applied.

Classification Methods

SVM

SVM (Cortes et al. 1995) is a widely used Machine Learning algorithm applied to a wide variety of biomedical classification problems including EEG classification, cancer identification and seizure prediction. Classification using SVM entails assigning all samples to two disjoint hyperplanes in a manner that maximizes the margin between the two hyperplanes. For two-dimensional data that is linearly separable this is an easy task. When the data is more complex and of higher dimensionality such as biomedical data, a linear approach may not be appropriate.

In the present study SVM was implemented using the scikit-learn available in Python. The radial basis function (RBF) kernel was selected over linear or polynomial kernels as it is best suited for higher dimensional data that are not linearly separable. The linear and polynomial kernels were also tested but produced very poor classification outcomes. The RBF kernel is a popular choice for SVM implementation due to its satisfactory results. The function of this kernel is the following:

$$\mathrm{K}(\mathrm{xi},\mathrm{xj}) = e^{\frac{-1}{2}\left(\frac{\chi-\mu}{\sigma}\right)^2}$$

where χ, μ are two data samples represented as feature vectors and $(\chi - \mu)^2$ is the squared Euclidian distance between χ and μ. All other parameters were set at the default values except for "C" and "gamma", which represent the penalty parameter and rbf kernel coefficient respectively. C controls the trade-off between correct classification of training examples against maximization of the decision function's margin. For instance, a higher value will allow for a smaller margin to be accepted if the decision function classifies all the training samples correctly. While a lower value will ensue a larger margin and as result a simpler decision function while possibly lowering training accuracy. Furthermore, gamma defines the amount of influence a single training sample will have on the model. The influence of these parameters on the overall classification accuracy was tested and is shown in the results section.

KNN

The KNN algorithm has been widely used for classification problems where there is little or no prior knowledge of the data distribution (Peterson 2009). The need for an algorithm to perform discriminant analysis when the probability densities of the data are unknown, was one of the main reasons that we chose to implement KNN. In our case, although feature selection methods were applied to the data, the dimensionality remained high. For this reason, a method that implements a basic concept such as Euclidean distance was considered as ideal. K-nearest-neighbor classifier, based on the input testing data x_i, takes the k closest training samples (points) using Euclidean distance $\|xi - xk\|$ and starts a voting process. In this process every training sample which belongs to this set gives its 'vote'. To be more specific, firstly, the algorithm takes the Euclidean distances from the given testing sample to every training sample and secondly it sorts them by ascending order. According to the value of k, a parameter of the classification function, KNN selects the k smallest values (Fig. 3). As for the 'voting' process, we know the prior label of each sample (supervised problem), so there is one to one correlation between samples and labels. The algorithm, for all the labels of the training data that belong to the set of k-closest neighbors y_k checks which label appears with the highest frequency. The label that collects the most votes is more likely to be the label of the testing data.

Finally, we compared the true label of the testing sample with the predicted label. The ratio of correct matches over the total attempts represents the model's classification accuracy. In our case, KNN algorithm was applied with a k value of 5, which is the value corresponding to the best observed classification accuracy.

RF

Random Forest is a combination of tree predictors such that each tree depends on the values of a random feature vector sampled independently and with the same distribution for all trees in the forest (Breiman 2001). RF is a widely used classification algorithm due to its flexibility and ease of use. In our case, we used it in both the classification and feature selection stages because it provides substantial classification separability as well as indices of relative feature importance. The random forest algorithm is not biased since there are multiple trees and each tree is trained on a subset of the data. Bias is reduced because the algorithm relies on the power of "the crowd".

In a decision tree the aim of each layer is to associate specific targets with specific values of a particular variable. A decision tree indicates a combination of values associated with a particular prediction ending up to leaf nodes, representing a class. Each non-leaf node (decision node) is a decision marker. Decisions regarding a particular feature at a given decision node are based on a threshold derived from a cost function (Gini) – a measure of inequality of a distribution. It is defined as a ratio with values between 0 and 1.

$$\text{Gini index} = 1 - \sum_{i \neq j} p(i) \cdot p(j)$$

A split is decided depending on the outcome of each decision node; this split determines if we continue our path to either the left branch or the right branch of this node. By randomly selecting the next feature and repeating this process, we construct a classifier (i.e. decision tree). Specifically, the number of decision trees (estimators) has a big impact on the classification performance. In our case, the RF algorithm was implemented with 60 estimators. A schematic illustration of the RF algorithm is shown in Fig 1.

After establishing all the decision trees based on the aforementioned process, we can start feeding test samples. Every testing sample x_i traverses all the possible constructed decision trees and ends up collecting 'votes' by the leaf nodes to whether it belongs to some class. The class receiving the majority of 'votes' is probably the proper class for the testing sample with a given probability. As we have previously discussed each sample comes with its label, so the predicted label which represent a class was compared with the true label of testing sample. The combination of correct and incorrect predictions corresponds to the overall classification score.

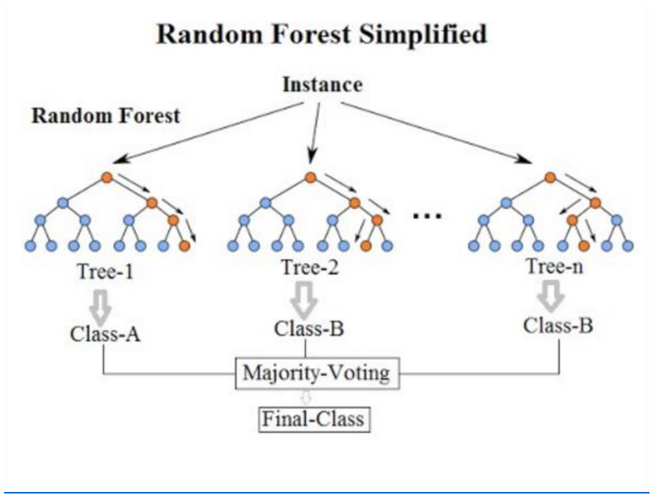


Fig 1. Structure of Random Forest

NB

Naïve Bayes classifiers are a family of supervised probabilistic classifiers based on Bayes' theorem with strong independence assumptions between features: $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

In spite of their simplicity, Naïve Bayes classifiers have been found to work well in real-world situations. Also, they are known to scale well to higher dimensional data such ours and are also very fast to train. Furthermore, concerning the theoretically limiting and challenging statistical independence assumptions made by the classifier, studies have shown that it can still be optimal if the dependencies distribute even in classes or if dependencies are present, but cancel each other out (Zhang 2004).

Classification Performance Metrics

Several indices were employed to compare classification methods. Apart from the accuracy score, which is calculated as the ratio of correctly classified samples by the total number of samples, showing the overall accuracy of the method, other metrics of classification performance were also used for the evaluation of the algorithms. In order to calculate these metrics, the number of True Positive (tp), False Positive (fp), False Negative (fn) and True Negative (tn) samples were computed. The sensitivity or Recall (true positive rate) of a test is its ability to determine the patient cases correctly. This is also obvious through its calculation: $TPR = \frac{tp}{tp+fn}$ (Baratloo at al. 2015). In disease classification studies, this is a very important measure. The misclassification of a patient as healthy is quite serious and is desirable to be as low as possible. This is estimated through the false negative rate or *1 – TPR*. It is because of this that Recall is considered of high importance in the evaluation of each classification method, especially in Biomedical problems such as this. The opposite, misclassifying a healthy subject as a patient is not as severe, it is measured by the inverse of Specificity (*1-TNR*). Specificity is the True Negative rate, $TNR = \frac{tn}{tn+fp}$ and is the ability of a test to determine the healthy cases correctly. Precision (positive predictive value) is calculated as: $PPV = \frac{tp}{tp+fp}$. Precision can be interpreted as the ability of the classifier not to label as positive a sample that is negative. The F1 score can be thought as a combination of Sensitivity and Precision as it is calculated using these two metrics: $F1 = \frac{PPV \times TPR}{PPV+TPR} = \frac{2tp}{2tp+fp+fn}$, as the harmonic average of precision and recall and is often used in Machine Learning applications as it provides a combined metric.

Results

Feature selection

Figures 2-10 illustrate the application of various feature selection approaches for hyperparameter tuning.

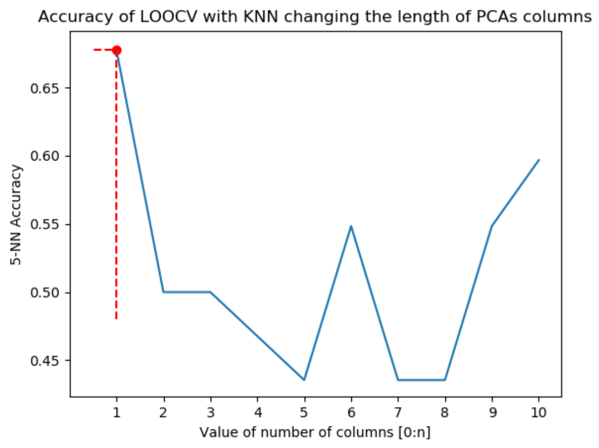


Fig 2. KNN. Accuracy across different number of PCA components.

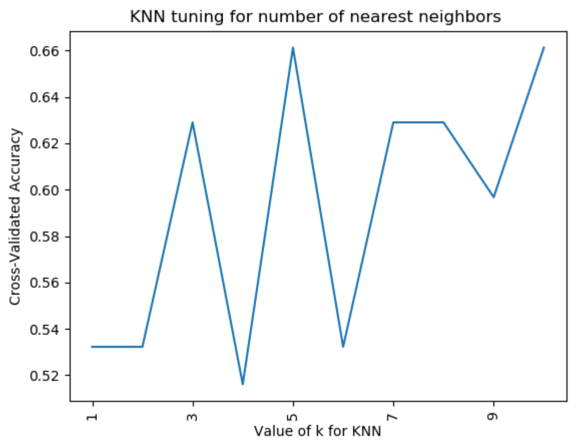


Fig 3. KNN. Accuracy across the optimal value of k parameter with SFS (number of neighbors).

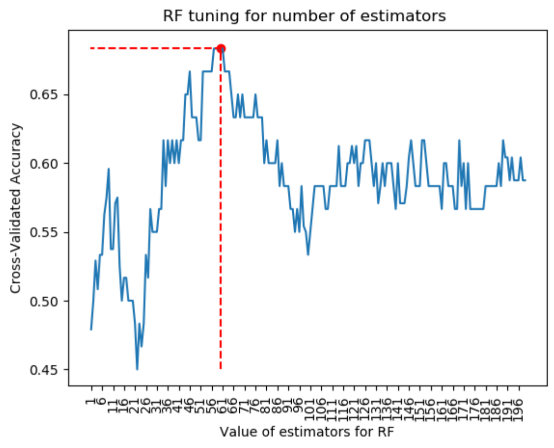


Fig 4. RF. Accuracy based on the number of decision trees.

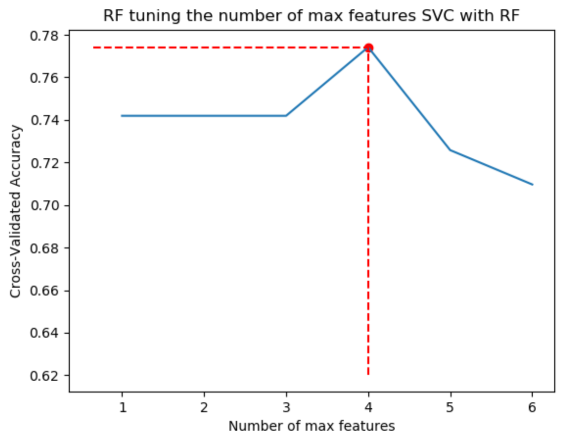


Fig 5. RF Max features of SFS.

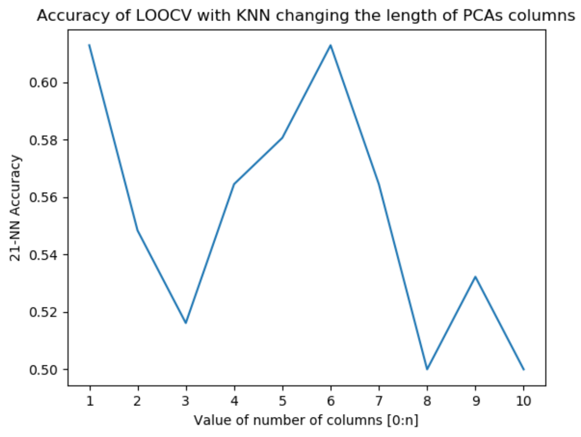


Fig 6. RF. Accuracy across different number of PCA components.

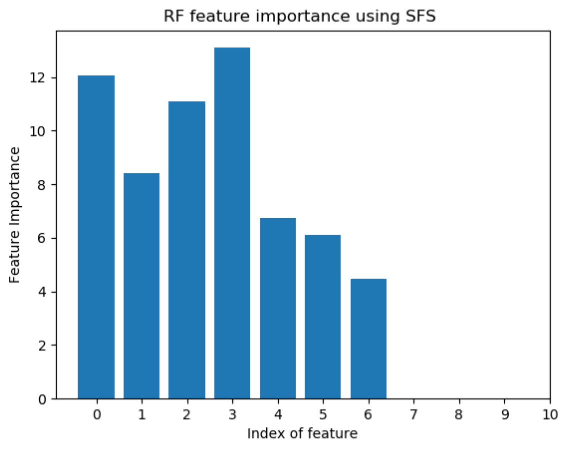
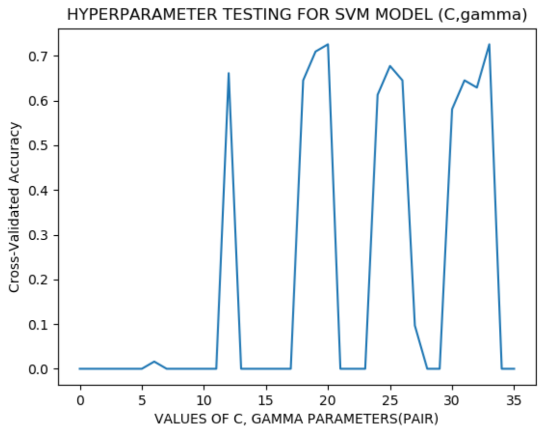


Fig 7. RF. Feature importance cumulative graph.



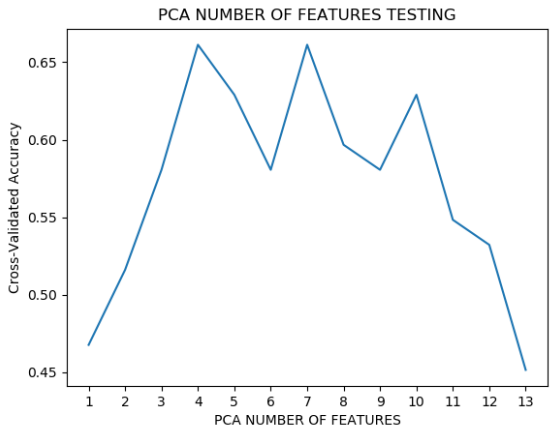


Fig.8 SVM (Pearson). Hyperparameter tuning, horizontal axis represent pairs of C and gamma values.

Fig.9 SVM (Pearson). Accuracy across different number of PCA components.

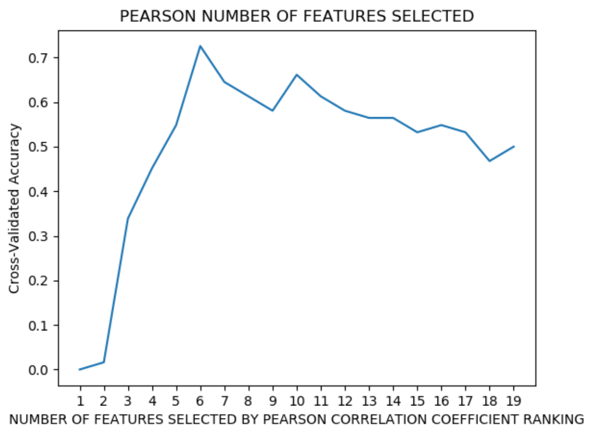


Fig.10 SVM. Accuracy scores for increasing number of Pearson correlated features selected

Cross Validation Results

As mentioned above, four different machine learning classifiers were trained with functional connectivity-based data using three different feature selection methods. Several cross-validation techniques were evaluated, including Leave-One-Out, Leave-p-Out, k-fold and stratified k-fold. Optimal results, based on accuracy scores, were obtained with the Leave-One-Out method (see Tables 1-2 and Fig. 11). This method is also more suitable for the relatively small sample size of the present data set.

Among all classifiers, Random Forest with SelectFromModel selection method has the best accuracy and recall, 77% and 87% respectively. Random forest has the lead in terms of accuracy because it generates an internal unbiased estimate of the generalization error as the forest building progresses. On the other hand, KNN method has the worst accuracy on average from all the other classification methods proving that it is an inefficient method to be used on this problem.

The SVM classifier showed respectable classification accuracies especially in the two supervised methods with scores of up to 74.2% combined with the SelectFromModel selection method. The Recall and Precision scores of SVM with the supervised selection methods were also good and as high as 77% with Pearson feature selection method.

The NB classifier performed surprisingly well, meeting and surpassing the results of SVM while using the supervised selection methods. As mentioned above regarding the nature of NB, it is often characterized by high scalability to higher dimensional data such as in our case, sometimes outperforming “sophisticated” classification methods. The NB Recall scores were also acceptable at 74.2% with 74.2% accuracy. NB outperformed SVM in Recall results which as mentioned above are of high importance in disease classifications such as this. It was surpassed only by RF.

Table 1: Leave One Out Cross Validation Accuracy and Recall (LOOCV) for all combinations of selection and classification methods

	Accuracy			*Recall (Sensitivity)*		
Method	PCA	SFS/SFM	Pearson	PCA	SFS/SFM	Pearson
kNN	59.67%	66%	64%	51.6%	61%	64%
RF	53%	77%	66%	45%	87%	67%
SVM	66.1%	74.2%	72.6%	48.4%	71%	77.4%
NB	62.9%	72.6%	74.2%	48.4%	74.2%	71%

Table 2: Leave One Out Cross Validation Precision, F1 score, Specificity (LOOCV) for all combinations of selection and classification methods.

	Precision			*F1 score*			*Specificity (Selectivity)*		
Method	PCA	SFS/SFM	Pearson	PCA	SFS/SFM	Pearson	PCA	SFS/SFM	Pearson
kNN	61.5%	67%	64%	56.1%	64%	64%	67.7%	70%	64%
RF	53%	72%	65%	49%	79%	66%	61%	67%	64%
SVM	75%	75.9%	70.6%	58.8%	73.3%	73.8%	83.9%	77.4%	67.7%
NB	68.2%	71.9%	75.9%	56.6%	73%	73.3%	77.4%	71%	77.4%

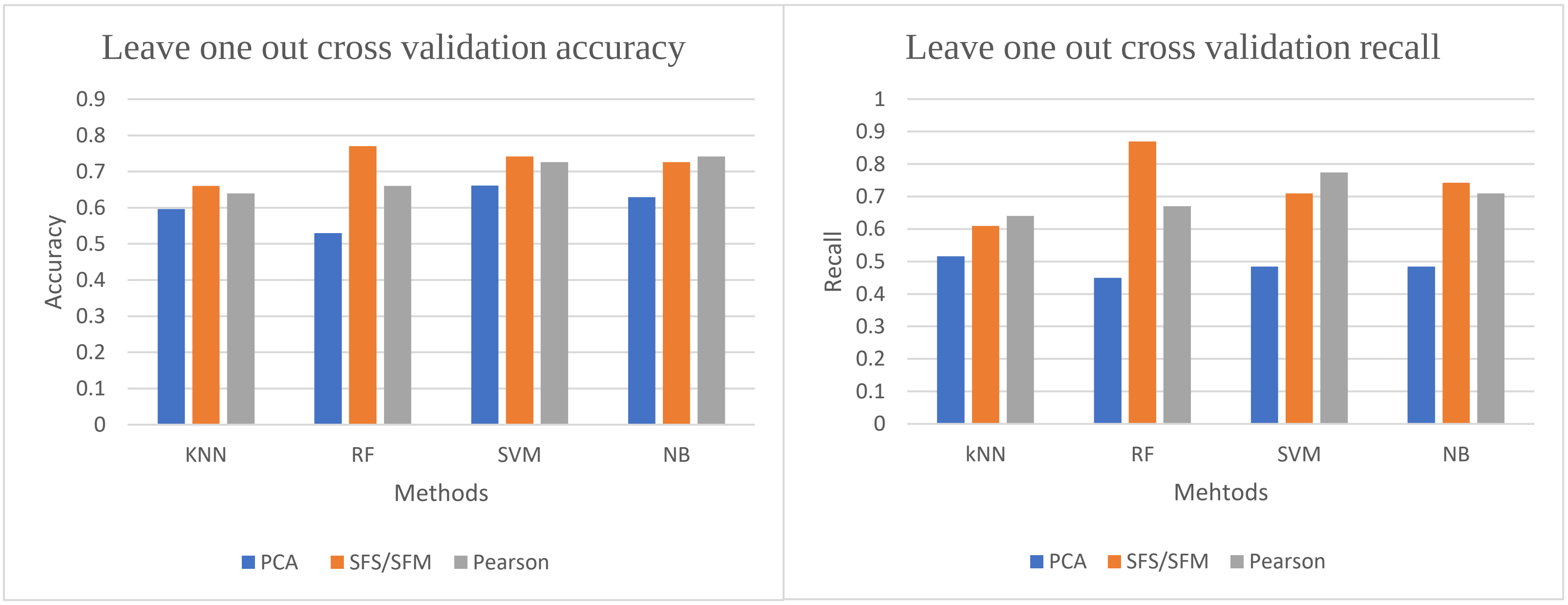


Fig 11. Accuracy (left panel) and recall (right panel) associated with each classification method in the leave one out scheme.

Discussion

Parameter Tuning of Classifiers

During the implementation process, certain challenges were met while tuning the hyperparameters of each classifier. After applying exhaustive testing methods, some conclusions were formed considering the optimal values of each classifier.

Regarding the KNN method, the most important values that we examined, were the number of neighbors (k-value), despite the feature selection, and the PCA feature vector size. Testing one parameter at a time, it can easily be observed in fig.1 that the optimal value for k is 9. We know that a small value of k means that noise will have a higher influence on the result and a large value will make it computationally expensive. Data scientists usually prefer an odd number, so as to avoid ties. (Peterson 2009). In our case, the number of classes are two so a simple approach to select k is the following: $k = \sqrt{number\ of\ samples} = \sqrt{61} = 7.8102$. So, we are looking for an odd number near 7.8. By observing Fig.1 we can conclude that the number of 5 neighbors would be suitable. Secondly, as already mentioned, through PCA an eigenvalue has been assigned to each eigenvector. After examining the offered information of a new added feature to the feature vector, we conclude that the first 14 features were enough for both low informational loss and low computational demand. So, it is obvious enough that the eigenvector (i.e. feature) with the bigger eigenvalue will carry information of higher importance compared to the others. After conducting an exhaustive search, we found that the feature with the highest eigenvalue is sufficient to separate the two classes with a cross validated score of 66.13% in KNN method. This outcome can be explained by randomness of the single feature and in no case, could be generalized as a rule.

Moreover, Random Forest has more or less the same number of parameters available for tuning but in a different manner than KNN. In RF, the two most important parameters that can be tuned base on the data, are the number of estimators (i.e. decision trees) and the maximum of number features in every decision tree. The remaining parameters such as max depth were set to default values based on the recommendations of sklearn. The maximum features parameter can be interpreted as the number of features to consider when looking for the best split. For every decision tree, the number of maximum used features is calculated by the square root of the feature vector size, resulting $\max\text{features} = \log_2 number\ of\ features = \log_2 7 = 2.8 \cong 3$. As shown in Fig.2 every decision tree should have a number close to three, in our case four features were used in order to make a decision. Also, every decision tree reaches the same cross validation accuracy for every number greater than seven. This can be explained from the fact that RF does not need any extra depth in order to effectively separate the classes.

Regarding the hyperparameter tuning of the SVM classifier, this was completed based on the results shown in Figure 8. The horizontal axis was set depict pairs of C and gamma values for ease of visualization, the ranges used for the parameters were 0.1 – 100 and 0.1 – 0.00001 respectively. These were deemed reasonable values by studying similar techniques conducted through various algorithms for Hyperparameter optimization. The classification scores observed are for 35 value pairs in total hence the 1-35 range of the horizontal axis. For each value in the range of C, each value from the range of gamma is tested in an exhaustive manner. The zero classification scores are due to incompatible pairs of values within the two ranges. The two highest scores correspond to the following pairs of values:

Pair 20: C = 10, gamma = 0.03, average classification accuracy observed = 72.58%
Pair 33: C = 100, gamma = 0.001, average classification accuracy observed = 72.58%

Similar processes were performed in the case of the SVM classifier for the remaining two feature selection/reduction methods. These processes resulted in the optimal (to some extent) classification results for the particular dataset.

Feature Selection and Reduction Methods

Overall PCA did not lead to the best classification results while paired with most of the classifiers tested except from KNN. This was to be expected due to the nature of the data and the unsupervised and linear nature of PCA. It has been shown from similar studies that PCA often fails to capture the complexity of the human brain connectome (He et al., 2018). Furthermore, PCA led to the lowest Recall and F1 scores, deeming it practically unusable for this particular classification task.

The two supervised (labeled) feature selection methods performed adequately across most classifiers especially when compared to PCA. The Pearson selection method, is solely based on choosing the features with highest correlation to the labels while SFS/SFM keeps/selects the features that perform the best preliminary classification results. As derived from the results, simply choosing or selecting the most suitable and important features without performing any kind of vector transformation (in the case of PCA) is able to lead to better overall results.

Conclusion and Future Work

Overall, the best performing method was Random Forest resulting in the highest accuracy (77%), Recall (87%) and F1 score (79%). These results indicate the existence of biomarkers in the Networks studied, able to distinguish NPSLE patients from healthy subjects with considerable accuracy. In conclusion, the proposed method could potentially provide physicians with useful insight in the diagnosis NPSLE.

As a continuation of the work presented here, where various ML algorithms were tested for their robustness on this classification problem, we would also aim to test the effectiveness of Neural Network techniques and Gaussian Mixture Models on this data set. Furthermore, the proposed method could be potentially effective in discriminating between NPSLE patients and Multiple Sclerosis (MS) patients, presenting a significant diagnostic problem in clinical practice. With the methods used currently this is often challenging so healthcare professionals and patients could benefit from an improved pipeline for diagnosis. Provided such data is available this would be a possible follow up study. Finally, the incorporation of additional features derived from fMRI time sequences as well as other clinical data for each patient, could potentially increase the classification strength of the proposed method. This also poses a possible point of advancing the work presented.

References

Saccà, V., Sarica, A., Novellino, F. et al. Brain Imaging and Behavior (2018). https://doi.org/10.1007/s11682-018-9926-9

Price T., Wee CY., Gao W., Shen D. (2014) Multiple-Network Classification of Childhood Autism Using Functional Connectivity Dynamics. In: Golland P., Hata N., Barillot C., Hornegger J., Howe R. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014. MICCAI 2014. Lecture Notes in Computer Science, vol 8675. Springer, Cham

Vivaldo JF, de Amorim JC, Julio PR, de Oliveira RJ, Appenzeller S. Definition of NPSLE: Does the ACR Nomenclature Still Hold?. *Front Med (Lausanne)*. 2018;5:138. Published 2018 May 31. doi:10.3389/fmed.2018.00138

Cortes, C. & Vapnik, V. Machine Learning (1995) 20: 273. https://doi.org/10.1023/A:1022627411411

Leif E. Peterson (2009) K-nearest neighbor. Scholarpedia, 4(2):1883.

Kevin P. Murphy (2006) Naïve Bayes classifiers.

Breiman, L. Machine Learning (2001) 45: 5. https://doi.org/10.1023/A:1010933404324

Kazeminejad, Amirali & Sotero, Roberto. (2019). Topological Properties of Resting-State fMRI Functional Networks Improve Machine Learning-Based Autism Classification. Frontiers in Neuroscience. 12. 1018. 10.3389/fnins.2018.01018.

Khazaee, A., Ebrahimzadeh, A. & Babajani-Feremi, A. Brain Imaging and Behavior (2016) 10: 799. https://doi.org/10.1007/s11682-015-9448-7

Guyon, I., Elisseeff, A., & Kaelbling, L. P. (Ed.). (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research, 3*(7-8), 1157-1182. http://dx.doi.org/10.1162/153244303322753616

Zhang, Harry. (2004). The Optimality of Naive Bayes. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004. 2.

Baratloo A, Hosseini M, Negida A, El Ashal G. Part 1: Simple Definition and Calculation of Accuracy, Sensitivity and Specificity. *Emerg (Tehran)*. 2015;3(2):48–49.

He, Lili & Li, Hailong & Holland, Scott & Yuan, Weihong & Altaye, Mekibib & Parikh, Nehal. (2018). Early prediction of cognitive deficits in very preterm infants using functional connectome data in an artificial neural network framework. NeuroImage: Clinical. 18. 10.1016/j.nicl.2018.01.032.