

An Evolutionary Approach for fMRI Big Data Classification

Amirhessam Tahmassebi[†], Amir H. Gandomi^{*}, Ian McCann[†], Mieke HJ Schulte[§], Lianne Schmaal[¶],
Anna E. Goudriaan[§], and Anke Meyer-Baese[†]

[†]Department of Scientific Computing, Florida State University, Tallahassee, Florida 32306-4120, USA
Email: at15b@my.fsu.edu , Web: www.amirhessam.com

^{*}BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, Michigan 48824, USA
Email: gandomi@msu.edu

[§]Amsterdam Institute for Addiction Research, Academic Medical Center, University of Amsterdam, Netherlands
Email: a.e.goudriaan@amc.uva.nl

[¶]Center for Youth Mental Health, The University of Melbourne, Melbourne, VIC, Australia

Abstract—Resting-state function magnetic resonance imaging (fMRI) images allow us to see the level of activity in a patient’s brain. We consider fMRI of patients before and after they underwent a smoking cessation treatment. Two classes of patients have been studied here, that one took the drug N-acetylcysteine and the ones took a placebo. Our goal was to classify the relapse in nicotine-dependent patients as treatment or non-treatment based on their fMRI scans. The image slices of brain are used as the variable and as results here we deal with a big data problem with about 240,000 inputs. To handle this problem, the data had to be reduced and the first process in doing that was to create a mask to apply to all images. The mask was created by averaging the before images for all patients and selecting the top 40% of voxels from that average. This mask was then applied to all fMRI images for all patients. The average of the difference in the before treatment and after fMRI images for each patient were found and these were flattened to one dimension. Then a matrix was made by stacking these 1D arrays on top of each other and a data reduction algorithm was applied on it. Lastly, this matrix was fed into some machine learning and Genetic Programming algorithms and leave-one-out cross-validation was used to test the accuracy. Out of all the data reduction machine learning algorithms used, the best accuracy was obtained using Principal Component Analysis along with Genetic Programming classifier. This gave an accuracy of 74%, which we consider significant enough to suggest that there is a difference in the resting-state fMRI images of a smoker that undergoes this smoking cessation treatment compared to a smoker that receives a placebo.

I. INTRODUCTION

Smoking cigarettes lead to illnesses such as heart disease, strokes and cancer. Smoking is the leading cause of preventable mortality in the United States with around 50% of lifelong smokers dying from one of the illnesses mentioned earlier [1] [2]. What drives people to continue smoking cigarettes is the nicotine dependency that smoking causes them to have. This dependency drives them to compulsively have to smoke in order to keep the withdrawal effects associated with smoking cessation away. Developing a cessation treatment that will reduce a patients dependency on nicotine as well as reduce the effects of withdrawal could help millions of people quit a dangerous habit. Previously, genetic algorithm has been

used widely for automatic segmentation of 3D MRI data [3], fuzzy feature selection of fMRI data [4], and dynamic casual modeling of fMRI data [5].

In this paper, as a new approach for classifications, Genetic Programming (GP) [6] has been applied to analyze data from a smoking cessation treatment, where subjects take a drug to reduce their nicotine dependence while still being allowed to smoke in order to keep off the effects of withdrawal. This is the preferred method as more people are likely to try it if they do not have to quit smoking immediately. The goal is to reduce the nicotine dependency to the point that it is easier for the subject to stop. The purpose of this paper is to prove that there is a difference in the resting-state [7] functional magnetic resonance imaging (fMRI) images of a smoker that undergoes this smoking cessation treatment compared to a smoker that receives a placebo. Functional Magnetic Resonance Imaging (fMRI) is a set of noninvasive techniques for functional brain mapping. By taking snapshots of the brain through time, we will come up with movies of the brain which would help us to find out which parts of the brain are activated after receiving the treatments [8] [9]. Areas of high activity are defined to be those where more oxygen-rich blood is flowing [10] [11] [12] and the fMRI is able to map these areas. Smokers should have similar networks in areas of the brain where addiction occurs. Therefore, it should be noticeable if there is a change in these areas in the brains of subjects who underwent the treatment. We have compared some machine learning algorithms [13] [14] with Genetic Programming to classify the subjects on whether or not they underwent the treatment. The accuracy of classification will rely heavily on how the data is reduced. Techniques on how to accomplish that will be discussed.

II. SUBJECTS & FMRI DATA

The data used in this research was obtained from a study done by the Amsterdam Center for Addiction and Research. The study consisted of 39 subjects who were regular smokers and had a nicotine dependence. The study sought to determine if administering the drug N-acetylcysteine (NAC) to a subject

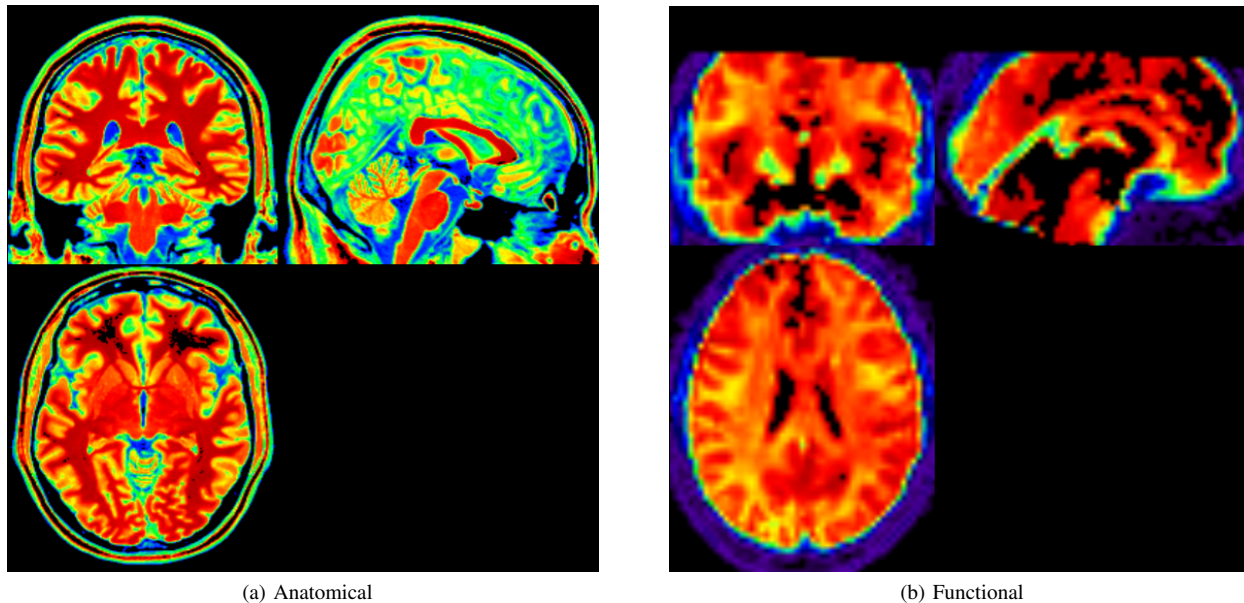


Fig. 1. Anatomical and Functional Image Slices of Brain of Subject.

would decrease their dependence on nicotine. In total 39 subjects partook in the experiments with 19 receiving the drug NAC and 20 receiving a placebo. All the subjects underwent a session of anatomical and functional scans of their brains before and after the experiment. This MRI data was obtained using a 3.0 T Intera MRI scanner (Philips Health care, Best, The Netherlands) equipped with a SENSE eight-channel receiver head coil. A gradient-echo planar image sequence sensitive to blood oxygen level-dependent contrast took 200 3-dimensional temporal images of the subjects brains. The subjects were asked to relax, keep their eyes closed and stay awake during the scan. The fMRI works by combining a strong magnetic field with radio waves in order to flip the magnetic fields of the nuclei in oxygen-rich blood. This allows us to create detailed maps of the flow of oxygen-rich blood to the brain, which we relate to areas with high activity. The measurement of blood flow, blood volume and oxygen use is called the blood-oxygen-level-dependent (BOLD) signal and this signal is what we study in this paper. The 3-dimensional anatomical data obtained from this was of size $240 \times 240 \times 220$ with a voxel size of 1mm. The 3-dimensional functional data was of size $80 \times 80 \times 37$ with a voxel size of 3mm. Figure 1 shows slices of the brain from one patient in all three axes for both the anatomical (a) data and the functional(b).

III. DATA PRE-PROCESSING

We are given the fMRI data in NIFTI (Neuroimaging Informatics Technology Initiative) formats which contains spatio-temporal slices. Due to the long process of the scans, and possible movements of our subject, the data will have artifacts. The fMRI data was analyzed using Statistical Parametric Mapping (SPM) and FMRIB Software Library (FSL). By getting the data into pre-processing phase, we are increasing the

BOLD contrast to noise. We start with motion correction, since the movement of the subject for 1% of the voxel size would make 1% change in the signal. This change can be greater than the BOLD signal we are going to extract as a features. The final voxel would not be as the same previous voxel. This is important due to the sensitivity of the statistical analysis of the residual noise in the image series. Then the images underwent segmentation, and realignment. In addition to this, scanners might acquire slices in interleaved fashion to avoid interfering with neighboring. Thus, temporal slice timing correction is needed. We need to correct and shift the slices back in order. The middle of the sequence is not necessarily middle of the brain. Regarding spatial normalization and spatial smoothing a Gaussian Full Width Half Maximum (FWHM) kernel has been employed. A kernel of 3 mm has been chosen for each voxel to be replaced by the weighted average of its neighbors. High pass or low pass filters based on the frequencies would be used for temporal filtering. Finally, we need to map functional and anatomical scans into a brain template to start analyzing these slices. To remove linear trends in each session of 200 images, the function data were band-pass filtered and de-trended. Then we have applied a mask [15] which is a 3-dimensional array of 0s and 1s, where a 1 signifies to keep the voxel in that position, and 0 indicates to ignore it on our data. Our mask is related to the parts of the brain that have to do with addiction. We have chosen to look at the limbic system which is where addiction occurs. Observing the average 3-dimensional image obtained earlier, we have constructed a 3-dimensional rectangular box around where we believe the average patient's limbic system was. Next, we wanted to only look at voxels that had "high activity". We considered this to be the voxels that had the highest values. We decided to reduce the matrix by 40% of its original size, so we found the highest voxel values that would

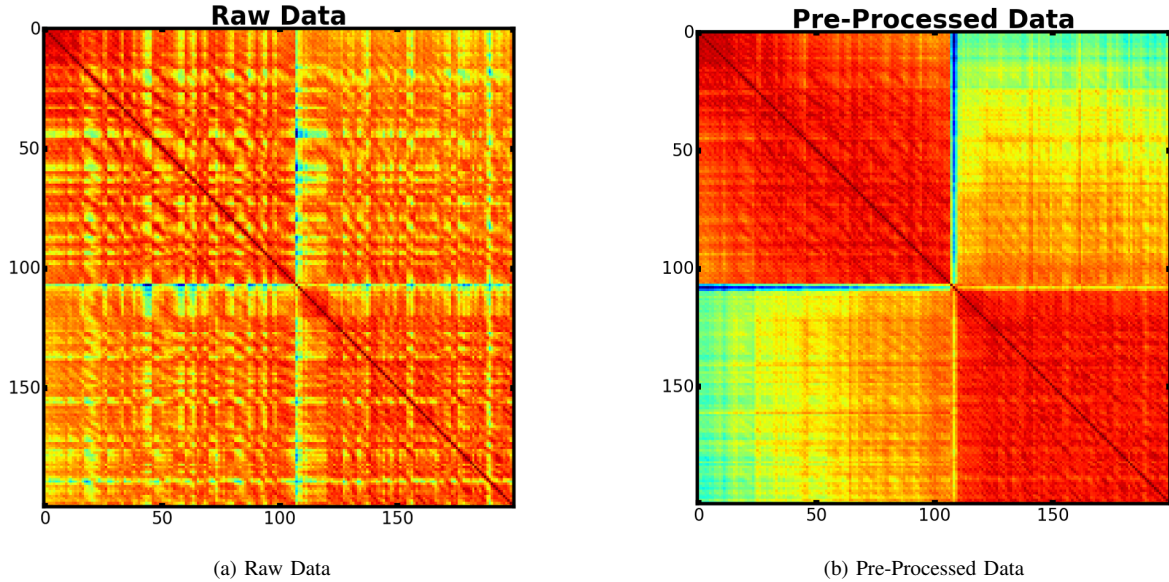


Fig. 2. Functional MRI Data

account for 40% of the size of the matrix. Fig. 2 demonstrates the correlation matrix of raw (a) and pre-processed (b) data.

IV. DATA REDUCTION

We are dealing with big data problem here which means it would be hard to come up with results with available computational equipment and also it will be computationally expensive. Our pre-processed matrix would be $39 \times 9.472 \times 10^9$ which is a huge number for feature vector. As we have stated before, we have 200 temporal snapshots before the treatment, and 200 after the treatment. Here, the average of temporal parts has been used. After this the size of the matrix would be $39 \times 236,800$. Then, a mask to extract the top 40% of voxels has been applied. The final matrix size would be $39 \times 94,720$ which is still high to be used as a feature vector for classification. Here, two methods have been employed to reduce the data and find the feature vector to feed the classifier.

A. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) [16] is a technique to separate a multivariate signal into multiple independent non-Gaussian signals. It should be noted that ICA has been used to extract the hidden spatio-temporal structure in neuroimaging. ICA has the assumption that these underlying signals are maximally independent of each other. It uses the fact that two random variables would be uncorrelated if they are independent. However, we cannot come up with these results from un-correlation to independence [17]. Typically, ICA model tries to extract a feature vector like U from our full rank matrix A . Let's assume we have N patients and M features for each, then our matrix size would be $[A]_{N \times M}$. We are trying to find a good approximate with respect to our source which we call S . Thus, for extracting q features out

of our M features, we will have:

$$[U]_{N \times q} = [A]_{N \times M} [S]_{M \times q}$$

Now, we can feed vector U into our classifiers with different number of features q . Here, 5, 10, and 15 independent components have been extracted. We hope that ICA will tell us where the regions of the brain are that share similar brain activity. ICA is also limited by the number of subjects. Figure 3 depicts the correlation matrix produced from getting the correlation of the independent components with each other. We see that the number of red (high correlation) values goes down as the number of independent components goes up.

B. Principal Component Analysis (PCA)

PCA orthogonally transforms data consisting of correlated of uncorrelated variables into linearly uncorrelated variables, which are called principle components. The principal components are ordered so that the first principal component will have the largest variance within the data set and the last will have the least variance. All principal components must also be orthogonal to one another, thus giving us an orthogonal basis set. Singular Value Decomposition (SVD) can be used to decompose the data matrix to transform principal components:

$$[A]_{N \times q} = [U]_{N \times N} [\Sigma]_{N \times q} [W]_{q \times q}^T$$

To project the matrix with respect the principal component we just simply need the score matrix T :

$$T = AW = U\Sigma W^T W = U\Sigma$$

which Σ are equal to the square roots of the eigen values of $A^T A$ [18]. The possible number of principle components is

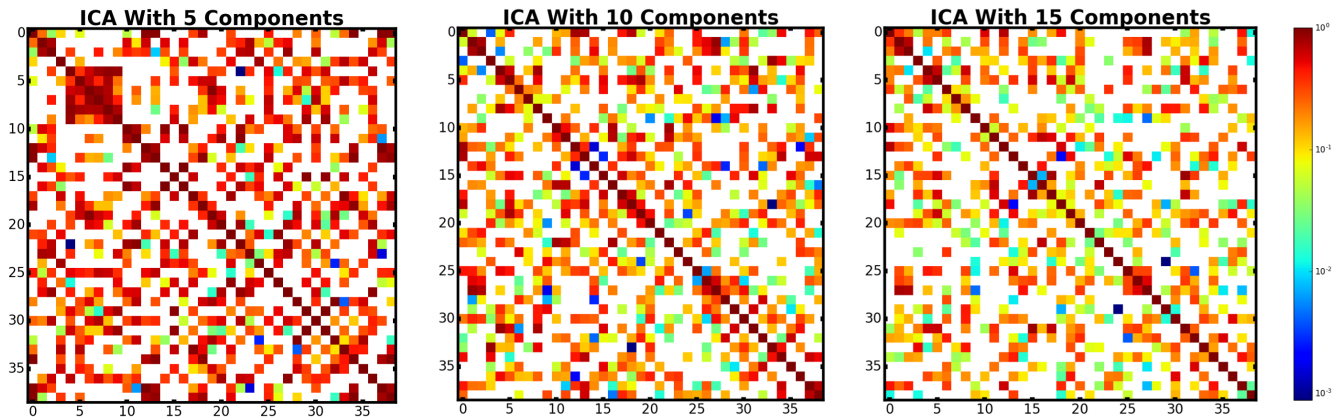


Fig. 3. Correlation Matrix For Different Number of Independent Components.

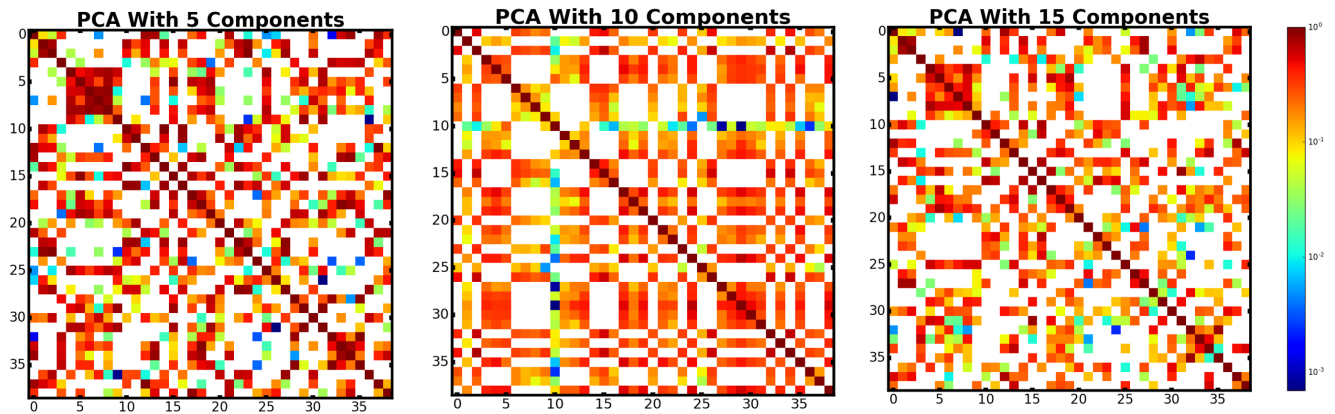


Fig. 4. Correlation Matrix For Different Number of Principal Components.

equal to or less than the number of subjects. Figure 4 shows the correlation matrix produced from getting the correlation of the principal components with each other. Again we see that the number of red (high correlation) values goes down as the number of principle components goes up.

V. CLASSIFICATIONS

Multivariate analyses have been employed based on applying different classifiers on fMRI data. Based on having two classes, some binary classifiers have been compared with each other [19]. We have tried to compare our Genetic Programming (GP) classifier with the other three Machine Learning (ML) classifiers: Logistic Regression, Naive-Bayesian, and K-th Nearest Neighbors.

A. Machine Learning Algorithms

1) *Logistic Regression (LR)*: Logistic regression is useful in problems that only involve two classes, such as in this paper. The method works by finding the relationship between

the classes and the features and it does this by estimating probabilities using a logistic function. These odds are used for the predictive model.

2) *Naive-Bayes (NB)*: The Naive Bayes classifier assumes the features to be independent and applies Bayesian Theorem to them. Bayesian theorem being the conditional probability (posterior), which is equal to the prior times the likelihood. The prior can be calculated by making an assumption about the features distribution called an event model. In this paper, we test two distributions for our event model: Bernoulli and Gaussian distributions.

3) *K-Nearest Neighbors (KNN)*: In KNN, the K stands for how many nearest neighbors we will consider in our algorithm. When classifying, we will look at the K nearest neighbors of our test set and use them to determine what class it belongs to. Here, we have set K=3 for our predictions.

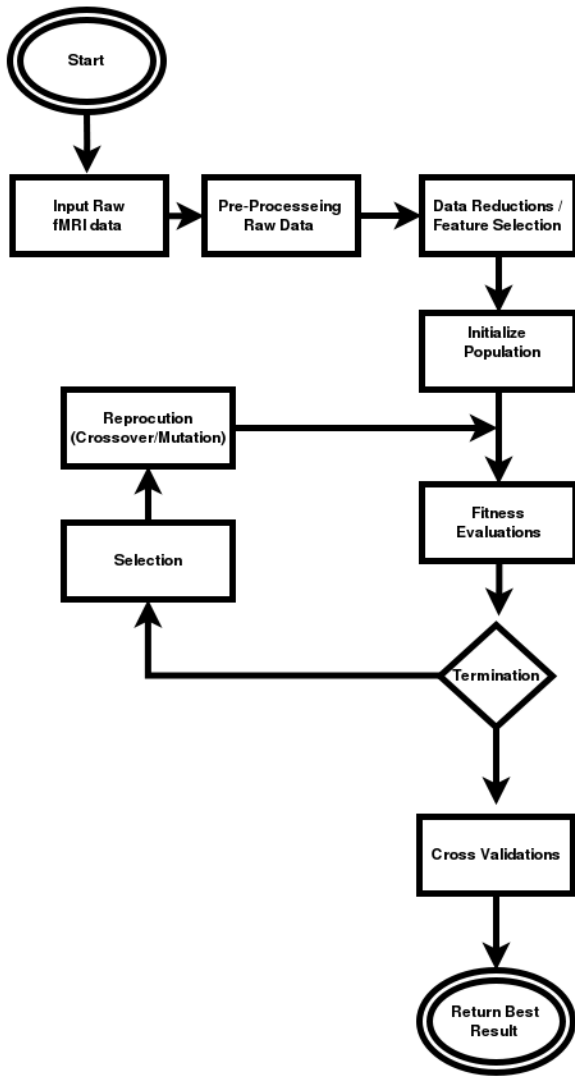


Fig. 5. Flowchart For the GP Classifier

B. Genetic Programming (GP)

As the main method of our study, Genetic Programming (GP) has been employed due to the selection of designs applies on fitness measurement phase [20]. GP has been formulated as a symbolic optimization technique originally based on functional programming language as an evolutionary method [21] to use computer programs for solving a problem following the principle of Darwinian natural selection [22]. Returning real values based on each tree and turning into class labels is the way that GP performs classification [23]. GP instead of using one candidate, uses a group of individuals (Population) and genetic operators to make new individuals (Generations) guided by a function which measures the quality of each individual (Fitness). In other words, having a higher probability of being selected for an individual at each generation would lead us to have better fitness measure [24]. It is always desired to solve a given problem in an efficient way. In this regard, the

fitness function has been calculated during evolution to have the most efficient guided GP [25] [26].

$$Fitness = \frac{Number\ of\ patients\ classified\ correctly}{Number\ of\ patients\ used\ for\ training}$$

In our GP model, to find the best mathematical formula, a Crossover operator [27] has been used to select and replace the winner of the tournament with a stochastic subtree [28]. In addition to this, to maintain the population diversity Subtree Mutations has been added to our GP model. It also could be done with Point Mutation, Hoist Mutation, and Reproduction operators in our model [29]. Figure 5 represents that the GP model receives the Pre-Processed data as an input. It initializes a population of individual solutions. Comparison will be applied by measuring fitness. The next phase will be selecting individual solution from population and modifying fitter individual using crossover and mutation. Based on the termination criteria, leave-one-out cross validation will be applied and classifications accuracy would be the output. Moreover, Table I lists the parameter setting used in GP model.

TABLE I
PARAMETERS SETTING FOR GENETIC PROGRAMMING (GP) CLASSIFIER

Parameter	Setting
Population Size	500
Number of Generations	2000
Hall of Fame	300
Tournament Size	20
P Crossover	0.9
P Subtree Mutation	0.01
P Hoist Mutation	0.01
P Point Mutation	0.01
P Point Replace	0.05
Function Set	<i>add, sub, mul, div, log, neg, inv, abs</i>
Parsimony Coefficient	0.0005
Max Samples	0.9
Random State	0
Number of Jobs	3

Through each evolution, GP model picks 300 (Hall of Fame) best programs from population size (500). Then, these programs will compete in a tournament and just 20 of them will be considered for the next generations. In this regard, the probability of crossover, subtree, point, and hoist mutation (Reproduction phase) has been performed on a tournament winner. Moreover, The fitness of large programs has less probability being selected based on the Parsimony Coefficient. In other words, parsimony coefficient might decrease the computation times by controlling the depth or length of the program to earn better estimation of fitness and stay away from Bloat phenomenon. The maximum distance from its root node to the furthest leaf node is known as depth and the number of nodes in the program is known as length of the program [29]. To decrease the cost of the evaluating the fitness of all programs, three cores (number of jobs) has been parallelized in the Python code to work on this part. It should be noted that the

maximum number of generations and also perfect score have been chosen as stopping criteria to terminate the evolution early.

VI. CROSS-VALIDATION

To test for accuracy in this paper, we used Leave-One-Out Cross-validation. This works by leaving one subject out of the training processes in order to use it in the predictive process. This is done 39 times in order to leave each subject out once and the accuracy is the number of times our model predicted correctly divided by the number of subjects, 39.

VII. RESULTS & DISCUSSIONS

A GP model with 2000 generations and 500 populations for classifications task with two major data reduction methods, namely ICA and PCA has been developed in Python [30] [29]. Figure 5 depicts the flowchart of the GP classifier. Data reduction has been done with 5, 10, and 15 independent components (IC) and principal components (PC). For each one, best fitness, average fitness, and average fitness for a different number of generations have been reported. The left Y-axis has been set to fitness, right Y-axis to length, and X-axis to generations in logarithmic scale. Classification accuracy of GP classifier has been compared with some machine learning algorithms, namely Logistic Regression, Naive-Bayesian with Gaussian and Bernoulli distribution, and 3rd nearest neighbor for ICA and PCA. To represent the classification accuracy better, an ROC curve has been plotted for each case as well. As we can see in the figures 6,7, and 8, as we increased the number of independent components, the average fitness has been increased too. Assigning a higher fitness score to the classifier is the way fitness function classifies more samples using a smaller batch of features [31]. It should be noted that, we found the best accuracy with 15 independent components as we can see in the Table II as the GP just used 75 as the depth length of the program. On the other hand, for 5, and 10 IC, the GP model struggles with increasing the depth of the model to increase the fitness factor. As we can see the average lengths for 5 and 10 independent components are around 90 and 100. Figures 9, 10, and 11 demonstrate that this result matches with the results we found in the figures 9, 10, and 11. Looking closer, classification accuracy for the two different data reduction methods might change significantly for a different number of components. Especially, the classification error which differs with data distribution for each method [21]. In addition to this, as we increased the number of principal components, the length of the model decreased, and the best classification accuracy has been found with 10 principal components.

The classifications accuracy for all the classifiers we used could be found in Tables II and III with ICA and PCA data reduction. We should take into account that the 3rd Nearest neighbor algorithm has produced mostly the same results for both reductions. On the other hand, logistic regression showed better accuracy with PCA than ICA. Since the number of patients in each class are relatively the same (19 and 20),

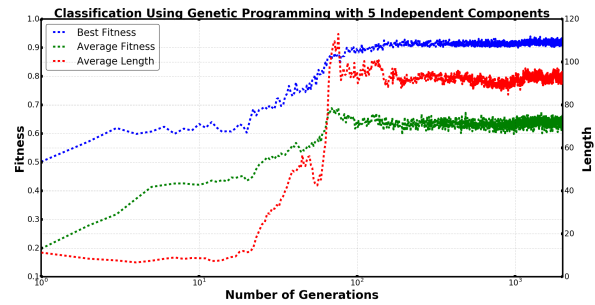


Fig. 6. GP Evolution For 5 Independent Components

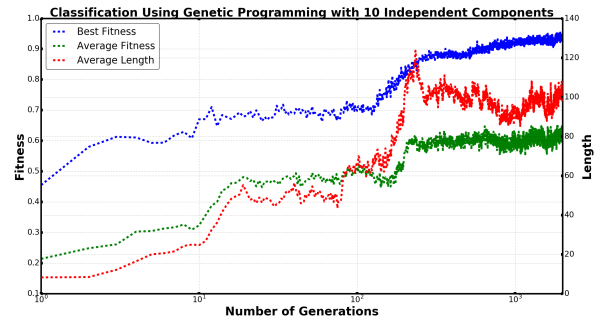


Fig. 7. GP Evolution For 10 Independent Components

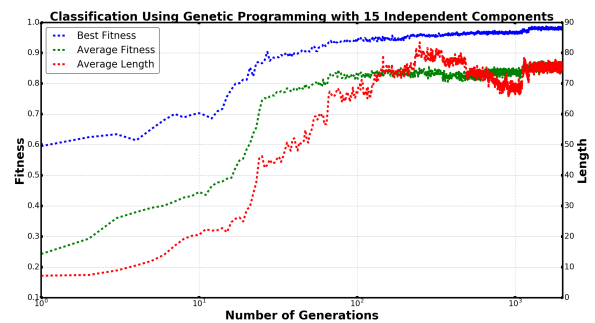


Fig. 8. GP Evolution For 15 Independent Components

predicting of 50% does not tell us anything important. Finding a better performance in our classification accuracy can be enough to tell us there is a significant change in the fMRI data based on the NAC and placebo.

TABLE II
CLASSIFICATION ACCURACY(%) FOR DIFFERENT NUMBER OF INDEPENDENT COMPONENTS (IC)

IC	LR	Bernoulli NB	Gaussian NB	KNN	GP
5	38.46%	66.66%	41.02%	61.53%	64.10%
10	35.89%	38.46%	48.71%	61.53%	64.10%
15	38.46%	48.71%	41.02%	51.28%	68.71%

Machine Learning classifiers usually employ a loss function to update the weights for each subject which is different than

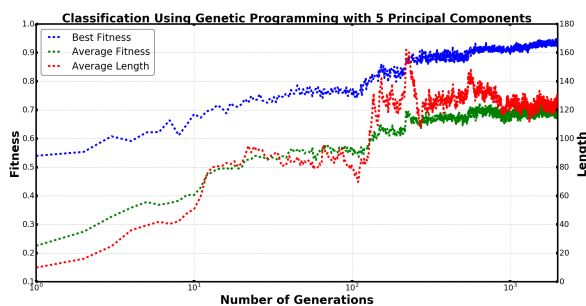


Fig. 9. GP Evolution For 5 Principal Components

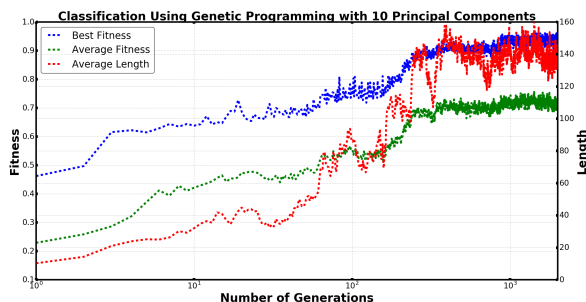


Fig. 10. GP Evolution For 10 Principal Components

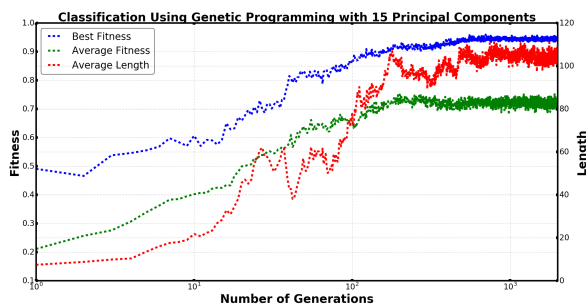


Fig. 11. GP Evolution For 15 Principal Components

TABLE III
CLASSIFICATION ACCURACY(%) FOR DIFFERENT NUMBER OF PRINCIPAL COMPONENTS (PC)

PC	LR	Bernoulli NB	Gaussian NB	KNN	GP
5	61.53%	58.97%	46.15%	61.53%	58.97%
10	46.15%	51.28%	43.58%	61.53%	73.46%
15	53.84%	61.53%	43.58%	58.97%	64.10%

Genetic Programming (GP) models where the learning process occurs during evolution. In fact, the fitness function which is the same score for Machine Learning classifiers, supervises the learning process through each generation. In this regard, better accuracy is expected for GP methods. Our results confirm the experimental results found by Wagner et. al. [20]. In addition to this, the ability of automatically extracting discriminant features by GP methods has shed lights on solving complex

pattern recognition problems with better accuracy. In contrast to machine learning algorithms, GP might provide us with a brighter point of view of the medical diagnosis [21].

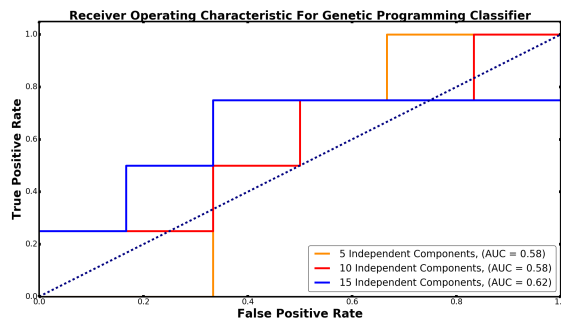


Fig. 12. ROC Curve For GP Classifier With Different Independent Components

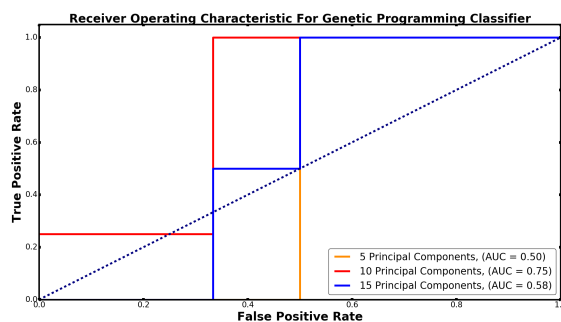


Fig. 13. ROC Curve For GP Classifier With Different Principal Components

The ROC curves have been employed to visualize the detection performance for both ICA and PCA in the figures 12 and 13. The ideal case would be the top left corner (100% sensitivity) which is not very realistic. As the number of features increases, the classifier’s performance increases until we reach the optimal number of features. Here, we can consider 10 principal and 15 independent components as the optimal number of features. We should take it into account that increasing the number of features would lead to overfitting due to learning of the exceptions related to training data by a classifier. In fact, by increasing the feature space, it will become way more sparse. It would be much easier to find a hyperplane based on the small likelihood of the training sample on the wrong side of the best hyperplane which classifies the data. The area under ROC curve for each component has been determined. That would be one of the elements to see how good our GP model classifies the data.

VIII. CONCLUSIONS & FUTURE RESEARCH

In this paper we have compared an evolutionary approach, Genetic Programming model with multivariate machine learning methods to conduct analyses on High Activity regions in the Limbic system of the fMRI data. The high dimensionality of fMRI data makes the classification task abstruse to employ

original pre-processed data as input. In this regard, a classification task with GP classifier has been applied for ICA and PCA since GP has the ability to discover features for each class without any knowledge of the statistical distribution of the data [25]. Based on our results, an enhancement of the classification accuracy due to the feature selection process has been reported. Also, due to the power of GP methods in classifications and flexible heuristic techniques [24], the GP classification accuracy is better than the common machine learning algorithms we used here such as, Logistic Regression, Naive-Bayes with Gaussian and Bernoulli distributions, and K-th Nearest Neighbors. One of the significant advantages of GP is the fitting flexibility to the data. In this regard, also ROC curves with an area under curves have been reported. The best accuracy has been found for 10 principal components with 0.75 as the area under curve. We have shown that the fitness is increasing through the number of generations which proves how good we optimize the loss function. On the other hand, in the other methods we used here, we are dealing with the possibility of under or over fitting based on the low number of subjects. In addition to this, since we have two class of "NAC" and "Placebo" here, we are dealing with binary classes where GP is much more powerful than multi-class problems. Moreover, it has been shown that our model can be trained successfully using features detected by our High-Limbic mask automatically from the studied fMRI data which contradicts previous approaches considering the region of interest based on anatomical data.

The impressive performance shown by GP compared with Machine Learning algorithms, we presented in this paper seems to suggest to combine deep learning algorithms such as Convolutional Neural Networks (CNN) with GP model and to take advantage of temporal components of the data before and after treatment in future work.

ACKNOWLEDGMENTS

This material is based in part upon work supported by the National Science Foundation under Cooperative Agreement No. DBI-0939454. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors also would like to thank Eitan Lees for the careful revision of the manuscript.

REFERENCES

- [1] A. Ehtemami, A. Smith, D. Fratte, A. Meyer-Baese, A. Goudriaan, L. Schmaal, M. Schulte, and O. Zavala-Romero, "Functional connectivity analysis of resting-state fMRI networks in nicotine dependent patients," In *SPIE Medical Imaging*, pp. 978827-978827, International Society for Optics and Photonics, 2016.
- [2] A. Tahmassebi, A.H. Gandomi, I. McCann, M. Schulte, L. Schmaal, A. Goudriaan, and A. Meyer-Baese, "fMRI Smoking Cessation Classification," 2017 (Submitted).
- [3] R. Moller, and R. Zeipelt, "Automatic segmentation of 3D-MRI data using a genetic algorithm," *Medical Imaging and Augmented Reality*, IEEE, 2001.
- [4] J. Lee, et al. "Efficient classification system based on FuzzyRough Feature Selection and Multitree Genetic Programming for intension pattern recognition using brain signal," *Expert Systems with Applications* 42.3 : 1644-1651, 2015.
- [5] M. Pyka, et al. "Dynamic causal modeling with genetic algorithms," *Journal of neuroscience methods* 194.2 : 402-406, 2011.
- [6] J.Koza, *Genetic Programming*, 1992.
- [7] H. Shen, L. Wang, Y. Liu, and D. Hu, Discriminative analysis of resting-state functional connectivity patterns of schizophrenia using low dimensional embedding of fMRI, *Neuroimage*, vol. 49, no. 4, pp. 31103121, 2010.
- [8] S. Huettel, A. Song, and G. McCarthy, "Functional Magnetic Resonance Imaging," Sinauer Associates, Inc., 2004.
- [9] P. Jezzard, P. Matthews, and S. Smith, "Functional MRI: an introduction to methods," Oxford University Press, 2001.
- [10] A. Tahmassebi, "Fluid Flow Through Carbon Nanotubes And Graphene Based Nanostructures," Diss. University of Akron, 2015.
- [11] A. Tahmassebi, and A. Buldum, "Fluid flow calculations of Graphene Composites," *APS March Meeting Abstracts*. vol. 1, 2015.
- [12] A. Tahmassebi, A. Buldum, M. Avon, and S.G. Kelly, "Fluid flow through graphene based nanostructures," *Journal of Applied Physics*, 2017 (Submitted).
- [13] I. Kononenko, *Machine learning for medical diagnosis: history, state of the art and perspective*, *Artificial Intelligence in medicine*, vol. 23, no. 1, pp. 89109, 2001.
- [14] A. Meyer-Baese, "Pattern recognition for medical imaging," Academic Press, 2004.
- [15] R. A. Poldrack, *Region of interest analysis for fmri*, *Social cognitive and affective neuro-science*, vol. 2, no. 1, pp. 6770, 2007.
- [16] A. Meyer-Baese, A. Wismueller, and O. Lange, "Comparison of two exploratory data analysis methods for fMRI: unsupervised clustering versus independent component analysis," *IEEE Transactions on Information Technology in Biomedicine* 8.3: 387-398, 2004.
- [17] V. Calhoun, J. Liu, and T. Adali, "A review of group ICA for fMRI data and ICA for joint inference of imaging, genetic, and ERP data," *Neuroimage* 45.1: S163-S172, 2009.
- [18] A.J. Roberts, *Linear Algebra Reformed for 21st C Application*, 2016.
- [19] J. Etzel, V. Gazzola, and C. Keysers, "An introduction to anatomical ROI-based fMRI classification analysis," *Brain research* 1282: 114-125, 2009.
- [20] T. Wager, and T. Nichols, "Optimization of experimental design in fMRI: a general framework using a genetic algorithm," *Neuroimage* 18.2 : 293-309, 2003.
- [21] M. Brameier, and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining," *IEEE Transactions on Evolutionary Computation* 5.1, 17-26, 2001.
- [22] A.H. Gandomi , and A.H. Alavi, "Multi-stage genetic programming: a new strategy to nonlinear system modeling," *Information Sciences* 181.23 : 5227-5239, 2011.
- [23] T. Loveard, and V. Ciesielski, "Representing classification problems in genetic programming," *Evolutionary Computation*, Proceedings of the 2001 Congress on. Vol. 2. IEEE, 2001.
- [24] P. Espejo, S. Ventura, and F. Herrera, "A survey on the application of genetic programming to classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C* 40.2, 121-144, 2010.
- [25] J. Kishore, et al. "Application of genetic programming for multicategory pattern classification," *IEEE transactions on evolutionary computation* 4.3, 242-258, 2000.
- [26] P. D. Heerman, and N. Khazenie, Classification of multispectral remote sensing data using a back propagation neural network, *IEEE Trans. Geosci. Remote Sensing*, vol. 30, no. 1, pp. 8188, 1992.
- [27] A. Tsakonas, et al. "Evolving rule-based systems in two medical domains using genetic programming," *Artificial Intelligence in Medicine* 32.3: 195-216, 2004.
- [28] R. Poli, et al. *A Field Guide to Genetic Programming*, 2008.
- [29] T. Stephens, "Gplearn Model, Genetic Programming," Copyright 2015.
- [30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. and 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.
- [31] R. Ramirez, and M. Puiggros, "A genetic programming approach to feature selection and classification of instantaneous cognitive states," *Workshops on Applications of Evolutionary Computation*, Springer, Berlin, Heidelberg, 2007.