An Evolutionary Online Framework for MOOC Performance using EEG Data

Amirhessam Tahmassebi[†], Amir H. Gandomi[*], and Anke Meyer-Baese[†]
[†]Department of Scientific Computing, Florida State University, Tallahassee, Florida 32306-4120, USA
Email: atahmassebi@fsu.edu , URL: www.amirhessam.com
[*]School of Business, Stevens Institute of Technology, Hoboken, New Jersey 07030, USA
Email: a.h.gandomi@stevens.edu

***Abstract*—Massive Online Open Course (MOOC) is a scalable, free or affordable online course which emerged as one of the fastest growing distance education platforms in the past decade. One of the biggest challenges that threatens distance education is abnormality in the overall level of consciousness of students while they are taking the course. In this paper, an evolutionary online framework was proposed to improve the performance of MOOCs via noninvasive electro-physiological monitoring methods such as electroencephalography (EEG). Based on the proposed platform, EEG signals can be recorded from users while they are wearing any EEG headsets. EEG measures a brain's spontaneous voltage fluctuations resulting from ionic current within the neurons of the brain via multiple electrodes placed on the scalp. A total of eleven extracted features from EEG signals were employed as the inputs of the evolutionary classification algorithm to predict two classes of confused and not-confused for each individual. An accuracy of 89% was considered significant enough to suggest that there is difference in the EEG signals of individuals with confusion versus not-confused individuals.**

I. Introduction

Massive Online Open Course (MOOC) has emerged as one of the fastest growing platforms for distance education in the past five years. An MOOC is a scalable, free or affordable online course which focuses on educational communities. In 2012, the New York Times published an article, "The Year of the MOOC" to reflect the importance of this phenomenon [1]. Stanford Online, Coursera, edX, Udacity, and Khan Academy are only a few notable providers of MOOC. Unlike traditional educations, MOOCs require additional skills including videography, instructional designs, and IT platforms. A typical MOOC instructor requires over 100 hours for recording online lecture videos and course preparation before class starts. The instructor also spends 8-10 hours per week on the course, including participation in discussion forums and Q & A sessions [2].

One of the biggest challenges that threatens online learning is abnormality in the overall level of consciousness of students while they are taking the course and watching the lecture videos. This is called clouding of consciousness, also known as brain fog or mental fatigue. Normally, the presence of the instructor in the class would lower the chance of this phenomenon. The instructor would see how the students in the class can respond to their lecture by mimicking students or asking questions. Depending on the severity of brain fog, it can interfere with work or school. There are numerous explanations for why brain fog occurs such as stress, lack of sleep, hormonal changes, diet, medications, and medical conditions. For example, chronic stress can increase blood pressure in the body, weaken the immune system, and trigger depression which might cause mental fatigue and make the brain exhausted. In addition to this, poor sleep quality can also decrease the brain functionality which decreases the overall level of focus and thinking [3].

Electroencephalography (EEG) is a noninvasive electrophysiological monitoring method to record electrical activity of the brain. EEG measures a brains spontaneous voltage oscillations resulting from ionic current within the neurons of the brain recorded from multiple electrodes placed on the scalp [4]. EEG has shown reasonable potential to be used for diagnosis in different applications including epilepsy, sleep disorders, depth of anesthesia, coma, brain death, and focal brain disorders [5][6]. Moreover, it was shown recently that EEG signals can be used to detect confusion in students while they are acquiring complex knowledge [7][8]. If learners resolve confusion, it can be beneficial to them to support the development of a deeper concept. However, if learners fail to overcome confusion its effect can ruin the learning process completely [9]. In this paper, an evolutionary framework was proposed to employ EEG signals from users in real-time to improve the quality of MOOC.

II. Evolutionary Online Framework

As discussed in section I, the instructor can see feedback of students for in-class education by asking them questions or judging the students from their gestures and body language signs such as furrowed brow and head scratching. In contrast, instructors in online education do not have the chance to get feedback from the students as mentioned. However, the students usually participate in different Q & A forums and also post blogs and feedback in the course page. In this paper, an evolutionary framework was proposed that aims at detecting the confusion level of students in real time using EEG signals. The recent availability of simple, low cost ($\sim$ $100 USD), portable EEG monitoring devices brings new opportunities to the online learning providers to improve the quality of online education by providing the users with EEG devices [7]. This

978-1-5090-6017-7/18/$31.00 ©2018 IEEE

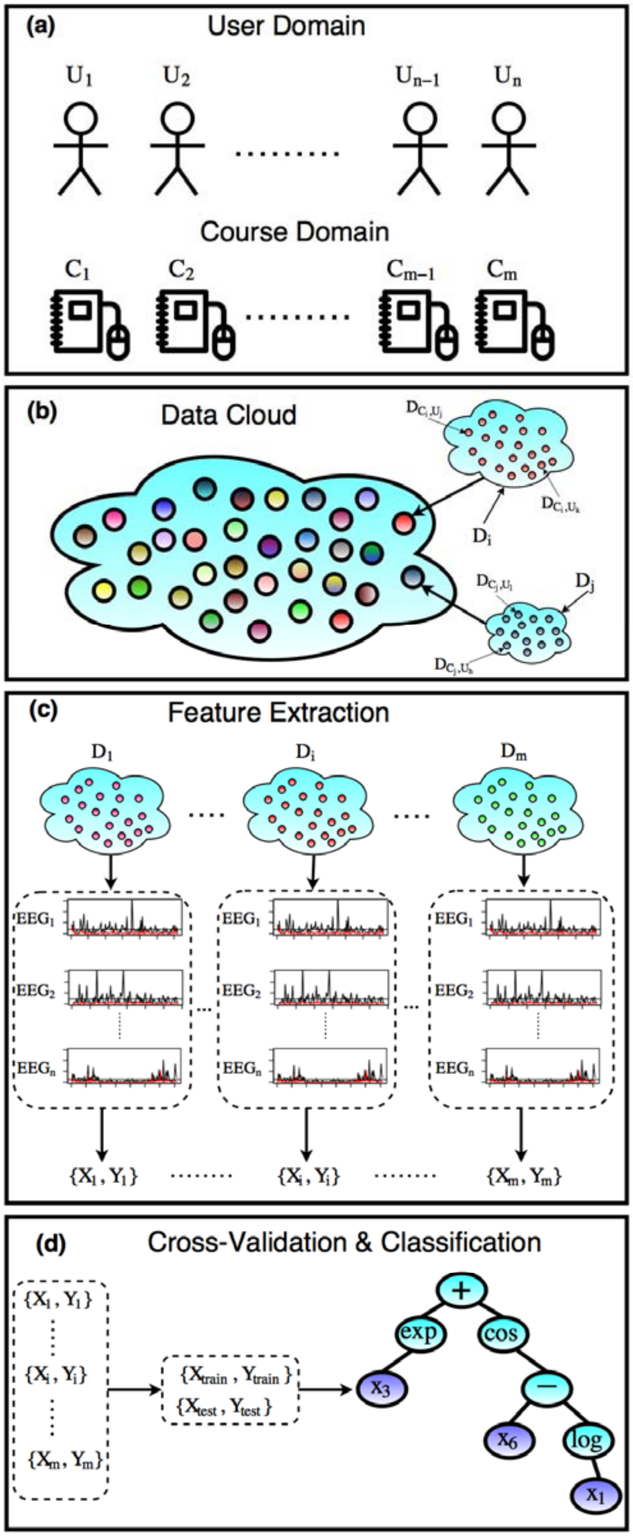


Fig. 1. Evolutionary online framework: (a) presents user domain with n members $\{U_1, \cdots, U_n\}$ and course domain with m members $\{C_1, \cdots, C_m\}$. (b) presents the data cloud comprises the data from user member U_i while taking course C_j. (c) presents the feature extraction process for each data cloud D_i. (d) presents the cross-validation procedure to feed the GP classifier.

would guarantee that students would learn the materials with higher probability and the online providers could also track the students' feedback to the course materials through the EEG signals.

As seen in Fig. 1, the proposed evolutionary framework contains four different stages. In the first stage the user domain and course domain were defined. The user domain contains n members $\{U_1, \cdots, U_n\}$ and course domain contains m members $\{C_1, \cdots, C_m\}$ (shown in Fig. 1a). As discussed, each user is supposed to wear a EEG device to capture the signals in which they are taking the course. Fig. 2 illustrates the diagram of NeuroSky MindSet which is a typical single-channel EEG headset.

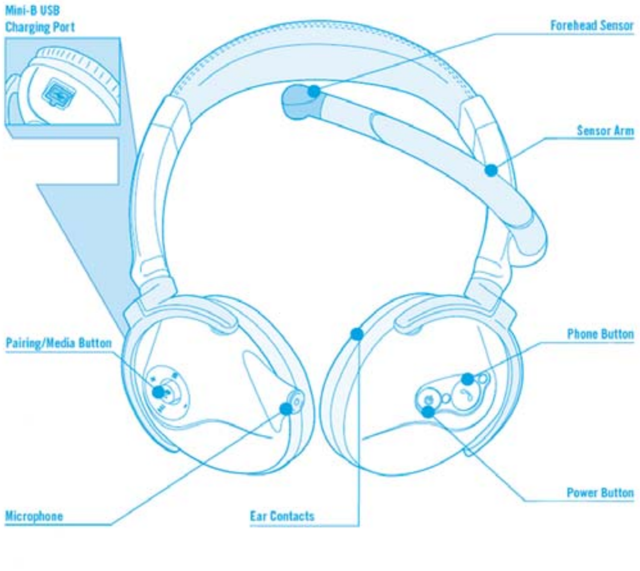


Fig. 2. NeuroSky MindSet diagram.[1]

To illustrate the above concepts, assume the user U_i is taking the course C_k and user U_j is also taking the course C_k. This would would mean all the data from U_i to U_j while the course C_k is being taken would make the data cloud D_k (shown in Fig. 1b). The total data cloud contains m smaller data clouds for each course and each small data cloud D_k contains the EEG signal for n users (shown in Fig. 1c). Note that it is not necessary for the whole user domain to take the course i. For instance, the data cloud D_h might have just one user member U_p. This indicates the proposed framework has the flexibility to handle all the possible combinations of users and courses. Based on the frequency bands, the features are extracted from each data cloud D_i and converted to vector sets $\{X_i, Y_i\}$ with continuous values. Fig. 3 demonstrates the comparison of EEG frequency bands. The EEG signals collection are done based on different standards depending upon demand. The NeuroSky API is a standard example for EEG signal streaming which is presented in Table I.

[1] http://support.neurosky.com/kb/mindset/mindset-diagram

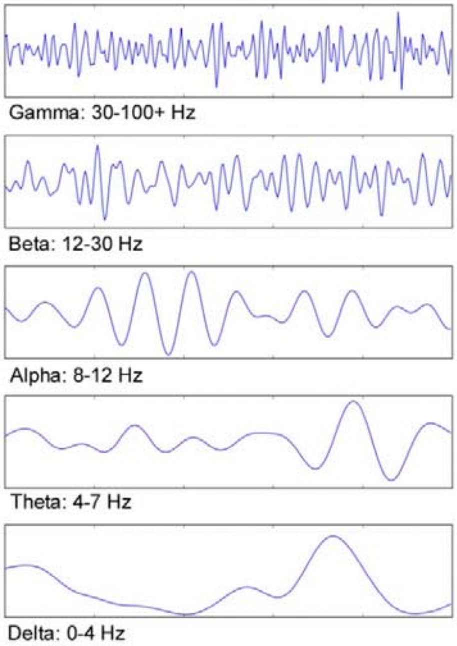


Fig. 3. Comparison of EEG frequency bands.

TABLE I
NeuroSky API for EGG signal streaming.

EEG Signal Type	Frequency
Raw	512 Hz
Attention	1 Hz
Meditation	1 Hz
Delta	1-3 Hz
Theta	4-7 Hz
Alpha	8-11 Hz
Beta	12-29 Hz
Gamma	30-100 Hz

There are two possible ways to create the class labels Y_i at first. The first proposed procedure is that the online provider assigns a level of confusion (for example 1 to 10) to the course contents. The second procedure is that the students send their feedback and assign a level of confusion for each part of course. This is necessary at the starting point of the framework. Once the data cloud contains enough EEG signals for each course, there is no need for the aforementioned procedures anymore. To expand more on this concept, once there are enough user members for each course there will be enough EEG signals for that specific course and then the GP function classifier would have enough inputs to fit a model. Thus, for each new user member U_i who is taking course C_j, the GP classifier can predict whether or not the student is confused. Once the prediction is done, the data according to the new user U_i who was taking course C_j will be added to the data cloud D_j as well. In this procedure, the data cloud also gets bigger which would increase the variability of the population of possible solutions. This leads to the extracted feature being combined together to have vector sets $\{X, Y\}$. Depending on the demands, the vector sets can be splitted into training and testing sets (shown in Fig. 1d). In addition to this, k-folds cross-validation can be employed to overcome possible overfitting. Once the prediction is done, a set of decisions can be made based on the results. For example, if the prediction is that the student is confused, the learning provider can ask the student to take a quiz about the material or re-watch the video. In this way, the quality of online education would be guaranteed. After the set of decisions are made, the new data can be added to the database to be used in the population. Adding the new data to the database would then cause the size of the data to get bigger. However, the number of inputs would be the same as before since it depends on feature extraction stage (Fig. 1c).

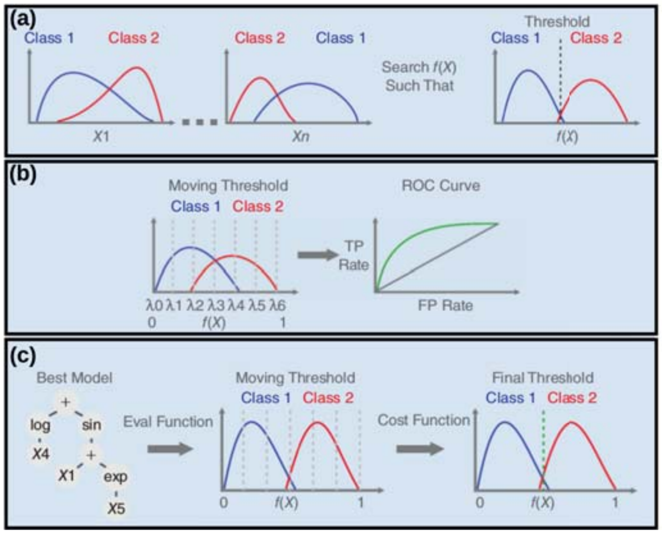


Fig. 4. GP function classifier: (a) search for the best nonlinear function with the highest discriminatory power between the two classes, (b) evaluation of the model for various thresholds to compute the area under ROC curve, and (c) threshold selection for the best final model [10].

III. GP Function Classifier

Genetic Programming (GP) [11] due to the selection of designs applying to the fitness and complexity measurements phase was formulated as a symbolic optimization approach originally based on functional programming language as an evolutionary strategy to use computer programs for solving a problem following the principle of Darwinian natural selection [12][13][14]. Returning real values based on each tree and turning into class labels is the way that GP performs classification. GP instead of using one candidate, uses a group of individuals (population) and genetic operators to make new individuals (generations) guided by functions which measure the quality of each individual (fitness and complexity). GP function classifier as shown in Fig. 4 [10] was implemented as a multi-objective genetic programming approach based on non-dominated sorting genetic algorithm II (NSGA-II) [15].

Algorithm 1 illustrates the details of the NAGA-II implementation [16]. For a given data X and classes H_0 and H_1, GP classifier searches for a non-linear function $y = f(\bar{X})$ such the distributions $p(f(\bar{X})|H_0)$ and $p(f(\bar{X})|H_1)$ are best separated (shown in Fig. 4a). Each output of the learning function will be sent to a decision rule $\hat{L_i}$.

$$\hat{L_i} = \begin{cases} 1 & if \quad y_i \geq \lambda \\ 0 & if \quad y_i < \lambda \end{cases}$$

where $0 \leq \lambda \leq 1$ is a threshold to find the highest discriminatory power between the two classes based on the area under Receiver operating characteristic (ROC) curve as the fitness function (shown in Fig. 4b). By varying the threshold λ, evaluating the GP tree on the training samples and finding the maximum and the minimum error, false positive and true positive rates can be calculated and the area under curve (AUC) can be easily computed. A weighted sum of calculated false positive and false negative rates was chosen as the cost. The best threshold was calculated by optimizing the cost function (shown in Fig. 4c). In addition to this, the subtree complexity measure [17] was employed as the second objective in the optimization process.

Algorithm 1: NSGA-II

```
Input: Generations $N$, Population $P$
Output: Best Model
Initialize population $P$;
Generate random population size $M$;
Evaluate objective values;
Assign ranking based on Pareto sort;
Generate child population;
Binary tournament selection;
Recombination and mutation;
for $i \in \{1, \ldots, N\}$ do
    for each Parent and Child in Population do
        Assign ranking based on Pareto sort;
        Generate sets of non-dominated solutions;
        Determine Crowding distance;
        Loop inside by adding solutions to next
          generation starting from the first front until $N$
          individuals;
    end
    Select points on the lower front with higher crowding
      distance;
    Create next generation;
    Binary tournament selection;
    Recombination and mutation;
end
```

IV. EEG Data & Feature Extraction

In 2013, Wang et al. [7] carried out a pilot study to collect EEG signal data of college students to see if it was possible to predict the confusion level of the students while they were watching MOOC videos. In this study, they employed ten students to watch ten MOOC videos with the topics including quantum mechanics, linear algebra, geometry, and stem cells. Each video last for two minutes with different level of confusion. Some of the videos were easier to understand in nature while others were more confusing. All the students were asked to wear a wireless single-channel MindSet as shown in Fig. 2 which measured the activity over the frontal lobe. The sensor arm is flexible to bend comfortably towards forehead since the forehead sensor must touch the user's forehead. The contacts of ear pads must touch the skin of the ears to ensure accurate brain wave reading as well. To run each session, each student were asked to stay calm and relaxed for 30 seconds before watching the videos. The EEG signal streams were collected based on NeuroSky API as shown in Table I.

TABLE II
Extracted features from EEG signals for GP classifier.

Feature	Sampling Frequency	Statistics	Variable
Attention	1 Hz	Mean	X_1
Meditation	1 Hz	Mean	X_2
Raw	512 Hz	Mean	X_3
Delta	8 Hz	Mean	X_4
Theta	8 Hz	Mean	X_5
Alpha.1	8 Hz	Mean	X_6
Alpha.2	8 Hz	Mean	X_7
Beta.1	8 Hz	Mean	X_8
Beta.2	8 Hz	Mean	X_9
Gamma.1	8 Hz	Mean	X_{10}
Gamma.2	8 Hz	Mean	X_{11}

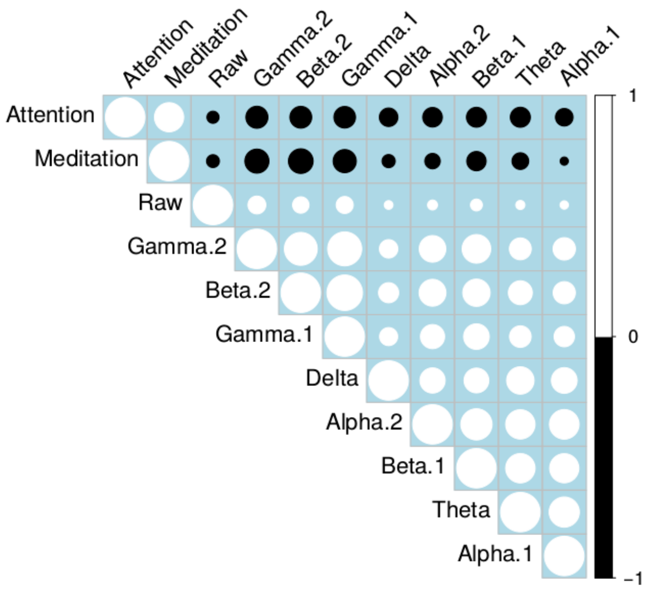


Fig. 5. Correlation matrix plot of the extracted features with hierarchical clustering.

The recorded waveforms reflect the cortical electrical activity. EEG signal intensity is quite small (μV), however, the main signal frequencies of the human EEG waves can be easily recorded. Frequency refers to rhythmic repetitive activity (in Hz). EEG waves have different frequency power bands as shown in Fig. 3. Voltage refers to the average voltage or peak voltage of EEG activity. Feature extraction was done based on the frequency bands and voltages of the recorded EEG waves. Table II presents the extracted features from EEG signals. For each of the signal time series, the mean value of the series was used as the feature. In addition to the frequency-based features, attention as the proprietary measure of mental focus and meditation as the proprietary measure of calmness were also extracted.

Total eleven features, $\{X_1, \cdots, X_{11}\}$, were extracted for the classification task. Moreover, a set of binary user-defined labels (confused or not-confused) was used for each sample. Fig. 5 depicts the correlation matrix of the extracted features with a color bar which highlights the probability of the correlation of each of the extracted features with each other. It is clear that the diagonal has the probability of one (white). As shown, to illustrate the correlation matrix, the hierarchal clustering method was employed. Positive correlations are displayed in white and negative correlations in black. The size of the circles are proportional to the correlation coefficients. Thus, as the circle gets progressively larger this indicates the features are more correlated which in turn can be both positive or negative (black or white). As seen, attention and meditation are inversely correlated with the rest of the features and Beta and Gamma signals showed the highest linear correlation among all the extracted features.

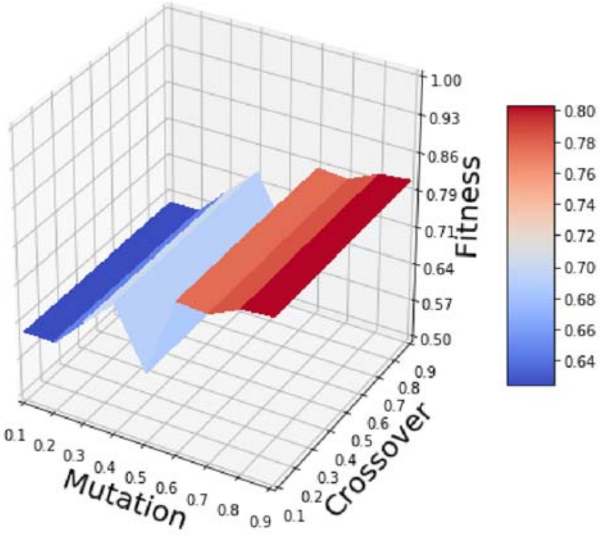


Fig. 6. 3D surface plot of the mutation rates and the crossover rates versus fitness measure.

V. Results & Discussion

A GP model based on the multi-objective genetic programming approach and NSGA-II was developed. A total of eleven extracted features were used as the inputs of the GP model for the classification task. To optimize the mutation and the crossover rates of the model, various rates ranging from 0.1 to 0.9 were employed for the developed GP model for 100 generations. Fig. 6 illustrates the three-dimensional surface plot of the mutation rates and the crossover rates based on the fitness measure. The highest fitness value was reported with 0.1 as the crossover rate and 0.9 as the mutation rate. These values were considered to run the developed GP model with 5000 generations and 1000 population for the classification task. The details of the parameters setting for GP classifier is presented in Table III.

TABLE III
Parameters setting for GP function classifier.

Parameter	Setting
Population Size	1000
Number of Generations	5000
Tournament Size	20
Number of Inputs	11
Crossover Rate	0.1
Mutation Rate	0.9
Number of Cross-Validation Folds	5
Number of CPU Threads	8
1^{st} Objective	AUC
2^{nd} Objective	Subtree Complexity
Population Initialization	Ramped-Half-and-Half
Function Set	$+, -, \times, /, \sqrt{\ }, (\ )^2, (\ )^3, (\ )^4$ $\log, \exp, \sin, \cos$

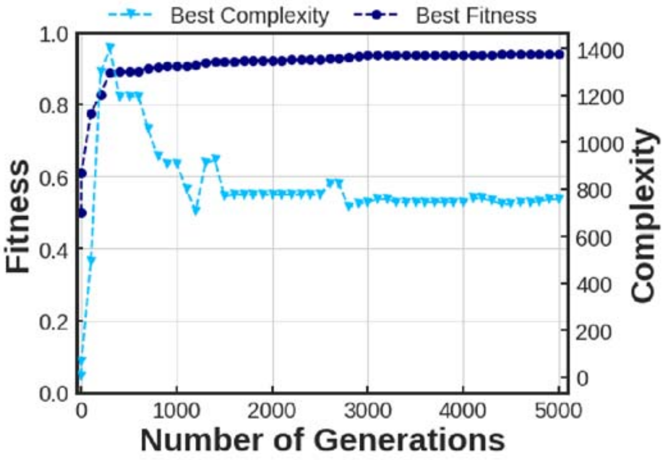
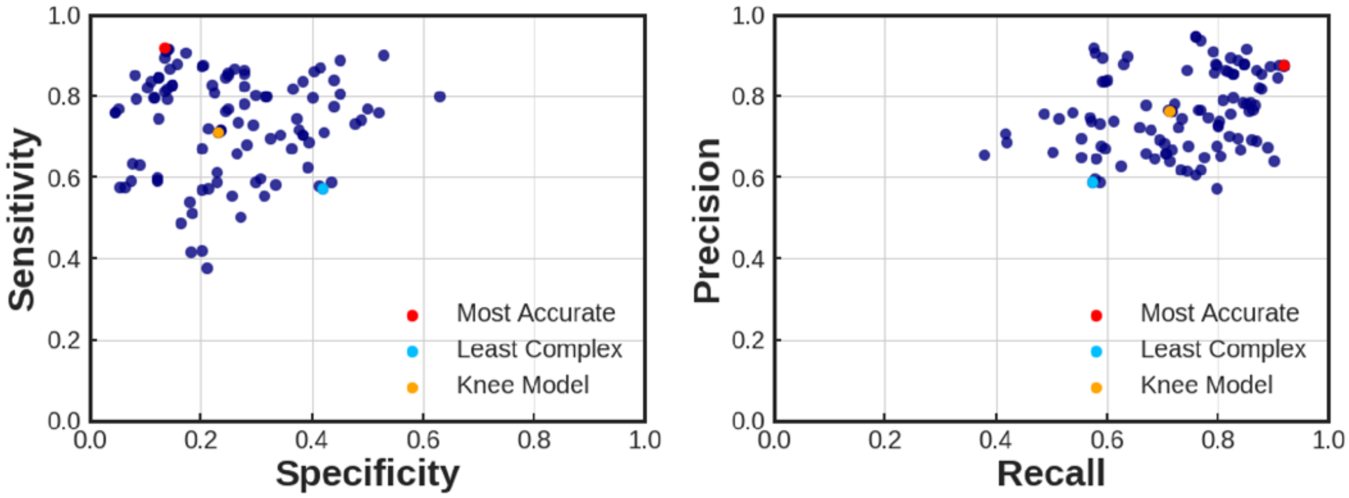


Fig. 7. GP evolution through generations.

At each generation, the fitness and the complexity were reported for the best individuals. Fig. 7 depicts the evolution of the GP model through generations for both fitness and complexity measures. The left Y-axis was set to fitness measure, the right Y-axis to complexity, and X-axis to number of generations. As seen, after 500 generations the model reached the highest complexity measure around 1400. However, as



(a) Sensitivity-Specificity

(b) Precision-Recall

Fig. 8. Scatter plots of (a) sensitivity-specificity and (b) precision-recall of the all models in the Pareto front (navy circles) with specifying the most accurate model (red circle), the least complex model (cyan circle) and the knee model (orange circle).

the number of generations increased, the complexity measure decreased dramatically. After 5000 generations the complexity reached a value of 756 which is roughly half of the highest complexity measure that was obtained for this model. In addition to this, the fitness measure increased through the number of generations and reached to a value of 0.94043. By considering the combinations of the fitness and complexity measures three different models can be defined. The first model, which is the least complex model which was obtained at the first generation with complexity measure of 3 and fitness value of 0.40. The second model, which is the most accurate model was obtained after 4850 generations with a fitness value of 0.94043 and complexity measure of 756. The third model is the knee model, which is the point at the model that the slope of fitness and complexity measures did not change through evolution of the generations. The knee model was found after 1493 generations with a fitness value of 0.92068 and complexity measure of 766. Thus, the knee model can be a great alternative to be used in the classification task with 80% less computational run-time.

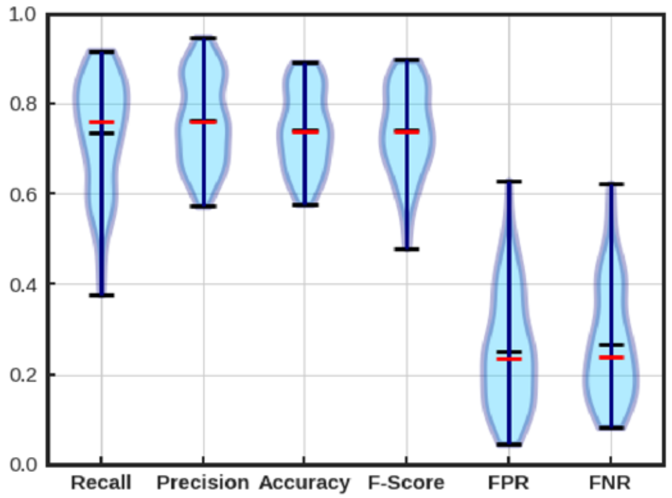


Fig. 9. Violin plots of the summary statistics of all the models in the Pareto front. Minimum, maximum, and mean values are presented with horizontal black lines. Red line indicates the median for each statistic.

As discussed, the developed GP model was based on multi-objective programming approach with optimizing both fitness and complexity simultaneously. The resulted Pareto front comprises more than 100 models with different characteristics from the least complexity to the highest accuracy. Fig. 8 visualizes the trade-off of all of the models in the Pareto front from two different aspects including sensitivity-specificity and precision-recall. Sensitivity, also known as true positive rate or recall is the proportion of positive cases that are correctly classified. Specificity, also known as false positive rate is the proportion of negative cases that are incorrectly classified as positive. The performance can be evaluated through how well a method separates the true positive rate from the false positive rate. As a straight forward measure, higher sensitivity with lower specificity is desired. Fig. 8a presents the scatter plots of sensitivity-specificity of all the models in the Pareto front. The least complex model, the most accurate model, and the knee model were presented in light blue, red, and orange circles, respectively. Recall measures how many truly relevant predictions are returned. Precision is defined as the number of correct positive predictions divided by the total number of positive predictions (true positive cases plus false positive cases). Precision is a measure of how many the samples predicted by the classifier as positive are indeed positive, which depend on how rare the positive class is. Fig. 8b presents the scatter plots of the precision-recall of the all models in the Pareto front. The least complex model, the most accurate model, and the knee model were presented in light blue, red, and orange circles, respectively. Additionally, Fig. 9 illustrates the distribution of the various score metrics including recall, precision, accuracy, F-score, FPR, and FNR of all the models in the Pareto front and their probability density estimation. A violin plot was used indicating the minimum, maximum, mean, and median values for each score.

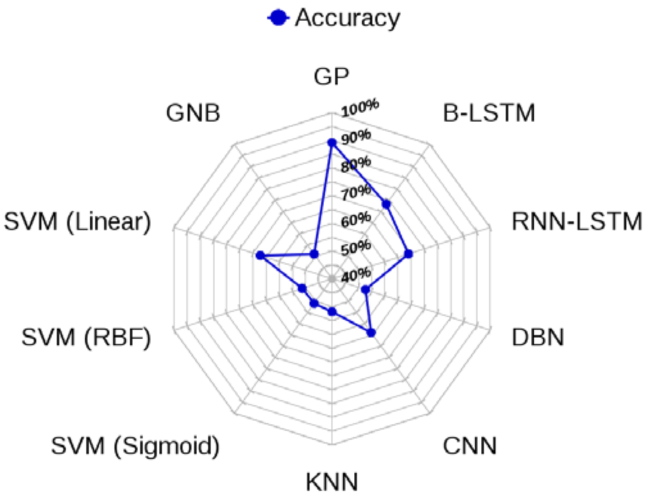


Fig. 10. Radar plot of the classification accuracy for various classifiers compared to the developed GP function classifier.

As a comparative study the result of the GP function classifier in terms of classification accuracy was compared to the results of Wang et al. [7] who employed a Gaussian Naive-Bayes (GNB) classifier and Ni et al. [8] who employed several machine learning algorithms including Support Vector Machines (SVM) with three linear, sigmoid and radial basis function (RBF) kernels, K-Nearest Neighbors (KNN), Convolutional Neural Networks (CNN), Deep Belief Network (DBN), Recurrent Neural Networks-Long Short Term Memory Networks (RNN-LSTM), and Bidirectional-Long Short Term Memory Networks (B-LSTM). Fig. 10 depicts the radar plot of the comparison of the classification accuracies of the various machine learning algorithms. It is clear that the developed GP model outperformed the other classifiers. The best classification accuracy after the GP model is 73.3%, which was obtained by Ni et al. using Bidirectional-LSTM algorithm.

VI. Conclusions & Future Work

This paper aims at developing an evolutionary online framework to improve the performance of MOOC using EEG signals. In this regard, a multi-objective genetic programming strategy based on non-dominated sorting genetic algorithm II with considering the optimization of area under ROC curve as the fitness and subtree complexity as the complexity measure simultaneously. The GP model ran for 5000 generations with 1000 population considering 5-folds cross-validation to overcome any possible over-fitting. Additionally, the mutation and crossover rates were optimized over 100 generations based on the fitness values. A total of eleven extracted features from EEG signals were employed as the inputs of the GP classifier. Table IV presents the summary statistics of the results of the GP classifier including the most accurate model, the least complex model, and the knee model.

TABLE IV
Classification score metrics of the selected models in the Pareto front.

Model	Accuracy	Precision	Recall	F-Score	FPR	FNR
Best	89.16%	87.67%	91.77%	89.67%	13.59%	8.24%
Knee	73.96%	76.43%	71.15%	73.70%	23.08%	28.86%
Least Complex	57.65%	58.94%	57.31%	58.11%	41.99%	42.70%
Most Accurate	89.16%	87.67%	91.77%	89.67%	13.59%	8.24%

The classification accuracy of the developed GP classifier was compared with the other machine learning algorithms. It was clearly shown that the GP classifier outperformed the other machine learning algorithms applied on this database, which suggest that there is difference in the EEG signals of individuals with confusion versus the not-confused individuals.

When considering the difference in regards to the GP classifier, the high performance of the developed nonlinear function suggest to employ linear or nonlinear transformation techniques to map the features to another subspace, in which the features are not correlated. For example, principal component analysis (PCA) is an orthogonal transformation that converts a set of possibly correlated features into a set of new features, which are linearly uncorrelated with each other [18]. In the field of signal processing, it is also called the discrete Karhunen-Loeve transform (KLT). Additionally, for high-dimensional multivariate data space PCA can transform the data into a lower-dimensional subspace while the maximum information of the data is kept. It is more practical to feed the evolutionary classifier with lower numbers of inputs to increase the efficiency and speed up the convergence [18].

In addition to PCA, independent component analysis (ICA) can be also employed to separate multivariate signals into additive subcomponents with an assumption that the resulted subcomponents are non-Gaussian signals, which are statistically independent from each other. In signal processing, independent component analysis (ICA) is a computational method as a special case of blind source separation, for separating a multivariate signal into additive subcomponents. This is done by assuming that the subcomponents are non-Gaussian signals and they are statistically independent from each other [18]. A common example application is applying ICA to noisy signals such as the ”cocktail party problem”. Tahmassebi et al. [19][20][21][22][23] have recently shown the practical ways of data reduction algorithms in big data problems employing evolutionary approaches and high performance computing (HPC). An additional alternative can be L_1-based feature selection algorithms. They usually have a sparse solution and must be used along with meta-transformers for selecting features based on importance weights and non-zero coefficients [24].

Acknowledgments

The authors would like to thank Eitan Lees and Shannon Duggar for the careful revision of the manuscript.

References

[1] L. Pappano, "The year of the mooc," *The New York Times*, vol. 2, no. 12, p. 2012, 2012.

[2] S. Kolowich, "The professors who make the moocs," *The Chronicle of Higher Education*, vol. 18, 2013.

[3] B. Schildkrout, *Unmasking psychological symptoms: How therapists can learn to recognize the psychological presentation of medical disorders*. John Wiley & Sons, 2011.

[4] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. Lippincott Williams & Wilkins, 2005.

[5] W. O. Tatum IV, *Handbook of EEG interpretation*. Demos Medical Publishing, 2014.

[6] C. C. Chernecky and B. J. Berger, *Laboratory tests and diagnostic procedures*. Elsevier Health Sciences, 2007.

[7] H. Wang, Y. Li, X. Hu, Y. Yang, Z. Meng, and K.-m. Chang, "Using eeg to improve massive open online courses feedback interaction." in *AIED Workshops*, 2013.

[8] Z. Ni, A. C. Yuksel, X. Ni, M. I. Mandel, and L. Xie, "Confused or not confused?: Disentangling brain activity from eeg data using bidirectional lstm recurrent neural networks," in *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. ACM, 2017, pp. 241–246.

[9] M. M. T. Rodrigo, R. S. Baker, M. C. Jadud, A. C. M. Amarra, T. Dy, M. B. V. Espejo-Lahoz, S. A. L. Lim, S. A. Pascua, J. O. Sugay, and E. S. Tabanao, "Affective and behavioral predictors of novice programmer achievement," in *ACM SIGCSE Bulletin*, vol. 41, no. 3. ACM, 2009, pp. 156–160.

[10] I. Arnaldo, K. Veeramachaneni, A. Song, and U.-M. O'Reilly, "Bring your own learner: A cloud-based, data-parallel commons for machine learning," *IEEE Computational Intelligence Magazine*, vol. 10, no. 1, pp. 20–32, 2015.

[11] J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.

[12] T. Loveard and V. Ciesielski, "Representing classification problems in genetic programming," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, vol. 2. IEEE, 2001, pp. 1070–1077.

[13] P. G. 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 (Applications and Reviews)*, vol. 40, no. 2, pp. 121–144, 2010.

[14] A. H. Gandomi, A. H. Alavi, and C. Ryan, *Handbook of genetic programming applications*. Springer, 2015.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[16] A. Tahmassebi and A. H. Gandomi, "Building energy consumption forecast using multi-objective genetic programming," *Measurement*, 2018. [Online]. Available: https://doi.org/10.1016/j.measurement.2018.01.032

[17] E. Y. Vladislavleva *et al.*, *Model-based problem solving through symbolic regression via pareto genetic programming*. CentER, Tilburg University, 2008.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley, New York, 1973.

[19] A. Tahmassebi, A. H. Gandomi, I. McCann, M. H. Schulte, L. Schmaal, A. E. Goudriaan, and A. Meyer-Baese, "An evolutionary approach for fmri big data classification," in *Evolutionary Computation (CEC), 2017 IEEE Congress on*. IEEE, 2017, pp. 1029–1036. [Online]. Available: http://doi.org/10.1109/CEC.2017.7969421

[20] A. Tahmassebi, A. H. Gandomi, and A. Meyer-Bäse, "High performance gp-based approach for fmri big data classification," in *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*, ser. PEARC17. New York, NY, USA: ACM, 2017, pp. 57:1–57:4. [Online]. Available: http://doi.acm.org/10.1145/3093338.3104145

[21] A. Tahmassebi, "ideeple: Deep learning in a flash," in *Disruptive Technologies in Information Sciences*, vol. 10652. International Society for Optics and Photonics, 2018.

[22] A. Tahmassebi and A. H. Gandomi, "Genetic programming based on error decomposition: A big data approach," in *Genetic Programming Theory and Practice XV*. Springer, 2018.

[23] A. Tahmassebi, A. H. Gandomi, I. McCann, M. H. Schulte, L. Schmaal, A. E. Goudriaan, and A. Meyer-Baese, "fmri smoking cessation classification using genetic programming," in *Workshop on Data Science meets Optimization*, 2017. [Online]. Available: http://ds-o.org/images/Workshop_papers/Gandomi.pdf

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.