EEG markers of STEM learning

Xiaodong Qu* xiqu@brandeis.edu

Yixin Sun* venusyixinsun@brandeis.edu

Robert Sekuler† sekuler@brandeis.edu

Timothy Hickey* tjhickey@brandeis.edu

*Computer Science Department, Brandeis University, Waltham, MA 02453, USA
†Volen Center for Complex Systems, Brandeis University, Waltham, MA 02453, USA

Abstract—**In this Innovative Practice Full Paper, we examined whether signals from inexpensive, wearable brainwave sensors could be used to identify the STEM learning task in which a student was engaged. Twelve subjects completed four different STEM learning tasks – two entailing passive learning (watching a video or reading), and two entailing active learning (solving problems based on the passive learning). There were two mathematics tasks (one active and one passive) and two Python programming tasks (one active, one passive). Subjects were fitted with wearable brainwave sensors that captured cortical oscillations from four scalp electrodes, and transformed the signals from each electrode into five distinct frequency bands. This yielded 10 samples per second within each frequency band and from each electrode. We then trained ensemble-based machine learning algorithms (boosting and bagging of decision tree learners) to classify various features of tasks and subjects from a single sample of brainwave activity. We explored several different types of training/testing regimes, and our results suggest that within a single session, brain activity patterns for each of these four types of learning are substantially different, but that the patterns do not generalize well between sessions. Importantly, the brainwave patterns differ greatly between individuals, which suggests that applications using such devices will need to rely on personalization to achieve high accuracy. The project is a first step toward developing apps that could use individualized EEG feedback to help subjects develop learning strategies that optimize their learning experience.**

I. Introduction

Wearable, inexpensive devices that can capture brain activity (electroencephalographic; EEG) are now widely available on the consumer market. This opens the possibility of incorporating EEG sensor data as a powerful input modality for next-generation web and mobile applications for the general population. We are particularly interested in how well these devices can categorize human cognitive activities, using machine learning to infer features of the cognitive activity from the brain sensor data. We are also interested in whether analysis of EEG signals could be an effective form of learning analytics for providing feedback to individual learners about their own cognitive state.

Our long-term goal is to analyze educationally important latent variables that are not easily quantifiable, such as classroom engagement, learning effectiveness, degree of concentration, level of anxiety, depth of creativity, etc. Given a large enough set of EEG sensor data tagged with these latent variables, we plan to use machine learning algorithms to connect the latent constructs with corresponding models in a suitable brainwave space. These models could be used for quantitative comparisons among different features of latent variables. For example, it might be possible to find a parameterized family of EEG patterns that correspond to different levels of engagement, by a single subject or across multiple subjects [1]. We focus on STEM learning activities (Science, Technology, Engineering, and Mathematics), because it is relatively easy to design and test learning activities in these domains in which skills can be learned and reliably tested in under five minutes. Moreover, there is considerable evidence that active learning is more effective than passive learning for STEM subjects [2], which suggests that it might be possible to distinguish those forms of learning using EEG data.

Here, we describe a first step toward these goals. Our study includes two sessions where each session consisted of four educational activities – two passive learning activities (listening to a lecture video and reading part of an online textbook) and two active learning activities (doing some online coding and solving mathematical problems with paper and pencil). Seven subjects took part in both sessions, and five subjects participated in just one session. Our data analysis technique then used a variant of Boosting classification (using decision trees as the basic learners) on the absolute power of five standard spectral bands (alpha, beta, gamma, delta, theta) for each of four electrodes on a wearable EEG sensor [3].

We show how boosting and bagging decision tree classification can accurately predict in which of four STEM learning activities a subject is engaged, using only a single data sample. These classifiers make predictions of various features for an EEG sample by discovering complicated arithmetic expressions involving the power values of the five bands for each of the 4 electrodes, and we refer to these machine learning classifiers as markers, in analogy with DNA markers, as their existence is correlated with the particular feature we are investigating.

Our primary research question was whether signals from a wearable EEG device could be processed with a machine learning pattern recognition approach, to yield an integrated system able to reliably distinguish different learning activities and different test subjects. We focused on relatively simple Science, Technology, Engineering, and Math (STEM) learning tasks, which could be introduced and completed within the five minutes our experiment allowed for each task. Our secondary research question was whether analysis of the EEG signals could identify which individual subject had generated a particular set of brain signals.

The long term goal for our research is to use EEG sensor data to build more effective educational technology applications. Such systems could have a wide range of pedagogical applications. For example, they could potentially indicate if the problem on which a student is working lies inside what Vygotsky called the student's Zone of Proximal Development [4]. That is, they could determine whether a problem was too easy for the student who therefore did not have to focus much attention, or whether the problem was too hard and the student would struggle ineffectively with the problem. Our approach could also help an instructor teaching a group of students to assess individual levels of engagement.

In the rest of this paper we describe collection and analysis of EEG data from 13 subjects who worked on four different STEM tasks. We then summarize what individual subject and task-features could be predicted from the EEG data, and speculate on ways in which this capability could be exploited in order to enhance STEM teaching and learning.

II. The Experiment

The data on which this paper is based came from an experiment with 13 subjects whose brainwave activity was measured with a Muse portable brainwave sensor [5]. During each of 20 minute sessions, subjects engaged in four different STEM activities: two involving mathematics problems and two involving programming problems in the Python language. Seven subjects participated in both sessions 1 and 2 and six in only one session (five in session 1 only, and one in session 2 only).

A. The Four STEM Learning Tasks

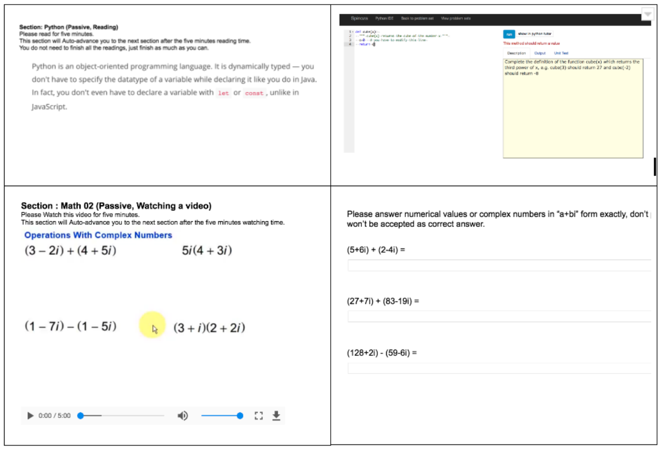
Subjects performed four learning activities from the STEM fields (Science, Technology, Engineering, and Math). As our experimental design afforded each subject only five minutes for each activity, the activities were designed to be relatively easy to learn. We refer to these activities by their two letter abbreviations: PP, PA, MP, MA:

PP For the Python Passive task, subjects read for five minutes from the first chapter of an online Python textbook. The chapter discussed basic data types, primitive operations, and syntax for variables and assignment statements. The upper left image in Fig. 1 shows a screen shot of the online Python text subjects read.

MP For the Math Passive task, subjects watched a five minute video about arithmetic operations on complex numbers. The lower left image in Fig. 1 is a screen shot from the video subjects watched.

PA For the Python Active task, students solved simple Python programming problems (*e.g.*, converting Fahrenheit to Centigrade), using the online programming tool Spinoza, which gives immediate feedback about program correctness and allows learners to resubmit attempted solutions multiple times [6]. The upper right image in Fig. 1 shows a screen shot from the Spinoza app. Subjects read the problem description in the righthand pane and entered their code in the lefthand pane. Pressing the "run" button generated unit test results, which were displayed in the righthand panel. Subjects could revise and resubmit as many times as they wanted.

MA For the Math Active task, subjects evaluated arithmetic expressions involving complex numbers with paper and pencil, and entered their answers into an online data collection system. The lower right image in Fig. 1 shows a screen shot of the web-based app that was used to present the mathematics problems and collect the subjects' answers. Subjects were not given immediate feedback but were allowed to work out their answer with paper and pencil before entering it into the online app.



Section: Python (Passive, Reading)
Please read for five minutes.
This section will Auto-advance you to the next section after the five minutes reading time.
You do not need to finish all the readings, just finish as much as you can.

Python is an object-oriented programming language. It is dynamically typed — you don't have to specify the datatype of a variable while declaring it like you do in Java. In fact, you don't even have to declare a variable with `let` or `const`, unlike in JavaScript.

```
def cube(x):
    """ cube(x) returns the cube of the number x """
    c=0  # you have to modify this line
    return c
```

This method should return a value

Complete the definition of the function cube(x) which returns the third power of x, e.g. cube(3) should return 27 and cube(-2) should return -8

Section : Math 02 (Passive, Watching a video)
Please Watch this video for five minutes.
This section will Auto-advance you to the next section after the five minutes watching time.

Operations With Complex Numbers

$(3-2i)+(4+5i)$ $5i(4+3i)$

$(1-7i)-(1-5i)$ $(3+i)(2+2i)$

Please answer numerical values or complex numbers in "a+bi" form exactly, don't
won't be accepted as correct answer.

(5+6i) + (2-4i) =

(27+7i) + (83-19i) =

(128+2i) - (59-6i) =

Fig. 1. Screen shots of displays used for each of the four activities. Upper row: PP and PA; lower row: MP and MA

After subjects performed all four tasks, machine learning algorithms were applied to find EEG markers for binary classifications: 'Activity'(Active or Passive) and 'Topics' (math or Python), as well as multiple label classification: 'Task' (PP, PA, MP, MA), and 'Subjects' (12 individual test subjects in session 1, 8 in session 2).

Six of the subjects completed a single 20 minute experimental session and seven completed two such sessions (five only participated in session 1, and one only participated in session 2). The order of activities for both sessions is shown in Table I

Order	Session 1	Session 2
1	Python Passive (PP)	Math Passive (MP)
2	Python Active (PA)	Math Active (MA)
3	Math Passive (MP)	Python Passive (PP)
4	Math Active (MA)	Python Active (PA)

TABLE I
Order of testing in Sessions 1 and 2

B. Structure of the study

After giving written informed consent and completing an intake survey, subjects were fitted with a brainwave sensor, the Muse headband (described below), and completed a 20 minute session during which they performed four STEM learning activities. Some key features of the design of each subject's

Name	Variable	Value
Activity	Active or Passive	0, 1 (0 for active)
Task	PP, PA, MP, MA	1,2,3,4
Subject	Subject id	1, 2, 3 to 8 or 12
Topic	Python or Math	0, 1 (0 for Python)

TABLE II
CLASSIFIERS:WHAT WE ARE TRYING TO PREDICT

session are described in Table III. The subjects used an online web application which gave them instructions, provided access to videos and text to read, and signaled them when it was time to move to the next task.

In Session 1, subjects first read the Python text for 5 minutes (PP), then the experimental platform signalled that they should move to the the next task (PA), in which subjects used the Spinoza platform [7] to solve several simple Python programming problems, *e.g.,* write a function to calculate the cube of an input x. At the end of this five minute section, the subjects were prompted to move to the next task (MP) no matter how many problems they had completed. In the MP task, subjects watched a five minute math video about working with complex numbers and then were prompted to complete the last task MA where they were ask to calculate sums, products, and quotients of complex numbers using scratch paper as necessary and to enter the results online. Session 1 concluded with an exit survey.

Name	Time (Minutes)	EEG Recorded
Informed Consent	2	No
Entrance Survey	2	No
Python Passive (PP)	5	Yes
Python Active (PA)	5	Yes
Math Passive (MP)	5	Yes
Math Active (MA)	5	Yes
Exit Survey	2	No

TABLE III
SEQUENCE OF ACTIVITIES IN SESSION 1 OF THE EXPERIMENT.

C. Subjects

Thirteen students were recruited for our study, eleven undergraduate students and two graduate students. Five subjects were Computer science majors; the remaining subjects were biology or psychology concentrators or had not yet decided on a field of concentration. The average age of the subjects was 20.5. There were seven male subjects and six female subjects. The survey of subjects' prior experience showed that six had some Python programming experience, and four had some Java but no Python programming experience. The remaining three had no prior programming experience. All subjects had completed a college level Calculus class. Seven subjects completed both Sessions 1 and 2; five completed only Session 1; one completed only session 2. This yielded a total of 20 sessions of EEG data from 13 different subjects.

D. Data collection.

The experiment used two data collection platforms, one is an online data collection system (Qualtrics), which the students used to perform all the tasks mentioned above except the Python Active learning (PA), which used an online programming and learning system, Spinoza [7].

We collected 40 minutes of EEG data from each of the seven subjects who completed both sessions, and 20 minutes from the remaining six subjects. The raw EEG data was collected at 220 Hz, and sent in bursts via bluetooth at about 10Hz, and the FFT data we analyzed was generated at 10 Hz and recorded the absolute power of the five standard bands (alpha, beta, gamma, delta, theta) which generated a total of 12,000 samples per subject per session. Each sample consisted of five relative power bands for each of four electrodes, so for each subject, each session, the original data is a 12,000 row, 21 column matrix, with one column for time stamps, and 20 columns for EEG power from the four electrodes in each of five frequency bands.

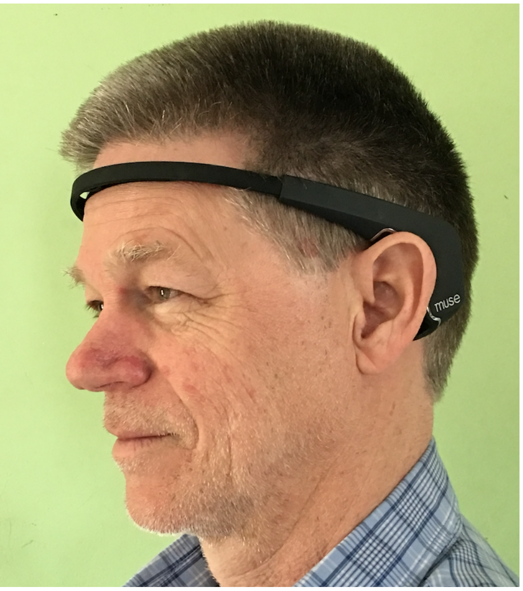


Fig. 2. A user who is outfitted with a Muse band wearable EEG sensor from Interaxon (Toronto, ON).

III. THE WEARABLE EEG SENSOR

EEG data were collected using wireless, bluetooth-enabled Muse® headsets, developed by Interaxon [5] and shown in Fig. 2. The Muse headsets were equipped with four dry sensors that made contact with the subjects' scalp. Two of the sensors were located just above the ears; the other two were located on either side of the forehead. This configuration positioned two electrodes over the brain's temporoparietal region, and the other two over the brain's frontal region. In standard, 10/20 EEG nomenclature, these correspond, to TP9, TP10, AF7 and AF8 locations. The EEG system downsampled sensor signals to 220 Hz, with 2uV (RMS) noise (Kovacevic et al., 2015; Hashemi et al., 2016). Spectral analysis was performed onboard the Muse device, and then transmitted wirelessly at 10 Hz to the experimenter's workstation using the Bluetooth protocol. The output is the EEG in the ranges shown in

Frequency Band	Frequency Range
delta	1-4 Hz
theta	4-8 Hz
alpha	7.5-13 Hz
beta	13-30 Hz
gamma	30-44 Hz

TABLE IV
FREQUENCY RANGES FOR THE FIVE SPECTRAL BANDS

Table IV. The boundaries of the frequency ranges are inclusive of the end values. Where two ranges overlapped, their shared frequency was included in both ranges.

The absolute band power for a given frequency range (for instance, alpha, 9-13 Hz) is the logarithm of the power spectral density of EEG signals summed over that frequency range. These frequency bands were computed onboard the Muse device by collecting the previous 256 raw EEG values for each electrode and using FFT to perform the spectral analysis on those values. As the raw EEG signals were sampled at 220 Hz, each FFT calculation summarizes about ∼1.16 (256/220) seconds of raw brainwave data.

IV. Method: Analysis of Data

As described in the previous section, for each of the subjects we collected 12,000 samples of absolute spectral power bands for cortical oscillation data using the Muse band during each 20-minute experimental session. Samples were taken over the course of four five-minute activities, presented one immediately after another. The four activities were PP, PA, MP, MA in session 1, and MP, MA, PP, PA in session 2. Each sample consisted of 21 numbers: the first was a timestamp (in milliseconds), followed by 20 numbers representing the log of the absolute power of each of the five spectral bands for each of the four electrodes. This resulted in a table of 12,000 rows and 21 columns.

A. Cleaning the Data

When collecting EEG data, one or more electrodes sometimes lost contact with a subjects' scalp. This resulted in multiple sequential samples from one or more electrodes that had exactly the same value. When we detected this anomaly, we removed that entire sample from the dataset, even if the anomaly was only detected on one electrode. This resulted in a loss of about 30% of all samples.

B. Machine Classification

For each of the four features in our data: Activity (Passive or Active), Task (PP, PA, MP, MA), Subject (1-12), and Topic (Python or Math), we used either boosting (for the binary properties Activity and Topic) or bagging (for Task and Subject) to train classifiers on a subset of the data and test them for classification accuracy on the rest of the data.

For the Activity and Topic features, we used the Matlab `fitcensemble` function with the `GentleBoost` method to perform boosting for binary classification. For Task and Subject, we used the `fitcensemble` function with the `Bag` method for multiclass classification. For all of these classifiers we used decision trees as basic learners.

Boosting [8] and Bagging [9] are instances of Ensemble Machine Learning Algorithms which use a set of weaker learners to make a more accurate classification. They are trained by forming a weighted average of the classifications of the weaker learners and then applying optimization techniques to maximize the accuracy of that weighted average. In our case, we used boosting and bagging of decision tree classifiers. The Matlab function we used, `fitcensemble`, employs a variant of the most widely used form of boosting algorithm called AdaBoost (adaptive boosting), which was developed by Freund and Schapire (1996) Boosting can produce good results even if the base classifiers perform only slightly better than chance. Therefore, the base classifiers are sometimes described as ”weak learners.”

Decision Tree classifiers [10] can be represented as trees whose nodes represent Boolean tests that determine whether the algorithm proceeds to the tree's left branch or right branch [11], [12], [13]. In our application, these Boolean tests are linear hyperplane conditionals that split a 20-dimensional region of brainwave data into two subsets with a linear boundary. There are many ways to create such trees, but any single tree is unlikely to generate a highly accurate classifier as the regions represented by the leaves of the classification tree are simple convex multidimensional polyhedra [14]. The basic learner used by `fitcensemble` uses a particular tree-based framework called ”classification and regression trees,” or CART (Breiman et al., 1984), although there are many other variants such as ID3 and C4.5 [15].

V. Training and Testing the Classifiers

We employed five training/testing strategies to explore various features of these machine classification algorithms when applied to brainwave data. Four of the approaches are variations on k-fold cross-validation in which the data is chunked into k segments, a classifier is trained on $k-1$ of the segments, and tested on the remaining segment. Accuracies from various ways of chunking the data are then averaged provide an estimate of the classifier's effectiveness. The fifth method we employed was to train on sparse subsets of samples. To do this, we took samples that were uniformly distributed with a distance of k between each consecutive pair of training examples. We then tested the classifier on all of the remaining samples.

We trained and tested four different classifiers using this approach.

- **Topic** classifier – predict the topic being studied when the sample was taken, mathematics or Python programming
- **Activity** classifier – predict whether a sample was from an active learning task or a passive learning task
- **Task** classifier – predict which of the four tasks (MP, MA, PP, PA) the subject was engaged with when the sample was taken
- **Subject** classifier – predict the subject from the sample

This section describes the various approaches in detail and reports the results. The next section we discuss and interpret these results.

A. Randomized five-fold cross-validation

As our goal was to identify EEG markers of STEM learning, we initially used a standard cross-validation approach. Data were divided into training sets (each with 80% of the samples) and test sets (each comprising the remaining 20% of samples). We processed each subject's data by partitioning it into five randomly selected subsets of equal size.

The cross-validation feature was generated by shuffling the data for each subject and then partitioning the resulting data into fifths. We then trained on four of the fifths, tested on the remaining subset, and averaged the results. The cross-validation feature allowed us to break the entire dataset into five disjoint subsets to be used for cross-validation.

The results of these cross-validation tests are shown in Tables V and VI. We see a surprisingly high prediction accuracy of over 90%. This means that given a single sample from the testing set, the corresponding feature could be predicted with over 90% accuracy. This is surprising because we would expect that many cognitive activities would appear in multiple of these STEM learning activities (e.g. reading words, making numerical estimates, etc.) and so the ability to predict which task a single sample corresponds to with 90%+ accuracy is surprising.

Feature	Value	Accuracy	Range	Random
Topic	Python, Math	0.946	0.938-0.950	0.50
Activity	Active, Passive	0.937	0.927-0.943	0.50
Task	PP PA MP MA	0.924	0.920-0.929	0.25
Subject	101 102 103 etc	0.950	0.94.5-0.954	0.083

TABLE V
RESULTS FROM RANDOMIZED 5-FOLD CROSS VALIDATION ON SESSION ONE DATA; 12 SUBJECTS

Name	Value	Accuracy	Range	Random
Topic	Python Mat	0.957	0.956-0.958	0.500
Activity	Active Passive	0.960	0.953-0.964	0.500
Task	PP PA MP MA	0.928	0.922-0.938	0.250
Subject	101 102 103 etc	0.960	0.955-0.963	0.125

TABLE VI
RESULTS FROM 5-FOLD CROSS VALIDATION ON SESSION TWO DATA: 8 SUBJECTS

This randomized 80/20 approach is a standard practice in the machine learning community, but Saeb et al. [16] observed that such record-wise cross-validation is susceptible to over-fitting when applied to time-series data. This occurs because there is a high probability that neighbors of a test data point in a time-series will be in the training data set. This artifact, which we call the Time Continuity Effect, may well have influenced results from our Randomized 80/20 split approach.

From one perspective, the Time Continuity Effect is actually beneficial. After all, if two data samples were close to one another in the 20 dimensional brainwave space (using the standard Euclidean distance measure), then they are likely to have arisen from similar cognitive states (such as Activity, Topic, Task, or Subject), which is a necessary feature for successful machine classification. It does not however guarantee that such a classifier would generalize to other subjects, to other tasks, or to tasks that, while superficially similar, incorporate distinct learning materials (*e.g.*, distinct Python problems). In the remainder of this section we will explore these questions by varying the training and testing subsets and examining the effect this has on prediction accuracy.

B. Regular interval training/testing

In this approach, which is not a cross-validation method, we tried to estimate the influence of the Time Continuity Effect by selecting data points that are separated by regular intervals as training data, and then testing the resulting classifier on all of the rest of the samples. We started by appending all of the samples from all of the subjects of session 1 into a single sequence of 12,000*12 = 144,000 samples, then for each k selected 144,000/k of those elements for training and tested it on the remaining elements. By varying the inter-training sample distance we can see how prediction accuracy is affected by the average distance of testing elements from the set of training elements.

For example, if the selected interval were $k = 10$, then the training set would consist of the each 10^{th} sample in the set, *i.e.*, one sample selected at the beginning of each second of data, and the average distance between a testing sample a the training set would be 2.5 samples. In practice we applied this only to the cleaned data, so all we can say is that for k=10 the successive samples would be at least 1 second apart, and possibly more if successive clean samples were separated in the time series by discarded, ”dirty” samples.

The prediction accuracies for k varying from 2 to 4096 are shown in figure 3. To clarify this training/testing regimen, lets look at the case for the interval size of k=128. In this case, we consecutively numbered all of the clean samples from all the tasks of all 12 subjects in Session 1, and then took as a training set, those samples whose indices were multiples of 128. This results in a training set containing only 1/128 = 0.78% of the data, and each pair of successive training samples is at least 128/10 = 12.8 seconds apart.

We then used the classifier trained on that subset to test the remaining the data (99.22% of the samples). The table shows that we obtained surprisingly high accuracy levels of around 65-70% for the Activity, Topic, and Subject features and about 45% for the Task (MP, MA, PP, PA) in the $k = 128$ case. These high accuracies indicate that the classifier is able to distinguish these particular activities with high accuracy, but it doesn't mean that the Topic classifier (Math/Python) will also have high accuracy for some other Math and Python programming activities.

One particularly surprising feature of the data in Fig. 3 is that the classifier which predicts which of the 12 subjects corresponds to a particular sample from the testing set has

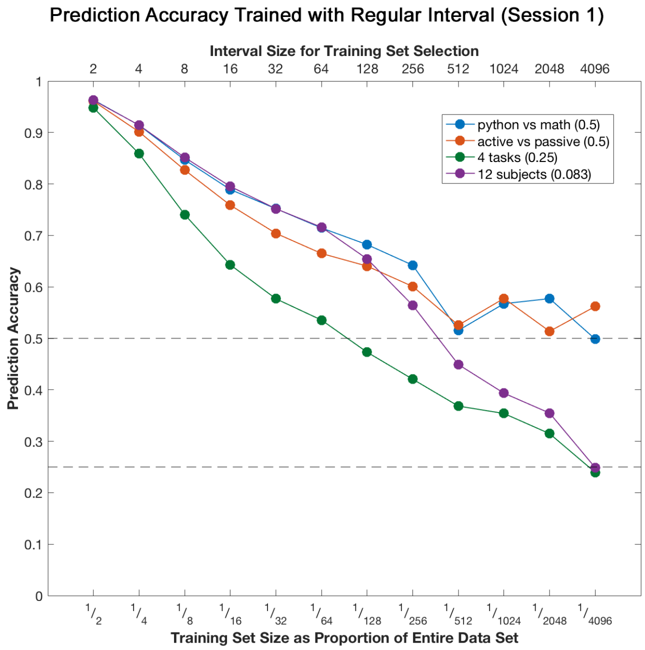


Fig. 3. Prediction accuracy for each of the four features (Topic, Activity, Task, Subject) when trained on a subset of the data where successive samples are separated by a fixed distance, k. The resulting classifer is tested on all of the remaining data. The dashed lines represent the expected prediction value for random guessing for Topic and Activity (at 0.5) and for Task at (0.25). Even when $k = 4096$ (which is only 2 or 3 samples per subject), the accuracy is twice the expected average of 0.125 for random guessing.

accuracy of about 70% for $k = 128$ when the random level is only 8.3%. This means that training the system on less that 1% of the data allows the system to correctly predict the individual for a single sample with accuracy of about 70%, and so the brainwave patterns for the individuals must be very different! Successive samples in the training set are at least 12.8 seconds apart.

To further explore possible differences between the subjects EEG data, we plotted the average power for each of the five bands and each of the four electrodes. for each of the 12 subjects. Fig. 4 shows that data and it is clear that there is a wide variety in these gross EEG features for the individuals.

C. Subject-wise twelve-fold cross validation

For this approach we trained the classifiers on 11 of the 12 subjects who completed session 1 and then tested on the remaining subject. This is the approach suggested by Saeb et al. [16] to counteract the Time Continuity effect. The training and testing used all of the data for all 12 subjects. We trained three classifiers. One for Active/Passive, one for Math/Python, and one for the four different tasks (MP, MA, PP, PA) Table VII shows the average prediction accuracy for these 12 cross-validation folds. The results are slightly above random, showing the method has some predictive capability, but is disappointingly low.

This low predictive capability is most probably because, as we saw from the Regular Interval Training classifiers, there is a great deal of variation among individuals' brainwave patterns for these four activities. This bespeaks the need for personalization in any application that would use EEG signals to categorize tasks, and then to guide individuals' performance.

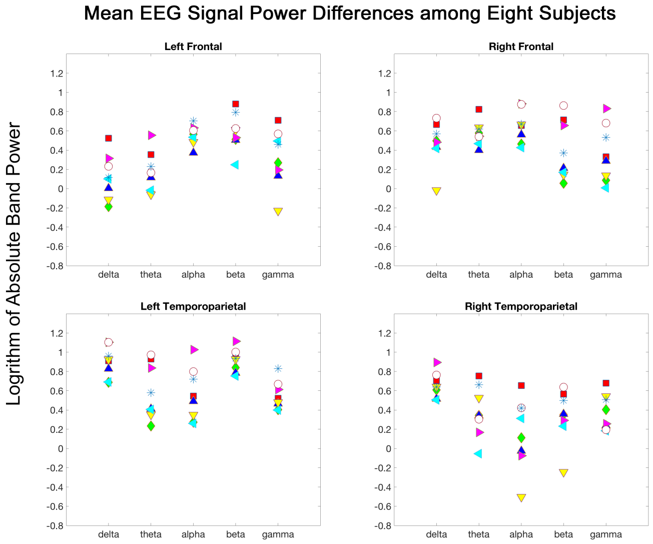


Fig. 4. These plots show the subject-wise differences of the mean values of the five EEG power bands for each of the four eletrodes. The four panels correspond to the left and right frontal elecrodes and the left and right temporoparietal electrodes. Each subject is represented by a different marker (e.g. circle, disk, square, diamond, etc.) and we can see that different subjects have wide variability in their mean power per band per electrode, which might partly account for the high accuracy in the subject classifiers which we have seen.

Feature	Value	Prediction Accuracy	Random
Topic	Python/Math	0.57±0.034**	0.50
Activity	Active/Passive	0.54±0.050 ns	0.50
Task	PP PA MP MA	0.34±0.033***	0.25

Note: ns = p>0.05: ** = p≤0.01; *** = p≤0.001.

TABLE VII
SUBJECT-WISE CROSS VALIDATION, SESSION ONE, 12 SUBJECTS

D. Time-wise five-fold cross validation

To lessen the over-fitting that comes from allowing test samples to be chronologically close to a training sample, we turned to a different cross-validation method. Since each of the tasks in a single session was five minutes, we could divide the samples for each task into five subsets corresponding to each of the five minutes. We could then train on four of the minutes (*e.g.*, 2,3,4,5) of all the tasks and test that classifier on the remaining minute (*e.g.*, minute 1) of all tasks. This provided 16 minutes of training data (9600 samples) for 4 minutes of testing data (2400 samples). We actually cleaned the data in each of those 1 minute intervals resulting in data sets with fewer that the 600 samples of raw, uncleaned data. Fig. 5 shows a graphical representation of this cross-validation approach where we are testing on the first minute of each task.

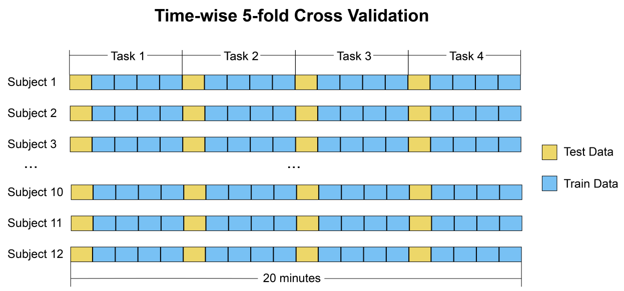


Fig. 5. Training/Testing decomposition for the Time-Wise cross validation approach. The four yellow boxes in each row correspond to the four testing minutes (the first minute of each task), and the remaining 16 blue boxes correspond to the four training minutes for each of the four tasks.

We performed 5-fold time-wise cross-validation of our machine learning classifiers separately for Session 1 data, and Session 2 data, and for each of the four features: Activity, Task, Subject, and Topic. The prediction accuracy for the session 1 classifiers are shown in Table VIII, session 2 classifier accuracy is in Table IX. These results are more what we expect from a non-overfitted classifier. We get prediction accuracies of over 50% for the four different tasks which is impressive seeing that a random classifier would generate about 25% accuracy, these tasks likely share many similar cognitive activities.

Feature	Values	Accuracy	Random
Topic	Python/Math	0.71±0.030****	0.50
Activity	Active/Passive	0.66±0.070**	0.50
Task	PP PA MP MA	0.54±0.081**	0.25
Subject	101 102 103 etc	0.75±0.042****	0.083

Note: * mark indicates the corresponded p value from t-test, which represents how significantly the results are different from random, ** means $p \leq 0.01$, *** means $p \leq 0.001$, and **** means $p \leq 0.0001$.

TABLE VIII
TIME-WISE CROSS VALIDATION, SESSION ONE, 12 SUBJECTS

Feature	Value	Accuracy	Random
Topic	Python Math	0.71±0.043***	0.500
Activity	Active Passive	0.77±0.080**	0.500
Task	PP PA MP MA	0.59±0.086***	0.250
Subject	101 102 103 etc	0.81±0.019****	0.125

Note: * mark indicates the corresponded p value from t-test, which represents how significantly the results are different from random, ** means $p \leq 0.01$, *** means $p \leq 0.001$, and **** means $p \leq 0.0001$.

TABLE IX
TIME-WISE CROSS VALIDATION, SESSION TWO, 8 SUBJECTS

E. Session-wise two-fold cross validation

In this approach we trained the classifiers on the data of all 7 subjects who participated in both sessions, and we trained the classifier on one session, and tested it on the other session.

The results of our session wise cross-validation are shown in Table X. This shows the average accuracy for prediction when training using the data of all 7 subjects who participated in both sessions, with one session for training and the other for testing. The results are substantially lower that when cross-validating within a session, but are still considerably above what a random classifer would produce. We suspect that if we had each subject participate in 10 sessions and if we carefully designed the sessions to require very similar cognitive activities, then a 10-fold cross-validation would produce much higher accuracies than we are seeing for this example.

Feature	Value	Accuracy	Random
Activity	Active Passive	0.63±0.076*	0.50
Task	PP PA MP MA	0.37±0.050*	0.25
Subject	101 102 103 etc	0.39±0.013***	0.14
Topic	Python Math	0.57±0.0054***	0.50

Note: * mark indicates the corresponded p value from t-test, which represents how significantly the results are different from random, * means $P \leq 0.05$, ** means $p \leq 0.01$, and *** means $p \leq 0.001$.

TABLE X
SESSION-WISE 2-FOLD CROSS VALIDATION, 7 SUBJECTS

VI. DISCUSSION

Our goal was to study the effectiveness of machine learning classification algorithms in distinguishing among different STEM learning tasks (active/passive, python/math) and in identifying different individuals by their brainwave patterns while they engage in STEM learning tasks.

Each of the brainwave samples corresponds to the power spectrum analyses of about 1.16 seconds of raw EEG data (256 raw samples collected at 220 Hz transformed with a 256 element Fast Fourier Transform and then binned into five distinct frequency bands).

Our hypothesis is that each of these samples represents a signature for some particular cognitive activity, *e.g.*, doing mental arithmetic, remembering some concept, comparing two objects for differences, decoding written text, etc. Thus, for each of these tasks, signatures of cognitive activities are being generated at the rate of 10 Hz in our EEG data. However, we do not know the details of those activities.

Our hypothesis though is that our four STEM tasks (MP, MA, PP, PA) differ somewhat in the sets of cognitive activities they require, *e.g.* looking for Python bugs might require different cognitive activities than calculating the product of two complex numbers. Many components of each set though are shared over multiple tasks, *e.g.*, all four tasks entail reading of text, and this will reduce the ability of any classifier to predict the task based on a single sample, as that sample could have been collected while the subject was involved in one of these common tasks. Thus, we would be surprised to discover that the tasks could be distinguished with very high accuracy based on a single sample. On the other hand, we expect that active and passive learning would produce some differences in the kinds of cognitive activity (especially since there is considerable data which shows active learning is superior to passive learning), so we would also be surprised to see near-random predictive values.

The results from the Randomized five-fold cross-validation, shown in Table V and Table VI, demonstrate that the classifier was able to attain an average accuracy of between 92 and 95 percent for the four features we examined, which is very surprising since we would expect that these tasks share many of the same cognitive activities. These are very high accuracies, and they suggest that the classifiers may be over-fitting the data. Indeed, it is easy to see that the probability that a data point in the testing set would be adjacent to a data point from the training set is very high (96%). Saeb et al. [16] mentioned that such record-wise cross-validation for time-series data may have an over-fitting effect, which we call the time continuity effect.

This effect tells us that when two samples are close in the 20 dimensional Euclidean space metric, there is a high probability that the samples share other features such as the task, activity, topic, or subject. This is welcome news if we are trying to build tools which recognize these features automatically, but does not guarantee that the classifier will generalize to other subjects, or other tasks.

Another approach for avoiding the time continuity effect was the Time-wise Cross-Validation method whose results are shown in Tables VIII and IX. Prediction accuracies are well above random, and are about equal to the accuracies from Regular Interval training with $k = 16$. However, here the testing intervals are at least 60 seconds wide, which would correspond to $k = 600$. This shows that that this style of training produces a more reliable classifier than the Regular Interval training approach does.

We also observe that the prediction accuracy for Subject-wise cross validation is barely above random. This result can perhaps be explained by the striking dissimilarity of brainwave patterns between individuals as shown by the very high accuracies with which almost all of the classifiers can predict which individual produced a particular sample.

Our final observation is that the within-session prediction accuracy of the Time-Sliced Cross-validation study was substantially higher that the between session cross-validation prediction accuracies. This is not surprising as the former was trained on four minutes of the same activity that it was tested on, while the latter was trained on five minutes of one activity and tested on a different activity, on a different day, albeit of the same type, *e.g.* Active or Passive learning.

VII. Limitations

This pilot study has a number of limitations that we will address in future research. The most pressing limitation is the relatively small size of the data set: only 13 subjects were tested, each in just one or two short sessions and in only four STEM learning activities. A larger number of subjects and a greater variety of activities will allow us to examine a number of important additional variables, including (i) the structural relationships among multiple types of STEM activities; (ii) the subject characteristics that affect the accuracy of predictions generated by machine learners.

VIII. Conclusions and future work

Our study demonstrated that machine learning can be applied to brain signals in order to fairly accurately predict which of four STEM learning activities a subject is engaged in. Moreover, machine learning achieves these levels of accuracy with just a single sample from the testing dataset. Our system was also able to use a single sample of EEG activity from the test set to pick out which individual, out of 12, had generated that sample, with accuracy of 75-80% for Time-Wise cross validation.

Our pilot study raises several interesting and important questions. For example, how general are the particular markers generated by these machine learning methods? Do the markers trained on one kind of activity (*e.g.,* Python programming) generalize to similar activities (*e.g.,* solving different Python programming problems, or programming problems in different languages). We showed that four STEM learning activities could be distinguished with relatively high accuracy (PP, PA, MP, MA), but how many distinct activities can be similarly distinguished. With enough data, we could also train and test classifiers on single subjects to see if restricting to a single subject increases the predictive accuracy of the machine classifier.

In this study, we focused on Python and Math but the same methodology could be applied to virtually any human activity in which cognition plays a major role, *e.g.,* reading and writing, musical performance, athletic performance, etc. In the future, we will look at other cognitive activities besides active and passive learning of Python and Mathematics. Moreover, we will also focus on the more difficult problem of using brainwave data to estimate the quality of the cognitive activity, that is, how effectively is the subject using their cognitive facilities to learn the particular skills and concepts. One approach would be to have students work on problems spanning a wide range of difficulty, and then attempting to train classifiers to use EEG samples to predict a problem's difficulty.

Finally, we intend to design, deploy and evaluate cognitively-based STEM coaches that exploit a student's brainwave signals to provide biofeedback about the quality of their engagement in the learning process. For STEM learning, there is considerable evidence that students learn more effectively through active, engaged learning than through passive learning. Our research could lead to the development of systems that can characterize the particular markers for active STEM learning and use these to provide feedback to students and/or instructors as to the effectiveness of a particular learning activity.

Our long term goal is to build a "Thinking Cap" application that could be trained to detect important latent variables key to learning's effectiveness (such as focus, engagement, effectiveness of working memory) and to use this information to discriminate between effective and ineffective problem solving activities. This could provide students with valuable, real-time neurofeedback that they could use to sharpen their learning skills. We believe that our results represent a first step toward this important goal.

IX. Acknowledgments

We thank Brandeis University's Provost's Research Fund for its generous support, and all the subjects who participated in our study.

References

[1] A. Hashemi, L. J. Pino, G. Moffat, K. J. Mathewson, C. Aimone, P. J. Bennett, L. A. Schmidt, and A. B. Sekuler, "Characterizing population EEG dynamics throughout adulthood," *ENeuro*, vol. 3, no. 6, pp. ENEURO–0275, 2016.

[2] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.

[3] X. Qu, M. Hall, Y. Sun, R. Sekuler, and T. J. Hickey, "A personalized reading coach using wearable EEG sensors - a pilot study of brainwave learning analytics," 2018.

[4] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Harvard University Press, 1980.

[5] Interaxon. (1999) Muse developer website. [Online]. Available: http://developer.choosemuse.com

[6] F. A. Deeb and T. Hickey, "Flipping introductory programming classes using spinoza and agile pedagogy," in *Frontiers in Education Conference (FIE)*. IEEE, 2017, pp. 1–9.

[7] T. Hickey and F. Abu Deeb, "Spinoza: In-class python problem solving with classroom orchestration," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, 2018, pp. 1112–1112.

[8] Y. Freund, "A more robust boosting algorithm," *arXiv preprint arXiv:0905.2138*, 2009.

[9] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[10] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[11] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[12] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 20, no. 8, pp. 832–844, 1998.

[13] D. Coppersmith, S. J. Hong, and J. R. Hosking, "Partitioning nominal attributes in decision trees," *Data Mining and Knowledge Discovery*, vol. 3, no. 2, pp. 197–217, 1999.

[14] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[15] W.-Y. Loh and Y.-S. Shih, "Split selection methods for classification trees," *Statistica sinica*, pp. 815–840, 1997.

[16] S. Saeb, L. Lonini, A. Jayaraman, D. C. Mohr, and K. P. Kording, "Voodoo machine learning for clinical predictions," *Biorxiv*, p. 059774, 2016.