# Data Mining Approach for Determining Student Attention Pattern

Sujan Poudyal, M. Jean Mohammadi-Aragh, and John E. Ball
*Department of Electrical and Computer Engineering*
*Mississippi State University*
Mississippi State, MS, USA

*Abstract*— **This Research Full Paper presents the approach of traditional engineering analysis techniques on education. Data mining techniques have been successfully employed to extract hidden information from large data sets within various contexts. We hypothesized that data mining techniques can similarly be applied to large educational data sets to extract and analyze patterns and create insights. Specifically, we examined the degree to which standard data mining techniques can distinguish between different student attention patterns in large lectures in which personal computers were actively used. Our data set consists of electronically captured student attention data (on-task, off-task) that was recorded at 20 second intervals throughout each course lecture over one semester. With Institutional Review Board (IRB) approval, methods involved capturing student data via a backend monitoring system to reduce student awareness of monitoring and reduce false behavior changes during data collection periods. The data were originally captured in the form of images (screenshots), and image processing techniques were applied to extract student attention patterns in the form of zero (off-task) and one (on-task). We conducted descriptive statistical analysis to add other features such as characterization information (e.g., total logged in attention, average class period attention to the recorded data sets). For the data mining analysis, we used three different supervised machine learning classification algorithms: Support Vector Machine (SVM), Decision Tree, and K-Nearest Neighbors (KNN) for classifying the students' dataset. We classified the students into one of four different classes based on their attention pattern in the lecture class. Before applying each classification algorithm, feature extraction was performed. For this purpose, we used Haar wavelets, Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA) dimensionality-reduction techniques before applying the classification algorithm. Their performance is compared. Our results indicate high classification accuracies can be obtained using these dimensional reduction algorithms followed by classification algorithms. Our result highlights the importance of applying traditional engineering analysis techniques to educational data in order to provide engineering education insights.**

***Keywords— Data mining, SVM, Decision tree, KNN, learning analytics***

## I. INTRODUCTION

Computers are ubiquitous in engineering classrooms. Students are regularly required to purchase laptops as part of the requirements of enrolling in engineering degree programs. Then, students routinely bring their laptops to class in order to actively engage in course material (e.g., complete in-class MATLAB activities). Further, even in courses where laptop use is not led by the instructor and embedded directly into the course activities, students bring their computers and complete tasks related to the course (e.g., note taking) and tasks unrelated to the course (e.g., responding to email, completing homework for another course). Laptops are especially prevalent in large lecture courses, which due to the sheer number of students sitting in a large auditorium format provide a degree of anonymity to students and arguably encourage off-task, unstructured laptop usage. Numerous researchers have focused on understanding the impact of laptops in classrooms. Multiple studies have produced results demonstrating learning gains that support the use of laptops in the classroom while other studies illustrate how laptops interfere with learning and recommend classroom-wide laptop bans. Our position is that computers will be in the classroom, whether they are large, visible laptops or small, hidden smart watches. Therefore, rather than focus on the dilemma of "to ban or not to ban", we should focus on understanding how we can encourage learning in classrooms where computers are ubiquitous.

The battle to improve learning in computer-embedded classrooms hinges on the fact that computers distract students from learning tasks. According to the Robert Gagne's Conditions of Learning, paying attention is a prerequisite for learning [1]. Computers and all their applications compete with the instructor for students' attention. Instructors have control over student attention [2] and can utilize pedagogies of engagement, such as active learning techniques, to help student re-engage with lecture content. However, knowing when to time an intervention is not an exact science. Commonly, instructors attempt to gauge the "pulse" of the class by making a real-time observation of class attention levels. However, as enrollments increase, and lecture sizes grow this becomes increasingly difficult. In a lecture with 30 students, an instructor can likely skim the class and use their intuition to determine whether students are paying attention or not, but in a lecture of 300, the sheer number of students and size of the lecture hall makes the decision less clear. In these large lecture halls, especially when computers are used for instructor-led activities, real-time data mining of network or server data could help inform the instructor of periods of inattention. We hypothesize that traditional data mining techniques can be applied to computer usage data to accurately indicate whether students are completing course-related activities or not. In this paper, we explore the application of six standard data mining techniques (three dimensionality-reduction techniques and three classification algorithms) in order to answer: *How accurately can standard data mining techniques distinguish student attention patterns as on-task or off-task?* With acceptable accuracy, employing traditional data mining techniques could be one step towards designing a data-driven classroom pulse that could alert an instructor to periods of low attention levels in their classroom.

### A. Short introduction to data mining techniques

Data mining techniques are used to extract the hidden information from large data sets within various contexts. We have used six data mining techniques for this research including three classification algorithms (Support Vector Machine, Decision Tree, and K-Nearest Neighbor) and three dimensionality-reduction techniques (Haar wavelets, Principal Component Analysis, and Linear Discriminant Analysis). Classification algorithms automatically categorize data. Dimension reduction techniques are used to obtain the data set that has lesser dimensions than the original one, but it should convey the similar information. Each of the six data mining technique are briefly described in this section.

*1) Support Vector Machine (SVM):* Multiclass SVMs are learning models that project the feature data to a higher dimensional space to classify the data. [3] Along with pattern recognition applications, SVM is also widely used for classification [4]. If SVMs are trained with sufficient training data they often provide high classification accuracies [4]. SVMs have been used in a variety of applications such as face detection, classification of images, handwritten recognition, and many more where data requires classification. In our work, the linear SVM and the kernel SVM are used. The linear SVM applies a linear projection to the data and is most useful if the data are linearly separable. The kernel SVM applies a nonlinear kernel to the data, then classifies using a linear classifier in the kernel-projected data space. The kernel SVM is more powerful, but requires more work to select the appropriate kernel.

*2) Decision Trees:* Decision tree classiers build a tree structure by dividing the dataset into subsets and incrementally developing the associated decision tree. The starting node of the decision tree is called the root node. We jump to the next node from the root node by comparing the values of the root attribute with that of the record's attribute. There are decision nodes (sub-nodes following the root node) and leaf nodes (terminal nodes). Decision nodes are further divided into the branches. The decision or the classification is given by the leaf node. Decision tree can give us 100% accuracy on the training data if it makes one leaf node for each observed data but may perform poor on the testing data. This condition is called overfitting and it can be often be remedied by pruning the decision tree, which means that one trims the branches of the decision tree in such a way that the overall accuracy is not harmed. Pruning algorithms have been developed to achieve the good classification accuracies [5].

*3) K-Nearest Neighbor (KNN):* KNN is a machine learning and data mining algorithm that is widely used as a prediction method because it is a simple and versatile method [6]. It is a versatile method because it can be used for classification, regression and search. KNN makes predictions by calculating the distance between the input samples and the training samples, and then classifying the input sample based on the shortest distance to training data (i.e., nearest neighbor). In our work, we use the Manhattan distance given in (1) as the distance metric. The value *K* in the KNN algorithm is the number of the nearest neighbors used for classification purposes. In our work, the value of *K* is one. KNN can be used for handwriting detection, image recognition, and video recognition, to name a few sample applications.

$$Manhattan\ Distance = \sum_{i=1}^{k} |x_i - y_i| \tag{1}$$

where, k is number of variables, $x_i$ and $y_i$ are the variables of vector x and y respectively.

*4) Haar wavelets:* The Haar wavelet is the simplest wavelet that takes the value of positive and negative unity within the defined ineterval and it dissapears outside that interval. Wavelet analsyis allows a function to be represented using an orthonormal basis of Haar wavelets. A simple two-level Haar wavelet is shown in Fig. 1. The two filters are applied repeatedly followed by subsampling the signal for analyzing the signal by discrete Haar decomposition, as shown in Fig. 1. There are many wavelets that can be utilized, but in our case, the data is most suited for the Haar wavelet.

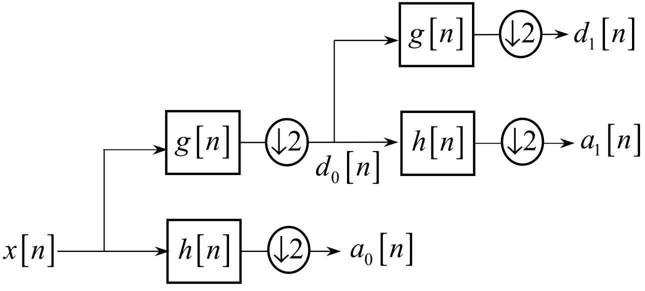

Fig. 1. Two-level Discrete Wavelet Transform(DWT) with Haar wavelet

The circles in Fig.1 repreesnt the downsamplers. Downsampling by two means we throw away every other data sample. The two filters are defined by eqs. (2) and (3) are the high pass and low pass filters respectively. The output of the high pass filter give the approximation coefficients (the high-frequency details in the signal) and the low pass filter gives the detail coefficients (the low-frequency content). The outputs $a_m[n]$ and $d_m[n]$ are the approximation and detail coeffienents respectively at level *m*, where *n* is the sample index. Haar wavelets provide time and frequency resolution simultaneously, and are used in a wide variety of data processing applications such as time-series analysis, image compressing [7] and denoising, and image and signal processing [8], to name a few.

$$\square[n] = \frac{1}{\sqrt{2}}[-1,1] \tag{2}$$

$$g[n] = \frac{1}{\sqrt{2}}[1,1] \tag{3}$$

*5) Principle Component Analysis (PCA):* PCA is a dimensionality reduction algorithm that evaluates the correlation among different input dimensions [9]. It decomposes a signal into orthogonal components and ranks them by their associated contribution to the overall signal power. In this way, one can create a lower-dimensional approximation to the original signal by only keeping the top *m* eigenvectors of the original signal. PCA is used extensively in signal processing as a dimensionality reduction tool.

*6) Linear Discriminate Analysis (LDA):* LDA is another dimensionality reduction technique that projects the data to the space that maximizes the class separability, by simultaneously maximizing the between-class separation and minimizing the within-class spreading (variance). According to [10], LDA is given by the vector v that maximizes eq. (4).

$$J(v) = \frac{v^T s_B v}{v^T s_W v} \tag{4}$$

where, $s_B$ and $s_W$ represent between-class scatter matrix and within-class scatter matrix respectively, and $v$ is the input vector. The superscript T denotes the transpose operator. LDA can be applied to bankruptcy prediction, face recognition, marketing, and prognosis of disease outcome in biomedical science, to name a few applications.

### *B. Features*

During the data collection, the attention pattern of each student was encoded as a binary value: zero or one, where a zero indicates off-task, and a one indicates on-task. Encoding is done using the method described in [11]. On-task represents that the students are paying attention in the classroom whereas off-task represents that the students are inattentive in the classroom. The Haar wavelets, PCA, and LDA techniques were utilized to extract the features from the attention pattern. These features were concatenated with the other 13 characterization features that were calculated manually from the obtained attention pattern for each student. The other characterization features are as follows: 1) count of on-task entries, 2) count of off-task entries, 3) on-task attention percentage, 4) class period attention, 5) total number of switching between the on-task and off-task, 6) average of the consecutive on-task entries, 7) minimum consecutive on-task entries, 8) maximum consecutive on-task entries, 9) average of the consecutive off-task entries, 10) minimum consecutive off-task entries, 11) maximum consecutive off-task entries, 12) time duration at which the first off-task occurred, 13) total time duration for the first on-task period.

## II. METHODS

### *A. Experimental Setup*

Educational data were captured electronically in a large classroom in a first-year engineering lecture. Students who enrolled in the first semester of an engineering course (n=256) and consented to participation (n=203) were included in the study. The lecture portion of the course was 50 minutes and met once per week in the early morning. Students also attended 110-minute laboratory sessions in groups of less than 30 students once per week. However, data were recorded during the lecture period only. Students demographics were typical for first-year engineering courses in the United States (i.e., predominately male, 18 years of age). For the data collection conducted with Institutional Review Board (IRB) approval, students were informed to bring their laptops and to log into the course software. There was a network connection between the student's and the instructor's computer so that the instructor could share the lecture content with all the students. The network link between the instructor and student computers was used to determine whether students were active within the required course software on their laptop (on-task) or were using their laptop for other activities (off-task). The data were captured electronically and were recorded in the form of the screenshot at 20 seconds interval throughout each 50-minute lecture over one semester. The original screenshot data was analyzed using MATLAB scripts and the student attention patterns were extracted in the form of zero (off-task) and one (on-task). After extracting each students' attention pattern, MATLAB scripts were used to calculate the thirteen other features listed in Section I.D. A detailed description of data collection methods is provided in [11] which includes a discussion of method validation, error rates, and limitations. We note that the methods involved capturing student data via a backend monitoring system to reduce student awareness of monitoring and reduce false behavior changes during data collection periods (i.e., students using a smart phone and leaving their computer set to "on task" software). Additionally, the electronic monitoring was supplemented by in-person observations of student behavior that quantified the method's mean percent error at 4.28% and estimated a standard error of 0.82.

The attention patterns of every student were plotted using MATLAB and the result was observed by an expert. The expert observed four general classes of attention. The four different classes are:

- Class 1: Attentive students,
- Class 2: Students with primarily attentive periods with few inattentive periods,
- Class 3: Students with primarily inattentive periods with few attentive periods, and
- Class 4: Inattentive students.

Figs. 2-5 shows representative samples for four different classes. In Figs. 2-5, the blank space in the plot represents the off-task period and the shaded portion represents the on-task period. The x axis in these figures are the sample indices when the attention was encoded. Fig. 2 shows the sample of attention pattern to represent the students of class 1. Here the students are mostly attentive throughout the lecture as represented by shaded portion in Fig. 2. Fig. 3 shows the sample of attention pattern to represent the students of class 2. Here, the students are mostly attentive represented by shaded portion and has few inattentive periods represented by blank space in Fig. 3. Fig. 4 shows the sample of attention pattern to represent the students of class 3. Here, the students are mostly inattentive throughout the lecture as represented by blank space in Fig. 4 with few attentive periods as represented by shaded portion in Fig. 4. Fig. 5 shows the sample of attention pattern to represent the students of class 4. Here the students are mostly inattentive throughout the lecture as represented by blank space in Fig. 2.

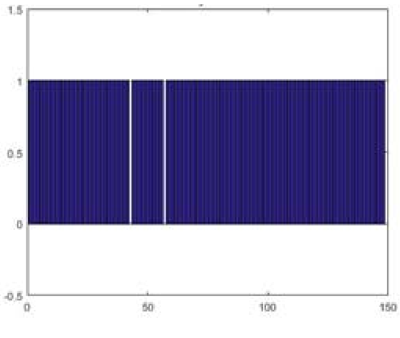

Fig. 2. Sample to represent students of Class 1: Attentive.

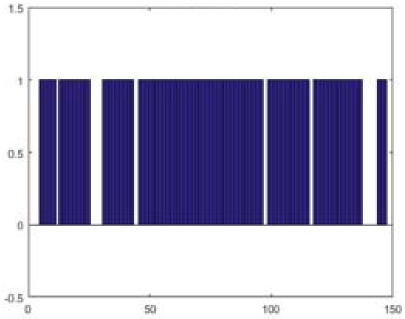

Fig. 3. Sample to represent students of Class 2: Attentive with Inattentive Periods.

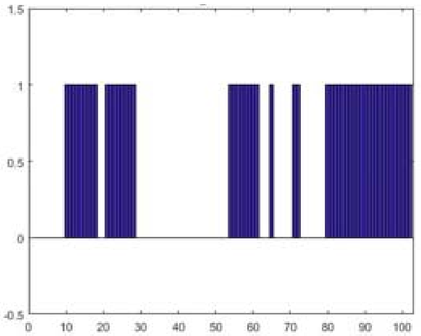

Fig. 4. Sample to represent students of Class 3: Inattentive with Attentive Periods.

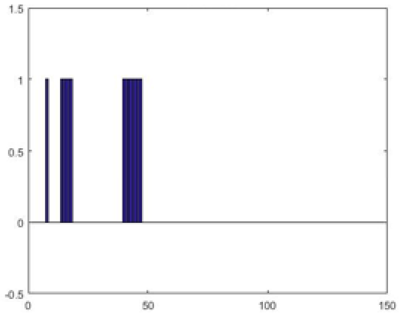

Fig. 5. Sample to represent students of Class 4: Inattentive.

All student attention patterns were classified by the expert into one of the four classes based on overall patterns. In a few cases, the student patterns did not fit within the four general classes and were marked as unclassified. The students whose attention pattern follows the sample as shown in Fig. 2 was classified as class 1, those that follow the sample as shown in Fig. 3 were classified as class 2 and so on. Based on the expert classification total students who participated in different classes for two different lectures are as shown in table 1. Data in table 1 only includes students who were present for the entire lecture; absences, tardiness, or failure to bring one's laptop would exclude the student from analysis for a particular lecture period. Additionally, since all data was collected from the same engineering course, each lecture shown in table 1 is a subset of the study participants and reflects participation in two different weekly lectures.

TABLE I. EXPERT CLASSIFICATION OF STUDENTS

| Classification group | Students per lecture | |
|---|---|---|
| | *First Lecture* | *Second Lecture* |
| Class 1 | 71 | 52 |
| Class 2 | 34 | 35 |
| Class 3 | 8 | 15 |
| Class 4 | 3 | 2 |
| Unclassified | 20 | 9 |
| Total | 136 | 113 |

## B. *Performance Measures*

After extracting the attention patterns, calculating the features, and completing expert classification manually, we applied each of the three dimension reduction techniques (Haar, PCA, and LDA) and used four automated classification algorithms (linear SVM, kernel SVM, Decision Tree, and KNN). We used a confusion matrix to analyze performance. A confusion matrix is an important tool for observing the performance of the classification algorithm. Confusion matrix of size $M \times M$ is used where $M$ is the number of classes. The rows of the confusion matrix represent the true sample and the columns of the confusion matrix represent the predicted sample. The diagonal values in the matrix represent the truly classified samples. The classification accuracy is obtained by dividing the total truly classified samples by the total samples as shown in (5).

$$Classification\ Accuracy = \frac{\sum_{m=1}^{M} C(m,m)}{\sum_{m=1}^{M}\sum_{n=1}^{M} C(m,n)} \qquad (5)$$

We also used the kappa coefficient to analyze performance. The kappa coefficient is another performance metric that measures how much the classification agrees to the truth values (expert classification). Its value ranges from zero to one where zero signifies no agreement and the one means perfect agreement. It is calculated as shown in (6).

$$Kappa\ (K) = \frac{NC-g}{N^2-g} \qquad (6)$$

where $N$ is the total number of samples, $C$ is the total sum of truly classified samples, and g is the sum of products of total true samples and total predicted samples for each class.

# III. RESULT

We performed two experiments. The first experiment consists of the data sets of 116 students from the first lecture as the training set and the data sets of 104 students from the second lecture as the testing set. For the second experiment, students' data sets were swapped, that is training set consists of data sets of 104 students from the second lecture and the testing set consists of 116 students from the first lecture. The results of both experiments are explained below.

## A. *Classification Accuracies*

For the first experiment, all the four classifiers (linear SVM, kernel SVM, KNN, Decision Tree) were trained with the datasets of the first lecture which had a total of 116 participants and the testing was done on 104 students' dataset of the second lecture. The classification algorithm is run on the datasets after the Haar coefficients have been extracted and the resulting confusion matrixes are shown in tables 2-5 for four different classification algorithms. The diagonal element in the matrix shows the truly classified samples which is used to calculate the classification accuracy. For example, table 2 has 52, 26, 15, and 1 as the diagonal elements for the confusion matrix. It means that 52 students have been correctly classified as class 1, 26 students have been correctly classified as class 2, 15 students have been correctly classified as class 3, and 1 student have been correctly classified as class 4.

TABLE II. CONFUSION MATRIX USING LINEAR SVM.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | 52 | 0 | 0 | 0 |
| **2** | 6 | 26 | 3 | 0 |
| **3** | 0 | 0 | 15 | 0 |
| **4** | 1 | 0 | 0 | 1 |

TABLE III. CONFUSION MATRIX USING KERNEL SVM.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | 52 | 0 | 0 | 0 |
| **2** | 9 | 26 | 0 | 0 |
| **3** | 0 | 1 | 14 | 0 |
| **4** | 0 | 0 | 0 | 2 |

TABLE IV. CONFUSION MATRIX USING KNN.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | 51 | 1 | 0 | 0 |
| **2** | 4 | 28 | 3 | 0 |
| **3** | 0 | 0 | 15 | 0 |
| **4** | 1 | 0 | 1 | 0 |

TABLE V. CONFUSION MATRIX USING DECISION TREE.

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| **1** | 52 | 0 | 0 | 0 |
| **2** | 7 | 28 | 0 | 0 |
| **3** | 0 | 3 | 9 | 3 |
| **4** | 0 | 0 | 0 | 2 |

The classification accuracies were calculated by using (5). We calculated the classification accuracies as 90.4%, 90.4%, 87.5%, and 91% for linear SVM, kernel SVM, KNN, and Decision tree respectively.

Similarly, the confusion matrix and the classification accuracies are obtained using other feature extraction techniques. The same technique was repeated for the second experiment to obtain the classification accuracies. The obtained classification accuracies for all the techniques for both experiments are shown in tables 6 and 7 respectively.

TABLE VI. CLASSIFICATION ACCURACY (FIRST EXPERIMENT)

| | Haar wavelets | PCA | LDA |
|---|---|---|---|
| **Linear SVM** | 90.4 | 89.4 | 92.3 |
| **Kernel SVM** | 90.4 | 90.4 | 75.0 |
| **KNN** | 87.5 | 91.3 | 87.5 |
| **Decision Tree** | 90.4 | 89.4 | 89.4 |

TABLE VII. CLASSIFICATION ACCURACY (SECOND EXPERIMENT)

| | Haar wavelets | PCA | LDA |
|---|---|---|---|
| **Linear SVM** | 81.0 | 78.4 | 81.0 |
| **Kernel SVM** | 83.6 | 85.3 | 82.8 |
| **KNN** | 88.8 | 81.9 | 89.7 |
| **Decision Tree** | 91.4 | 84.5 | 90.5 |

## *B. Accuracy vs Kappa graph*

To find out how much the classification agrees to the truth values (expert classification), we calculated the Kappa values using (6). Figs. 6-11 show the graph of classification accuracy and Kappa values for different algorithms.

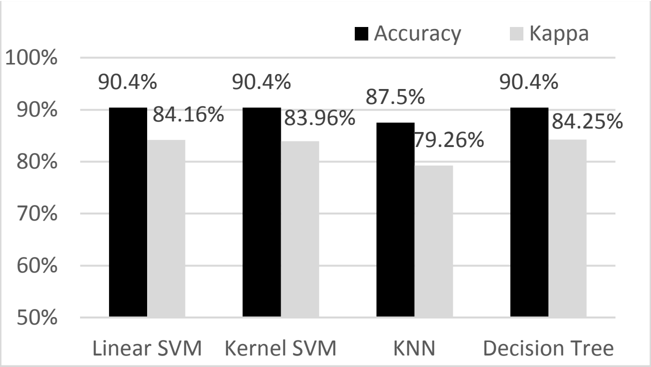

Fig. 6. Accuracy and Kappa plot for four different classification algorithms using Haar features (First experiment).

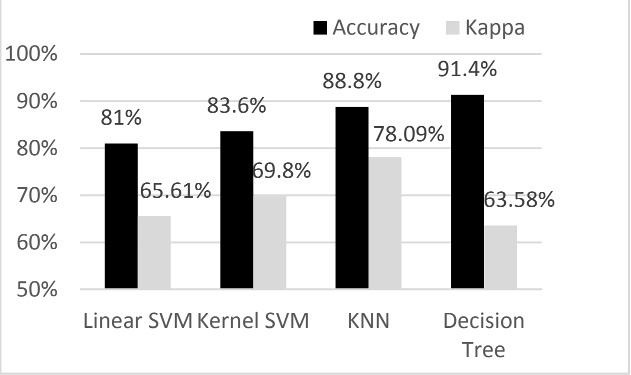

Fig. 7. Accuracy and Kappa plot for four different classification algorithms using Haar features (Second experiment)

Figs. 6-7 show the case when using the Haar features. Fig. 6, for first experiment, shows that the Decision tree has the highest classification accuracy (91%) with the highest Kappa value (84.25%). For second experiment, Fig. 7 also shows that the Decision tree has the highest classification accuracy (91.4%) and the KNN produced the highest Kappa value (78.09%).

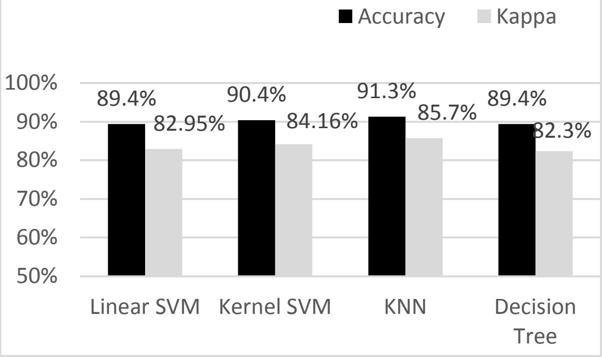

Fig. 8. Accuracy and Kappa plot for four different classification algorithms using PCA features (First experiment)

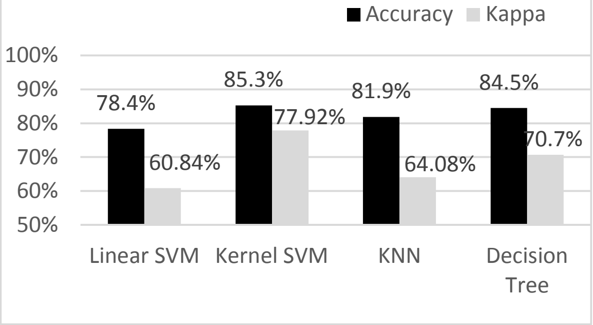

Fig. 9. Accuracy and Kappa plot for four different classification algorithms using PCA features (Second experiment).

Figs. 8-9 show the case when using the PCA features. Fig. 8, for first experiment, shows that KNN has the highest classification accuracy (91.3%) and the highest Kappa value (85.7%). For second experiment, Fig. 9 shows that kernel SVM has the highest classification accuracy (85.3%) with the highest Kappa value (77.92 %).

When the LDA features are used, Figs. 10-11 show the plot of accuracy and Kappa values for first and second experiments respectively. Fig. 10 shows that for first experiment, linear SVM has the highest accuracy (92.3%) with the highest Kappa value (87.66%). Fig. 11 shows that for second experiment, the Decision tree has the highest accuracy value (90.5%) with the highest Kappa value (82.07%).

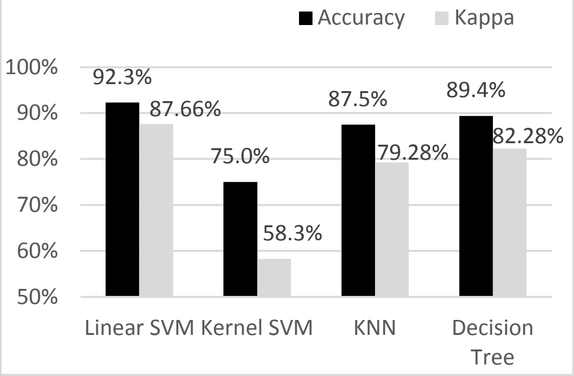

Fig. 10. Accuracy and Kappa plot for four different classification algorithms using LDA features (First experiment)

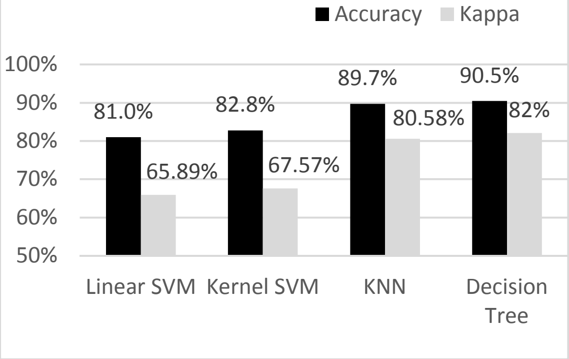

Fig. 11. Accuracy and Kappa plot for four different classification algorithms using LDA features (Second experiment)

## IV. DISCUSSION

The result section shows that all the classification algorithm that we have used produce good results with regards to determining the attention pattern of the students. Before the classification was done by our classification algorithm, feature extraction was done using different dimensional reduction algorithm. As explained in section II, we conducted two different experiments. Table 6 shows the classification accuracies for the first experiment. When we used the Haar wavelets as the dimensional reduction technique, the three algorithms linear SVM, kernel SVM, and the decision tree provided the highest accuracies. But in the case of using the PCA as dimensional reduction technique, KNN outperformed all other classification algorithm with the highest classification accuracy. When we used the LDA as the dimensional reduction technique, linear SVM performed the best. From the first experiment, we observed that linear SVM performed the best with the features extracted using LDA. kernel SVM did the best with the features extracted using Haar wavelets and PCA. Finally, for the decision tree, the features extracted using Haar wavelets favored to obtain the highest classification accuracy.

The classification accuracies shown in table 7 for the different classification algorithms highlights that the case is different for the second experiment. We can see that the classification accuracies are reduced in almost all cases in comparison with the first experiment. The reason for this may be due to the smaller number of samples in the training data set (104 samples) than the testing dataset (116 samples). Because of this the model could not fit well the training data and as a result the classification accuracy decreased for the testing set. When we used the Haar wavelets as the dimensional reduction technique, decision tree produced the highest classification accuracy. kernel SVM produced the highest classification accuracy when we used the PCA as the dimensional reduction technique. When we used the LDA as the dimensional reduction technique, decision tree outperformed all other classification technique in case of classification accuracy.

We also observed the effect of the dimensionality-reduction techniques on the individual classification algorithm. For the linear SVM, features extracted using Haar wavelets and LDA produced the highest accuracies. Our results show that in the case of kernel SVM, PCA is good choice to obtain the highest accuracy. KNN produced the highest classification accuracy with the features extracted using the LDA. Finally, for the decision tree, the features extracted using Haar wavelets favored to obtain the highest classification algorithm.

If we consider the results for both experiments, it shows that for the linear SVM the features extracted using the LDA technique produces the best result. In case of kernel SVM, PCA is the best dimensional reduction technique. For the KNN we obtained highest accuracy with the features extracted using PCA and LDA for the first and second experiment respectively. In case of decision tree, features obtained from the Haar wavelets performed the best for both the experiment.

The kappa coefficient shows how much the classification agrees the truth values. High overall classification accuracies do not guarantee that the classification model is perfect because overall accuracy considers only the diagonal values of the confusion matrix. The strength of the kappa coefficient is that it not only considers the diagonal values but also considers the non-diagonal values of the confusion matrix for its calculation. The famous interpretation of Kappa explains that if kappa is between 0-0.2 it is slight agreement, 0.21-0.4: fair agreement, 0.41-0.60: moderate agreement, 0.61-0.8: substantial agreement, 0.81-0.99: almost perfect agreement. When using the Haar and PCA features for the first experiment the value of kappa coefficients from the Fig. 6 and Fig. 8 shows that all the classification models have almost perfect agreement with the truth values and for the second experiment, the value of kappa coefficients from the Fig. 7 and Fig. 9 shows that all the classification models have substantial agreement. When the LDA features are used for the first experiment, the value kappa coefficients from Fig. 10 shows that kernel SVM has moderate agreement with the truth values, KNN has substantial agreement, linear SVM and Decision tree have the almost perfect agreement. For the second experiment using the LDA features, the values of kappa coefficients from the Fig. 11 shows that Linear and kernel SVM has substantial agreement, KNN and Decision tree has the almost perfect agreement with the truth values.

## V. Conclusion

At the beginning of the research, we had hypothesized that data mining techniques can be applied to large educational data sets to extract and analyze patterns and create insights. This hypothesis is supported by the acceptable classification accuracies as shown in the result section (Section III). Based on the result, the instructor makes decision whether the students are paying attention in the classroom or not. For learning interventions, we do not need 100% accuracy in order to have the instructor make decisions about their lecture. For instance, if we are 90% accurate, and we state that only 45% of students are paying attention, that means roughly 40-50% of students are paying attention. That could be an actionable range for instructional intervention. Tables 6 and 7 shows that the obtained classification accuracies are in the range of 75% to 91.4%. So, all the accuracies that we have obtained may be in an acceptable range for instructional intervention. Data mining techniques can help us understand how we can encourage learning in classrooms where computers are ubiquitous.

The classification errors during the process suggest further exploration to understand the reason for such errors. First, the expert completed the classification by observing the students' attention pattern as shown in Figs. 1-4. During the attention pattern plot, there were some attention patterns that did not fit on any of the class and such students were removed from the observations. In the future, we could increase the total number of classes so that all the students can fit into one of the several classes. Second, for classification purposes, we only used the traditional machine learning algorithms. In the future, we could use advanced techniques like neural networks or deep learning that may produce higher classification accuracies than traditional methods.

The students' attention records are classified into different classes with satisfactory classification accuracies. This has motivated us to continue our pursuit of developing classifiers for educational contexts. As we have explained in the introduction section, our goal is to ultimately develop tools that would provide real-time feedback to educators to help them monitor and guide student learning when using computers in large lectures. We could also extend this so that our tools could provide feedback to the students explaining their academic standing. Attention in the classroom has a direct impact on learning. Using the classification pattern of each student, their learning pattern could be monitored throughout the semester. This could help to understand the learning behavior of individual students. Also, the attention pattern of students from one course could be compared with other courses and help the instructor to understand which courses the students enjoy must, seeing their attention pattern. It could also help instructors adapt instructional elements from those courses which have more attentive students as seen by the attention pattern of the students. We are excited to continue exploring the prospects that real-time mining of educational data can provide instructors who are seeking to maximize learning in the large lecture environments.

## References

[1] Gagné Robert M., *The conditions of learning*. New York, New York: Holt, Rinehart and Winston, 1977.

[2] M. L. Fleming, "Displays and communication", in *Instructional Technology: Foundations*, R. M. Gagne, Ed., Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1987, pp. 234-260

[3] I. Guler and E. D. Ubeyli, "Multiclass Support Vector Machines for EEG-Signals Classification," in IEEE Transactions on Information Technology in Biomedicine, vol. 11, no. 2, pp. 117-126, March 2007.

[4] D. He and H. Leung, "CFAR Intrusion Detection method Based on Support Vector Machine Prediction," IEEE Intl. Conf. on Comp. Intell. for Meas. Systems and Intell., 14 Jul 2004, pp. 10-15.

[5] X. Long and Y. Wu, "Application of Decision Tree in Student Achievement Evaluation," 2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, 2012, pp. 243-247.

[6] Tanner, Tuomas & Toivonen, Hannu. (2010). Predicting and preventing student failure – using the k-nearest neighbour method to predict student performance in an online course environment. IJLT. 5. 356-377. 10.1504/IJLT.2010.038772.

[7] A.N. Akansu, W.A. Serdijn, and I.W. Selesnick. "Emerging applications of wavelets: A review." *Physical communication* 3.1 (2010): 1-18.

[8] T. Li, Tao, et al. "A survey on wavelet applications in data mining." ACM SIGKDD Explorations Newsletter 4.2 (2002): 49-68.

[9] A. Alkandari and S. J. Aljaber, "Principle Component Analysis algorithm (PCA) for image recognition," 2015 Second International Conference on Computing Technology and Information Management (ICCTIM), Johor, 2015, pp. 76-80.

[10] S. Mika, G. Ratsch, J. Weston, B. Scholkopf and K. R. Mullers, "Fisher discriminant analysis with kernels," Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing

Society Workshop (Cat. No.98TH8468), Madison, WI, USA, 1999, pp. 41-48.

[11] M.J. Mohammadi-Aragh, “Characterizing student attention in technology-infused classrooms using real-time active window data”, 2013.