Applying classification techniques on temporal trace data for shaping student behavior models

Zacharoula Papamitsiou
IPPS in Information Systems
University of Macedonia
156 Egnatia Avenue
Thessaloniki, 54621, Greece
+30 2310 891-768
papamits@uom.edu.gr

Eirini Karapistoli
IPPS in Information Systems
University of Macedonia
156 Egnatia Avenue
Thessaloniki, 54621, Greece
+30 2310 891-768
ikarapis@uom.gr

Anastasios A. Economides
IPPS in Information Systems
University of Macedonia
156 Egnatia Avenue
Thessaloniki, 54621, Greece
+30 2310 891-768
economid@uom.gr

ABSTRACT
Differences in learners' behavior have a deep impact on their educational performance. Consequently, there is a need to detect and identify these differences and build suitable learner models accordingly. In this paper, we report on the results from an alternative approach for dynamic student behavioral modeling based on the analysis of time-based student-generated trace data. The goal was to unobtrusively classify students according to their time-spent behavior. We applied 5 different supervised learning classification algorithms on these data, using as target values (class labels) the students' performance score classes during a Computer-Based Assessment (CBA) process, and compared the obtained results. The proposed approach has been explored in a study with 259 undergraduate university participant students. The analysis of the findings revealed that a) the low misclassification rates are indicative of the accuracy of the applied method and b) the ensemble learning (treeBagger) method provides better classification results compared to the others. These preliminary results are encouraging, indicating that a time-spent driven description of the students' behavior could have an added value towards dynamically reshaping the respective models.

Categories and Subject Descriptors
K.3 [Computers and Education]: General

Keywords
Learner behavioral modeling, assessment analytics, computer-based testing, supervised learning classification.

1. INTRODUCTION
The landscape on Learning Analytics (LA) research over the last five years has rapidly changed. Lately, the educational research community has shifted towards exploring different, multiple, more complex and more information rich data sources (e.g. tangible and wearable computing, immersive learning environments, shared workspaces, massive open online courses, etc.), in order to identify new suitable measures of learning and success (e.g. affect, attention, attitudes, community structure, degrees of competence, expectations, satisfaction, social dynamics, etc.) and develop applications that are expected to enable personalized learning on a large scale (e.g., [1], [2], [3], [4], [5], [6]).

Detecting undesirable learner behaviors, and profiling learners are among the core objectives of LA research. Moreover, differences in learners' behavior have a deep impact on their performance. As apparent, there is a need to detect and identify these differences and build suitable learner models accordingly. These models will further assist in improving the personalization of educational services at a larger scale.

In this paper, we present a method for dynamic student behavioral modeling based on the analysis of temporal, student-generated trace data. The goal was to unobtrusively classify students according to their time-spent behavior during assessment processes. In particular, we explored a large range of supervised learning classification algorithms (SLA) (namely Artificial Neural Networks-ANNs, Support Vector Machines-SVMs, Naïve Bayes-NB, k-Nearest Neighbors-*k*NN and treeBagger) on a dataset consisting of time-based data (including total time to answer correctly, total time to answer wrongly, total idle time, effort and goal expectancy) using as target values (class labels) the students' performance during a Computer-Based Assessment (CBA) process. Next, we present the results from a study with 259 undergraduate participant students from a Greek University. For the study, we employed the LAERS assessment environment [7]. We discovered that a) the low misclassification rates, as well as the high sensitivity and performance measures are indicative of the accuracy of employing temporal trace data for student behavioral modeling and b) the ensemble learning (treeBagger) method provides better and more solid classification results compared to the other SLA algorithms.

The rest of the paper is organized as follows: in section 2, we briefly review existing work regarding student modeling methods, approaches and results. In section 3, we explain the motivation and rationale of our research. In section 4, we present the experiment methodology, the data collection procedure and the analysis methods that we applied, while in section 5, we analyze the results from the case study. Finally, in section 6, we discuss the major findings and conclude with future implications.

2. RELATED WORK
Student modeling can be defined as the process of information extraction from different sources into a profile representation of student's knowledge level, cognitive and affective states, and

DOI: http://dx.doi.org/10.1145/2883851.2883926

meta-cognitive skills on a specific domain or topic [8, 9]. The goal is to describe and/or predict particular behavioral patterns. Identification and modeling of students and students' learning behavior is a primary educational research objective. A student model is used to represent multiple student's characteristics – either static (e.g., age, gender, etc.), or dynamic. The most common among the dynamic characteristics include student's personality traits, performance, goals, achievements, prior and acquired domain knowledge, [10], learning strategies, preferences and styles [11], making decisions and analyzing abilities, critical thinking and communication skills, collaborative skills [12], errors and misconceptions, motivation, emotions and feelings, self-regulation, self-explanation and self-assessment [13], [31].

More recently, the time dimension has been explored for modeling user behavior [e.g., 14, 15, 16]. Barua et al. [14] included temporal data in order to construct a model for long-term goal setting representation, while Belk et al. [15] also used response-times in combination with the given answers in order to discriminate "Verbal" users from "Imagers". The work in [16] is yet another example of time-spending exploitation during an experiment with eye-tracking technology for classification of users in a user-independent fashion.

In the educational research domain, Shih, Koedinger and Scheines [17] used worked examples and logged response times to model the students' time-spent in terms of "thinking about a hint" and "reflecting on a hint". The goal was to capture behaviors that are related to reasoning and self-explanation during requesting hints. It was found that specifying the moments that teacher should intervene requires to better distinguish between students who use worked examples, how they use them and their response times.

Additional studies have shown that the temporal interpretation of students' engagement in task solving during testing, could be used for predicting their progress [7, 18]. In particular, it was found that Total Time to Answer Correctly (TTAC), Total Time to Answer Wrongly (TTAW) and goal expectancy (GE) are strong determinants of Actual Performance (AP), and Total Idle Time (TIT) is an indicator of students' effort (EFF) [30] during testing.

3. MOTIVATION AND RARIONALE OF THE RESEARCH

As stated in the introduction, differences in learners' behavior have a deep impact on their performance. As apparent, there is a need to detect and identify these differences and build suitable learner models accordingly. These models will further assist in improving personalization of educational services. Consequently, it is important for systems developers to identify the parameters that could be used for fully adapting the assessment items to the examinees' level of ability/expertise or for providing personalized feedback during CBA (e.g., the recommendation of the most appropriate next testing item).

Further, since the temporal interpretation of the students' behavior has been found to explain satisfactorily their actions [14, 16] and to provide statistically significant explanation of students' performance [7, 18], additional research should be conducted regarding whether temporal, user-generated trace data could be used for student behavioral modeling purposes.

In this paper, we suggest a method for dynamic student behavioral modeling. In particular, we propose the use of temporal, student-generated trace data that are unobtrusively and seamlessly tracked during a CBA procedure. Such mechanisms for tracking temporal data are cost-effective, consume low computational resources, and can be easily implemented in any CBA system. Moreover, our methodology applies multiple supervised learning algorithms (SLAs), including ANNs, SVMs, *k*NN, NB and treeBagger for classifying and analyzing this type of data.

4. METHODOLOGY

4.1 Research participants and data collection

In this study, data were collected from a total of 259 undergraduate students (108 males [41.7%] and 151 females [58.3%], aged 20-27 years old (M=22.6, SD=1.933, N=259) from the Department of Economics at University of Macedonia, Thessaloniki, Greece. 12 groups of 20 to 25 students attended the midterm exams of the Computers II course (related to databases, information systems and introduction to e-commerce). For the purposes of the examination, we used 34 multiple choice questions. Each question had two to four possible answers, but only one was the correct. The participants could skip or re-view the questions and/or alter the submitted answer. Finally, the participation to the midterm exams procedure was optional. As an external motivation to increase the students' effort, we set that their score would participate up to 30% to their final grade.

4.2 The LAERS Assessment Environment

In our study, we used the LAERS assessment environment [7], which is a CBA system that we are developing in order to automate the provision of personalized recommendations of most appropriate next task as adaptive feedback service.

At the first phase of the implementation, we configured a testing mechanism and a tracker that logs the students' temporal data. The testing unit displays the multiple choice quiz tasks delivered to students. Each task is displayed separately and one-by-one. The students can temporarily save their answers on the tasks, before submitting the quiz, and they can change their initial choice by selecting the task to re-view from the list underneath. They submit the quiz answers only once, whenever they estimate that they are ready to do so, within the duration of the test.

Table 1. Features from the raw log files

Feature	
1. student ID	*2. the task the student works on*
3. the answer the student submits	*4. the correctness of the submitted answer*
5. the timestamp the student starts viewing a task	*6. the timestamp the student chooses to leave the task (saves an answer)*
7. the total time the student spends on viewing the tasks and submitting the correct answers	*8. the total time the student spends on viewing the tasks and submitting the wrong answers*
9. the idle time the student spends viewing each task	*10. how many times the student views each task*
11. how many times the students change the answer they submit for each task	*12. the student's total time on task*
13. the student's GE	*14. the effort required (EFF)*
15. the student's AP	

The second component of the system records the students' activity data during testing. We also embedded into the system a pre-test questionnaire (consisting of 3 questions) in order to measure each student's goal expectancy (GE) (a measure of student self-confidence and goal orientation regarding the use of a CBA, proposed in Computer Based Assessment Acceptance Model (CBAAM) [19]). GE has two dimensions: the students'

preparation to take the CBA and the desirable level of success for each student. GE actually measures if the learners are fulfilled with their preparation. The students, before taking the CBA, set a goal regarding a percentage of correct answers that provides them a satisfying performance. The three items from the questionnaire that measure GE were: a) GE1: Courses' preparation was sufficient for the CBA, b) GE2: My personal preparation for the CBA, and c) GE3: My performance expectations for the CBA. GE was found to be a direct strong determinant of the temporal variables and concurrently an indirect strong determinant of AP [7]. The overall features/attributes of students' activity either tracked, computed and/or measured are listed in Table 1.

4.3 Feature Subset Selection

The initial raw log file contained a sample of the 15 attributes (features) to be used in this study. Our pre-experimental thoughts were that some of the attributes were ''noisy''; i.e. contain signals not related to the target of classification. Therefore, we first attempted to remove spurious attributes using *feature subset selection*. Feature selection reduces the dimensionality of data by selecting only a subset of features (i.e., predictor variables) to create a model. Selection criteria usually involve the minimization of a specific measure of predictive error for models fit to different subsets. Algorithms search for a subset of predictors that optimally model measured responses, subject to constraints such as required or excluded features and the size of the subset. Note that the number of attributes to select is crucial in the analysis of the data. In this experiment, we ranked the 15 attributes from most to least informative. The attributes were ranked using the *sequential feature selection* method of MATLAB. This method has two components: a) an objective function, called *the criterion*, which the method seeks to minimize over all feasible feature subsets, and b) a *sequential search algorithm*, which adds or removes features from a candidate subset while evaluating the criterion.

4.4 Analysis Methods

The machine learning techniques are divided into those performed without supervision (unsupervised learning), and those that take place under supervision (supervised learning). The algorithms that belong to the first category build a model without knowing the desired outputs for the training set. Typical example of unsupervised learning is association rules mining between the values of characteristics within the learning vectors. However, most of the research activity in the field of machine learning concerns learning with supervision (supervised learning), typical example of which is the categorization or classification problems.

Suppose there is a data set containing observations with measurements on different variables (called predictors) and their known class labels. If predictor values are obtained for new observations, could the classes those observations belong to be determined? This is the problem of *supervised classification*: the task of assigning objects to one of several predefined classes. In other words, it is the task of learning a target function f to map each input attribute set x to one of the predefined class labels y.

Each classification technique employs a *learning algorithm* to identify a model that best fits the relationship between the attribute set and the class label of the input data. These techniques operate in two phases: the *training phase* and the *testing phase*. [20, 21]. In the present study, we explored 5 different advanced supervised learning techniques for classifying students based on their time-based characteristics (predictors) and according to their actual performance (class label). In particular, we tried Artificial Neural Networks (ANNs), Support Vector Machines (SVM), Naïve Bayes (NB), k-Nearest Neighbors (*k*NN) and the treeBagger method. These are some of the well-known classifiers used in the machine learning field, and the most common approaches explored with a plurality of different attributes in the learning analytics and educational data mining research domain.

Artificial Neural Networks (ANNs) are computational systems based on the structure, processing method and learning ability of the brain [29]. When performing classification analysis with an existing dataset, a commonly adapted approach, named *holdout validation*, is used to split the data into a larger set for *training* the ANN and a smaller set for *testing* the model [23]. In this work, a Feed Forward neural network has been created and trained.

Support Vector Machines (SVM) is a supervised learning method for linear modeling. For classification purposes, *nonlinear kernel functions* are often used to transform the data into a feature space of a higher dimension than that of the input before attempting to separate them using a linear discriminator [24, 25]. In this work, a third degree polynomial kernel function was employed.

Naïve Bayes (NB) are a family of simple *probabilistic classifiers* based on applying Bayes' theorem with strong independence assumptions between the predictors within each class. During the training step, the method estimates the parameters of a probability distribution. Next, during the prediction step, the method computes the posterior probability of that sample belonging to each class, and classifies the test data accordingly [20].

k-Nearest Neighbors (kNN) is a *non-parametric method* used for classification. Given an unknown sample, a kNN classifier searches the pattern space for the k training samples that are closest to the unknown sample. kNN is based on the principle that the samples within a dataset will generally exist in *close proximity* to other samples that have similar properties [26, 27].

Bagging Bagging (Bootstrap Aggregating) is an *ensemble method* that creates separate samples of the training dataset and creates a classifier for each sample. In fact, bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The aggregation does a plurality vote when predicting a class. The multiple versions are formed by making bootstrap replicates of the learning set and using these replicates as new learning sets [20, 28].

4.5 Measures and Performance Criteria

Evaluation of the performance of a classification model is based on the counts of test records correctly and incorrectly predicted by the model, tabulated in a confusion matrix. Generally speaking, the *(i,j)* element in the confusion matrix is the number of samples whose known class label is class i and whose predicted class is j. The diagonal elements represent correctly classified observations.

However, the confusion matrix is not convenient to compare the performance of different models. *Accuracy* is a single-value performance metric defined as the proportion of correct predictions to the total predictions. Further, the performance of a model can be expressed in terms of its *error rate*, which is given as the proportion of wrong prediction to the total predictions [20, 21]. The errors committed by a classification model are generally divided into two types: *resubstitution errors* (training errors) and *test errors* (generalization errors). The resubstitution error is the proportion of misclassified observations on the training set, whereas the test error is the expected prediction error on an independent set. A good model must have low resubstitution error as well as low test error [20, 22].

A method commonly used to evaluate the performance of a classifier is *cross validation*. The *k*-fold cross validation method segments the data into k equal-sized partitions. This procedure is repeated n times so that each partition is used the same number of

times for training and exactly once for testing. We used a stratified k=10-fold cross validation with n=100 iterations for estimating the misclassification (test) error [20, 21, 22]. Moreover, *sensitivity analysis* is a method for identifying the "cause-and-effect" relationship between the inputs and outputs of a prediction model. This method is often followed in machine learning techniques to rank the variables in terms of their importance according to the sensitivity measure [22]. Finally, *F-score* (or F-measure) is a measure of a test's accuracy. It considers the precision and the recall of the test to compute the score. In simple terms, high precision means that an algorithm returned substantially more relevant results than irrelevant, while high recall means that an algorithm returned most of the relevant results [21, 22]. The F-score can be interpreted as a weighted average of the precision and recall, where an F-score reaches its best value at 1 and worst score at 0 [20].

5. RESULTS

Table 2 outlines the SLA methods we applied on the input data, the number of classes being predicted (i.e., the different categories of students' performance results), the overall accuracy of the prediction (for training and testing respectively) together with the respective sample sizes (90% for training and 10% for testing for all SLA methods), and the tool used during the analysis.

Table 2. A summary of our experiment

SLA used	*# of classes predicted*	*Sample size*	*Accuracy of prediction*	*Simulation tool used*
ANNs, SVMs, NB, kNN, treeBagger	7-class	259 samples in total 233 for training 26 for testing	100% for training 76% for testing	MATLAB

5.1 Exploratory data analysis

Table 3 illustrates the variables (features) used to train and test the machine learning networks, during the experiment, as well as the range of their values:

Table 3. Features used for training and testing

	Variable	*Description*	*Type*	*Value Range*
Temporal	TTAC	Total time to answer correctly	Simple	≥0 (msec)
	TTAW	Total time to answer wrongly	Simple	≥0 (msec)
	TIT	Total idle time	Simple	≥0 (msec)
	EFF	Effort	Composite	0-1
Time-varying	AP*	Actual Performance	Simple	0-1.5
Self-reported	GE	Goal expectancy	Latent	0-5

**AP: target (output)-dependent variable*

Further to that, Table 4 illustrate the covariance matrix for all five input variables. As it can be seen, there are no strong correlations between the variables.

Table 4. Covariance matrix for all predictor variables

	TIT	*TTAC*	*TTAW*	*EFF*	*GE*
TIT	1.000				
TTAC	-0.082	1.000			
TTAW	-0.313	0.357	1.000		
EFF	-0.353	0.056	0.259	1.000	
GE	0.128	0.098	0.055	0.564	1.000

5.2 Classification results

In this study, we initially explored the previously described methods with an input dataset consisting of three variables (predictors): TTAC, TTAW and GE. We chose to examine these variables based on the formerly reported results from [7]. Table 5 presents the performance results (resubstitution error, true test error, sensitivity, and F-score) for all the methods used to develop a classification model in this study with these three features and with testing sample size 10% of the initial dataset.

Table 5. Performance metrics for cross-validation 10% with three features

Test Set Size	*|cvpartition| = 10% (k-fold=10)*				
Classifier	**ANN**	**SVM**	***k*NN**	**NB**	**ENS****
Resub Error	0.34	NaN	0.30	0.38	0.00
True Test Error*	0.24	0.27	0.28	0.24	0.24
Sensitivity	0.96	0.95	0.95	0.96	0.96
F-score	0.87	0.85	0.86	0.85	0.88

**True test error=cross-validation error, **ENS:ensembles of decision trees*

These results demonstrate that all methods achieve high classification performance, since the true test error varies from 0.24 (ENS method) to 0.28 (*k*NN method). Further to that, the sensitivity measure is close to 1 in most cases (0.95-0.96) and the F-score is also high (0.85-0.88). Moreover, from this table it becomes apparent that the ENS method provides better classification results compared to the other methods, while the kNN and NB methods also achieve satisfactory results.

Based on this finding, we examined how the highly performing methods (ENS, *k*NN and NB) change their output when applied to more input variables (predictors). We explored this question with two additional features: TIT and EFF. Table 6 illustrates the performance metrics for the ENS, *k*NN and NB methods with 4 (initially we added TIT) and 5 (finally we added EFF) features and testing sample set to 10% of the initial dataset.

Table 6. Performance metrics for test set size 10% with 4 and 5 features

Forward Feature Selection	*'TTAW', 'TTAC', 'GE', EFF'*			*'TIT', 'TTAW', 'TTAC', 'EFF', 'GE'*		
Classifier	***k*NN**	**NB**	**ENS**	***k*NN**	**NB**	**ENS**
Resub Error	0.30	0.37	0.00	0.30	0.36	0.00
True Test Error	0.28	0.28	0.28	0.24	0.32	0.24
Sensitivity	0.95	0.95	0.95	0.96	0.94	0.96
F-score	0.85	0.86	0.85	0.88	0.82	0.84

As seen from this table, ENS does not seem to be affected by the additional features, providing results similar to the previous ones (conducted using with three features only). On the contrary, the performance of the other two methods is slightly reduced when the number of predictors increases.

6. DISCUSSION AND CONCLUSIONS

In this paper, we explored student-generated temporal trace data for modeling students' behavior during a CBA procedure according to the students' performance score. Our goal was to unobtrusively and seamlessly identify the students' time-spent behavioral patterns in order to dynamically shape the respective models. The motivation for our experimentation was based on previous research studies that analyzed temporal parameters for user modeling and reported significant results. During our experimentation, we applied 5 advanced SLA techniques.

Our findings verify formerly reported results [15], [17] regarding the capability of temporal data to represent, describe and model the students' behavior. In particular, our findings indicate that the total time to answer correctly and the total time to answer wrongly in combination with the goal expectancy could satisfactorily be used for classification of students during computer-based testing. The low misclassification rates are indicative of the accuracy of the proposed method. Further to that, from tables 5 and 6 it becomes apparent that the ensemble learning (treeBagger) method provided the most accurate classification results compared to the other methods. However, an interesting finding that requires more investigation is that most algorithms perform worse when two additional features are included in the analysis. We still have to explore why this is happening and weather these additional features are appropriate for classification purposes.

Based on the findings, we suggest that one can identify a set of functional temporal (or behavioral) factors/parameters that could constitute the core components of a system's architecture. For example, TTAC, TTAW, TIT, EFF and GE are only indicative variables that could be embedded into a testing system in order to

model the test-takers and to guide adaptation and personalization of services. Systems like that would aim at personalizing the deliverable service according to their user's model. For example, such a service could be the recommendation of the next most appropriate task according to the student's model and detected level of expertise (based on the corresponding timely predicted performance). In this case, the system should be "trained" in order to "recognize" and model its current users based on their temporal and behavioral data. Then, it should "choose" the appropriate task (among the collection of tasks from an item bank) that best corresponds to the needs and meets the abilities of the user, in order to improve the expected outcome. Finally, the system should inform the users about their progress and either suggest the selected task (as a CAT system) or allow the users to make their own choice of the next task (as a CBT system).

The approach suggested in this paper was applied on a dataset collected during an assessment procedure in the context of mid-term exams. However, the nature of the data collected (time-based parameters) and the general-purpose methodology followed for the analysis of these data (SLA), render this approach replicable and/or transferable to other contexts, and eliminate the restriction of using it only during testing. The temporal factors are not contextualized to the LAERS assessment environment, but a similar tracker could be embedded in any adaptive learning system. For example, time-related parameters (time-spent) could be tracked to measure the duration of solving/implementing sub-activities or sub-tasks in the context of project-based learning, or to measure the duration of studying and exercising with learning modules during inquiry-based learning, etc., along with the number of repeating the intermediate, facilitating steps (e.g. watching educational videos, opening and using educational resources, participating in discussions, etc.).

As a next step, we are planning to deeper explore the patterns of these classes in terms of time-spent, i.e., which are the specific characteristics of the time-spent behavior of the examinee that belong to each one of the classes. Moreover, we plan to investigate other patterns within the students' time-spent behavior aiming at identifying unwanted behaviors that affect the assessment results, by employing other suitable mining techniques, like process mining. Finally, we envisage creating the learner model simultaneously, while the student takes the test, in a stream mining fashion, which would enrich the profile modeling with a notion of dynamics, allowing for adaptive question sequencing.

7. REFERENCES

[1] Aramo-Immonen, H., Jussila, J., Huhtamäki, J. 2015. Exploring co-learning behavior of conference participants with visual network analysis of Twitter data, *Comput Human Behav*, 51, 1154-1162.

[2] Agudo-Peregrina, Á. F., Iglesias-Pradas, S., Conde-González, M. Á., Hernández-García, Á. 2015. Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning, *Comput Human Behav*, 31, 542-550.

[3] Kovanović, V., Gašević, D., Joksimović, S., Hatala, M., Adesope, O. 2015. Analytics of communities of inquiry: Effects of learning technology use on cognitive presence in asynchronous online discussions, *Internet High Educ*, 27, 74-89.

[4] Tempelaar, D. T., Rienties, B., & Giesbers, B. 2014. In search for the most informative data for feedback generation: Learning analytics in a data-rich context, *Comput Human Behav*, 47,157-167.

[5] van Leeuwen, A., Janssen, J., Erkens, G., Brekelmans, M. 2015. Teacher regulation of cognitive activities during student collaboration:Effects of learning analytics, *Comput Educ*, 90, 80-94.

[6] Veletsianos, G., Collier, A., & Schneider, E. 2015. Digging Deeper into Learners' Experiences in MOOCs: Participation in social networks outside of MOOCs, Notetaking, and contexts surrounding content consumption. *Brit. J. Educ. Technol*. 46(3), 570-587.

[7] Papamitsiou, Z., Terzis, V. Economides, A. A. 2014. Temporal Learning Analytics during computer based testing, In *Proceedings of the 4th International Conference on Learning Analytics and Knowledge (LAK'14)*, Indianapolis, USA, 31-35.

[8] McCalla, G. 1992. The central importance of student modeling to intelligent tutoring. In E. Costa [Ed.], *New Directions for Intelligent Tutoring Systems*. Berlin: Springer Verlag.

[9] Thomson, D., Mitrovic, A. 2009. Towards a negotiable student model for constraint-based ITSs. *17th International Conference on Computers in Education*, Hong Kong, 83–90.

[10] Self, J. A. 1990. Bypassing the intractable problem of student modeling. In C. Frasson & G. Gauthier (Eds.), *Intelligent-tutoring systems: At the crossroads of AI and education*, 107–123, Norwood, NJ: Ablex

[11] Peña-Ayala, A. 2014. Educational data mining: A survey and a data mining-based analysis of recent works. *Expert Syst Appl*, 41(4), 1432–1462

[12] Mitrovic, A., Martin, B. 2006. Evaluating the effects of open student models on learning. *2nd international conference on adaptive hypermedia and adaptive web-based systems*, 296–305

[13] Peña, A., Kayashima, M. 2011. Improving students' meta-cognitive skills within intelligent educational systems: A Review. *6th International Conference on Foundations of Augmented Cognition*, Orlando, Florida, USA, 442–451

[14] Barua, D., Kay, J., Kummerfeld, B., Paris, C. 2014. Modeling long term goals, *22nd International Conference on User Modeling, Adaptation and Personalization*, Aalborg, 1-12

[15] Belk, M., Germanakos, P., Fidas, C., Samaras, G. 2014. A personalization method based on human factors for improving usability of user authentication tasks, *22nd Int. Conf. on User Modeling, Adaptation and Personalization*, Aalborg, 13-24,

[16] Bixler, R. D'Mello, S. 2014. Toward fully automated person-independent detection of mind wandering, *22nd Int. Conf. on User Modeling, Adaptation and Personalization*, Aalborg, 37-48

[17] Shih, B., Koedinger, K.R., Scheines, R. 2008. A response time model for bottom-out hints as worked examples. In R. de Baker, T. Barnes, J. Beck (Eds), Proc. *1st International Conference on Educational Data Mining*, Montreal, 117–126

[18] Papamitsiou, Z., Economides, A. A. 2014. Students' perception of performance vs. actual performance during computer-based testing: a temporal approach, *8th Int. Technology, Education and Development Conference*, Valencia, 401-411

[19] Terzis, V., Economides, A. A. 2011. The acceptance and use of computer based assessment, *Comput Educ*, 56(4), 1032–1044

[20] Tan, P-N., Steinbach, M., Kumar, V. 2005. *Introduction to Data Mining*, (1st Edition). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA

[21] Alpaydin, E. 2010. *Introduction to Machine Learning*. MIT Press

[22] Mitchell, T. 1997. *Machine Learning*, Mcgraw-Hill, New York

[23] Arlot S., Celisse A. 2010. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79

[24] Cortes, C., Vapnik, V. 1995. Support-vector networks. *Machine Learning*, 20 (3), 273

[25] Cristianini, N., Shawe-Taylor, J. 2000. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, London, UK.

[26] Altman, N. S. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.*, 46 (3), 175-185

[27] Cover, T. Hart, P. 1967. Nearest neighbor pattern classification. *IEEE Trans on Information Theory*, 13 (1), 21–27

[28] Breiman, L.1996.Bagging predictors. *Machine Learning*,24,123-140

[29] Haykin, S. 1998. Neural networks: A comprehensive foundation (2nd ed.). Prentice Hall.

[30] Papamitsiou, Z., Economides, A.A. 2015. A temporal estimation of students' on-task mental effort and its effect on students' performance during computer based testing, *IEEE 18th Int. Conf. on Interactive Collaborative Learning (ICL2015)*

[31] Economides, A. A. 2009. Adaptive context-aware pervasive and ubiquitous learning. *International Journal of Technology Enhanced Learning*, 1(3), 169-192.