Artif Intell Rev (2019) 52:381–407
https://doi.org/10.1007/s10462-018-9620-8

Using machine learning to predict student difficulties from learning session data

Mushtaq Hussain[1] · **Wenhao Zhu**[1] · **Wu Zhang**[1] · **Syed Muhammad Raza Abidi**[1] · **Sadaqat Ali**[1]

Published online: 10 February 2018

Abstract The student's performance prediction is an important research topic because it can help teachers prevent students from dropping out before final exams and identify students that need additional assistance. The objective of this study is to predict the difficulties that students will encounter in a subsequent digital design course session. We analyzed the data logged by a technology-enhanced learning (TEL) system called digital electronics education and design suite (DEEDS) using machine learning algorithms. The machine learning algorithms included an artificial neural networks (ANNs), support vector machines (SVMs), logistic regression, Naïve bayes classifiers and decision trees. The DEEDS system allows students to solve digital design exercises with different levels of difficulty while logging input data. The input variables of the current study were *average time, total number of activities, average idle time, average number of keystrokes* and *total related activity* for each exercise during individual sessions in the digital design course; the output variables were the student(s) grades for each session. We then trained machine learning algorithms on the data from the previous session and tested the algorithms on the data from the upcoming session. We performed k-fold cross-validation and computed the receiver operating characteristic and root mean square error metrics to evaluate the models' performances. The results show that ANNs and SVMs achieve higher accuracy than do other algorithms. ANNs and SVMs can easily be integrated into the TEL system; thus, we would expect instructors to report improved student's performance during the subsequent session.

Keywords Machine learning · Educational data mining (EDM) · Decision support tools · E-learning · Neural networks (NN) · Support vector machine (SVM)

✉ Wenhao Zhu
whzhu@shu.edu.cn

1 School of Computer Engineering and Science, Shanghai University, Shanghai, China

1 Introduction

The educational advantages of e-learning include online teaching and course delivery, which do not require physical classrooms for students. Compared to traditional modes of learning, e-learning is less expensive, and a larger number of students can register for online courses. However, in e-learning, there is no direct communication between students and teachers. Therefore, e-learning poses some challenges. First, it is difficult for instructors to assess the effectiveness of a course. Second, the dropout rate of students in e-learning courses is much higher than that in traditional modes of learning. Third, assessing student's performance is difficult. Fourth predicting at-risk students in new courses is also difficult. Finally, teachers are interested in predicting students' expected results on upcoming assessments (Lykourentzou et al. 2009; Pahl and Donnellan 2002; Smith-Gratto 1999; Kuzilek et al. 2015; Bakki et al. 2015).

Web-based learning environments such as massive open online courses (MOOC), digital electronics education and design suite (DEEDS) and learning management systems (LMSs) allow teachers to study student performances using logged student data, but teachers may have difficulty analyzing the student logs. MOOCs and LMSs are popular types of web-based learning platforms; they provide free higher education to the entire world and offer courses from different universities. Furthermore, they provide administration, documentation, content assembly, student management and self-services (He et al. 2015). LMSs are online portals for both students and teachers that facilitate teacher-student interactions and allow them to perform their educational tasks and activities. More-over LMSs deliver courses to students, and the students can select their own courses through a course selection process (Imran et al. 2014). MOOCs are free web-based learning platforms that supply all their courses online. Students can register and attend these courses from any location (Kloft et al. 2014). These web-based learning environments affect how teachers and students think during class, and they can be used to predict a student's performance during the next class or a student's behavior at different times. In addition, these environments can be used to improve course-related content (Chen et al. 2000).

Predicting a student's progress in a class or session through, for example, quizzes, assignments, exams, and session activities can provide instructors with in-depth information on the progress of students throughout the course. To achieve this goal, researchers have applied various machine learning and statistical techniques to data acquired from both traditional and online universities.

In traditional universities, researchers mostly use a student's educational history (e.g., quizzes, midterm exams, degrees, and attended schools) and demographic information (country, sex, race and zip code) to predict student's performance.

Acharya and Sinha (2014) forecast students' performances using machine learning techniques (e.g., C4.5, sequential minimal optimization (SMO), Naïve bayes, 1-NN (1-Nearest Neighborhood), and MLP (multi-layer perceptron) with input features (e.g., gender, income, board marks and attendance). They applied correlation-based feature selection (CBFS) techniques to improve the model performances and determined that SMO achieves a higher effective average testing accuracy (66%) than do other methods.

De Albuquerque et al. (2015) employed artificial neural networks (ANNs) to predict student's performance. These models achieved high accuracy (85%) using input features such as grades, periods of study and school scores.

Marbouti et al. (2016) used logistic regression, support vector machines (SVMs), decision trees (DTs), ANNs and a Naïve bayes classifier (NBC) to identify at-risk students in advance of the next course. This study used input features, such as grades, attendance, quizzes, weekly homework, team participation, project milestones, mathematical modeling activity tasks, and exams from an offline course. Analysis of the results found that the NBC algorithm provided satisfactory accuracy (85%).

Huang and Fang (2013) performed a study that used machine learning techniques to predict student academic performance in engineering courses. In this study, the input features included course grades from all semesters and the output variable was exam scores. The researchers observed that SVMs are suitable for predicting an individual student's performance and that multilinear regression is suitable for forecasting the performance of all students in a course.

Abu Saa (2016) performed a study to find the best classifier to predict student's performance in higher education using social and personal input features.

Some probabilistic models (i.e., Bayesian knowledge tracing) have been used to predict student's performance by analyzing logs compiled during student computer gameplay (Käser et al. 2017). However, these models do not predict hidden patterns of students.

Furthermore, in traditional universities, some statistical methods have been used to predict student's performance; these include linear mixed-effect models (LMEM) and survival analysis techniques that use variable multimodal data (heart rate, step count, weather condition and learning activity) as input along with cumulative student pre-enrollment and semester-wise information (Di Mitir et al. 2017; Ameri et al. 2016).

Currently, most universities provide courses using e-learning systems accessible from any location. Scientists use input features common to these e-learning systems (e.g., time, activity, assessment and online discussion forums) to forecast student performance.

Kotsiantis et al. (2003) predicted student's performance on final exams using machine learning techniques (e.g., Naïve bayes, 3-NN, RIPPER C4.5 and WINNOW). They used demographic features as inputs (e.g., sex, age, marital status, and number of children) along with performance-related input features (e.g., meetings and assignment grades) from an e-learning system and found that the Naïve bayes approach achieved a higher average accuracy (73%) than did the alternatives.

Hu et al. (2014) developed a student warning system using e-learning system features such as course login time, average login time and delay in reading the assignment. They found that C4.5 and CART achieved satisfactory accuracy (93 and 94%, respectively).

Kaur and Kaur (2015) examined student difficulties in a course on mathematics, system analysis, and design using data mining techniques. They used test grades as input features and determined that AdaBoost was the best classifier for predicting the difficulties that students would experience in subjects.

Vahdat et al. (2015) used process mining (PM) and complexity matrix (CM) methods to analyze the relationship between grades and students' learning processes using DEEDS data. They concluded that the average student grades are positively correlated with the CM and that difficulty is negatively correlated with the CM. In addition, they determined that process discovery using PM and CM models provides valuable information regarding student learning processes.

Chen et al. (2000) used database systems and decision tree techniques on e-learning system logs to check the performance of students using an approach helpful for teachers.

Hlosta et al. (2017) introduced a self-learning system using machine learning algorithm to find at-risk students in a new course without any previous history data. This study demonstrated that XGBoost achieved the best performance.

He et al. (2015) provided early predictions of at-risk students in a MOOC course using an LR technique by analyzing assignment and lecture features.

Arnold and Pistilli (2012) developed a learner analytics system that allowed teachers to give real-time support to students and solved student retention problems. Moreover, this system depends on student demographic characteristics, past academic history, student efforts and student grades. The results showed that students using the analytics system achieved higher grades compared to those who did not use the system.

Liu and d'Aquin (2017) used a supervised learning algorithm to predict student's performance. They investigated how demographic variables and online learning activities affect student's performance. Furthermore, they used the k-prototypes clustering algorithm to find the group of weak students who needed additional help from the teacher. They concluded that the successful groups of students mostly came from privilege and most of these students complete their higher education.

The authors Kai et al. (2017) used the J-48 and J-Rip classifiers to identify students who do not continue past the course orientation stage and found that these models provide good information to teachers that can aid in student retention.

Another study Elbadrawy et al. (2015) predicted student grades using a collaborative multi-regression model based on students' performances, activities and Moodle interactions as features. The results revealed that the performance of a collaborative multi-regression model using Moodle interaction features is comparable to that of a linear regression model.

Studies have also been conducted that use ANNs with only slight modifications to classify students based on to their final grades using web-based education system features (VOD-watching times, courseware download times and BBS posting times) (Zheng et al. 2013). Some commonly used machine learning techniques have been investigated to predict student's performance and identify at-risk students in e-learning course (Kuzilek et al. 2015).

Recently, an early predictive model was developed using student demographic, LMS data, and aptitude-related features. The authors developed a learning analytic system with an applied LR model that sent emails to high-risk students (Jayaprakash et al. 2014).

The majority of early studies that focused on predicting student grades and learning behaviors in upcoming course sessions used datasets from traditional universities and e-learning systems. However, their input features did not reflect the students' performances during in-session problem-solving exercises or projects. None of the existing studies predicted students' performances in a technology-enhanced learning (TEL) domain. Most studies used academic input features (e.g., GPA (grade point average) and grade and semester marks) and non-academic input features (e.g., age and gender), which are less effective for making timely predictions concerning student's performance in a TEL system. Moreover, it may be costly to collect these data. Some of the early studies that used training and test data from the same course suffered from the same difficulty; such methods do not help the teacher correctly evaluate the model accuracy in the succeeding session because student course outlines and activities change from one class to another. Performance prediction regarding future coursework sessions based on log data is not a straightforward task because every session has its own difficulties and unique problems. Prediction also depends on course features and teaching techniques; thus, it is important to build an intelligent TEL data system that forecasts the difficulty of the upcoming session.

The first step in improving student learning is being able to predict the difficulty that students will have with the subsequent class session. Predicting student difficulty for the next session using DEEDS logs is important to both the instructor and to the students in MOOCs and TEL systems. However, because teachers of MOOCs and TEL systems are not machine learning experts, they cannot easily interpret the DEEDS log data. A data interpretation feature can be easily integrated into a TEL system or a MOOC to identify the students' difficulties, improve their learning performances, and prevent performance degradation in the subsequent session. In addition, such predictions allow the instructor to use the DEEDS logs in the learning model to determine the probability that a student will encounter difficulties in the next session and to provide feedback to the student in real time. Overall, by using this method, the instructor is better able to prepare students who experience difficulties before they start their subsequent sessions. Thus, this approach is expected to increase retention and provide advance information about the challenges that individual students experience.

This study used machine learning algorithms to predict individual students' difficulties in the subsequent session of a TEL system when the students performed different activities (problem-solving exercises, laboratory assignments, reading course-related materials, etc.) during the course session. These data can be easily integrated into DEEDS and MOOCs to assist teachers in identifying potential student difficulties in upcoming sessions. There does not appear to be any related prior research on using TEL and machine learning techniques to predict student difficulties in a subsequent session of a digital design course.

In the current study, we used log data obtained from the DEEDS (https://www.digitalelectronicsdeeds.com), a TEL tool and virtual digital electronic laboratory used by instructors at the University of Genoa, Italy, both in and out of the classroom to improve student learning. Students remember concepts better when reading course-related materials using the TEL system than when reading course-related materials without the TEL system (Vahdat et al. 2015). Additionally, the DEEDS is an e-learning environment used by students to complete various laboratory assignments in electronic and information engineering classes at the University of Genoa (Donzellini and Ponta 2007). By applying the DEEDS to massive open online courses (MOOCs) and learning management systems (LMSs), teachers can easily track students activities and provide students with news, guidelines and feedback (Donzellini and Ponta 2007).

Our main goals were as follows:

- To identify the most appropriate machine learning algorithms for predicting the difficulty an individual student would have in the next session of a digital design course based on prior session activities and the current session.
- To investigate which machine learning algorithms used in the current study are appropriate for predicting student difficulty in the next session of digital design course while using the fewest features.

The results of the current study show that SVMs and ANNs are appropriate machine learning models to predict a student's performance as well as the difficulty a student will experience over the entire next session in a digital design course. The remainder of this paper is organized as follows: Section 2 includes problem formulation. Section 3 describes the materials and methods, Sect. 4 presents the experimental results, and Sect. 5 presents conclusions and describes future work.

2 Problem formulations

The DEEDs is a technology-enhanced learning and virtual digital electronic laboratory used to improve student learning. The problem of predicting student difficulty in the DEEDs involved investigating the most appropriate machine learning algorithm to predict student difficulties in terms of the grades they would earn in the subsequent session of digital design course exercises and assignments. Providing advance warnings of the difficulties students are likely to face in an e-learning system is a challenging task for instructors. To solve this problem, we developed an early warning system that allows teachers using TEL systems to monitor the performances of students in problem-solving exercises and laboratory assignments throughout multiple sessions. Using the developed tool, teachers can identify students' areas of difficulty in the course in advance and warn those students about their weaknesses. The components of the problem formulation are discussed below.

2.1 Participants

The dataset used in this study was collected from first year BSc students at the University of Genoa, as they solved various digital design course exercises in the DEEDS over five sessions in which the students devoted time to and performed different activities while completing each exercise. Before developing their solutions, students read material related to the exercises. At the end of each session, they earned a grade. The data included records of 100 students completing 6 digital design exercises in each of five sessions while using the DEEDs (Vahdat et al. 2015).

2.2 Procedure

Data collection is a fundamental step in the machine learning field. Student data can be obtained from sources (such as MOOCs, online tutoring systems and course management systems (CMSs)). The present study used data collected from for first-year University of Genoa BSc students as they interacted with a TEL tool called DEEDS. The study data is available from the UCI machine learning repository (Murphy and Aha 1995). The raw information from each session is arranged as follows:

> session_Id, student_Id, exercise, activity, start_time, end_time, idle_time, mouse_wheel, mouse_wheel_click, mouse_click_left, mouse_click_right, mouse_movement, keystrokes.

Such real-time log information can be used as input to construct more accurate machine learning models that can predict the difficulties that students will encounter in the next session. These data characterize the learning processes of students as they interact with the DEEDS. The DEEDS is a TEL tool that provides instruction (lectures, exercises, laboratory assignments, etc.) to students through a specified browser. Using this tool, instructors can enhance student learning. Students solve problems that reflect various levels of difficulty. When using the TEL system, first-year students in the digital design course communicate with the DEEDS, which then provides material from the digital design course to those students. After studying the material, the students must solve exercises in the DEEDS. The current study dataset included five laboratory sessions of course material in digital design along with data from the sequential activities that students performed for each session. Each session was related to a specific part of the digital design course. In this study, the midterm examination grade depended on the exercises solved in each session (Vahdat et al. 2015). As the first step, the DEEDS collects and logs a large amount of data as each student interacts with the system. There is no machine learning readable format, but a format readable by a specific machine learning algorithm instead. We needed to preprocess the raw data in order to apply our machine learning techniques. We used MATLAB to preprocess the raw data.

We extracted the following input features for the 6 exercises of the digital design course for each student over 5 sessions:

The average time, *the total number of activities, the average idle time, the average number of keystrokes and the total related activity.*

A brief discussion of the features used to build the prediction model follows. Two important features are the average time and the total related activities. The former is the average time spent by the student to complete each exercise, and the latter reflects the total number of related activities (such as viewing/reading material related to the exercise and answering exercise questions inside the DEEDS) that the student performed while completing the exercise. These features play important roles in predicting student difficulty in the next session of the e-learning system because students who engage in fewer related activities and expend less average time are not focusing well on the current session. A third important feature is the average number of keystrokes the student made while completing the exercise. This value reflects the level of student engagement while completing the exercise. Most students who receive high grades displayed high engagement during the sessions.

A fourth feature, idle time, is also related to a student's performance; it reveals the average time during which the student is idle during the exercise, i.e., the time during which the student does nothing. Increased idle time suggests that the student has some difficulty in answering the exercise question. Usually, better students have less idle time because they perform many activities during the session. The final best feature is the total number of activities (activities related to the exercise plus activities unrelated to the exercise), which can also affect a student's performance in the session. A student who performs a greater number of activities while completing an exercise is more likely to be busy answering the questions posed in the exercise.

Each record of the five sessions was stored in a separate matrix (XS_1, XS_2, XS_3, XS_4, and XS_5). The output labels were the grades of every student during every session after completing the various exercises; these data were also stored in separate matrixes (GS_1, GS_2, GS_3, GS_4, and GS_5) that represent the student's level of knowledge in the digital design course.

Finally, the best features for predicting student difficulty in next session were selected using the Alpha-investing feature selection method, which reduces the number of input features used for the machine learning model.

2.3 Size, power and precision of the sample

In this study, 100 students solved 6 digital design exercises in each of five sessions. We extracted 30 features for every student in each session. We extracted five features for each exercise for all students in each session. The data from sessions 1–4 (361 student records) were used for training and the data from session 5 (85 student records) were used for testing.

2.4 Measurements and covariates

MATLAB was the source of the machine learning algorithms used to predict student's learning difficulties. The input variables of the current study are related to student's activities while solving exercise and assignments. To compare the performances of the various machine learning algorithms used in this study, in addition to accuracy, precision, F1-score, and recall, we adopted standard metrics such as RMSE, the Kappa statistic, and the ROC as performance indicators.

2.5 Design of the investigation

In this study, we used MATLAB to extract features from raw data and build predictive models. The machine learning models (ANNs, LR, NBCs, SVMs and DTs) were trained on the training data and tested on the testing data.

2.6 Experimental manipulations or interventions

In these steps, we randomly divided the dataset into training and test datasets (80% and 20% of the total data, respectively) to predict the difficulties that students may encounter in the next session. We trained the machine learning classifiers on the training dataset using the input features described above and then tested them on the testing dataset.

2.7 Statistical analysis

For the statistical analysis, SPSS (Statistical Package for the Social Sciences) (https://www.ibm.com/products/spss-statistics) was used to conduct descriptive analyses and regression analyses between the input features and grades of students.

3 Materials and methods

This study analyzed the effectiveness of machine learning methods to help a teacher predict a student's performance during the next session of a TEL system.

3.1 Preprocessing

Because preprocessing is important in machine learning, before training the classifier, it is essential to clean and prepare the DEEDS log data using preprocessing techniques. Thus, we can extract the input variables that are most related to student difficulty. Moreover, the preprocessing steps are significant because the performance of the models depends on the preprocessing methods.

3.1.1 Feature extraction

Feature extraction is a process that creates new features from the original features. Feature extraction is a key step in classification because the learning model performance is dependent on the significant features (input variables) that describe student characteristics and can be used to predict a student's performance. In previous studies, researchers predicted traditional university students' overall performances using features such as student demographic information, student gender and degree. It is difficult for the present study to predict a student's performance in a TEL system using these features because the domain is different; therefore, we extracted 30 features for every student in each session.

We extracted five features (X_1, X_2, X_3, X_4, and X_5) for each exercise for all students in each session and stored them in separate matrices (XS_1, XS_2, XS_3, XS_4 and XS_5) as shown in Table 1. The feature extraction steps are presented in Algorithm 1 and (Fig. 1). Finally, we extracted all the student grades from every session using the MATLAB platform and stored these in a separate matrix (GS_1, GS_2, GS_3, GS_4 and GS_5).

Table 1 List of attributes (features) used in this study

Feature Name	Description
X1. Average time in the exercise	Average time spent between the start time and the end time while completing the exercise
X2. Total activities in the exercise	Total number of activities performed by a student while completing the exercise
X3. Average idle time in the exercise	Average idle time during the total time spent on completing the exercise
X4. Average keystrokes in the exercise	Average number of key strokes executed while completing the exercise
X5. Total related activities in the exercise	Total number of related activities performed by a student while completing the exercise

```
Extract features from CSV file
For i=1 to size (stud id)
    Find total time for each exercise
End
Find unique activity and student ID
For m=1 to size (unique stud id )
    For j=1 to size (unique exercise)
        For i=1 to size (Exercise of student m)
            Find the average time of student for each exercise
            Count the total number of activities of student for each exercise
            Find the average idle time of student for each exercise
            Calculate the average keystrokes of student for each exercise
            Find the related activity of student for each exercise
        End
    End
End
```

Notes: CSV (Comma-separated values)

Algorithm 1: Feature extraction

3.1.2 Feature selection

Feature selection is the process of selecting subsets of relevant variables. This preprocessing step reduces the number of predictor variables used in the machine learning algorithms. Additionally, it helps simplify the machine learning models, reduce the computation time of models, enhance their generalization abilities, prevent overtraining, and reduce the amount of resources (memory and CPU time) required to predict a student's difficulty in a session. Feature selection involves finding all the possible combinations of attributes that increase the accuracy of predictive models. To do this, we used various streaming feature selection (SFS) methods in which features are sequentially selected for the predictive model. Finally, the Alpha-investing method selected 14 of the 30 available features in our dataset (Zhou et al. 2005; Ungar et al. 2005). The results are presented in the Sect. 4.

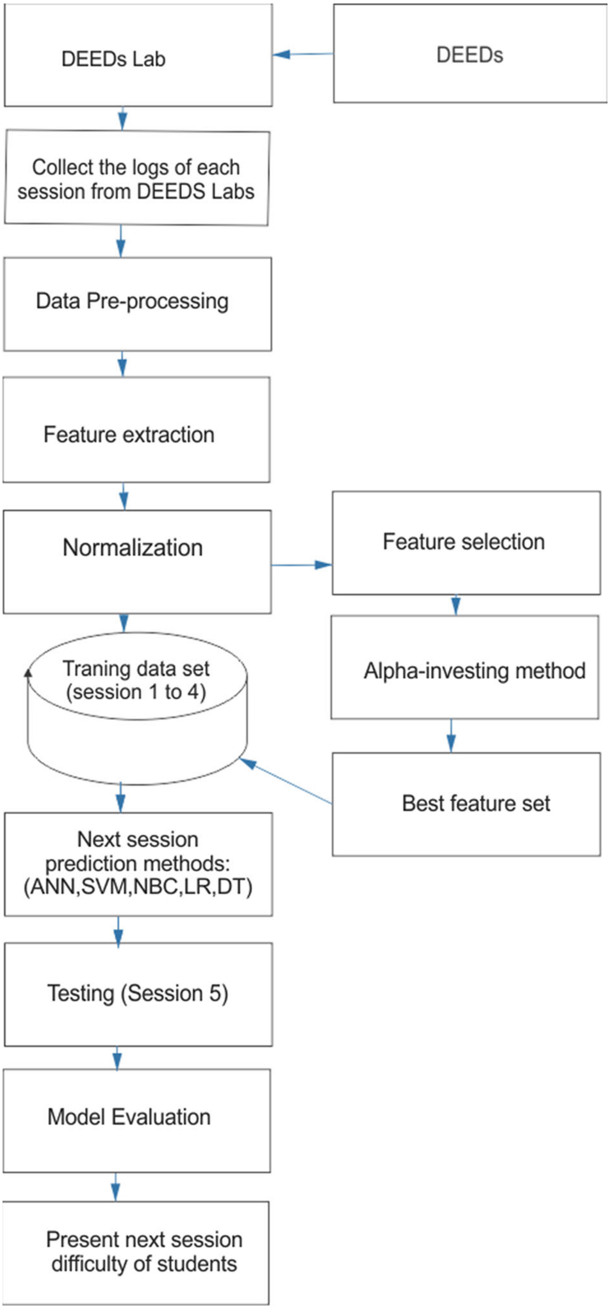

Fig. 1 Flow diagram of the proposed students' performance prediction model. *Note:* Digital electronics education and design suit (DEEDS), artificial neural network (ANN), support vector machine (SVM), Naïve bayes classifier (NBC), logistic regression (LR) and decision tree (DT)

3.1.3 Feature normalization

In this study, the extracted features were initially at different scales; therefore, the data were normalized by dividing the values in each session matrix column by the mean of that column. Thus, each column value is located around the mean.

3.1.4 Grade normalization

In this study, five session exam grades (from the DEEDS) were used as an indicator of a student's performance in the digital design course. The grade is a single number that indicates how well a student did in a specific course (Meier et al. 2016). The students' grades in the five sessions (e.g., $\mathrm{GS}_1,\ \mathrm{GS}_2,\ \mathrm{GS}_3,\ \mathrm{GS}_4,$ and GS_5) varied from 1 to 5. Before developing the machine learning predictive models, the grades of the students were normalized within the range [0, 1]. For example, when a student received a grade of five for a session, his/her normalized grade was 5/6 = 0.83 (Huang and Fang 2013).

3.2 Predicting student difficulty in the next session

This tool predicts the student's difficulty during the next session, which depends on the assignment completed in the current session, and then assists the students according to their difficulty level. To predict each student's difficulty in a digital design course when using the TEL system, this study uses several different well-known machine learning algorithms, including ANNs, LR, SVMs, an NBC and DTs. These are relatively simple classification algorithms that perform well when the attributes are numeric and there is little correlation between attributes. In addition, they are less sensitive to overfitting and require less training data (Hämäläinen and Vinni 2010).

These classifiers learn from past student log data and then inform the student and the teacher of the difficulty predicted for the upcoming session. Teachers can use this information to assess the level of difficulty the student will experience in the upcoming session. To predict each student's difficulty in the next session of the DEEDS, we extracted well-defined input features of the exercises for all the students from the TEL system DEEDS, as shown in Table 1. The training dataset for the classifiers contained exercise-related features of the students in sessions #1–5, and the testing dataset included the mean grades of the students in the same sessions. The results show that the ANN and SVM models can be straightforwardly integrated with TEL and that the instructor can supply feedback in real time to students before the start of the next session. A flow diagram of the proposed framework is shown in (Fig. 1). Three primary components of our study are discussed below.

3.2.1 Combination of predictor variables

This study developed machine learning predictive models using datasets from five sessions. The matrix $X(train) \in R^{n\times m}$ included datasets from sessions #1–4; the dataset for session #5 was saved for testing in the matrix $X(test) \in R^{n\times m}$. The input features of this study are shown in Tables 1 and 2 contains the dataset used in the current study. Each row of Table 2 represents all the features of the student for all the exercises in a session; n = 361 is the number of student records in a session, and m = 30 is the number of student features. Finally, each column represents a single feature for all students.

Table 2 Dataset used to train and test machine learning classifiers

ATE1	TAE1	AITE1	AKE1	TRAE1	ATE2	……….	TRAE6
0.06	0.5088	7.98E−05	0	0.6513	0.0839	……….	0
0.37	0.5088	0.0114	1.7751	1.3289	0.3984	………..	0
0.1	1.271	2.45E−05	0.8439	1.5066	0.104	……….	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.

Average time on exercise (ATE); Total activities in exercise (TAE); Average idle time in exercise (AITE); Average Keystrokes in exercise (AKE); Total related activities in exercise (TRAE); The times of all exercises are calculated in seconds

Student grades are essential parameters for detecting a student's level of understanding of a session. Therefore, the student grades from sessions #1–4 were combined in the matrix $Y(train) \in R^n$, and the grades for session #5 were stored in $Y(test) \in R^n$

At the conclusion of the study, we divided the students' grades for a session into two levels: "no difficulty" and "have difficulty." When a student's grade was $\geq$ 2, that student had no difficulty with the session, and the flag was set to 1. When a student's grade was $<$ 2, that student did have difficulty with the session, and the flag was set to 0.

3.2.2 Model training

Before predicting a student's difficulty with the next session of the digital design course, we must train the machine learning algorithms (ANN, LR, SVM, NBC and DT) to minimize the difference between the actual measured value and the predicted value. In this study, the ANN, SVM, LR, NBC and DT models were developed using the MATLAB platform. A neural network is a commonly used machine learning algorithm that contains a set of input and hidden layers that work in parallel to achieve an overall goal (Haykin 1999). Its performance is affected by the number of hidden layers, the activation function and an alpha value (Haykin 1999). We trained the neural network models using an appropriate number of hidden layers and set the alpha value so that the models could find a pattern between the input and output variables. In addition, we used the gradient function of the neural network model to minimize the cost function and find an optimal value of theta such that the differences between a student's actual and predicted grades were minimized.

Next, we constructed the SVM model using MATLAB's LIBSVM tool and used a grid-search method to locate the optimal values of C and sigma. The SVM is a machine learning algorithm that can learn from a small dataset (Pai and Hong 2005).

To compare the performances of the SVM and the ANN, we built the LR, NBC and DT models, and then trained each model with the same dataset. Logistic regression is the most widely used machine learning algorithm today. We randomly divided the data into two components: a training dataset and a test dataset. We trained the LR model using different alpha values; finally, we obtained the best performing model. We calculated the appropriate value of theta utilizing the gradient function of the logistic regression and then estimated the correct model.

The Naïve Bayes and DT classifiers are also good alternatives for use in predicting a student's performance. DT is a popular and effective technique used for classifying and forecasting (Chaudhuri 1998). The decision tree (classification tree) algorithm generates a classification tree using $X(train) \in R^{n \times m}$ as input variables (predictors) and $Y(train) \in R^n$

(session grade) as the output. Finally, we trained the NB classifier to our data to calculate the probabilities of weak and good students. We evaluated our Naïve Bayes classifier using default distribution functions (normal (Gaussian) distribution) and eventually obtained good performance. Because the mathematical details of the models used in the current study (ANN, SVM, LR, NBC and DT) are complex, they are not included in this paper.

3.2.3 Model evaluation

After building the ANN and SVM models, we evaluated our models on unseen test data (session #5). We used three types of evaluation measures as follows: root mean square error (RMSE), the receiver operator characteristic (ROC) curve and Cohen's kappa coefficient (Fawcett 2004; Kaur and Kaur 2015; Cohen 1960). These evaluation methods can be used to find the goodness of fit between data and a model. The RMSE value indicates how well a model fits unseen test data, and the ROC curve is defined by the relationship between true positives and false positives (Fawcett 2004; Pelanek 2015). Cohen's kappa coefficient compares the observed accuracy to the expected accuracy (Kaur and Kaur 2015; Cohen 1960). In addition, we used the four performance parameters of accuracy, precision, recall, and F1 score.

Accuracy characterizes the degree to which a predicted value agrees with an actual value and can help identify weak students in the early stages of a course (Devasia et al. 2016). Precision identifies the probability of a positive test result. High precision values indicate that the probability of the test set being accurately classified will be high. Because we were predicting weak students in a digital design course session, precision indicates the fraction among them who truly were weak students. Recall evaluates the number of true positives of the actual class predicted by the models (Sweeney et al. 2016; Ge et al. 2011). We computed the recall measure, which represents the fraction of all the students in the data set who truly did not achieve a good grade that the classifiers accurately identified as weak. The recall of the current study model can be interpreted as follows: Higher recall scores indicate better classifier performances. Together, recall and precision indicate how well an algorithm performs.

The third measure is the F1-score, which is a single evaluation metric that indicates which algorithms performed best. Using F1-scores, it is possible to make rapid decisions about relative algorithm performances. These matrices can be obtained follows (Fernandez-Delgado et al. 2014; Marquez-Vera et al. 2015; Rovira et al. 2017):

$$Precision = \frac{True\ positive}{True\ Positive + False\ Positive} \tag{1}$$

$$Recall = \frac{True\ positive}{True\ Positive + False\ Negative} \tag{2}$$

$$F1\ Score = \frac{2PR}{P + R} \tag{3}$$

where P represents precision and R represents recall

$$Accuracy = \frac{True\ positive + True\ Negative}{True\ Positive + False\ Positive + False\ Negative} \tag{4}$$

Algorithms with high F1-scores are considered to be good algorithms. The overall results for precision, recall, and F1-score are listed in Table 4. Sometimes, when the data are imbalanced, the measured accuracy is a poorer measure of performance than are precision or recall; thus, we computed the ROC curve. The results of the current study are presented in Sect. 4.

4 Results and discussion

In the following sections, we describe the parameters used to predict the difficulty that students would have in the next course session and summarize the experimental results.

In this section, the student's difficulty in the digital design course was predicted by conducting three experiments. To understand the DEEDS dataset, it is important to explore the dataset both statistically and visually. This step is essential in machine learning and data mining because it allows researchers to understand the dataset before applying the machine learning algorithms (Ramesh et al. 2013; Abu Saa 2016). Therefore, to find the significance of the input variables for predicting the difficulty students would have with the next session, we conducted a statistical analysis of the student data in the TEL system using SPSS (Statistical Package for the Social Sciences) (https://www.ibm.com/products/spss-statistics) and found a connection between the students' independent variables (features of the exercise) and the dependent variable (the students' grade at the end of the session) using a significance level of 0.05. The summary statistics are shown in Table 3. The mode shows the value with the highest frequency (Abu Saa 2016). For this study, Table 3 reveals that of the 30 variables, 9 variables have a significant ($P < 0.05$) correlation with session grades. In statistics, the correlation coefficient (R) value is always between -1 and $+1$ and reflects the strength of the linear relationship between two variables. The correlation results show that 4 variables (*Total number of related activities in exercise 6, total number of activities in exercise 6, total number of related activities in exercise 3 and total number of activities in exercise 3*) were moderately correlated (R = 0.40–0.5) with session grades and 5 variables (*total average keystrokes in exercise 6, total number of related activities in exercise 4, total number of activities in exercise 4, total number of related activities in exercise 2 and total number of activities in exercise 2*) had a weaker correlation (R = 0.20–0.30) with session grades (Zacharis 2015).

We then plotted the number of keystrokes and its relationship to the students' grades during the sessions. Figure 2 shows the average number of keystrokes for each student in exercise, which reflects their engagement. The plot shows the average number of keystrokes related to different student' grades. This well-known variable is used to compute a student's online effort in a session and it is important for determining the difficulty a student will have in the next session of the digital design course. Figure 3 shows the distribution if student grades in the digital design course.

Because this analysis is statistical, it does not provide hidden information about students and does not provide much insight into a student's performance. Because 9 variables (*Total number of activities in exercise 2, Total number of related activities in exercise 2, Total number of activities in exercise 3, Total number of related activities in exercise 3, Total number of activities in exercise 4, Total number of related activities in exercise 4, Total number of activities in exercise 6, average keystrokes in exercise 6, Total number of related activities in exercise 6*) had a significant predictive value in this study, we applied machine learning algorithms using these variables as input to predict the difficulty a student would have in their next session. We used ANN, LR, SVM, NBC and DT classification models to classify the relationship between student's performance in the exercises during all sessions and the grades they received at the end of a session. We used the MATLAB programming language to construct machine learning predictive models and used two model evaluation methods (cross-validation and RMSE). The cross-validation method was used for model selection and to determine how the model works with succeeding session features. The RMSE value is calculated between the actual and the predicted grades of a student in a session (Elbadrawy et al. 2015).

Table 3 Regression analysis and descriptive statistics for input features (X) and grades (Y)

	Variables	R	R^2	*P*	r	Mode	SD	Mean	Max.	Min.
1	average time in exercise 1	0.020	0.000	0.679	0.081	0.06	1.11	0.39	16.1	000
2	Total number of activities in exercise 1	0.050	0.003	0.311	0.041	1.00	5.91	3.67	72.0	001
3	average idle time in exercise 1	0.007	0.000	0.899	− 0.094*	000	3.84	3.23	7.08	000
4	average keystrokes in exercise 1	0.080	0.008	0.093	− 0.010	000	4.30	6.52	31.9	000
5	Total number of related activities in exercise 1	0.040	0.002	0.424	− 0.018	185	97.4	2221	573	041
6	average time in exercise 2	0.037	0.001	0.482	− 0.090	0.04	1.06	0.39	16.7	000
7	Total number of activities in exercise 2	0.340	0.120	0.000*	− 0.176**	2.00	64.2	90.9	416	001
8	average idle time in exercise 2	0.001	0.000	0.983	− 0.0093	000	3.82	3.23	7.08	000
9	average keystrokes in exercise 2	0.090	0.009	0.710	0.034	2.60	4.17	6.66	32.2	0.19
10	Total number of related activities in exercise 2	0.360	0.131	0.000*	− 0.177**	1.00	34.5	46.8	212	000
11	average time in Exercise 3	0.017	0.000	0.744	− 0.003	0.02	1.08	0.39	16.7	000
12	Total number of activities in exercise 3	0.550	0.254	0.000*	− 0.243**	32.0	72.5	98.7	365	000
13	average idle time in exercise 3	0.003	0.00	0.960	0.016	000	3.38	3.23	7.08	000
14	average keystrokes in exercise 3	0.072	0.005	0.174	0.033	000	4.12	6.45	32.2	000
15	Total number of related activities in exercise 3	0.483	0.234	0.000*	− 0.246**	19.00	40.8	52.9	202	000
16	average time in exercise 4	0.005	0.000	0.927	0.003	000	1.07	0.37	16.7	000
17	Total number of activities in exercise 4	0.329	0.108	0.000*	− .186**	000	103	99.5	500	000
18	average idle time in exercise 4	0.007	0.000	0.902	0.017	000	3.83	3.09	7.08	000
19	average keystrokes in exercise 4	0.093	0.009	0.790	0.001	000	4.66	6.69	33.0	000
20	Total number of related activities in exercise 4	0.322	0.104	0.000*	− 0.175**	013	58.4	54.6	309	000
21	average time in exercise 5	0.093	0.009	0.076	− 0.021	000	0.60	0.26	8.68	000
22	Total number of activities in exercise 5	0.031	0.001	0.567	0.074	000	69.8	69.3	376	000
23	average idle time in exercise 5	0.055	0.003	0.299	− 0.013	000	8.95	9.35	1.65	000
24	average keystrokes in exercise 5	0.072	0.005	0.169	0.112*	000	4.38	5.75	20.0	000

Table 3 continued

	Variables	R	R^2	*P*	r	Mode	SD	Mean	Max.	Min.
25	Total number of related activities in exercise 5	0.017	0.000	0.750	− 0.143**	000	35.1	33.6	200	000
26	average time in exercise 6	0.154	0.210	0.006	0.052	000	0.56	0.14	8.67	000
27	Total number of activities in exercise 6	0.503	0.253	0.000*	0.267**	000	111	66.2	605	000
28	average idle time in exercise 6	0.060	0.004	0.258	− 0.006	000	8.87	7.46	1.65	000
29	average keystrokes in exercise 6	0.294	0.087	0.000*	0.253**	000	3.08	1.95	20.0	000
30	Total number of related activities in exercise 6	0.468	0.219	0.000*	1.000	000	36.9	21.6	257	000

Standard deviation (SD); minimum (Min); maximum (Max); Significance level ($*P < 0.05$); **correlation is significant at the 0.01 probability level (2-tailed); *correlation is significant at the 0.05 probability level (2-tailed); Pearson's correlation (r); coefficient of correlation (R^2);
Y (output features): grades of student in sessions
X (input features): average time, total activities, average idle time, average keystrokes and total related activities in each exercise of all students

The input features of the first experiment were five student variables. These features were *the average time*, *the total number of activities, the average idle time, the average number of keystrokes* and *the total related activity* on each exercise; the output variables were the mean grades of students in the sessions. The datasets were randomly divided into training and test data at percentages of 80 and 20%, respectively, so that the training set contained all students' class performance data. If the size of the training dataset is too small or too large, the performance of the models will be affected (Ward et al. 2010). We used a training dataset from previous data sessions ($X(train) \in R^{n\times m}$, $Y(train) \in R^{n}$) to train the machine learning algorithms (LR, NBC, SVM, ANN and DT) and tested our models on unseen test data from the next session ($X(test) \in R^{n\times m}$, $Y(test) \in R^{n}$).

In the logistic regression model of this study, we used a cost function for the logistic regression model to determine the most appropriate value of theta and then used logistic gradient descent with regularized parameters to decrease the error between actual values and predicted values. The best performance of the logistic regression model was achieved using the parameters alpha = 0.04 and iterations = 1,000,000.

Next, we constructed an ANN model. The overall performance of an ANN depends on the number of hidden layers, activation functions and learning rates (Haykin 1999). We identified an optimal number for the hidden layer size because it affects the model capacity and generalization. We started with one hidden layer and increased the size of the hidden layer until we achieved satisfactory accuracy using alpha = 0.04 and 25 hidden layers (Ducher et al. 2005).

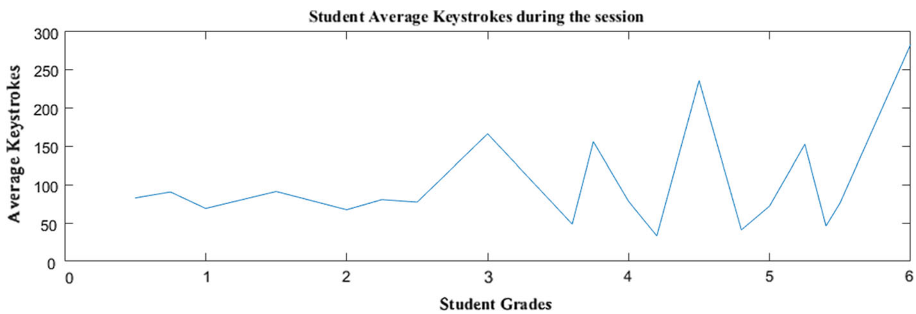

Fig. 2 Keystrokes during an exercise in the digital design course

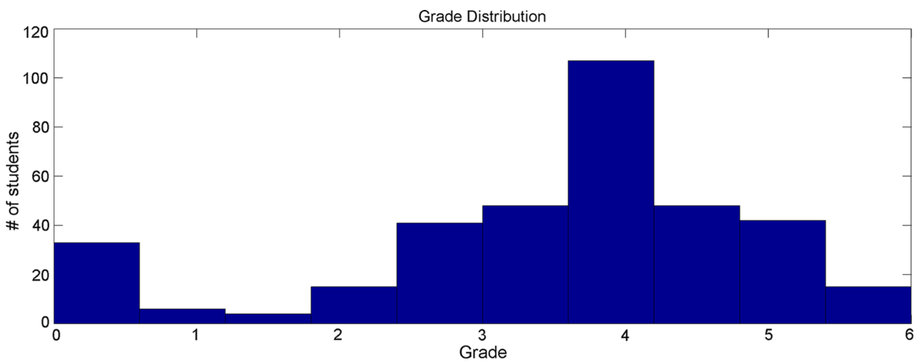

Fig. 3 Histogram showing the distribution of students' grades in the digital design course

Table 4 Comparative results of artificial neural network (ANN), logistic regression (LR), Naive bayes classifiers (NBC), support vector machine (SVM), decision tree (DT) when predicting a student's performance in the next session using the random division method

Classifier	Avg. RMSE	Avg. precision	Avg. recall	Avg. F1 score	Avg. acc. (%)
ANN	0.48	0.8	0.91	0.85	75
LR	0.5	0.79	0.9	0.84	73
NBC	0.49	0.82	0.9	0.85	75
SVM	0.48	0.8	0.91	0.85	75
DT	0.54	0.79	0.83	0.81	69

Average (Avg); root mean squared error (RMSE); accuracy (Acc). All the above values represent the average values of all machine learning models over all trials in the first experiment

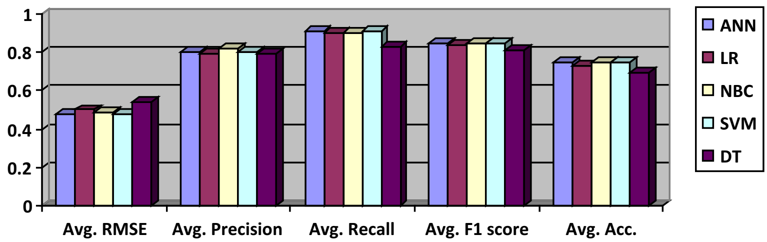

Fig. 4 Visualization of the comparative results obtained using artificial neural network (ANN), logistic regression (LR), Naïve bayes classifier (NBC), support vector machine (SVM), and decision tree (DT) algorithms for predicting a student's performance in the next session under the random division method. *Note*: average (Avg); root mean squared error (RMSE); accuracy (Acc)

To predict student difficulty using SVMs, we developed an SVM model using MATLAB's LIBSVM tool. We used a grid-search method and found that the optimal parameter values were C = 8 and sigma = 0.125. We adopted the Gaussian kernel method for feature mapping.

Finally, we compared the performances of the ANN and SVM models with those of the DT and NBC models. To do this, we randomly divided the dataset into training and test data with percentages of 80 and 20% respectively. Then, we tested the trained NBC models with new session data ($X(test) \in R^{n\times m}$, $Y(test \in R^n)$. In DT, the decision-making process begins at the root node and moves to child nodes, and the last child node contains the predicted label. We tested the trained DT model on unseen session data ($X(test) \in R^{n\times m}$, $Y(test \in R^n)$. We repeated the first experiment five times for each classifier model. The detection accuracies of the ANN, LR, NB, SVM and DT classifiers were 75, 73, 75, 75 and 69%, respectively. The average precision, recall, accuracy and F1 scores are shown in Table 4 and Fig. 4.

All the Table 4 average precision, recall, F1-score and accuracy values were calculated using Eqs. 1, 2, 3 and 4, respectively. Even when a model achieves high accuracy, it may not be a good model because high accuracy may be inadvertently obtained when the dataset is unbalanced; thus, we found the RMSE through cross-validation and ROCs (Moseley and Mead 2008). We used five-fold cross-validation to assess the models performances using the RMSE evolution metric. The results are shown in Table 5.

We compared the prediction results of all classifiers evaluated during the digital design course. These experimental results showed that significant improvements were obtained in terms of model accuracy as measured by the RMSE error when using the preprocessing method and employing the input features identified as valuable in the present study. The

Table 5 Comparative results (RMSE) of artificial neural network (ANN), logistic regression (LR), Naïve bayes classifier (NBC), support vector machine (SVM), and decision tree (DT) for predicting a student's performance in the next session under the n-fold cross-validation technique

No.	Models	RMSE (root mean square error)
1	ANN	0.5
2	LR	0.51
3	NBC	0.65
4	DT	0.52
5	SVM	0.01

primary experimental results showed that both ANN and SVM achieved impressive performances (75%) relative to the alternatives (LR, NBC and DT), as shown in Table 4. The results also showed that the accuracy of the logistic regression and decision tree classifiers were unsatisfactory. We obtained the lowest RMSE values using the ANN and SVM through five-fold cross-validation (0.5 and 0.01, respectively) as shown in Table 5. This result is encouraging because session #5 has a different difficulty level than do sessions #1–4 and because the number of students and topics for this session are also different. The differences in session difficulty levels makes it impossible to achieve perfect accuracy in this study. LR and the NBC are well-known algorithms, but the current study results show that, compared to alternative methods, both classifiers perform poorly when predicting student difficulty in the next session of a course because they are more sensitive to data imbalance and missing datasets.

For this experiment, we also plotted the ROC curve and computed the area under the curve (AUC), plotting the true positive rate against the false positive rate for different discrimination thresholds. Using the ROC method, a model is considered to offer satisfactory performance when its AUC value is closer to 1 than to 0.5 (Barata et al. 2016). Figure 5 shows that the SVM and ANN have higher ROC values than the alternatives and thus exhibit better performances.

In the second experiment, we also compared the performance of the proposed algorithms (ANN, LR, SVM, NBC and DT) using Cohen's kappa coefficient method. The kappa statistic and its evaluation matrix compare the observed accuracy with the expected accuracy. The kappa statistic value of 0.5 for both the SVM and ANN models indicates their favorable performance relative to the other models (LR, NBC and DT) (Cohen 1960; Kaur and Kaur 2015). The results are shown in Table 6.

In the third experiment, we applied a feature selection method (Alpha-investing) to rank the subset of attributes (Zhou et al. 2005; Ungar et al. 2005). All the classifiers in the current study were trained with these features and tested using the test data. This last experiment demonstrates that the SVM achieved the highest accuracy (80%) using features obtained from the Alpha-investing method. The results are shown in Table 7 and Fig. 6.

Finally, we determined that both ANNs and SVMs offered satisfactory prediction performances in all the experiments, indicating that both are appropriate algorithms for predicting the difficulty that students will have in the next session of the digital design course. The experiments demonstrated that using the current study variables, SVM and ANN model success predicted the student's difficulty in the next session of the digital design course with and accuracy 75%.

Compared to the performances of the other models, the ANN and SVM performances were more favorable for the following reasons: (1) An ANN detects all types of interactions with

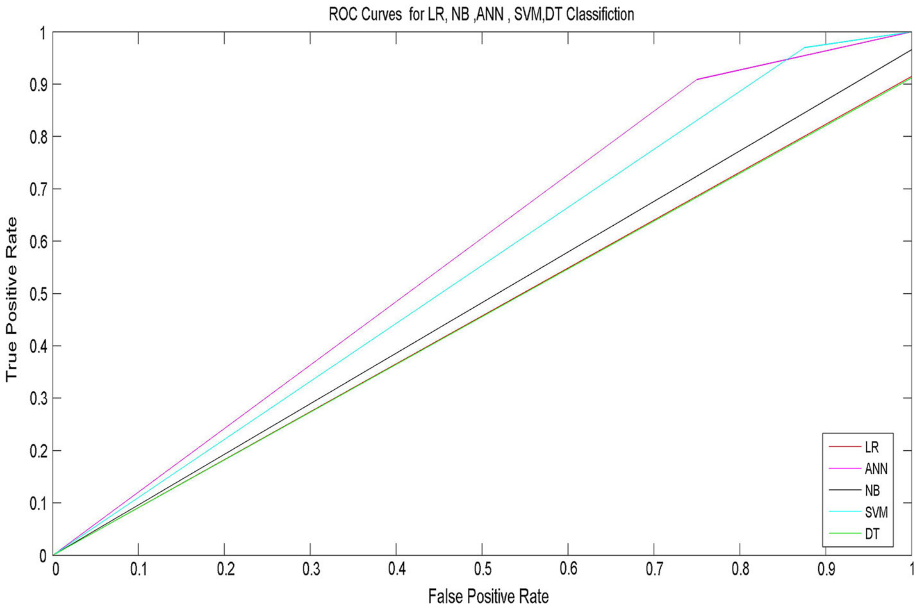

Fig. 5 Receiver operator characteristic (ROC) curves of the artificial neural network (ANN), logistic regression (LR), Naïve bayes classifier (NBC), support vector machine (SVM) and decision tree (DT) classifiers for predicting a student's performance in the next session

Table 6 Kappa statistics for artificial neural network (ANN), logistic regression (LR), Naïve bayes classifier (NBC), support vector machine (SVM), and decision tree (DT)

No.	Models	Kappa statistic
1	ANN	0.52
2	LR	0.43
3	NBC	0.4
4	DT	0.42
5	SVM	0.5

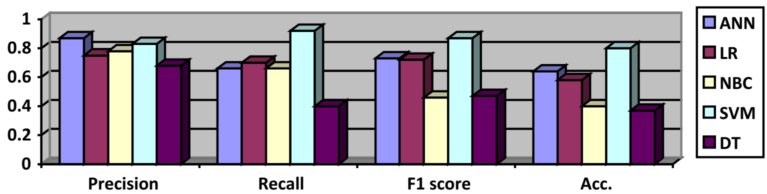

Fig. 6 Acc. (accuracy) visualization of the comparative results of artificial neural network (ANN), logistic regression (LR), Naïve bayes classifier (NBC), support vector machine (SVM), and decision tree (DT) for predicting a student's performance in the next session using the Alpha-investing feature selection method

Table 7 Comparative results of artificial neural network (ANN), logistic regression (LR), Naïve bayes classifier (NBC), support vector machine (SVM), and decision tree (DT) for predicting a student's performance in the next session using the alpha-investing feature selection method

Classifier	Precision	Recall	F1 score	Accuracy (%)
ANN	0.87	0.66	0.73	64
LR	0.75	0.7	0.72	58
NB	0.78	0.66	0.46	40
SVM	0.83	0.92	0.87	80
DT	0.68	0.4	0.47	37

Table 8 Number of weak students (grade < 2) predicted by artificial neural network (ANN), and support vector machine (SVM) in each session

Classifier	Session 1	Session 2	Session 3	Session 4	Session 5
ANN	3	23	3	4	39
SVM	3	41	6	1	38

the output variable and is suitable for handling all types of nonlinear relationships between the input variables and the output variable; (2) An ANN can learn from noisy training data and correctly predict the output of unseen data (Lykourentzou et al. 2009); (3) ANN prediction is trained on unseen data and can be quickly compared to other models (Lykourentzou et al. 2009); and (4) both ANNs and SVMs can learn from a small dataset (Huang and Fang 2013). When determining the optimal parameters in the current study, it was found that the convergence speed of the ANN is slower than that of the alternatives (Kaur and Kaur 2015).

As shown in Table 7, SVM is a good choice for predicting student difficulty in the next session when the features are selected using a sequential feature selection technique. In this experiment the SVM achieved a better performance (80%) than did the other tested classifiers (ANN, LR, NBC and DT).

At the end of all the experiments, we investigated the difficulty of the sessions using the SVM and ANN models and compared our results with those obtained in a previous study (Vahdat et al. 2015) in which the instructors were manually interviewed to determine the difficulty of the sessions during the digital design course. During the manual interviews, most instructors agreed that, compared with other sessions, sessions #4 and #5 were more difficult for the students (Vahdat et al. 2015).

The results shown in Table 8 and Fig. 7 reveal that compared to the manual interview, the proposed model using student logs provided more insight and more precise results: the manual interviews were less reliable. Additionally, the results show that session #2 and session #5 are more difficult for students. These results indicate that our model can help teachers determine the difficulty of sessions that they did not realize were difficult for the students. From all the above experiments two important conclusions can be made:

1. If instructors want to predict an individual's results as well as the difficulty of an entire session for digital design students when completing different exercises and assignments during a course, then SVM and ANN models are most appropriate for the DEEDs and other e-learning systems. This conclusion is reached because SVMs and ANNs achieve

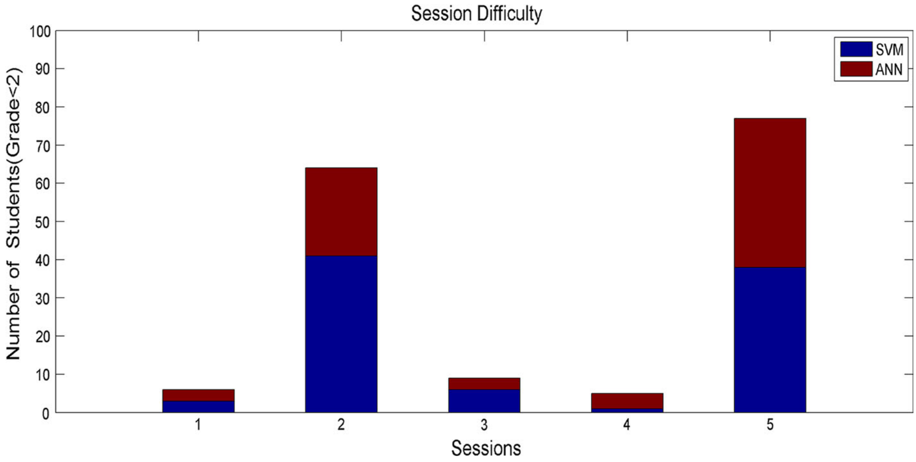

Fig. 7 Number of weak students (grade $<$ 2) predicted by artificial neural network (ANN), and support vector machine (SVM) in each session

high accuracy (75%), a good kappa statistic value (0.5), a high ROC value and low RMSE scores (0.5 and 0.01, respectively) compared to other models (NBC, LR and DT).

2. If teachers want to predict a student's difficulty in the next session of a digital design course using the fewest features, then an SVM should be used with the Alpha-investing feature selection technique. This conclusion is reached because the SVMs obtained the highest accuracy (80%) compared to other models (NBC, LR and DT).

The evaluation results of all the experiments show that integrating the SVM and ANN models with the TEL system can improve the prediction of the difficulty students will have in a subsequent DEEDS session. The relationship between the DEEDS and the system for predicting student difficulties is shown in (Fig. 8), which illustrates how teachers utilize the logged student data during interactions with the DEEDS. The proposed model consists of the four following main components: a logs module, a pre-processing module, machine learning predictive modules and a knowledge base module.

Logs module The logs module contains the student-side log from the DEEDS system. As mentioned in the data description, the teacher presents various exercises and laboratory assignments to the students during different sessions through the DEEDs system. Each session is associated with a different part of the course (digital design), and students complete these tasks. The student-side computer in the DEEDs lab generates student activity logs, which the teacher can then use to evaluate student behavior from a different perspective.

Pre-processing module Typically, the teacher cannot interpret the DEEDS logs. Thus, the pre-processing module converts student logs into a format that is readable by the machine learning methods. Additionally, the pre-processing module extracts features from the student logs and sends them to the machine learning predictive module.

Machine learning predictive module The machine learning predictive modules (utilizing an SVM or ANN) predict student difficulties during the next session. They play an important role in the delivery of information in the form of a graph that allows the instructor to go online and diagnose student or class difficulties in a specific part of the

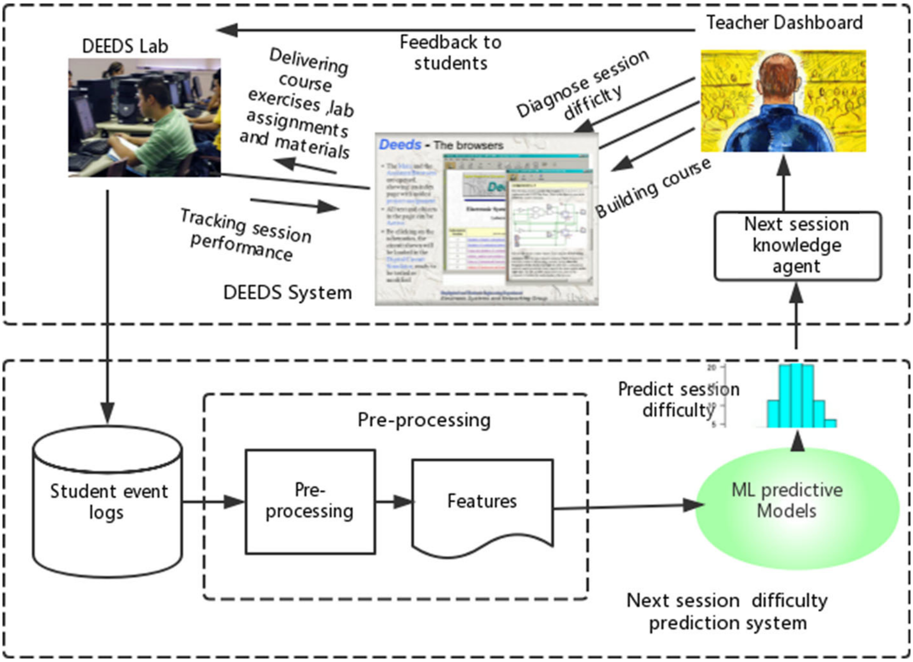

Fig. 8 The proposed DEEDS model. *Note:* machine learning (ML); digital electronics education and design suite (DEEDS)

course or laboratory assignment. The teacher can then dedicate more time, include more material and/or devote more effort to those parts of the course.

Knowledge base module The knowledge base module contains knowledge that is extracted from the DEEDS log by the machine learning models. Based on the results of the current research, a DEEDS developer could develop a teacher dashboard to help teachers predict student difficulties in the next session; then this knowledge could be moved to the teacher dashboard. Using this dashboard, the teacher would receive information about individual students and the difficulty of an session, allowing them to make appropriate decisions about students.

When all students experience difficulty during a session, the quality of the session may be questionable such results indicate that students are not performing well in that session. This information can increase the teachers' awareness, allowing them to balance the difficulties of the exercises with the students' academic backgrounds and to find students that need help as well as identify where they need help. Additionally, instructors could spend more time on those topics during the session to identify which activities and exercises are most important for the student.
The instructors of an e-learning system or a TEL system can use this tool to compare the expected average score and the actual score and thus improve his or her teaching method and style.

Another important capability of adding this model to the DEEDS is that the teacher can use the prediction results to warn students (e.g., an email warning) of their likely difficulties in advance (before the beginning of the next session); thus, better preparing the student to work harder in the next session (or to repeat the previous session to improve their knowledge level).

More specifically, DEEDS teachers need to train the proposed model only once using the data from the first session. After training, the proposed model can be used to identify difficulties in upcoming sessions and to predict a student's difficulty during the course.

This enhancement can assist instructors in planning lessons based on the difficulties students may face and to identify underprepared students at the beginning of a session. Moreover, this tool can help instructors monitor student progress more frequently and provide students with real-time feedback before the final exam. These results will help the student to change their study strategies.

Using the mechanism described above, the overall quality and effectiveness of teaching may be improved, and the student failure rate may be reduced.

5 Conclusions

This study investigated the ability to predict a student's difficulty for a subsequent coursework session using a TEL system and the MATLAB programming language. We extracted students' input features and output variables (e.g., the mean grades of students in a session) from the TEL system.

First, we trained the models (LR, ANN, SVM, NBC and DT) using training data (from sessions #1–4) that were based on all input features and tested the models on testing data (from session #5). Next, we applied two approaches to assess the performance of the models: (five-fold cross-validation and random division of the data into portions). The results showed that the ANN and SVM models achieved high accuracy (75%) in predicting the difficulty a student will have with the next session of the digital design course. We then trained all the models used in the current study on features selected via the Alpha-investing method and attained an accuracy of up to 80% using the SVM model.

The current study revealed that SVM and ANN models can be integrated into TEL systems and MOOCs to help a student select an appropriate session for study and to work more effectively throughout the class rather than waiting to study before the final exam. Such an enhancement can assist teachers in identifying poorly prepared students in advance of the next session so that they can make appropriate decisions regarding those students. Moreover, the teacher can also determine the learning behaviors of students during different exercises and laboratory assignments and determine session difficulty in advance. Thus, using an SVM or an ANN can improve teaching, learning and student success.

In future work, we plan to use the K-means algorithm to study the learning behaviors of students during their interactions with the DEEDS with the goal of helping instructors improve the performance of underprepared students.

Acknowledgements The work of this paper is supported by National Natural Science Foundation of china (Nos.61572434, 91630206 and 61303097) and the National Key R&D Program of China (No. 2017YFB0701501).

Compliance with ethical standards

Conflicts of interest The authors have no conflicts of interest.

References

Abu Saa A (2016) Educational data mining and students' performance prediction. Int J Adv Comput Sci Appl. https://doi.org/10.14569/IJACSA.2016.070531

Acharya A, Sinha D (2014) Early prediction of students performance using machine learning techniques. Int J Comput Appl 107(1):37–43. https://doi.org/10.5120/18717-9939

Ameri S, Fard MJ, Chinnam RB, Reddy CK (2016) Survival analysis based framework for early prediction of student dropouts. In: 25th Procedding of the ACM conference information and knowledge management, pp 903–912. https://doi.org/10.1145/2983323.2983351

Arnold KE, Pistilli (2012) Course signals at purdue: using learning analytics to increase student success. In: 2nd International conference on learning analytics and knowledge (LAK'12), pp 267–270. https://doi.org/10.1145/2330601.2330666

Bakki A, Oubahssi L, Cherkaoui C, George S (2015) Motivation and engagement in MOOCs: How to increase learning motivation by adapting pedagogical scenarios? Desing for teaching and learning in a network world. Lecture notes in computer science 9307:556–559

Barata G, Gama S, Jorge J, Goncalved D (2016) Early prediction of student profiles based on performance and gaming preferences. IEEE Trans Learn Technol 3(9):272–284. https://doi.org/10.1109/TLT.2016.2541664

Chaudhuri S (1998) Data mining and database systems: Where is the intersection? Data Eng Bull 21(1):1998

Chen G-D, Liu C, Ou K-L, Liu B-J (2000) Discovering decision knowledge from web log portfolio for managing classroom processes by applying decision tree and data cube technology. J Educ Comput Res 23(3):305–332. https://doi.org/10.2190/5JNM-B6HP-YC58-PM5Y

Cohen J (1960) A coefficient of agreement for nominal scales. Educ Psychol Meas 20(1):37–46. https://doi.org/10.1177/001316446002000104

De Albuquerque RM, Bezerra AA, de Souza DA, do Nascimento LBP, de Mesquita sa JJ, do Nascimento JC (2015) Using neural networks to predict the future performance of students. In: IEEE international symposium on computers in education (SIIE) 2015, pp 109–113. https://doi.org/10.1109/SIIE.2015.7451658

Devasia MT, Vinushree, HV (2016) Prediction of students performance using educational data mining. In: International conference on data mining and advanced computing (SAPIENCE). https://doi.org/10.1109/SAPIENCE.2016.7684167

Di Mitir D, Scheffel M, Drachsler H, Börner D, Ternier S, Specht M (2017) Learning pulse: a machine learning approach for predicting performance in self-regulated learning using multimodal data. In: 2017 seven international conference on learning analytics and knowledge, pp 188–197. https://doi.org/10.1145/3027385.3027447

Donzellini G, Ponta D (2007) A simulation environment for e-learning in digital design. IEEE Trans Ind Electron 54(6):3078–3085. https://doi.org/10.1109/TIE.2007.907011

Ducher M, Cerutti C, Marquand A, Mounier VC, Hanon O, Girerd X, Ader C, Juillard L, Fauvel JP, Club DJ (2005) How to limit screening of patients for atheromatous renal artery stenosis in two-drug resistant hypertension? J Nephrol 18(2):161–165

Elbadrawy A, Studham RS, Karypis G (2015) Collaborative multi-regression models for predicting students' performance in course activities. In: 5th International conference on learning analytics and knowledge (LAK '15), pp 103–107. https://doi.org/10.1145/2723576.2723590

Fawcett T (2004) Roc graphs: notes and practical considerations for researchers. HP Laboratoreis, Palo Alto. 31(8):1–38

Fernandez-Delgado M, Mucientes M, Vazquez-Barreiros B, Lama M (2014) Learning analytics for the prediction of the educational objectives achievement. In: 44th IEEE Frontiers in Eeducation conference (FIE), pp 2500–2503. https://doi.org/10.1109/FIE.2014.7044402

Ge X, Liu J, Qi Q, Chen Z (2011) A new prediction approach based on linear regression for collaborative filtering. In: 8th International 2011 conference on fuzzy systems and knowledge discovery (FSKD), pp 2586–2590. https://doi.org/10.1109/FSKD.2011.6020007

Hämäläinen W, Vinni M (2010) Classifiers for educational data mining. Handbook of educational data mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series,CRC Press, pp 57–74. https://doi.org/10.1201/b10274-7

Haykin S (1999) Neural networks: a comprehensive foundation, 2nd edn. Prentice-Hall, Upper Saddle River

He J, Bailey J, Rubinstein BIP, Zhang R (2015) Identifying at-risk students in massive open online courses. In: 29th AAA conference on artificial intelligence 2015, pp 1749–1755

Hlosta M, Zdrahal Z, Zendulka J (2017) Ouroboros: early identification of at-risk students without models based on legacy data. In: 7th International conference on learning analytics & knowledge (LAK'17), pp 6–15. https://doi.org/10.1145/3027385.3027449

Hu Y-H, Lo C-L, Shih S-P (2014) Developing early warning systems to predict students online learning performance. Comput Human Behav 36:469–478. https://doi.org/10.1016/j.chb.2014.04.002

Huang S, Fang N (2013) Predicting student academic performance in an engineering dynamics course: a comparison of four types of predictive mathematical models. Comput Educ 61:133–145. https://doi.org/10.1016/j.compedu.2012.08.015

Imran H, Hoang Q, Chang T-W, Kinshuk, Graf S (2014) A framework to provide personalization in learning management systems through a recommender system approach. In: Intelligent information and database system. ACIIDS 2014. Lecture notes in computer science 8397, pp 271–280. https://doi.org/10.1007/978-3-319-05476-6_28

Jayaprakash SM, Moody EW, Lauria E, Regan JR, Baron JD (2014) Early alert of academically at-risk students: an open source analytics initiative. J Learn Anal 1(1):6–47. https://doi.org/10.18608/jla.2014.11.3

Kai S, Miguel J, Andres L, Paquette L, Baker RS, Molnar K, Watkins H, Moore M (2017) Predicting student retention from behavior in an online orientation course. In: 10th International conference on education data mining

Käser T, Hallinen NR, Schwartz DL (2017) Modeling exploration strategies to predict student performance within a learning environment and beyond. In: 17th International conference on learning analytics and knowledge 2017, pp 31–40. https://doi.org/10.1145/3027385.3027422

Kaur K, Kaur K (2015) Analyzing the effect of difficulty level of a course on students performance prediction using data mining. In: 1st international conference on next generation computing technologies 2015, pp 756–761. https://doi.org/10.1109/NGCT.2015.7375222

Kloft M, Stiehler F, Zheng Z, Pinkwart N (2014) Predicting MOOC dropout over weaks using machine learning methods. In: Proceeding of the EMNLP 2014 workshop on analysis of large scale social interacion in MOOCs, pp 60–65

Kotsiantis S, Pierrakeas C, Zaharakis I, Pintelas P (2003) Efficiency of machine learning techniques in predicting students performance in distance learning systems. Recent advances in mechanics and related fields. University of Patras Press, pp 297–306

Kuzilek J, Hlosta M, Herrmannova D, Zdrahal Z, Vaclavek J, Wolff A (2015) OU analyse: analysing at-risk student at the open university. Learn Anal Rev 15(1):1–16

Liu S, d'Aquin M (2017) Unsupervised learning for understanding student achievement in a distance learning setting. In: IEEE global engineering education conference (EDUCON), pp 25–28. https://doi.org/10.1109/EDUCON.2017.7943026

Lykourentzou I, Giannoukos I, Mpardis G, Nikolopoulos V, Loumos V (2009) Early and dynamic student achievement prediction in e-learning courses using neural networks. J Am Soc Inf Sci Technol 60(2):372–380. https://doi.org/10.1002/asi.v60:2

Marbouti F, Diefes-Dux HA, Madhavan K (2016) Models for early prediction of at-risk students in a course using standards-based grading. Comput Educ 103:1–15. https://doi.org/10.1016/j.compedu.2016.09.005

Marquez-Vera C, Cano A, Remero C, Noman YM, Fardoun HM, Ventura S (2015) Early dropout prediction using data mining :a case study with high school grade. Expert Syst 33(1):107–124. https://doi.org/10.1111/exsy.12135

Meier Y, Xu J, Atan O, Van Der Schaar M (2016) Predicting grades. IEEE Trans Signal Process 64(4):959–972. https://doi.org/10.1109/TSP.2015.2496278

Moseley LG, Mead DM (2008) Predicting who will drop out of nursing courses: a machine learning exercise. Nurse Educ Today 28(4):469–475. https://doi.org/10.1016/j.nedt.2007.07.012

Murphy PM, Aha DW (1995) UCI repository of machine learning databases, (Machine Readable Data Repository). Dept. Inf. Comput. Sci., Univ. California, Irvine, CA

Pahl C, Donnellan D (2002) Data mining technology for the evaluation of web-based teaching and learning systems. In: 7th International conference on e-learning in business, government and higher education, pp 15–19

Pai P-F, Hong W-C (2005) Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. Electric Power Syst Res 74(3):417–425. https://doi.org/10.1016/j.epsr.2005.01.006

Pelanek R (2015) Metrics for evaluation of student models. J Educ Data Min 7(2):1–19

Ramesh V, Parkavi P, Ramar K (2013) Predicting student performance: a statistical and data mining approach. Int J Comput Appl 63(8):35–39. https://doi.org/10.5120/10489-5242

Rovira S, Puertas E, lgual L (2017) Data-driven system to predict academic grades and dropout. PloS ONE 12(2):e0171207. https://doi.org/10.1371/journal.pone.0171207

Smith-Gratto K (1999) Best practices and problems. Report to the distance education evaluation task force distance educaiton. North Carolina A & T state University, Raleigh

Sweeney M, Rangwala H, Lester J, Johri A (2016) Next-term student performance prediction: a recommender systems approach. J Educ Data Min 8:1–27

Ungar LH, Zhou J, Foster DP, Stine BA (2005) Streaming feature selection using iic. In: Proceedings of the 10th international conference on artificial intelligence and statistics

Vahdat M, Oneto L, Anguita D, Funk M, Rauterberg M (2015) A Learning analytics approach to correlate the academic achievements of students with interaction data from an educational simulator. In: Conole G et al (eds): 10th International European conference on technology enhanced learning (EC-TEL) 2015. pp 352–366. https://doi.org/10.1007/978-3-319-24258-326

Ward ME, Peters G, Shelley K (2010) Student and faculty perceptions of the quality of online learning experiences. Int Rev Res Open Distrib Learn 11(3):57–77. https://doi.org/10.19173/irrodl.v11i3.867

Zacharis NZ (2015) A multivariate approach to predicting student outcomes in web-enabled blended learning courses. Internet High Educ 27:44–53. https://doi.org/10.1016/j.iheduc.2015.05.002

Zheng J, Chen Z, Zhou C (2013) Applying NN-based data mining to learning performance assessment. In: 13th IEEE joint international computer science and information technology conference (JICSIT). https://doi.org/10.1109/ANTHOLOGY.2013.6784924

Zhou J, Foster D, Stine R, Ungar L (2005) Streaming feature selection using alpha-investing.In: 11th ACM international conference on knowledge discovery in data mining, pp 384–393. https://doi.org/10.1145/1081870.1081914