Early Prediction of Students' Final Grades in a Gamified Course

Amir Hossein Nabizadeh, Daniel Gonçalves, Sandra Gama, and Joaquim Jorge, *Senior Member, IEEE*

***Abstract*—The main challenge in higher education is student retention. While many methods have been proposed to overcome this challenge, early and continuous feedback can be very effective. In this article, we propose a method for predicting student final grades in a course using only their performance data in the current semester. It assists students in analyzing how much effort they need to put into the course to obtain the desired grades while helping course instructors identify student types at the early stages of the course to provide better support for them. Our method, initially clusters students into several groups based on experience points (XP) that they obtain during a semester. Then, we estimate cluster size and balance clusters by generating and adding virtual students to the smaller clusters. Finally, we drop unimportant student attributes using a feature selection technique. We then predict their final grades via three different algorithms. We have compared the performance of our method with other approaches using data collected from a course for nine years, using data collected from 679 students. The results indicate that our method outperformed the others while achieving 78.02% average accuracy only four weeks after starting the course. It shows we can effectively predict final grades, which will potentially enhance students' learning outcomes.**

***Index Terms*—E-learning, gamification, grade prediction, random forest (RF), virtual students.**

I. Introduction

STUDENT success (learning and passing courses) is a main goal in higher education. However, many factors can hinder this goal. According to [16] and [34], one of the reasons for failure is the complexity of the course materials and not having enough support for taking them. Therefore, students may not achieve the grades they desire, causing dissatisfaction, decreasing engagement while increasing drop-out rates (i.e., withdrawal from courses). Hence, it is essential to implement methods that help students learn the required skills and knowledge while assisting them in completing the courses in a timely fashion, and achieving the desired grades.

Thank to information technologies, diverse and abundant student data are available. This makes educational data mining (EDM) and machine learning (ML) methods to become ever more common tools to support students in improving their outcomes. Using EDM and ML techniques and approaches to analyze large amount of learning data has produced interesting, interpretable, useful, and novel information about students [39]. EDM is the application of data mining (DM) techniques on learning activities data produced in educational environments [9], [76]. These methods and techniques are applied to provide different types of support, such as predicting student successes (passing a course) and failures in completing courses [33], [43], [60], predicting how well they perform by analyzing their interactions with e-learning systems [40], [79], or developing early warning methods for monitoring student performance during a course [4].

Early prediction of final grades is one such support. Early feedback helps students in analyzing how much effort they need to put into a course in order to achieve the desired performance. Additionally, it assists course instructors to identify students that show poor performance at the early stages of a course, taking remedial actions (i.e., providing additional learning materials that fit their competency level) and providing better support for them.

Although there are several studies on predicting student grades, such as [26], [71], [78], and [88], in this article, we focus on how early we can predict the student final grades with enough accuracy. To this end, we initially collected student data from a gamified course [58], where learners could continuously increase their grades via XP (see course description in Section IV-A1). We then use the XP obtained during the course to classify them into three four groups (depending on data) via a k-means clustering algorithm. Our motivation to cluster students is to tailor the gamified experience to different groups according to common profiles. Next, we calculate cluster size (i.e., number of students) and balance clusters by generating and adding virtual students to the smaller clusters. Balancing sizes gives equal priority to each cluster and enhances the performance of prediction algorithms [13]. For that, we use the resampling approach mentioned in [13]. Later on, we apply a feature selection technique to drop insignificant and irrelevant student attributes from

Manuscript received 1 May 2020; revised 25 September 2021 and 6 March 2022; accepted 20 April 2022. Date of publication 27 April 2022; date of current version 5 August 2022. This work was supported in part by national funds through Fundação para a Ciência e a Tecnologia (FCT), Portugal, under Grant UIDB/50021/2020 and in part by Project GameCourse under Grant Portugal-PTDC/CCI-CIF/30754/2017. *(Corresponding author: Amir Hossein Nabizadeh.)*

Amir Hossein Nabizadeh is with INESC-ID, 1000-029 Lisbon, Portugal and also with the Medical Informatics Research Center, Institute for Futures Studies in Health, Kerman University of Medical Sciences, Kerman 7616913555, Iran (e-mail: amir.nabizadeh@tecnico.ulisboa.pt).

Daniel Gonçalves, Sandra Gama, and Joaquim Jorge are with INESC-ID and Instituto Superior Técnico, University of Lisbon, 1649-004 Lisbon, Portugal (e-mail: daniel.goncalves@inesc-id.pt; sandra.gama@ist.utl.pt; joaquim.jorge@inesc-id.pt).

Digital Object Identifier 10.1109/TLT.2022.3170494

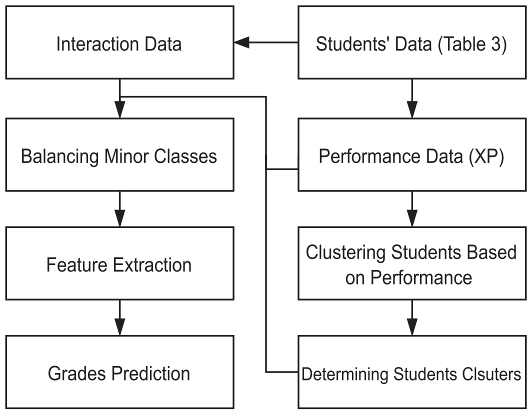


Fig. 1. Steps of our method.

the data. Finally, we predict student grades using naive Bayes (NB), K-nearest-neighbor (KNN), and random forest algorithms. Fig. 1 presents the different steps of our method.

The main highlights of this article are as follows.

1) Providing a review on the recent studies for the grade prediction of students.
2) Collecting and analyzing data from a gamified course called Multimedia Content Production (MCP) taught at *Instituto Superior Técnico*, University of Lisbon over nine years. This data are employed to demonstrate the performance of our prediction approach.
3) Analyzing how early we can identify the type of students in order to provide proper support for them.
4) Testing and comparing the performance of various methods in predicting the students' final grades. These approaches use the standard form of NB, random forest (RF), and KNN algorithms.
5) Using only the data of the students in the current semester (not previous data) for predicting their final grades, which was available to a course instructor.
6) Generating virtual students to balance the students' clusters and enhance the accuracy of the prediction results.
7) Analyzing how early we can have a promising prediction for the final grades.

The rest of this article is organized as follows. Section II introduces related work and Section III clarifies our problem in detail. Our method is described in Section IV. The data description, evaluation methodology, and the results are discussed in Section V. Section VI highlights the limitations of our method while Section VII suggests several research directions to other researchers working in the same area. Finally, Section VIII concludes this article.

II. Related Work

During the last decade, various methods and algorithms were designed and implemented in order to improve students' learning and support them toward their success (passing a course). These methods addressed different problems, such as predicting how well students perform by analyzing their interactions with the system [40], or early warning approaches to inform the students and course instructors [4], or recommending learning materials that students can timely complete [52], [53], [55], [67]. Among all these studies, several aimed at predicting the students' final grades for a course [26], [71], [88]. Students' grades were predicted using different algorithms, methods and techniques, such as using matrix factorization (MF) algorithms [63], [83], a neural network [59], [88], a decision tree (DT) [19], [59], an RF [64], [83]. These algorithms and methods can be divided into two groups. The methods that use only students' performance data (e.g., grades, GPA) for the prediction task and the ones that use also courses' or students' metadata, such as students' demographic data or behavioral patterns.

A. Methods Using Students' Performance Data

There are approaches that used only students' grades for the prediction task. In [72], authors presented a method called MFCTI to predict a student's grade on a course. In MFCTI, courses and students were presented in a latent knowledge space, while a student's grade on a course was modeled as the similarity of their latent representation in the knowledge space. In this method, course-wise effect was taken into account as an extra factor for the grade prediction. As another example, a few MF-based approaches were proposed by Sweeney *et al.* [82] to predict next-term grades of students. The performance of these approaches, including factorization machine [74], SVD [5], SVD-KNN [61] were compared to the several baseline methods. The evaluation results showed that MF-based approaches often outperformed the baselines. Three methods were also presented in [63] to predict a future course grade based on sparse linear and low-rank matrix factorization models. These methods were course-specific matrix factorization, student-specific and course-specific regressions, which were inspired by the content based recommendation methods [68], [72]. The experimental results indicated that the course-specific regression method outperformed the rest of approaches. In this method, students' grades were predicted using a sparse linear combination of the students' grades on the past courses.

Elbadrawy *et al.* [25] presented a domain-aware grade approach that categorized students considering their majors and academic levels. It also categorized courses regarding their levels and subjects. This approach presumed similar biases for the students/courses that were in the same category. Therefore, the biases for each course and student category were modeled within an MF framework. The experiment results showed that this method outperformed the baselines on grade prediction. Huang and Fang [32] compared the performance of four models to predict the final grades of courses for students using three midterm exam grades as students' performance indicators during a semester. In this article, the students' CGPA and their grades on four prerequisite courses (for the target course) were considered as their performance indicators before starting the course. The grades were predicted using the support vector machine (SVM) [11], [50], multilayer perception network (MPN) [65], radial basis function (MBF) network [28], and multiple linear regression (MLR) [57] models, and the best results were obtained by the SVM. The authors showed that the prerequisite courses' grades and CGPA did not significantly enhance the accuracy of the results.

Polyzou and Karypis [64] also predicted students' grades using various methods including RF. To concentrate on the poor performance students, the authors considered the grade prediction as a binary classification task, where two students' groups were identified using their performance (good and poor performance students). Given the historical grading data, the authors extracted various features to represent possible factors of a poor performance student. These features could be classified into three different groups: course-specific, student-specific, and course- and student-specific features. Then, by identifying the features, two base classifiers: a linear SVM and a DT, and two ensemble classifiers: an RF and a gradient boosting (GB) [56] were tested. The results showed that the RF and GB outperformed the base classifiers, while RF had less training time than GB. NB is another method that was used for the prediction task. The standard-based grading (in-semester performance data) along with seven different methods including the NB, KNN, SVM, and four other methods were used for the grade prediction [41]. The feature selection technique was also applied to enhance the prediction accuracy and generalizability of these models and methods. The evaluation results presented that the NB and ensemble models outperformed the other approaches.

Meier *et al.* [44] introduced an algorithm for a timely and personalized prediction of students' final grades considering their scores in early performance assessment, like midterm exams, quizzes, or homework assignments. This algorithm generated predictions for a student whenever the prediction accuracy was sufficient. Such a timely prediction occurred by learning from the previous performance of the students in a course. The performance of this algorithm was compared against five prediction approaches using data collected from a course taught at the UCLA over seven years. The evaluation results showed that this algorithm outperformed the rest of approaches. In [77], a method was presented to address the problem of grade prediction of a student given the previous grades. For that, a collaborative filtering method using baseline adjustment was implemented due to its ability to deal with large and sparse data. In this method, initially a sparse matrix was generated including the students-courses grades. Then, a student grade for a course was predicted by finding a set of students that already took that course and detecting similar courses to the target course that were already selected by a target student. This similarity was estimated using the KNN algorithm with Pearson similarity as a distance measure. The performance of this method was compared with regression models, like linear regression [47] and support vector regression [10]. The experiments using fivefold cross-validation with data from three courses (law, mathematics, and computer science) presented that the introduced method outperformed the other approaches.

Another grade prediction method was presented in [86]. This method generated predictions considering the progressive performance of students. It was designed in two layers—a base and an ensemble predictor layers. In the base layer, several base predictors generated local predictions using the current performance of the students in every academic semester. The input for this layer was a set of relevant courses, which were determined by an introduced clustering method based on a probabilistic matrix factorization [45]. In the second layer, an ensemble predictor predicted the future performance of the students by incorporating the base layer predictions (local predictions) along with the ex-semester ensemble predictions. The performance of this method was compared with three methods on two different courses, and this method obtained the best results.

B. Methods Using Students'/Courses' Metadata

There are studies that besides students' grades, use metadata (whether from students or courses) for the prediction task. For instance, grade prediction of secondary school students was addressed in [19]. For this purpose, the students' data, such as previous courses grades, social, demographic, and other school related data, were collected using questionnaires and school reports. Then, two subjects (Portuguese and mathematics) were modeled with three different data mining goals (binary/5level classification and regression) and using four data mining algorithms, which were SVM, RF [21], neural network [23], and DTs [31]. In this article, various input selections were also explored, such as using or not-using past grades. The evaluation results showed that the high predictive accuracy could be achieved when the grade from the first and/or second school period were known. As another example, a wide variety of regression (e.g., RF, KNN, multilinear regression) and factorization methods (e.g., SVD, factorization machine) were applied by [83] for predicting the next-term students' grades. After testing these methods, a hybrid of the RF algorithm and the MF-based factorization machine had the best performance for the prediction. The key success of this hybrid method was applying a new feature selection method called mean absolute deviation importance (MADImp), which was inspired by [27]. This hybrid method was able to predict grades for both returning and new students, and could be used for all type of courses (i.e., new and existing). It outperformed the other evaluated approaches.

Morsy and Karypis [48] presented a cumulative knowledge-based regression model (CKRM) for the next-term grade prediction of courses. The CKRM modeled a student using a vector including the cumulative knowledge gained by him/her in previous courses while a course was modeled using a vector consisting of the knowledge components provided by the course. To predict a student's grade on a course, CKRM computed a per-course linear model for capturing the required knowledge components for having a well performance on the course. Finally, a dot product between the student's knowledge vector and the computed linear model for the course resulted in the grade prediction. Unlike other methods to predict the next-term grades, like [26] and [73], CKRM considered the effect of cotaken courses on performance of the students. Next-term grade prediction problem was also tackled by Ren *et al.* [73]. For that, they introduced two additive latent effect (ALE) models within an MF framework. Inspired by CKRM, ALE modeled latent factors of a student using the accumulated knowledge from the courses that were already taken by him/her as well as the grades for the courses. In addition, ALE incorporated student and course instructor academic level influences together with the student global latent factor for having a precise grade prediction. The performance of ALE models were compared with five different approaches and ALE models achieved better prediction accuracy.

NB classifier that works based on the Bayesian theorem [75] was also used by various researchers, such as [36], [42], [59], and [70], to predict students' performance and grades. In [59], three supervised data mining algorithms (NB, DT, and neural network) were used on the preoperative assessment datasets for predicting success in a course (i.e., fail/pass). In this article, these algorithms were assessed considering their prediction accuracy and comprehensible characteristics (i.e., easy to understand). The experimental results showed that the NB classifier obtained better prediction results than the other algorithms. In [70], authors also used the NB classifier as well as a few data mining techniques and algorithms, such as multilayer perception algorithm [84], to predict students' performance. For that, they implemented a method based on opted input variables collected by questionnaires. Then, they analyzed and identified the most influencing variables for the grade prediction. The evaluation results showed that the multilayer perception algorithm outperformed the other approaches in grade prediction. Like [59] and [70], NB and KNN algorithms are used for the prediction task with the educational data from secondary schools [3]. In this article, the Rapid Miner software was employed to perform the prediction and evaluate the results. The experimental results indicated that the NB had the best performance.

A time series neural network algorithm was presented in [88] for a behavioral-based grade prediction. This algorithm predicted the total grades of students in a massive open online course (MOOC) [12] as they proceeded on learning materials of a course. The inputs of this algorithm were the prior assessment performance and the prior video watching behavior of the students. For the watching behavior, definite behavioral quantities, such as average playback rates, fraction completed, and number of rewinds, which were correlated with a quiz success (pass a quiz) were estimated from the clickstream of the students [15]. For the evaluation, two versions of this algorithm were assessed using two MOOC datasets. One version was learning only from quizzes (FTSNN) and another one was learning from both quizzes and behavioral features of students (IFTSNN). Finally, these two algorithms were tested against two baseline methods, one considering past performance average and one based on the lasso regression [29]. The introduced methods could outperform the baselines.

Besides the aforementioned groups, we also classified all the mentioned studies within two classes. The first class addresses methods and algorithms that used only the data from current semester (called in-semester data), and the second class refers to the ones that used the data from current and previous semesters for the prediction task. We summarized all these studies and presented in Table I. As shown in this table, only a minor group of studies used in-semester data. So, we found it of interest to focus on this type for the grade prediction task.

TABLE I
PREDICTION METHODS USING CURRENT AND PREVIOUS SEMESTERS' DATA

Classes	References
Current Semester (in-semester)	[19],[3],[41],[42],[88]
Current & Previous Semesters	[72],[82],[83],[63],[25],[32],[64],[48],[73],[26],[36],[59],[70],[44],[77],[86],[18]

III. PROBLEM DEFINITION

In this article, our main goal is to predict the students' final grades at the early stages of a course. It benefits both course instructors and students. On one hand, it assists the course instructors to identify the students' types more precisely as soon as starting the course for providing better support for them. On the other hand, it assists students to realize how much effort they need to put into the course to achieve their desired grades. To this end, in this article, we are faced with three main challenges. We first need to collect and determine the types of data that are significant for the prediction task. We then need to examine what prediction algorithm achieves the highest accuracy results. Finally, we require to analyze how soon a selected algorithm can predict students' final grades with a promising accuracy.

For this purpose, we break down our challenges into a set of smaller ones. Here, usage data are referred by D, a target student is presented by S and XP_s is the experience points of a target student. P depicts the grade prediction task, G_P is the predicted grade of a student, while G_s is his/her actual final grade.

The subchallenges that need to be tackled are the following.

1) Data collection: Collecting students' data, including their performance, behavior, and demographic data.
2) Dropping insignificant data attributes: Ignoring unimportant students' attributes from the collected data. It can be done using feature selection techniques.
3) Predicting students' final Grades: For the prediction task, we intend to initially discretize the grades (0 to 20) and then predict in which performance bucket (grade category) a students' grade would end up. Predicting task can be done using algorithms, such as RF, and NB algorithms. Finally, we select an algorithm with the highest accuracy.
4) Finding how early we can predict the final grades: Assessing the selected prediction algorithm using different chunks of data to find how soon the predictions achieve a promising accuracy. Data chunks are explained in Section V-C.
5) Evaluating the results: We aim to assess the quality of the predictions using offline evaluation approaches and metrics, such as Accuracy and Kappa.

After detailing all subchallenges that need to be tackled, we formalize our main challenge in the form of expression 1

$$P\left(G_p|S,D,XP_s\right) \quad \rightarrow \quad G_p = G_s. \tag{1}$$

IV. PREDICTION APPROACH

In this section, we detail our grade prediction approach, describe what type of data is collected for the grade prediction, how our method generates the students' clusters and creates the virtual students. In addition, we explain how it uses a feature selection technique to drop the insignificant data attributes and predicts the final grades of the students. Fig. 1 shows a general view of our approach and the steps, which it takes to predict the students' final grades.

A. Data Collection

In this article, we use the data collected from the MCP course since the academic year 2010–2011 until 2018–2019. The MCP is selected since in this course students' grading is made in a continuous manner throughout the semester (continuous accrual of XP), which makes the prediction model feasible. The collected data do not include the previous data of the students, such as their CGPA, or their data from other courses. It only includes the data from the current semester of the students, which are available for the course instructors. To this end, we collect different type of data from students, like their XP, attendance, type of actions, and their frequencies. In Section IV-A1, the MCP course is explained briefly.

1) Multimedia Content Production Course: MCP is a gamified M.Sc. course in Information Systems and Computer Engineering field at Instituto Superior Tecnico (IST), Portugal. It employs a blended learning approach in such a way that students attend both practical labs and theoretical classes while participating in discussions and completing online projects and assignments in Moodle[1]. Theoretical classes are to cover multimedia concepts, such as multimedia standards, file formats, production techniques and editing. Lab sessions are to explore concepts and use tools for images, audio, and video manipulation [7].

In the MCP course, students attend parallel lectures in two university campuses, and the course is conducted identical and synchronized across both campuses having the same Moodle platform. This course is in English and it is only conducted during the second semester of each academic year (semester duration: $\approx$ 17 weeks), having two theoretical classes and one practical lab per week.

In the MCP, students are graded using the experience points (called XP), considering the course activities they complete. These activities include lab assignments, quizzes, a multimedia presentation, and several other activities, such as attending lectures or finding bugs in class slides. Some of these activities result in gaining badges. For instance, students can get the *Lab Master* badge if they participate in a specified amount of labs. In addition, this course includes a Skill Tree, which is a precedence tree. In this tree, every node refers to a thematic task that results in XP upon completion. Initially, six nodes are unlocked and posterior nodes could be unlocked when the anterior ones are completed [8].

Students start the MCP having the default XP (i.e., 500 XP) while obtaining every 1000 XP by a student results in gaining one experience level. The topmost level is 20 while 10 is the required level to pass the course. At the end of a semester, levels are converted to a 20-point grading system, which is the grading standard of the University of Lisbon. Fig. 2 shows the course leaderboard, which includes the students' names, XP, levels, and badges. It allows the students to compare themselves with the others while motivates them to work harder for raising their grades.

[1] [Online]. Available: https://www.moodle.org

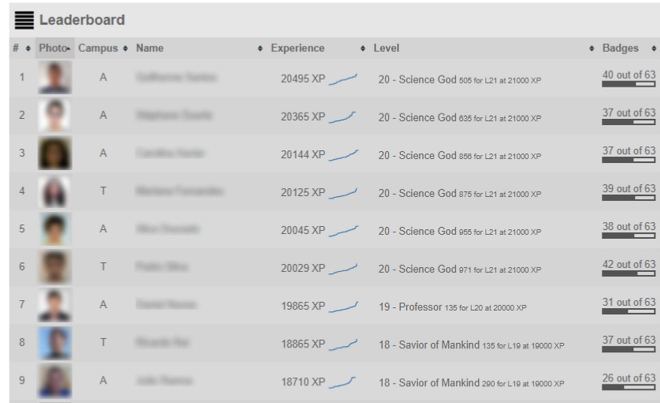


Fig. 2. MCP leaderboard [6].

In the nongamified MCP course, students were graded based on regular quizzes, lab assignments, online participation, and a final exam. Nonetheless, the students often concentrated on the major assessment elements, and ignored the online participation. In the gamified MCP, to further captivate students and engage them with the course, regular quizzes (occurring often every other week) and collective achievements (obtained considering the course participation throughout a semester) are used. It allows to continuously study and analyze how students' behavior and performance are affected by the gamification elements.

B. Clustering Students

After collecting all the required data, we use the students' XP to cluster them into various groups. This clustering enables us to later on tailor the gamified experience in the way that best suits different students' groups. XP data are used for clustering since in the MCP course a student grade is calculated using the XP. Hence, we can conclude that the XP is the most informative data for identifying the type of students. Therefore, we have used the students' XP along with a k-means clustering algorithm to classify them into different groups. The k-means algorithm is selected due to its efficiency and simple implementation [51].

We initially cluster the students into four groups. The reason for generating four clusters is already justified in [8] and [67]. In the case that there is a cluster with a small number of students (i.e., 1 or 2 students), we generate three clusters (i.e., good, average, and naive students). Algorithm 1 presents our clustering algorithm that works based on a k-means algorithm.

C. Generating Virtual Students

After clustering the students, we have noticed that in some cases the clusters are highly imbalanced. Regarding the fact

Algorithm 1: Our Clustering Algorithm.

Input: All students S_a, XP data.
Output: Students Clusters.
$C_a \leftarrow$ Divide S_a into 4 groups using XP data via a k-means.
if Size of a Cluster in $C_a <= 2$ **then**
  $C_a \leftarrow$ Divide S_a into 3 groups using XP and a k-means.
Return C_a

Algorithm 2: Virtual Students Generation.

Input: All students S_a, All data D, C_a from Alg 1.
Output: Balanced clusters using virtual students.
for *(1 to i (number of clusters))* **do**
  $L_i \leftarrow$ Size C_{a_i};
$L_b \leftarrow$ Max L_i;
$L_s \leftarrow$ Min L_i;
for *(1 to i (number of clusters))* **do**
  $R_i \leftarrow round(\frac{L_i}{L_b})$; ▷R is a ratio that shows how much a cluster needs to enlarge.
$n \leftarrow L_s - 1$;
$Class_B \leftarrow$ SMOTE $(D, R, dist ='' Euclidean,'' k = n)$;
Return $Clus_B$;

that some of the predictive models are highly sensitive to this matter [2], this problem needs to be tackled. Hence, we balance the clusters to give equal priority to each cluster and increase the performance of prediction models [13]. For this purpose, we have used a resampling approach, which is explained in [13]. In [13], a utility-based learning (UBL) tool is proposed, which handles imbalanced classification issues using a synthetic minority over-sampling technique (SMOTE) [17]. SMOTE technique enhances the prediction accuracy over the minority clusters by generating synthetic instances of them. An instance is generated for a minor cluster using *k* members of that cluster. Members are selected randomly taking into account the required amount of over-sampling.

In this article, the selected number of members is determined as 1 size smaller than the size of the smallest cluster (i.e., *n* = smallest cluster - 1). This size is opted since it enables us to take as many members as possible into account for generating more realistic instances while it makes the task feasible (larger *n* causes errors). In addition, in order to generate less artificial instances (i.e., virtual students), we only generate the virtual instances and add them to the small clusters to have similar size as the largest cluster. Fig. 3 presents the running flow of our balancing approach. Table II shows how the size of each cluster was changed before (B) and after (A) balancing.

To balance the clusters, we have considered all data that we have collected from the students. Table III classifies the collected attributes that were used to generate new instances for the minority clusters. Although the number of attributes that were used for each year was around 2000, all could be categorized in one of the mentioned clusters in Table III.

Algorithm 2 presents our solution to balance the generated clusters using virtual students. This algorithm works based on the students' clusters generated by the Algorithm 1 and an

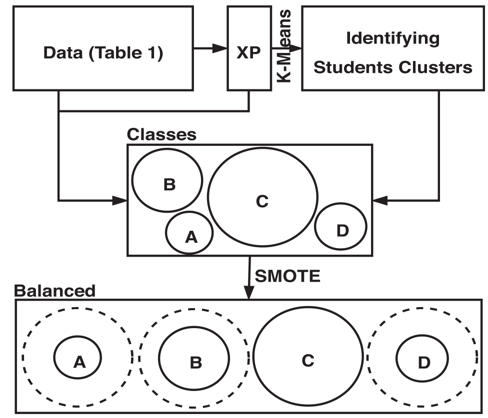


Fig. 3. Running flow of our balancing approach.

TABLE II
SIZE OF EACH CLUSTER BEFORE (B) AND AFTER (A) BALANCING FOR EACH YEAR

	2010-11		2011-12		2012-13		2013-14		2014-15		2015-16		2016-17		2017-18		2018-19	
Clusters	**B**	**A**	**B**	**A**	**B**	**A**	**B**	**A**	**B**	**A**	**B**	**A**	**B**	**A**	**B**	**A**	**B**	**A**
One	14	14	23	23	13	26	22	22	5	25	29	29	24	24	52	52	26	52
Two	14	14	9	27	11	22	29	29	26	26	15	30	29	29	18	54	42	42
Three	7	14	9	27	7	21	8	32	16	32	32	32	15	30	25	50	28	56
Four	-	-	11	22	23	23	17	34	14	28	9	36	22	22	22	44	13	39

TABLE III
DATA ATTRIBUTES

Attributes	Remark	Value
XP	Quizzes, Labs, Multi Media Presentation, Quest	0,50,100,150
Badges	Collected Badges	0,1,2,3
Attendence	Attending Classes	Binary
Gender	Student's Gender	Binary
Activity	Activity Frequencies	Frequency

SMOTE technique. In Algorithm 2, the $Clus_B$ refers to a balanced dataset.

D. Data Attribute Selection

Balancing the clusters of datasets results in creating large data having many attributes. To identify, which attributes are important for the grade prediction task, we have used an attribute selection technique. Attribute selection is a technique to determine the relevant and informative data attributes while discarding the redundant and the irrelevant ones [22]. It has several benefits. First, the irrelevant attributes might cause the over-fitting problem [30]. For example, in our case, if we use the students' ID as an input variable for the grade prediction, an over-tuned machine learning (ML) algorithm might conclude that the final grade can be determined by the students' ID. Second, it has impact on the interpretability (easy to understand) of the obtained results of a predictive model [24]. Attribute selection can also enhance the learning speed of inductive learner algorithms (e.g., NB), reduce their storage volume, and decrease the noise caused by redundant and irrelevant attributes [46].

All attribute selection algorithms fall in two categories: *filter* and *wrapper*. *Wrappers* assess attributes by means of estimated

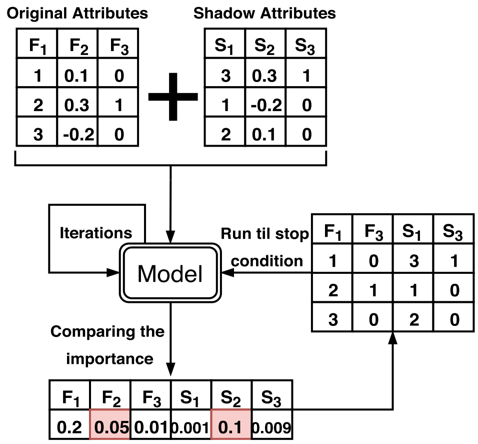


Fig. 4. Running flow of the Boruta algorithm.

accuracy provided by a target algorithm. *Filters* apply general data characteristics for attribute assessment, and it works separately from any learning algorithm [22]. In this article, we have used a Boruta algorithm [38], which is applied in several Kaggle competitions for selecting the important attributes [37].

Boruta is a *wrapper* that employs an RF algorithm. It iteratively compares the importance of the original attributes with the shadow ones, which are generated by shuffling the original attributes. The original attributes that have lower importance than the shadow ones are dropped while the higher ones are accepted as the confirmed attributes. The shadow attributes are regenerated in each iteration. Boruta stops when no confirmed attributes are left or the stop criteria is met (i.e., max iteration) [38]. The running flow of the Boruta algorithm is shown in Fig. 4.

E. Grade Prediction

After identifying the informative attributes and dropping the irrelevant ones from the data, we have used the data to predict the final grades of the students. For that, we have employed the regular form of three predictive algorithms, which are NB, RF, and KNN.

1) *Naive Bayes*: NB is a classifier, which works based on the Bayesian theorem [75] and a naive assumption that claims all data attributes are independent [87]. It has several benefits. First, it has a simple design process due to using a very intuitive technique (i.e., not having various parameters to be tuned). Second, it does not need much data to train the model. Third, it is computationally fast to make decisions. Finally, since it returns probabilities as results, it is more straightforward to use the results for a wide range of tasks than if a random scale was employed [80].
2) *Random Forests*: They are a combination of DT predictors (i.e., an ensemble learning approach). The trees are depending on the values of random vectors, which are sampled independently while having the same distribution [14]. RF are widely used by researchers and have various advantages. Some of the advantages are: having high classification accuracy, ability to model complex interactions among predictor attributes, dealing with data having numerous attributes, reducing the over-fitting issue, ability to train the trees in parallel, and finally its insensitivity to emissions in data because of random sampling [66], [81].
3) *K-Nearest Neighbor*: It is one of the main and simplest classification approaches, and should be one of the first choices when the data distribution is unknown [62]. KNN classifies an instance using a majority vote of its K neighbors. Therefore, it works by estimating the distance among the instances using a distance function [41]. Besides its simplicity, KNN has other advantages. First, it is easy to implement and debug due to having a transparent process. Second, there are several techniques to reduce the noise that only work for KNN, which result in enhancing the classification accuracy. Finally, in a situation that the output clarification is needed, KNN can be highly effective by analyzing the neighbors [20].

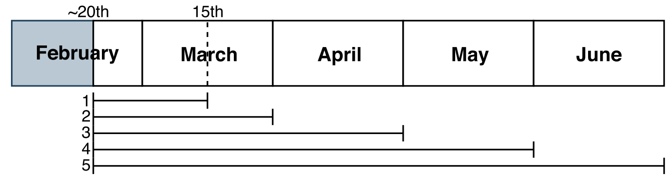


Fig. 5. Datasets used for grade prediction. In MCP, students became active (getting XP) around February 20th.

To predict the students' final grades using the mentioned algorithms, we first categorized the final grades of the students. In our university, the final grades are calculated based on a 20-points grading system. To this end, after our discussion with the MCP course instructors, we came up with the following categories:

$$\textit{For Each Final Grade } (G) = \begin{cases} \textbf{if} \;\; 18 \leqslant G \;\leqslant 20 \rightarrow A \\ \textbf{if} \;\; 14 \leqslant G \;< 18 \rightarrow B \\ \textbf{if} \;\; 10 \leqslant G \;< 14 \rightarrow C \\ \textbf{if} \;\;\; 0 \leqslant G \;< 10 \rightarrow D \; (\textit{Failed}). \end{cases}$$

For the grade prediction, our task is to predict the label of each category (i.e., A, B, C, or D). The data that are used to predict these labels are detailed in Section V. As a brief explanation, for each academic year we use five datasets, collected until the middle and end of March, end of April, May, and June (see Fig. 5). It allows us to analysis how early we can predict the final grades having a promising accuracy.

V. Evaluation and Discussion

In order to assess the performance of our prediction approach, we used the data collected from the MCP course since the academic year 2010–2011. For each academic year, we have considered five subsets, accounting for a different time span (columns 4 to 8 in Table IV). Fig. 5 shows a graphical view of these subsets (datasets). Using these datasets enabled us to monitor which approach consistently worked better than the rest of approaches.

TABLE IV
MCP DATASETS STATISTICS

			Data Sparsity				
Years	# Students	1st Taken XP	Mid March	End March	End April	End May	End June
2010-11	35	15 Feb	0.69	0.68	0.68	0.67	0.68
2011-12	52	17 Feb	0.73	0.71	0.70	0.69	0.70
2012-13	54	18 Feb	0.64	0.63	0.64	0.64	0.67
2013-14	76	19 Feb	0.75	0.72	0.72	0.71	0.71
2014-15	61	18 Feb	0.76	0.73	0.73	0.73	0.73
2015-16	85	16 Feb	0.75	0.74	0.73	0.74	0.74
2016-17	90	20 Feb	0.74	0.71	0.72	0.72	0.72
2017-18	117	22 Feb	0.76	0.75	0.75	0.72	0.74
2018-19	109	18 Feb	0.73	0.71	0.72	0.71	0.73

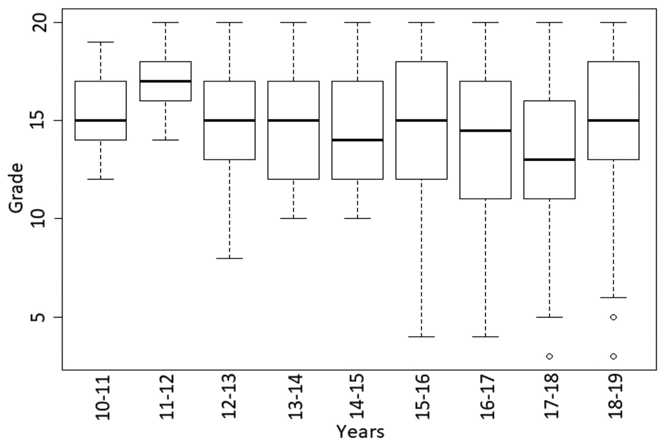


Fig. 6. Students' final grades in different years.

The details of these datasets are presented in Table IV. As presented in this table, the sparsity of datasets for all years was around 70%. The third column of this table presents when the first XP was obtained (students became active). In this article, we ignored those students that obtained less than 3000 XP or their final grades were less than 3.

In Fig. 6, we compared the final grades of the students in different years. As shown in this figure, the average grades were around 15. In addition, there were several students that failed the course mostly after the year 2015–2016. Fig. 7 presents the average XP that is obtained by each cluster every year. Three groups of students were only identified in 2010–11. In 2011–12, there was no limitation for obtaining XP, therefore the average XP was higher than the other years. The same pattern can be seen in Fig. 6.

Fig. 8 presents the density of students' activities during a semester. For that, every single activity by a student is considered one, such as posting or opening a page. At the beginning of a semester, students were exploring the course, therefore, we had a high density while in April the density dropped due to having several holidays in this month. At the end of all semesters, the density is raised again since students were trying to enhance their grades, which influenced the density of their activities.

A. Evaluation Metrics

To evaluate prediction methods, we predicted the grade categories of the students (i.e., A, B, C, or D) using various

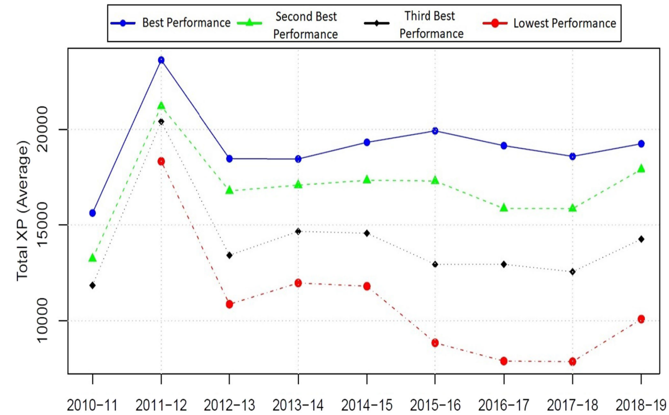


Fig. 7. Average clusters' XP in each year.

datasets. As mentioned already and also shown in Fig. 5, we used five subsets of the dataset for each academic year, accounting for a different time span. After the grade prediction, the results were compared with the actual grades of the students and the accuracy was calculated. The overall accuracy was defined as the total number of correct predictions divided by the total number of predictions, as presented in the following [35]:

$$Overall\ Accuracy = \frac{\sum_{i=1}^{K} n_i}{n}. \tag{2}$$

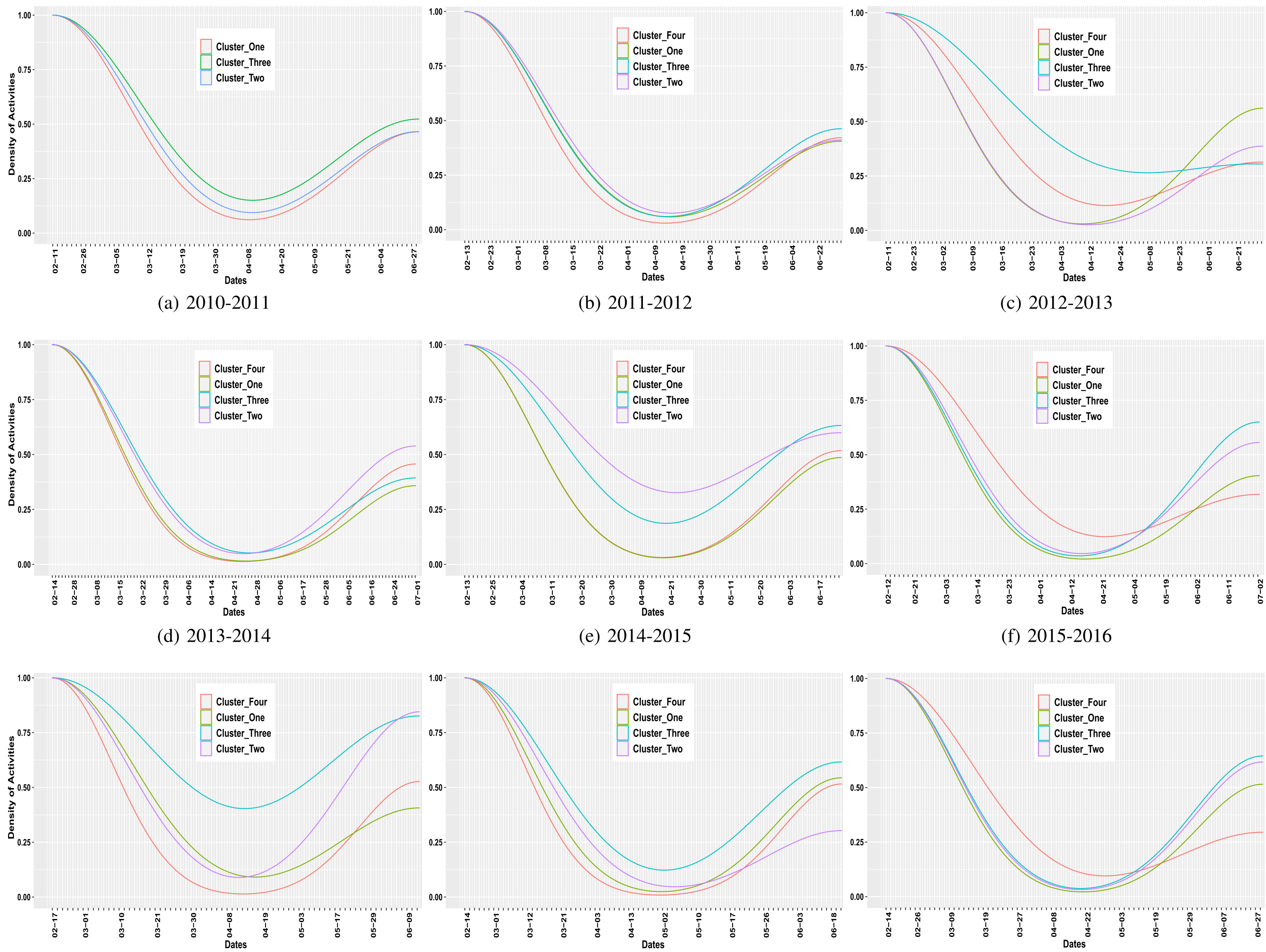
We also estimated the Kappa statistic (Kappa coefficient) for the results. It basically shows the reliability of the prediction results (not obtained by chance), which was calculated using the following [85]:

$$\hat{K} = \frac{P_o - P_e}{1 - P_e} \tag{3}$$

In (3), P_o indicates observed agreement among actual grades and the predicted ones while P_e is the probability of chance agreement between the predicted and the actual grades. In Kappa, 1 presents perfect agreement while 0 indicates agreement equivalent to chance.

B. How Early for Grade Prediction

Besides predicting the students' final grades, we also need to answer one of the main questions of this article, which is: "How early can we predict the students' final grades having a promising accuracy?" To answer this question, we decided to analyse the students' performance through each semester and check when their performance became stable and their ranks got steady. Our assumption by that was as soon as having the stability in the students' ranks and performance, our grade prediction become less noisy and more accurate. To this end, we estimated the students' ranks through the entire semesters using the accumulated XP obtained by each student. The XP was selected since the students' final grades were calculated based on it.



(a) 2010-2011 (b) 2011-2012 (c) 2012-2013

(d) 2013-2014 (e) 2014-2015 (f) 2015-2016

(g) 2016-2017 (h) 2017-2018 (i) 2018-2019

Fig. 8. XP acquisition density by each group for all semesters. The reasons for clusters are explained in Section IV-B.

Fig. 9 presents our analysis for the students' ranks through each semester. In this figure, the *x*-axis present the duration of each semester while the left *y*-axis indicate the students' ranks. In each year, a rank of a student is plotted as a line. For all years, the students were clustered into 4 groups (reason is explained in Section IV-B) except 2010–2011 that we only had three clusters (clusters are presented with different colors). The average of students' final grades of each cluster are mentioned on the right *y*-axis of plots. As presented in Fig. 9, by enhancing the students' ranks their average final grades were also getting better.

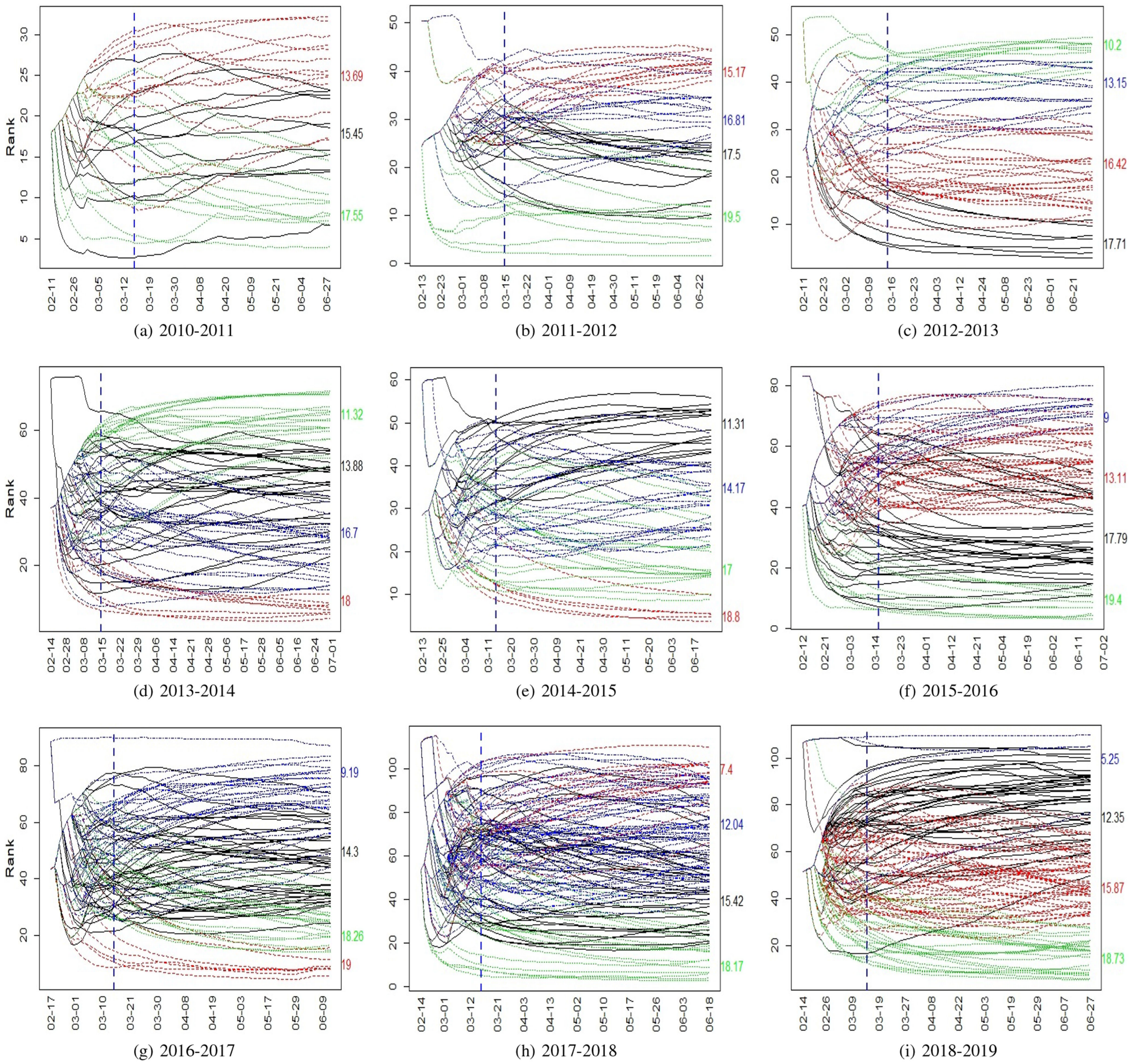
For each semester, we considered similar initial ranks for all students. Their ranks got diverse by obtaining XP. As shown in Fig. 9, in some years, there were a few students that their initial ranks were different than the others, such as the year 2011–2012 that there were a few students with the initial rank 50. These were the ones that registered and started the course later than the others.

In Fig. 9 (like in [49]), we see that the students' ranks varied a lot in the first month of each semester (up to the vertical dash-line in each plot). It can be due to the reason that the students were faced a new course and did not know how to study it. So, they could not perform enough tasks, and subsequently not obtained enough points (XP). Interestingly, we see that the ranks became significantly stable after the first month (after middle of March). In other words, before the middle of March (left side of the vertical dash-lines) the students' ranks were fluctuated a lot while after the first month (right side of the dash-lines) there was not much variation and the ranks got notably steady. Hence, we expect to have a promising prediction around the middle of March.

C. Prediction Results

As mentioned in the previous sections, our method works based on three algorithms to predict the final grades of the students, which are: NB, RF, and KNN. In order to be sure about the quality of our method, we compared its performance with the following methods.

1) *Baselines:* In these methods, the NB, RF, and KNN algorithms were used on the datasets to predict the final grades of the students. Here, the data were not balanced and the feature selection technique was not used.
2) *Confirmed Features:* In these methods, initially the irrelevant and unimportant data features were dropped using a



(a) 2010-2011 (b) 2011-2012 (c) 2012-2013

(d) 2013-2014 (e) 2014-2015 (f) 2015-2016

(g) 2016-2017 (h) 2017-2018 (i) 2018-2019

Fig. 9. Students' ranking during nine semesters. The right *y*-axis of each figure presents the average final grade of each students' groups. We often have four student groups while for the year 2010–2011 we only have three groups.

feature selection technique, and then the final grades were predicted using the NB, RF, and KNN algorithms.

3) *Reverse Methods:* These methods took the same steps as our method while they first used the feature selection technique and then balanced the datasets but in our method we first balanced the data and then used the feature selection technique.

All the mentioned prediction methods were generated using 75% of the data as training set and 25% as test, and their prediction ability was tested using a tenfold cross-validation technique [54]. The accuracy results are presented in Table V, and the best results are bold. As shown in this table, the best results were mostly made by our method. Our justification for outperforming the reverse methods is that our method used more data to balance the datasets while the reverse methods balanced the data after dropping the attributes that were detected as the irrelevant ones.

In this article, our main focus is on the early prediction of the final grades, hence, the mid march results are more significant than the other results for us. Although our method outperformed the baselines, confirmed features, and reverse methods, it is still not clear which version of our method (i.e., RF-based, NB-based, or KNN-based) had the best performance using the mid march data. To this end, we estimated the average accuracy of the results of these three versions of Our Method and presented in Table VII. As presented in this table, the average accuracy of the RF-based algorithm was 78% while it was descended to 72.7% and

74.7% using the NB-based and KNN-based algorithms, respectively. Therefore, our method based on the RF had a better performance than the ones using the NB and KNN algorithms.

In Table V, we also see that the accuracy results (for the mid march) for the years 2010–2011 and 2011–2012 were higher than the rest of the years. It can be due to the reason that the initial version of the course was without the Skill Tree (explained in Section IV-A1). So, it was more simple and straight forward (less confusing) for the students to learn and collect XP, while enabling us to identify their types having a better accuracy in comparison with the rest of the years. After 2011–2012, the results slightly got worse (for the mid march) due to adding different type of learning activities and tasks to the course, and subsequently made it more complicated for the students to learn and obtain XP.

In order to be sure about the reliability of the obtained accuracy, we presented the Kappa statistic results in Table VI. As shown in this table, our method outperformed the other ones for almost all years. Like the accuracy results, we estimated the average Kappa of our method and presented in Table VII. The average Kappa (for the mid march) using the RF-based algorithm was 62.9% while for the NB-based and KNN-based algorithms were 57.6% and 56.2%, respectively. Again, our method based on RF outperformed the other ones.

VI. Discussion

In the previous section, we have presented that our method outperformed the other approaches in predicting the final grades of the students. In spite of being successful to predict the grades with a promising accuracy, in this section, we aim to highlight the limitations of this article.

A. *Static Students' Clusters*

In this article, we often considered four clusters among the students and predicted their final grades based on that. Although other studies, such as [8] and [67] also identified four students' groups, four number of clusters can not be highly promising, since in some years, such as 2017–18 and 2018–19, five also can be a good candidate for the number of clusters. In addition, in our case, the number of clusters can vary for different years since the number of students increased during the years while new tasks and activities were added to the course in different semesters.

In addition, having a static number of clusters might influence the accuracy of the prediction results. It is of interest to consider the dynamic number of clusters during a semester.

B. *Additional Student Data*

Although in this article we only used the data from the current semester of the students, it is significant to analyze how various type of data, such as the students' CGPA or their grades on the past main courses, influence the accuracy of the grade prediction.

C. *Experiment Size*

A study like ours needs an appropriate sample size for the evaluation. A small sample might not be challenging enough to show all drawbacks of the method while a large sample might be costly both in terms of money and time and not necessary. Although we had datasets from nine years, our article could benefit from datasets having more students, which also could be useful to assess the evaluated methods in the case of scalability.

D. *Type of Course*

In this article, we have used the data from the MCP course. In this course, students' grading is made in a continuous manner throughout a semester, which makes our prediction model possible. However, this course is very different from the traditional ones, where a student grade is made using a few midterms tests, some graded labs, and a final exam. It is of interest to implement a model that is able to predict the final grades of the students using the data from the traditional type of courses.

VII. Design Implications

Our method as well as the evaluation results and the mentioned drawbacks propose additional research directions. The most significant ones are the following.

Student learning time: In spite of using various type of data, which is explained in Table III, we did not apply students' learning time to predict their final grades. Students' learning time is significant, since different students might collect similar XP from the same task but it might take different amount of time for them. So, it clarifies how good are the students in learning the course.

Personalization: In this article, we predicted the students' final grades having a promising accuracy at the early stages of the course. It is of interest to recommend various learning materials covering different level of difficulties based on the predicted grades for the students.

Gender is not significant: In this article, the students' gender was always one of the first data attributes that was detected by the feature selection technique as an irrelevant and unimportant students' feature. So, the researchers might ignore it for the grade prediction task.

Data availability: In e-learning, the students' data often can not be released due to the privacy concern. Therefore, there is always the problem of lack of public data, which makes it difficult to compare the methods proposed by different researchers. Having such datasets, in particular the large scale ones, can promote the researches in this field significantly.

Supplementary data: One of our tasks in this article was to predict the final grades using only the students' data in the current semester. It is of interest to collect and apply various type of data, such as understanding degree of the students, their learning style [69], objectives, motivation, or the difficulty level of the learning materials, to predict the final grades of the students.

TABLE V
ACCURACY RESULTS OF PREDICTION METHODS

	Baselines					Confirmed Features					Reverse Method					Our Method					
	Mid March	End March	End April	End May	End June	Mid March	End March	End April	End May	End June	Mid March	End March	End April	End May	End June	Mid March	End March	End April	End May	End June	
2010-2011	0.677	0.73	0.76	0.776	0.834	0.771	0.841	0.81	0.835	0.905	0.762	0.874	0.739	0.903	0.926	**0.839**	0.839	**0.83**	0.846	**0.934**	Random Forest
2011-2012	0.702	0.722	0.788	0.899	0.931	0.739	0.834	0.824	0.909	0.956	0.811	0.828	0.822	0.932	0.964	**0.844**	**0.89**	**0.888**	0.924	0.961	
2012-2013	0.551	0.605	0.611	0.652	0.845	0.608	0.702	0.713	0.805	0.939	0.835	0.656	0.663	0.78	0.935	0.796	**0.733**	**0.789**	**0.874**	0.926	
2013-2014	0.56	0.624	0.674	0.733	0.89	0.664	0.678	0.761	0.784	0.939	0.717	0.748	0.773	0.769	0.948	**0.776**	**0.802**	**0.852**	**0.833**	**0.954**	
2014-2015	0.54	0.655	0.667	0.698	0.869	0.658	0.659	0.758	0.785	0.9289	0.732	0.782	0.739	0.838	0.947	**0.809**	**0.837**	**0.793**	**0.849**	**0.948**	
2015-2016	0.535	0.574	0.673	0.746	0.888	0.571	0.644	0.729	0.836	0.965	0.774	0.679	0.689	0.881	0.969	**0.78**	**0.741**	**0.763**	**0.842**	**0.972**	
2016-2017	0.475	0.521	0.68	0.77	0.915	0.466	0.571	0.6944	0.798	**0.967**	0.727	0.663	0.708	0.826	0.95	**0.735**	**0.709**	**0.796**	**0.867**	0.947	
2017-2018	0.539	0.581	0.613	0.723	0.899	0.523	0.583	0.666	0.78	**0.96**	0.688	0.81	0.764	0.778	0.958	**0.758**	**0.816**	**0.808**	**0.825**	0.954	
2018-2019	0.566	0.584	0.61	0.78	0.91	0.596	0.64	0.631	0.821	**0.979**	**0.691**	0.681	0.766	0.86	0.977	0.685	**0.741**	**0.82**	**0.86**	0.962	
2010-2011	0.506	0.525	0.53	0.621	0.664	0.727	0.681	0.807	0.793	0.77	0.708	0.672	0.625	**0.884**	0.828	**0.876**	**0.836**	**0.795**	0.81	**0.837**	Navie Bayes
2011-2012	0.721	0.702	0.72	0.814	0.844	0.784	0.786	0.803	0.877	0.954	0.794	0.76	0.818	**0.903**	0.926	**0.868**	**0.88**	**0.868**	0.893	**0.959**	
2012-2013	0.416	0.456	0.489	0.587	0.682	0.541	0.578	0.649	**0.758**	**0.929**	**0.71**	0.549	**0.701**	0.744	0.863	0.709	**0.63**	0.691	0.746	0.884	
2013-2014	0.491	0.548	0.634	0.678	0.764	**0.636**	0.609	0.752	**0.787**	0.913	0.55	0.68	0.741	0.747	0.923	0.586	**0.687**	**0.816**	0.754	**0.952**	
2014-2015	0.482	0.569	0.67	0.688	0.746	0.625	0.71	0.741	0.76	0.9284	0.577	0.716	0.787	0.83	0.923	**0.767**	**0.798**	**0.811**	**0.853**	**0.942**	
2015-2016	0.605	0.636	0.639	0.71	0.795	0.558	0.582	0.643	0.77	0.918	0.675	0.632	0.714	0.788	0.898	**0.702**	**0.75**	**0.718**	**0.844**	**0.936**	
2016-2017	0.374	0.44	0.559	0.705	0.768	0.461	0.517	0.6947	0.82	0.932	0.72	0.551	0.667	0.817	**0.936**	**0.7315**	**0.682**	**0.73**	**0.86**	0.91	
2017-2018	0.387	0.358	0.459	0.629	0.733	0.591	0.601	0.513	0.786	**0.935**	0.505	**0.635**	0.576	0.711	0.933	**0.607**	0.546	**0.695**	**0.817**	0.919	
2018-2019	0.534	0.54	0.544	0.682	0.728	0.602	0.611	0.612	0.803	**0.928**	0.682	0.675	0.684	**0.855**	0.9186	**0.703**	**0.715**	**0.76**	0.847	0.924	
2010-2011	0.533	0.542	0.636	0.766	0.761	0.692	0.781	0.851	0.845	0.847	0.738	0.816	0.774	0.855	0.851	**0.7545**	**0.819**	**0.854**	**0.872**	**0.906**	KNN
2011-2012	0.713	0.773	0.773	0.823	0.832	0.783	0.816	0.853	0.9095	0.946	0.813	0.799	0.808	0.91	0.949	**0.82**	**0.845**	**0.88**	**0.941**	**0.952**	
2012-2013	0.585	0.625	0.616	0.607	0.652	0.626	0.687	0.693	0.755	0.911	0.764	0.717	0.693	0.777	**0.919**	**0.794**	**0.776**	**0.736**	**0.821**	0.885	
2013-2014	0.512	0.553	0.609	0.655	0.667	0.67	0.622	0.77	0.791	0.912	0.677	**0.692**	0.788	0.773	**0.939**	**0.682**	0.663	**0.839**	**0.834**	0.934	
2014-2015	0.474	0.483	0.543	0.61	0.641	0.698	0.713	0.736	0.784	0.924	0.711	0.73	0.809	0.853	**0.939**	**0.789**	**0.794**	**0.813**	**0.859**	0.931	
2015-2016	0.503	0.53	0.542	0.527	0.552	0.593	0.641	0.73	0.75	0.883	0.766	0.64	0.735	**0.844**	**0.899**	**0.776**	**0.733**	**0.736**	0.828	0.897	
2016-2017	0.375	0.436	0.407	0.494	0.509	0.476	0.505	0.701	0.813	**0.929**	**0.746**	0.525	0.736	0.821	0.928	0.7317	**0.659**	**0.786**	**0.876**	0.899	
2017-2018	0.513	0.518	0.551	0.61	0.643	0.572	0.593	0.613	0.798	0.923	0.664	**0.794**	0.713	0.799	**0.929**	**0.679**	0.729	**0.722**	**0.82**	0.924	
2018-2019	0.501	0.534	0.557	0.615	0.659	0.582	0.607	0.644	0.806	0.881	0.675	0.653	0.732	**0.872**	0.942	**0.698**	**0.704**	**0.777**	0.848	**0.943**	

The best results were bold.

TABLE VI
KAPPA RESULTS OF PREDICTION METHODS

	Baselines					Confirmed Features					Reverse Method					Our Method					
	Mid March	End March	End April	End May	End June	Mid March	End March	End April	End May	End June	Mid March	End March	End April	End May	End June	Mid March	End March	End April	End May	End June	
2010-2011	0.131	0.266	0.332	0.364	0.523	0.444	0.651	0.522	0.547	0.74	0.511	**0.676**	0.38	**0.819**	0.83	**0.704**	0.524	**0.573**	0.722	**0.861**	Random Forest
2011-2012	0.351	0.413	0.564	0.784	0.85	0.444	0.643	0.633	0.804	0.909	0.559	0.657	0.643	**0.856**	0.928	**0.658**	**0.779**	**0.773**	0.832	**0.923**	
2012-2013	0.264	0.365	0.376	0.448	0.757	0.376	0.526	0.553	0.703	0.906	**0.741**	0.47	0.508	0.678	0.903	0.679	**0.584**	**0.693**	**0.813**	**0.891**	
2013-2014	0.215	0.33	0.418	0.529	0.807	0.417	0.45	0.582	0.632	0.896	0.482	0.566	0.613	0.596	0.915	**0.554**	**0.669**	**0.751**	**0.71**	**0.927**	
2014-2015	0.267	0.455	0.47	0.518	0.793	0.465	0.469	0.619	0.658	0.889	0.591	0.671	0.608	**0.757**	0.92	**0.709**	**0.753**	**0.691**	0.722	**0.922**	
2015-2016	0.315	0.372	0.52	0.63	0.835	0.377	0.481	0.607	0.762	0.948	0.601	0.528	0.54	**0.828**	0.954	**0.617**	**0.608**	**0.647**	0.771	**0.959**	
2016-2017	0.237	0.312	0.55	0.682	0.881	0.243	0.395	0.575	0.723	**0.954**	0.596	0.543	0.592	0.768	0.933	**0.613**	**0.609**	**0.711**	**0.822**	0.929	
2017-2018	0.2	0.273	0.329	0.546	0.834	0.235	0.317	0.462	0.65	0.937	0.496	**0.715**	0.659	0.664	**0.942**	**0.593**	0.714	**0.717**	**0.733**	0.936	
2018-2019	0.349	0.377	0.418	0.673	0.866	0.401	0.468	0.455	0.735	**0.969**	**0.5464**	0.533	0.661	0.795	0.968	0.537	**0.627**	**0.74**	**0.799**	0.945	
2010-2011	0.245	0.235	0.241	0.343	0.42	0.334	0.408	**0.586**	0.541	0.476	0.484	0.415	0.315	**0.801**	**0.709**	**0.7546**	**0.549**	0.5858	0.677	0.691	Navie Bayes
2011-2012	0.369	0.349	0.408	0.618	0.667	0.5216	0.57	0.609	0.752	0.904	0.515	0.52	0.634	**0.805**	0.852	**0.71**	**0.758**	**0.736**	0.78	**0.917**	
2012-2013	0.154	0.211	0.268	0.401	0.53	0.352	0.385	0.474	**0.649**	**0.891**	0.559	0.339	**0.576**	0.638	0.806	**0.564**	**0.466**	0.552	0.638	0.835	
2013-2014	0.207	0.292	0.404	0.471	0.621	0.412	0.378	0.585	**0.653**	0.864	0.367	0.489	0.586	0.6	0.878	**0.398**	**0.509**	**0.702**	0.612	**0.924**	
2014-2015	0.191	0.329	0.481	0.511	0.596	0.405	0.531	0.594	0.618	0.89	0.367	0.575	0.683	0.745	0.884	**0.654**	**0.697**	**0.717**	**0.777**	**0.912**	
2015-2016	0.433	0.47	0.483	0.581	0.705	0.382	0.413	0.489	0.67	0.885	0.492	0.487	0.591	0.703	0.855	**0.533**	**0.632**	**0.597**	**0.777**	**0.907**	
2016-2017	0.159	0.244	0.391	0.588	0.677	0.276	0.339	0.578	0.756	0.908	0.589	0.405	0.536	0.755	**0.915**	**0.626**	**0.573**	**0.621**	**0.813**	0.879	
2017-2018	0.105	0.0917	0.157	0.424	0.586	0.343	0.339	0.273	0.678	0.901	0.294	**0.475**	0.421	0.583	**0.908**	**0.377**	0.372	**0.567**	**0.7329**	0.889	
2018-2019	0.321	0.335	0.338	0.535	0.603	0.406	0.436	0.432	0.713	**0.895**	0.5463	0.538	0.551	**0.791**	0.885	**0.573**	**0.597**	**0.656**	0.781	0.892	
2010-2011	0.108	0.147	0.285	0.44	0.42	0.116	0.446	0.576	0.55	0.533	0.382	0.439	0.408	0.709	0.678	**0.541**	**0.458**	**0.5854**	**0.767**	**0.793**	KNN
2011-2012	0.335	0.505	0.527	0.643	0.658	0.5214	0.619	0.697	0.812	0.88	0.525	0.596	0.611	0.815	0.898	**0.585**	**0.683**	**0.76**	**0.869**	**0.903**	
2012-2013	0.324	0.384	0.368	0.347	0.425	0.38	0.481	0.49	0.621	0.865	0.623	0.554	0.548	0.673	**0.878**	**0.675**	**0.643**	**0.605**	**0.729**	0.836	
2013-2014	0.166	0.239	0.327	0.403	0.422	0.435	0.335	0.586	0.631	0.847	**0.442**	**0.466**	0.631	0.612	**0.901**	0.376	0.424	**0.723**	**0.705**	0.893	
2014-2015	0.173	0.186	0.273	0.374	0.424	0.511	0.542	0.575	0.656	0.883	0.561	0.596	0.712	0.778	**0.907**	**0.682**	**0.688**	**0.719**	**0.786**	0.896	
2015-2016	0.251	0.295	0.315	0.294	0.332	0.396	0.469	0.603	0.635	0.829	0.594	0.475	**0.605**	**0.772**	**0.852**	**0.606**	**0.589**	0.5972	0.75	0.849	
2016-2017	0.121	0.213	0.171	0.281	0.302	0.244	0.297	0.584	0.739	0.902	**0.622**	0.358	0.629	0.76	**0.903**	0.6	**0.543**	**0.692**	**0.834**	0.865	
2017-2018	0.086	0.099	0.161	0.305	0.372	0.276	0.287	0.334	0.669	0.877	**0.444**	**0.693**	**0.585**	0.696	**0.901**	0.443	0.577	0.582	**0.7231**	0.894	
2018-2019	0.262	0.31	0.335	0.421	0.491	0.375	0.416	0.468	0.717	0.825	0.521	0.49	0.609	**0.812**	**0.9182**	**0.552**	**0.572**	**0.677**	0.779	0.917	

The best results were bold.

TABLE VII
AVERAGE ACCURACY AND KAPPA OF OUR METHOD

	Mid March	End March	End April	End May	End June	
Random Forest	**0.78**	**0.789**	**0.815**	**0.857**	**0.95**	ACC
Naive Bayes	0.727	0.724	0.764	0.824	0.918	ACC
KNN	0.747	0.746	0.793	0.855	0.919	ACC
Random Forest	**0.629**	**0.651**	**0.699**	0.769	**0.921**	Kappa
Naive Bayes	0.576	0.572	0.637	0.731	0.871	Kappa
KNN	0.562	0.575	0.66	**0.771**	0.871	Kappa

The best results were bold.

VIII. CONCLUSION

In this article, we predicted the final grades of the students having a promising accuracy at the early stages of a course. For that, we initially clustered the students into different groups using the accumulated XP that they obtained during the course. Then, the clusters were balanced by generating and adding virtual students to the smaller clusters. Finally, the irrelevant students' attributes were dropped from the data and their final grades were predicted using three algorithms. Our method was evaluated using the data collected from a gamification course for nine years. The results showed that for all years we could predict the students' final grades at the end of the fourth week of a semester.

In the future, we plan to extend our grade prediction approach for personalizing the course materials for the students, and also consider dynamic number of students' clusters during a semester.

REFERENCES

[1] K. M. Abiodun, E. A. Adeniyi, D. R. Aremu, J. B. Awotunde, and E. Ogbuji, "Predicting students performance in examination using supervised data mining techniques," in *Proc. Int. Conf. Inform. Intell. Appl.*, 2021, pp. 63–77, doi: 10.1007/978-3-030-95630-1_5.

[2] J. Akosa, "Predictive accuracy: A misleading performance measure for highly imbalanced data," in *Proc. SAS Glob. Forum*, 2017, pp. 2–5.

[3] I. A. A. Amra and A. Y. Maghari, "Students performance prediction using KNN and naïve Bayesian," in *Proc. 8th Int. Conf. Inf. Technol.*, 2017, pp. 909–913, doi: 10.1109/icitech.2017.8079967.

[4] K. E. Arnold and M. D. Pistilli, "Course signals at purdue: Using learning analytics to increase student success," in *Proc. 2nd Int. Conf. Learn. Anal. Knowl.*, 2012, pp. 267–270, doi: 10.1145/2330601.2330666.

[5] K. Baker, "Singular value decomposition tutorial," *Ohio State Univ.*, vol. 24, pp. 1–24, 2005.

[6] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Early prediction of student profiles based on performance and gaming preferences," *IEEE Trans. Learn. Technol.*, vol. 9, no. 3, pp. 272–284, Jul.–Sep. 2016.

[7] G. Barata, S. Gama, J. Jorge, and D. Gonçalves, "Studying student differentiation in gamified education: A long-term study," *Comput. Hum. Behav.*, vol. 71, pp. 550–585, 2017, doi: 10.1016/j.chb.2016.08.049.

[8] G. Barata, S. Gama, J. A. Jorge, and D. J. Gonçalves, "Relating gaming habits with student performance in a gamified learning experience," in *Proc. 1st ACM SIGCHI Annu. Symp. Comput.- Human Interact. Play*, 2014, pp. 17–25, doi: 10.1145/2658537.2658692.

[9] T. Barnes, M. Desmarais, C. Romero, and S. Ventura, "Educational data mining 2009," in *Proc. 2nd Int. Conf. Educ. Data Mining*, Córdoba, Spain, 2009, pp. 1–338.

[10] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Inf. Proc.- Lett. Rev.*, vol. 11, pp. 203–224, 2007.

[11] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," in *Proc. Data Mining Techn. Life Sci.*, 2010, pp. 223–239, doi: 10.1007/978-1-60327-241-4_13.

[12] R. Boyatt, M. Joy, C. Rocks, and J. Sinclair, "What (use) is a MOOC?," in *Proc. 2nd Int. Workshop Learn. Technol. Educ. Cloud*, 2014, pp. 133–145, doi: 10.1007/978-94-007-7308-0_15.

[13] P. Branco, R. P. Ribeiro, and L. Torgo, "UBL: An R package for utility-based learning," 2016, *arXiv:1604.08079*.

[14] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[15] C. G. Brinton and M. Chiang, "MOOC performance prediction via clickstream data and social learning networks," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2299–2307, doi: 10.1109/infocom.2015.7218617.

[16] C. Callender and R. Feldman, "Part-time undergraduates in higher education: A literature review," Higher Educ. Careers Services Unit, Manchester, U.K., Tech. Rep., 2009.

[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.

[18] L. Q. Chen, M. T. Wu, L. F. Pan, and R. B. Zheng, "Grade prediction in blended learning using multisource data," *Sci. Program.*, vol. 2021, pp. 1–15, 2021, doi: 10.1155/2021/4513610.

[19] P. Cortez and A. M. G. Silva, "Using data mining to predict secondary school student performance," in *Proc. 5th Annu. Future Bus. Technol. Conf.*, 2008, pp. 5–12.

[20] P. Cunningham and S. J. Delany, "k-nearest neighbour classifiers," *Multiple Classifier Syst.*, vol. 34, pp. 1–17, 2007.

[21] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Proc. Springer Ensemble Mach. Learn.*, 2012, pp. 157–175, doi: 10.1007/978-1-4419-9326-7_5.

[22] G. Demisse, T. Tadesse, and Y. Bayissa, "Data mining attribute selection approach for drought modelling: A case study for greater horn of africa," in *Proc. Int. J. Data Mining Knowl. Manage. Process*, 2017, pp. 1–16, doi: 10.5121/ijdkp.2017.7401.

[23] H. B. Demuth, M. H. Beale, O. De Jess, and M. T. Hagan, *Neural Network Design*, 2nd ed. Boston, MA, USA: PWS Publishing Co. 2014, Art. no. 1012.

[24] K. Deng, "Omega: On-line memory-based general purpose system classifier," *Ph.D. dissertation, Sch. Comput. Sci., Carnegie Mellon Univ.*, Pittsburgh, PA, USA, 1988.

[25] A. Elbadrawy and G. Karypis, "Domain-aware grade prediction and top-N course recommendation," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 183–190, doi: 10.1145/2959100.2959133.

[26] A. Elbadrawy, A. Polyzou, Z. Ren, M. Sweeney, G. Karypis, and H. Rangwala, "Predicting student performance using personalized analytics," *Computer*, vol. 49, pp. 61–69, 2016, doi: 10.1145/2959100.2959133.

[27] A. Elbadrawy, S. Studham, and G. Karypis, "Personalized multi-regression models for predicting students performance in course activities," Dept. Comput. Sci., Univ. Minnesota, Minneapolis, MN, USA, *Rep. UMN CS* 14–011, 2014, doi: 10.1145/2723576.2723590.

[28] N. Flyer and B. Fornberg, "Radial basis functions: Developments and applications to planetary scale flows," *Comput. Fluids*, vol. 46, pp. 23–32, 2011, doi: 10.1016/j.compfluid.2010.08.005.

[29] C. Hans, "Bayesian Lasso regression," *Biometrika*, vol. 96, pp. 835–845, 2009, doi: 10.1093/biomet/asp047.

[30] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, pp. 1–12, 2004, doi: 10.1002/chin.200419274.

[31] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali, "A comparative study of decision tree ID3 and C4.5," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, pp. 13–19, 2014, doi: 10.14569/specialissue.2014.040203.

[32] S. Huang and N. Fang, "Predicting student academic performance in an engineering dynamics course: A comparison of four types of predictive mathematical models," *Comput. Educ.*, vol. 61, pp. 133–145, 2013, doi: 10.1016/j.compedu.2012.08.015.

[33] C.-S. Hwang and Y.-C. Su, "Unified clustering locality preserving matrix factorization for student performance prediction," *IAENG Int. J. Comput. Sci*, vol. 42, pp. 245–253, 2015.

[34] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran, "Machine learning based student grade prediction: A case study," 2017, *arXiv:1708.08744*.

[35] J. Jenness and J. J. Wynne, "Cohen's kappa and classification table metrics 2.0: An arcview 3X extension for accuracy assessment of spatially explicit models,wc /+," Southwest Biol. Sci. Center, US Geological Surv., *Tucson, AZ, USA, Open-File Rep. 2005-1363*, 2005, doi: 10.3133/ofr20051363.

[36] S. T. Jishan, R. I. Rashu, N. Haque, and R. M. Rahman, "Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique," *Decis. Anal.*, vol. 2, 2015, Art. no. 1, doi: 10.1186/s40165-014-0010-2.

[37] A. Kordeczka, "Boruta - modern dimension reduction algorithm," 2018. [Online]. Available: http://rstudio-pubs-static.s3.amazonaws.com/369273_87ccc31e36c44bb886a5dfbf5865bb1c.html

[38] M. B. Kursa and W. R. Rudnicki, "Feature selection with the boruta package," *J. Statist. Softw.*, vol. 36, pp. 1–13, 2010, doi: 10.18637/jss.v036.i11.

[39] J. López-Zambrano, J. A. L. Torralbo, and C. Romero, "Early prediction of student learning performance through data mining: A systematic review," *Psicothema*, vol. 33, pp. 456–465, 2021, doi: 10.7334/psicothema2021.62.

[40] J. Luo, S. E. Sorour, K. Goda, and T. Mine, "Predicting student grade based on free-style comments using Word2Vec and ANN by considering prediction results obtained in consecutive lessons," *Int. Educ. Data Mining Soc.*, 2015, pp. 396–399.

[41] F. Marbouti, H. A. Diefes-Dux, and K. Madhavan, "Models for early prediction of at-risk students in a course using standards-based grading," *Comput. Educ.*, vol. 103, pp. 1–15, 2016, doi: 10.1016/j.compedu.2016.09.005.

[42] M. Mayilvaganan and D. Kalpanadevi, "Comparison of classification techniques for predicting the performance of students academic environment," in *Proc. IEEE Int. Conf. Commun. Netw. Technol.*, 2014, pp. 113–118, doi: 10.1109/cnt.2014.7062736.

[43] T. McKay, K. Miller, and J. Tritz, "What to do with actionable intelligence: E. 2 coach as an intervention engine," in *Proc. ACM 2nd Int. Conf. Learn. Anal. Knowl.*, 2012, pp. 88–91, doi: 10.1145/2330601.2330627.

[44] Y. Meier, J. Xu, O. Atan, and M. Van der Schaar, "Predicting grades," *IEEE Trans. Signal Process.*, vol. 64, no. 4, pp. 959–972, Feb. 2016.

[45] R. R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264, doi: 10.5555/2981562.2981720.

[46] L. C. Molina, L. Belanche, and À. Nebot, "Feature selection algorithms: A survey and experimental evaluation," in *Proc. IEEE Int. Conf. Data Mining*, 2002, pp. 306–313, doi: 10.1109/icdm.2002.1183917.

[47] D. C. Montgomery, E. A. Peck, and G. G. Vining, *Introduction to Linear Regression Analysis*, vol. 821. New York, NY, USA: Wiley, 2012.

[48] S. Morsy and G. Karypis, "Cumulative knowledge-based regression models for next-term grade prediction," in *Proc. SIAM Int. Conf. Data Mining*, 2017, pp. 552–560, doi: 10.1137/1.9781611974973.62.

[49] A. H. Nabizadeh, J. Jorge, S. Gama, and D. Gonçalves, "How do students behave in a gamified course?-A ten-year study," *IEEE Access*, vol. 9, pp. 81008–81031, 2021, doi: 10.1109/access.2021.3083238.

[50] A. H. Nabizadeh, J. P. Leal, H. N. Rafsanjani, and R. R. Shah, "Learning path personalization and recommendation methods: A survey of the state-of-the-art," *Expert Syst. With Appl.*, vol. 159, 2020, Art. no. 113596, doi: 10.1016/j.eswa.2020.113596.

[51] A. H. Nabizadeh, D. Gonçalves, S. Gama, J. Jorge, and H. N. Rafsanjani, "Adaptive learning path recommender approach using auxiliary learning objects," *Comput. Educ.*, vol. 147, 2019, Art. no. 103777, doi: 10.1016/j.compedu.2019.103777.

[52] A. H. Nabizadeh, A. Jorge, and J. P. Leal, "Long term goal oriented recommender systems," in *Proc. 11th Int. Conf. Web Inf. Syst. Technol.*, 2015, vol. 1, pp. 552–557, doi: 10.5220/0005493505520557.

[53] A. H. Nabizadeh, A. M. Jorge, and J. P. Leal, "Estimating time and score uncertainty in generating successful learning paths under time constraints," *Expert Syst.*, vol. 36, 2018, Art. no. e12351, doi: 10.1111/exsy.12351.

[54] A. H. Nabizadeh, A. M. Jorge, S. Tang, and Y. Yu, "Predicting user preference based on matrix factorization by exploiting music attributes," in *Proc. 9th Int. C* Conf. Comput. Sci. and Softw. Eng.*, 2016, pp. 61–66, doi: 10.1145/2948992.2949010.

[55] A. H. Nabizadeh, A. Mário Jorge, and J. Paulo Leal, "Rutico: Recommending successful learning paths under time constraints," in *Proc. Adjunct Pub. 25th Conf. User Model., Adapt. Personalization*, 2017, pp. 153–158, doi: 10.1145/3099023.3099035.

[56] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Front. Neurorobot.*, vol. 7, 2013, Art. no. 21, doi: 10.3389/fnbot.2013.00021.

[57] L. L. Nathans, F. L. Oswald, and K. Nimon, "Interpreting multiple linear regression: A guidebook of variable importance," *Practical Assessment, Res. Eval.*, vol. 17, pp. 1–19, 2012, doi: 10.7275/5FEX-B874.

[58] A. Nogueira, A. H. Nabizadeh, and D. Gonçalves, "DynaGame: A rule editor for gamified education," in *Proc. IEEE Int. Conf. Graph. Interact.*, 2021, pp. 1–7, doi: 110.1109/ICGI54032.2021.9655275.

[59] E. Osmanbegovic and M. Suljic, "Data mining approach for predicting student performance," *Econ. Rev.: J. Econ. Bus.*, vol. 10, pp. 3–12, 2012.

[60] Z. A. Pardos and N. T. Heffernan, "Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset," *J. Mach. Learn. Res. W. CP*, 2010, pp. 1–12.

[61] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDD Cup Workshop*, 2007, vol. 2007, pp. 5–8, doi: ACM 978-1-59593-834-3/07/0008.

[62] L. E. Peterson, "K-nearest neighbor," *Scholarpedia*, vol. 4, 2009, Art. no. 1883, doi: 10.4249/scholarpedia.1883.

[63] A. Polyzou and G. Karypis, "Grade prediction with models specific to students and courses," *Int. J. Data Sci. Anal.*, vol. 2, pp. 159–171, 2016, doi: 10.1007/s41060-016-0024-z.

[64] A. Polyzou and G. Karypis, "Feature extraction for classifying students based on their academic performance," in *Proc. 11th Int. Conf. Educ. Data Mining*, 2018, pp. 356–362.

[65] M.-C. Popescu, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, "Multilayer perceptron and neural networks," *WSEAS Trans. Circuits Syst.*, vol. 8, pp. 579–588, 2009.

[66] Y. Qi, "Random forest for bioinformatics," in *Proc. Ensemble Mach. Learn.*, 2012, pp. 307–323, doi: 10.1007/978-1-4419-9326-7_11.

[67] A. H. N. Rafsanjani, "A long term goal recommender approach for learning environments," Ph.D. dissertation, Univ. Porto, Porto, Portugal, 2018.

[68] A. H. N. Rafsanjani, N. Salim, A. R. Aghdam, and K. B. Fard, "Recommendation systems: A review," *Int. J. Comput. Eng. Res.*, vol. 3, pp. 47–52, 2013.

[69] M. Raleiras, A. H. Nabizadeh, and F. A. Costa, "Automatic learning styles prediction: A survey of the state-of-the-art (2006-2021)," *J. Comput. Educ.*, pp. 1–93, 2021, doi: 10.1007/s40692-021-00215-7.

[70] V. Ramesh, P. Parkavi, and K. Ramar, "Predicting student performance: A statistical and data mining approach," *Int. J. Comput. Appl.*, vol. 63, pp. 35–39, 2013, doi: 10.5120/10489-5242.

[71] Z. Ren, X. Ning, A. Lan, and H. Rangwala, "Grade prediction based on cumulative knowledge and co-taken courses," in *Proc. 12th Int. Conf. Educ. Data Mining*, 2019, pp. 158–167.

[72] Z. Ren, X. Ning, and H. Rangwala, "Grade prediction with temporal course-wise influence," 2017, *arXiv:1709.05433*.

[73] Z. Ren, X. Ning, and H. Rangwala, "Ale: Additive latent effect models for grade prediction," in *Proc. 2018 SIAM Int. Conf. Data Mining*, 2018, pp. 477–485, doi: 10.1137/1.9781611975321.54.

[74] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000, doi: 10.1109/icdm.2010.127.

[75] I. Rish, "An empirical study of the naive Bayes classifier," in *Proc. IJCAI Workshop Empirical Methods Artif. Intell.*, 2001, vol. 3, pp. 41–46.

[76] C. Romero and S. Ventura, "Educational data mining and learning analytics: An updated survey," *Wiley Interdiscipl. Rev.: Data Mining Knowl. Discov.*, vol. 10, 2020, Art. no. e1355, doi: 10.1002/widm.1355.

[77] S. Rovira, E. Puertas, and L. Igual, "Data-driven system to predict academic grades and dropout," *PLoS One*, vol. 12, 2017, Art. no. e0171207, doi: 10.1371/journal.pone.0171207.

[78] P. H. Santoso, S. Bahri Wahyudi, and J. Syahbrudin, "Predicting engineering students' grade on introductory physics using machine learning," in *Proc. 5th Int. Conf. Curr. Issues Educ.*, 2022, pp. 97–102, doi: 10.2991/assehr.k.220129.018.

[79] S. E. Sorour, T. Mine, K. Goda, and S. Hirokawa, "A predictive model to evaluate student performance," *J. Inf. Process.*, vol. 23, pp. 192–201, 2015, doi: 10.2197/ipsjjip.23.192.

[80] M. Stern, J. Beck, and B. P. Woolf, "Naive Bayes classifiers for user modeling," *Center Knowl. Commun., Comput. Sci. Dept., Univ. Massachusetts*, Boston, MA, USA, 1999.

[81] S. Subbotin, "A random forest model building using a priori information for diagnosis," in *Proc. Comp. Model. Intell. Syst.*, 2019, pp. 962–973.

[82] M. Sweeney, J. Lester, and H. Rangwala, "Next-term student grade prediction," in *Proc. IEEE Int. Conf. Big Data*, 2015, pp. 970–975, doi: 10.1109/bigdata.2015.7363847.

[83] M. Sweeney, H. Rangwala, J. Lester, and A. Johri, "Next-term student performance prediction: A recommender systems approach," 2016, *arXiv:1604.01840*.

[84] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2015.

[85] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: The kappa statistic," *Family Medicine*, vol. 37, pp. 360–363, 2005.

[86] J. Xu, K. H. Moon, and M. Van Der Schaar, "A machine learning approach for tracking and predicting student performance in degree programs," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 5, pp. 742–753, Aug. 2017.

[87] S. Xu, "Bayesian naïve bayes classifiers to text classification," *J. Inf. Sci.*, vol. 44, pp. 48–59, 2018, doi: 10.1177/0165551516677946.

[88] T.-Y. Yang, C. G. Brinton, C. Joe-Wong, and M. Chiang, "Behavior-based grade prediction for MOOCs via time series neural networks," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 5, pp. 716–728, Aug. 2017.

Amir Hossein Nabizadeh received the M.Sc. degree in information technology management from the University Technology Malaysia (UTM), Johor Bahru, Malaysia, in 2013, and the Ph.D. degree in computer science from the University of Porto, Porto, Portugal, in 2018.

He is currently a Senior Researcher with INESC-ID, Lisbon, Portugal and the Institute for Futures Studies in Health, Kerman University of Medical Science, Kerman, Iran. His main research interests include machine learning, artificial intelligence, artificial intelligence in medicine, data mining, recommender systems, and E- Learning.

Daniel Gonçalves received the B.Sc. degree in computer science in 1996, the M.Sc. degree in electrical and computer engineering in 2000, and the Ph.D. degree in computer science in 2007.

He currently a Researcher with the Graphics and Interaction (IG) Group, INESC-ID, Lisbon, Portugal, and an Associate Professor of Computer Science with Instituto Superior Técnico (IST/UL), Lisbon, Portugal. His research interests include the areas of information visualization, personal information management, and human–computer interaction and accessibility.

Dr. Gonçalves is a Member of the ACM and the Portuguese Computer Graphics Group (the national Eurographics chapter).

Sandra Gama received the B.Sc. degree in computer science in 2007, the M.Sc. degree in computer science in 2009, and the Ph.D. degree in computer science in 2015.

She currently a Researcher with the IG Research Group, INESC-ID, Lisbon, Portugal and an Assistant Professor of Computer Science with the Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal. Her research interests include information visualization, multimodal user interfaces, and human–computer interaction.

Joaquim Jorge (Senior Member, IEEE) received the Ph.D. degree in computer science from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1995.

He coordinates the IG Research Group, INESC-ID, Lisbon, Portugal. He is currently a Professor of Computer Graphics and Multimedia with Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. His research interests include multimodal interfaces, AR/VR, and advanced learning techniques.

Dr. Jorgehe is the Editor-in-Chief of the *Computers & Graphics Journal*, Fellow of the Eurographics Association, ACM Distinguished Member and Speaker, and chairs the ACM/SIGGRAPH External Relations Committee.