Ruipérez-Valiente, J. A., Muñoz-Merino, P. J., Alexandron, G., and Pritchard, D. E. Using Machine Learning to Detect 'Multiple-Account' Cheating and Analyze the Influence of Student and Problem Features. IEEE Transactions on Learning Technologies, vol. 12, no. 1, pp. 112-122, 1 Jan-March 2019

https://doi.org/10.1109/TLT.2017.2784420

https://ieeexplore.ieee.org/document/8219749/

Using Machine Learning to Detect 'Multiple-Account' Cheating and Analyze the Influence of Student and Problem Features

José A. Ruipérez-Valiente, Pedro J. Muñoz-Merino, *Senior Member, IEEE,*
Giora Alexandron and David E. Pritchard

Abstract—One of the reported methods of cheating in online environments in the literature is CAMEO (Copying Answers using Multiple Existences Online), where *harvesting* accounts are used to obtain correct answers that are later submitted in the *master* account which gives the student credit to obtain a certificate. In previous research we developed an algorithm to identify and label submissions that were cheated using the CAMEO method; this algorithm relied on the IP of the submissions. In this study we use this tagged sample of submissions to i) compare the influence of student and problems characteristics on CAMEO and ii) build a random forest classifier that detects submissions as CAMEO without relying on IP, achieving sensitivity and specificity levels of 0.966 and 0.996, respectively. Finally, we analyze the importance of the different features of the model finding that student features are the most important variables towards the correct classification of CAMEO submissions, concluding also that student features have more influence on CAMEO than problem features.

Index Terms—Academic dishonesty, educational data mining, machine learning, MOOCs.

1 Introduction

Academic dishonesty - defined as any type of fraudulent action in any academic work [1] - is a serious problem in education. With the increased use of online learning platforms, the study of academic dishonesty (which most authors call 'cheating') in online systems has increased, so some authors make a division between the most traditional cheating methods (like those that were used in the classroom) and more contemporary methods (those which incorporate Internet or new electronic devices) [2]. This issue becomes even more evident in online education where there is no ID confirmation about who took an exam or what they did during that exam [3]. Therefore, it is important that instructors try to create an honest culture and basic beliefs that positively contribute to student learning in online environments [4], e.g. such as the Honor Code [1] in which students must not create several accounts or share solutions with their peers.

MOOCs (Massive Open Online Courses) are free courses taken by many students from different parts of the world [5]. Over the last few years MOOCs have attracted a lot of attention providing both formal and informal education [6]. As they have become important, MOOCs have also brought a new form of "academic dishonesty", termed CAMEO [7], [8], [9], [10] in which students use multiple accounts to *harvest* the correct solution which is used later in their master account to earn a certificate. We denote the account that is used to get the solutions as *harvesting* account (or *harvester*), and the account that is used to insert the correct solutions and with which the student earns a certificate, as the *master* account; we note that in CAMEO these two accounts are run by the same person. Gaming the system is a phenomenon where the learner tries to get credit in a learning environment (e.g. obtaining a good score) by exploiting some of its properties instead of actually trying to learn [11]. CAMEO can be seen as a particular case of gaming the system since students are exploiting the system properties e.g. creating multiple accounts and using the feedback in quizzes to acquire a certificate without actually learning the materials. Gaming the system has usually been studied in Intelligent Tutoring Systems (ITSs). However, previous works in ITSs did not include CAMEO as a type of gaming the system because the issue of multiple accounts is not present in those environments, however it is common problem in MOOCs. In addition, gaming the system is not necessarily related to academic dishonesty, but CAMEO is more directly related as students need to agree to comply with a Code of Honor before enrolling into a course.

CAMEO is important for three reasons: it is related to poor learning [12], it undermines the credibility of MOOC certificates, and it may systematically affect educational research studies. Therefore, it is important to study how to decrease the prevalence of CAMEO in MOOCs and to know the causes of CAMEO. In this direction the implementation of models that are able to classify CAMEO submissions correctly as well as the study of which factors and variables affect CAMEO more importantly, can be helpful to find ways to decrease it. In this vein, this research addresses the following research questions:

- *J. A. Ruipérez-Valiente and P. J. Muñoz-Merino are with the Telematics Departament, Universidad Carlos III de Madrid, 28911 Leganés, Spain. J. A. Ruipérez-Valiente is also with Institute IMDEA Networks, 28918 Leganes, Spain (e-mail: jruipere@it.uc3m.es; pedmume@it.uc3m.es).*
- *G. Alexandron and D. E. Pritchard are with the Physics Departament, Massachusetts Institute of Technology (MIT), Massachusetts Ave 77, 02139 Cambridge (MA) USA (e-mail: giora@mit.edu; dpritch@mit.edu).*

Manuscript received April 19, 2005; revised August 26, 2015.

1. https://www.edx.org/edx-terms-service

1) Is CAMEO more influenced by the student or by the question? How do these results compare to similar studies in gaming the system?
2) Can we make an accurate classification of which submissions are CAMEO relying only on student, problem and submission features?
3) Which features of the classification model have a higher importance, thus providing more information regarding the detection of CAMEO submissions? Are these results in line with the findings of the first research question? Which features could be removed to simplify the model without having an impact on the performance?

The remainder of the manuscript is organized as follows: Section 2 reviews the related work in academic dishonesty and gaming the system, Section 3 explains the methodology followed and describes the data collection. Section 4 analyzes the importance of CAMEO on the student and the problem comparing with other similar studies in gaming the system. Section 5 implements and reports the results of a random forest (RF) model based on student, problem and submission features to classify if a correct submission of a student is CAMEO or not. Section 6 discusses the importance of the features of the model connecting also with the findings of Section 4. Section 7 presents conclusions and directions for future work.

2 Related Work

The analysis of the influence of student features and content features on student behaviors has been a topic of interest of research in recent years, especially on gaming the system. Gaming the system can be defined as trying to take advantage of the system properties in order to get credit by a learning environment [13]. The analysis of some works suggest that lesson features have more influence on gaming the system than student features, e.g. a 9% of variability was reported by student features [14] but a 54% variability was reported by lesson features [15]. These findings can be due to the fact that troubles with exercises, hints or content structure might make students to game the system in order to overcome these difficulties with the materials. However, student features such as student knowledge about different skills make an influence on gaming the system [16]. Indeed, recent studies suggest that gaming might be more affected by student features than by problem features [17], [18].

An interesting open question is whether different types of academic dishonesty are affected more by features of the students or the content, and to what extent academic dishonesty can be predicted by these features. It has also been reported that perspective and demographic factors affect cheating also [19]. Most previous cheating research was based on retrospectively self-reported survey data, thus there is a need of more objective data-based evidence regarding what features are more influential. Results from an interview study show that students' perceptions about cheating are related to some user features but also to content ones [20]. Previous studies revealed that various student features, such as that the strategy of students in games has a relationship with cheating [20], that different student personality factors have an effect on cheating [21], or that it is possible to base prediction models (e.g. decision trees) that depend on the students' moral [22] or other user indicators [23].

Although most cheating studies are based on user features, there are some studies that show the dependence of cheating on contents, instructional goals, and the availability of the copy&paste functionality [24]. In this paper, we investigate the question of whether CAMEO (a specific type of academic dishonesty in MOOCs) is more associated with student or problem features, and we compare these results with related work on gaming. Afterwards we build a machine learning model based on student, problem and submission features. A main goal of this work is to explore the importance of the different variables on the detection of CAMEO submissions without the use of IP address which was a necessary variable in previous studies [7], [8], [10]. Furthermore, the use of IP for multiple account cheating might not always be reliable. For example, students working within the same network (e.g. dormitories, universities) might share the same public IP which is the actual one that is logged within the web learning environment. Additionally, once this detection method becomes public knowledge, students can easily hide themselves behind a proxy connection, hence changing their IP. Therefore, it is important to explore alternative solutions to detect these patterns.

3 Methods

In this section we review the methodology and tools that this study applies. Section 3.1 describes the case study, Section 3.2 gives a quick overview of the detection method applied to obtain a tagged (CAMEO or non-CAMEO) sample and Section 3.3 describes the data collection that we are using.

3.1 MOOC and participants

We study an introductory physics MOOC called 8.MReV [2] and run on edX.org in Summer 2014 by MIT faculty. The MOOC received the enrollment of about 13500 participants of which 502 managed to earn a certificate. The course lasted for 14 weeks and there were 12 mandatory units and 2 additional optional units on advanced materials. The course contained about 1000 problems and 69 videos. These problems are organized as checkpoints embedded within e-text and videos, and homework and quiz problems at the end of each unit. The weight of these different types of assignments towards the final grade is different (Quiz $>$ Homework $>$ Checkpoint).

3.2 Overview of the Detection of CAMEO

The algorithm that we use to detect CAMEO submissions has been reported in detail previously [8], [10], thus for simplicity here we provide only a rough overview with the base ideas that we apply to tag submissions as CAMEO or not. The algorithm takes as input the clickstream data (tracking logs) where all the actions of users are stored as interaction events. Particularly relevant events for the algorithm are *problem_get* (when a student gets a problem), *problem_check*

2. https://www.edx.org/course/mechanics-review-mitx-8-mrevx

(when a student submits a solution) and *show_answer* (when a student asks to see a solution). The algorithm searches for students that shared the same IP address at some stage during the course, and we assigned those accounts linked through a shared IP to a single IP group. The algorithm searches for pairs of accounts in each IP group that have submitted questions that fulfill the following criteria: As first step, for each submission done by each account a_1 to each question q, the algorithm checks if there is any other account a_2 within the IP group of a_1, that obtained the correct answer to q in the previous 24h, either by using *show_answer* or exhaustively searching with several *problem_check* attempts. If found, we add the triplet $< a_1, a_2, q >$ as one of the potential CAMEO events. As second step, in order to ensure that we are not detecting false positives, we apply the following set of additional filters to the collected events detected in the first step:

1) The harvester does not earn a certificate.
2) At least 10 potential harvesting events have been detected for the master-harvester couple.
3) At least 5% of the master's correct submissions are CAMEO.
4) We find evidence of 'inhumanly fast' submissions.
5) More than 55% of the questions solved by the harvester account, were actually used by the master account.
6) The harvesting account does not act as a master account (or *vice versa*) at any time, since this is not a reciprocal relationship.

The thresholds in items 2-5 were chosen using statistical criteria described in [8], [10]. We note that the criteria that we established are very strict. Additionally, in our previous work [10] we manually analyzed the logs of a random sample of master accounts, concluding that in our judgment all of them were real CAMEO users. These results allowed us to establish a high confidence interval in terms of our precision. Consequently, the sample detected as CAMEO submissions is conservatively trustworthy. However, we should also highlight that since we are very strict, our recall might be low and some CAMEO submissions might be labeled as non-CAMEO, which might have a small noisy effect. As data is really imbalanced towards non-CAMEO we believe this effect to be non-important, and hence we can use the data sample to train the machine learning model in the current work.

3.3 Data Collection

The user interactions of students with the described MOOC are logged as a sequence of click-stream events. The data is then transformed into other variables for the analysis of this work. Note that we only take into account correct submissions, since that is a necessary feature of CAMEO; we completely discard incorrect submissions. Using the detection method described in 3.2 we are able to tag each correct submission of the dataset with the following variable:

- *harvested*: It is a binary variable with values 0 and 1, labeled as non-CAMEO and CAMEO, indicating if a specific student harvested or not a correct submission on a specific problem.

Then, for each correct submission by any user we have computed the triplet $S_{i,j,h}$, which represents the correct submission S done by the student i, in the question j, and h represents *harvested* variable (labeled using the criteria explained in Subsection 3.2 above). The criteria to include a correct submission event within the sample are:

- We keep correct submissions from students that completed at least 5% of the questions in the course.
- We remove from the data sample the submissions of users who were initially detected but did not surpass all the filtering in step 2 of Section 3.2. The rationale is that we are not certain whether those students harvested part of their solutions or not, thus their submissions could introduce noise when training the model.
- We remove submissions which belong to non-graded sections, such as initial survey, post survey, introduction, etc.
- We remove submissions that belong to harvesting accounts as these represent an outlying and noisy behavior.

Finally, these criteria leave us with a dataset of 470939 correct submissions, of which 27232 (6.13%) were labeled CAMEO. Furthermore, 65 (12.9%) accounts that earned a certificate and 84 (7.7%) that did not earn it, were detected as CAMEO.

4 Dependence of CAMEO on Student and Problem

Given that a CAMEO event involves the choice of a student on a particular problem, we now investigate the dependence of CAMEO on the student and problem. We use the three methodologies that were proposed in [18] because we specifically want to compare with the results published in this study related to gaming the system. Additionally, the three methods diverge in their basic foundations, which is adequate to obtain more general conclusions i.e., histograms are based on a visual analysis, logistic regression is a linear parametric model and a Bayesian Network (BN) is a probabilistic graphical model. The following subsections include the considered variables and the three analyses to compare student vs problem parameters as predictors of CAMEO.

4.1 Considered Variables

This analysis of the dependence of CAMEO on student and problem is based in related work in gaming the system [17], [18]. Therefore, since we want to replicate their study within our CAMEO context, the four variables considered in this section are exactly the ones proposed by this previous work [17], [18] for this particular analysis.

- *avg_student*: Number of CAMEO submissions (i.e. $harvested = 1$) by the student divided by the total number of problems that the student solved correctly. This variable gives an idea of the level of CAMEO for each student.
- *avg_problem*: Number of CAMEO submissions (i.e. $harvested = 1$) for a problem divided by the number of students who answered correctly that problem.

This variable gives an idea of the level of CAMEO for each problem.

- *level_student*: 0 is for masters with a low level of CAMEO and 1 for a high level. To differentiate low and high level masters, we establish a threshold as the median of the variable *avg_student* for those accounts which are detected as master.
- *level_problem*: 0 is a problem with a low level of CAMEO and 1 for a high level, we establish the threshold as the median of the variable *avg_problem*.

4.2 Histograms

Figure 1a represents a histogram of the distribution of the percentage of CAMEO questions per student, i.e. the *y*-axis represents the number of students who harvested the *x*-axis percentage of their answers. In a similar way, Figure 1b represents the number of problems with a given percentage of harvested solutions. We use the data collection from Subsection 3.3, therefore only accounts and submissions that passed that criteria are included in the plots.

The student distribution looks bimodal where we can mainly find students with low or high levels of CAMEO, with a dip in the middle. The students with low levels of CAMEO started late in the course. For the problem distribution, all of the problems are included, even though a few of them that were never harvested. We can see that the distribution has one main mode looking more Gaussian whereas the student distribution showed two main modes that addressed two different types of students. This is a preliminary indicator that harvesting depends more on student differences than problem differences.

Comparing these results of CAMEO cheating with the ones of gaming in [18] reveals similarity: their student distribution of gaming was bimodal, differentiating low and high gaming students. In addition, just one mode was detected for the problem distribution of gaming as in that study. The differences in the percentages where the modes are located are due to the different ways of computation of 'gaming events'.

4.3 Logistic Regression

We use a logistic regression model in which the dependent variable is *harvested* and the predictor variables were *avg_student* and *avg_problem*. This is quite similar to the analysis presented in [18] but we apply a logistic regression instead of a linear regression because *harvested* is a binary variable (a student either harvested the answer or not in a given problem) while the gaming variable in [18] was a quantitative variable because there were different methods for gaming in a problem so the level of gaming of a student in a given problem is considered as a quantitative variable and not just a binary variable. The results of the logistic regression gave a McFadden's pseudo R^2 of 0.73, which is a high level of explanation for the model. We got a Wald statistic of ($Z = 283.85, p < 2e-16$) for *avg_student*, and of ($Z = 57.65, p < 2e-16$) for *avg_problem* indicating the importance of both the student and problem. As Z is greater for *avg_student* than for *avg_problem*, then it appears that student parameters influence more than problem parameters. Comparing these results with the ones obtained for gaming in [18], the explained variability of gaming in the prediction model (60.8%) implied also a high level of explanation of the model with student features having also a greater influence than problem parameters in the regression model.

TABLE 1
Conditional probability table of the Bayesian network.

P(*harvested*)	*level_student* = 0	*level_student* = 1
level_problem = 0	0.019	0.78
level_problem = 1	0.040	0.91

4.4 Bayesian Network

As suggested in [18], a BN is proposed with three nodes to analyze the effect of problem and student on harvesting. All three variables are binary: being *harvested*, *level_student*, and *level_problem*. With all the data available, we inferred the conditional probability tables of the BN using the *bngrain* library in R, obtaining the following results in Table 1:

These results show a similar pattern as in the case of gaming the system, and again confirm the heavier influence of students on harvesting. The provided results support that student parameter is a better predictor of CAMEO events than problem parameter. The probability of CAMEO is greater when the student is a high CAMEO user than when the problem is highly cheated upon. From Table 1, we can infer that when a student is a low cheater, then the fact of addressing a problem which is highly cheated, only increase the amount of cheating from 1.9% to 4%. And when the student is a high cheater, the difference on the amount of cheating depending on whether the problem is highly cheated goes from 78% to 91%. However, the difference in the amount of cheating depending on whether the user is low or high cheater is from 1.9% to 78% (when the problem is lowly cheated) and from 4% to 91% (when the problem is highly cheated). Therefore, the effect on the amount of CAMEO is also significantly dependent on the problem.

5 Classification Model of CAMEO based on Student, Problem, and Submission Features

Motivated by the statistical importance that we found on student and problem features in previous section, we design a classification model using a machine learning methodology, which identifies whether a submission was harvested or not based on the selected student, problem and also including *submission* features. This model could potentially be used to detect CAMEO on run-time, without relying on users' IP address. The general idea is that student features will characterize the interaction of students with the platform (e.g. time watching videos), problem features define characteristics of each specific problem (e.g. type of response or maximum number of attempts) and submission features specify information related to the specific interaction between student and problem (e.g. time invested in the attempt) The first two following subsections describe the considered variables as well as the methodology and model training, the third one validates the model and describes the results, and a final one where we apply the model to the 'suspicious' submissions that were neither included in the training nor test datasets of the model.

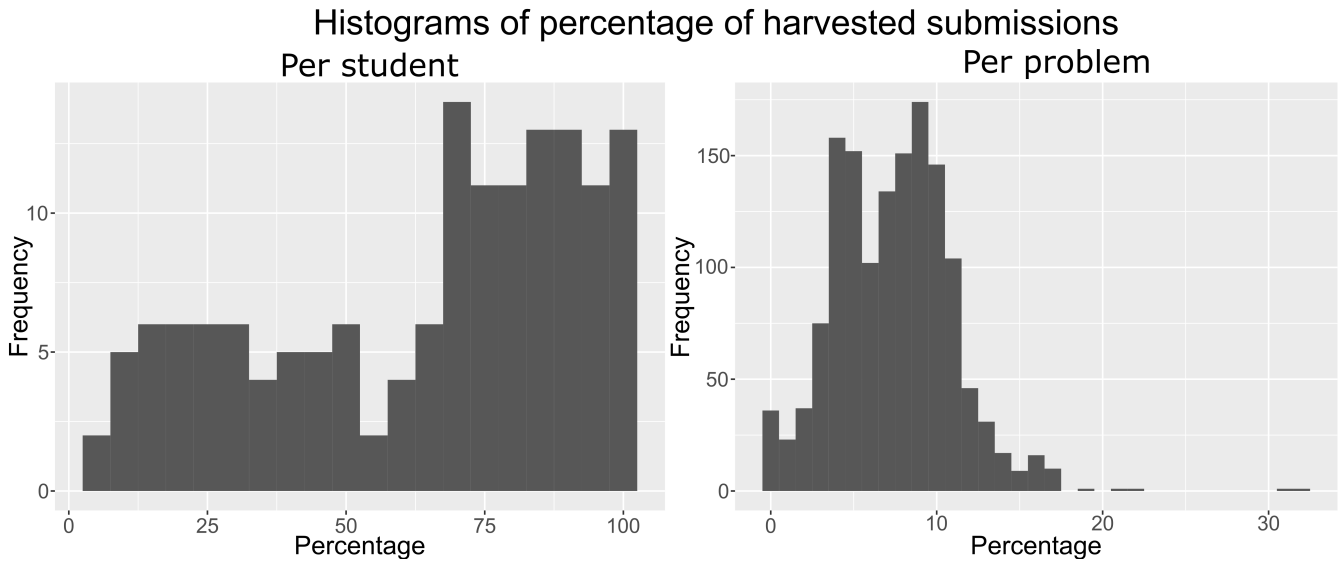

Fig. 1. Histograms showing: (a) average amount of CAMEO per student and (b) average amount of CAMEO per problem. Only information regarding master accounts due to the fact that all non-master accounts obviously have 0% as percentage of CAMEO submissions.

5.1 Considered Variables

The dependent variable that we aim to detect is *harvested* as was described in Subsection 3.3. We do not consider any variables that cannot be extracted from the MOOC data (e.g., prior knowledge or interest of students), or data that has have been input by students and might not be reliable (e.g., surveys or demographics), as this would limit the applicability of this model to other MOOCs where such data is not available. For this analysis, the considered independent variables are divided in those related to students, problem and submission features. The selection of these features has been based on lessons learned and conclusions drawn in our previous research [8], [9], [10]. We acknowledge that there might be other useful variables, but further improvements would lie more on a feature engineering project. First, the student features are the following:

- *performance_first*: Percentage of problems that were correctly submitted on the first attempt of the student.
- *sum_video_time*: Summation of the time spent in videos by the student.
- *questions_attempted*: Percentage of questions in the course attempted by the student.
- *attempts_correct_answer*: Average number of attempts required by the student to submit a correct response, including wrong responses to questions that were never correctly answered.
- *avg_time_correct*: Average time required by the student to submit a correct response, including wrong responses to questions that were never correctly answered.
- *sum_time_page*: Amount of time spent in course pages by the student.

The problem features are the following:

- *type_assignment*: Factor variable that indicates whether the problem was a 'Quiz', 'Homework' or 'Checkpoint' as described within Subsection 3.1.
- *type_response*: Factor variable that defines the type of response of each problem (e.g. multiple choice, fill the blank, formula, etc).
- *show_answer*: Factor variable that defines the configuration of the 'show answer' button. It can be available always, only after exhausting all your attempts, or only after the due date.
- *location*: Location of the problem within the course structure indicating with an integer the chapter where the problem is located.
- *random*: Binary variable indicating if the problem contains random variables or not.
- *max_attempts*: This variable specifies the maximum number of attempts allowed in the problem.

Finally, the considered submission features are the next:

- *time_to_deadline*: Difference of minutes between the submission deadline for the problem and the actual timestamp when the student submitted the problem.
- *attempt_duration*: Number of minutes elapsed between the event when the student got the problem (*problem_get*) and the submission of the problem (*problem_check*).
- *attempts_required*: Number of attempts that the student performed previous to the current submission, required to finally answer correctly the problem.

5.2 Methodology and Model Training

We have selected the RF algorithm [25] because it performs well on diverse types of data and also is useful for ranking the importance of predictors. As first step we divide the data reported in Subsection 3.3 into training (80%) and test (20%) datasets while maintaining a similar ratio of CAMEO and non-CAMEO submissions in both datasets. This leaves a dataset as shown in Table 2.

We use R software and specifically the *caret* package to build the model. We apply *train* function from *caret* package and RF algorithm from *randomForest* package, and

TABLE 2
Number of submissions after the dataset partitioning.

	Non-CAMEO submissions	CAMEO submissions
Train	354966	21786
Test	88741	5446

TABLE 3
Confusion matrix applying the model to the test dataset.

	Reference	
Classification	**Non-CAMEO**	**CAMEO**
Non-CAMEO	93.852%	0.194%
CAMEO	0.366%	5.588%

we configure it to perform a 10-fold cross validation and repeat 3 times to evaluate the results on the training set as well as select the tuning parameters for the RF model, the target quality metric that we seek to maximize is the Area Under the ROC Curve (AUC). We also configure the *train* function to pre-process the features by scaling and centering the numeric variables. The selected model is implemented with 500 trees (*ntree* parameter) and 10 variables sampled at each split (*mtry* parameter), the rest of configuration parameters are maintained as default. RF does not handle missing values, thus features that had missing values for some cases, have been filled with the mode value of the factor feature (this happened in few cases).

As a summary, each row of the model has the dependent variable *harvested*, and all the student, problem and submission features as specified in Subsection 5.1 We train the model obtaining value of AUC close to 1 (0.99993) on the training set and clearly improving the baseline prediction. The next subsection shows the results of applying this model on the test dataset.

5.3 Validation of the Model and Results

To validate the model, we use the test dataset, which is the portion of cases (20%) that were separated and not used to train the model. Table 3 shows the percentage confusion matrix (N = 94187 submissions) and Table 4 shows some quality metrics regarding the model when applied to the test dataset. We report the AUC, sensitivity, specificity, Kappa coefficient and accuracy (although taking into account that data is really unbalanced, this is not a very reliable measure). We can see a clear improvement with respect to the baseline accuracy which would be the classification of all submissions as not CAMEO.

The results show that the predictor has very good quality metrics when applied to the test data with an AUC value close to 1, high sensitivity (96.64%) and high specificity (99.61%). These results are quite encouraging since they suggest that it would be possible to implement a detector that can predict on run-time CAMEO with high probability and without relying on IP address. This removes the two primary limitations of our previous approach [8], [10].

TABLE 4
Quality metrics of the RF model applied to the test dataset.

AUC: 0.9993	**Sensitivity**: 0.9664	**Specificity**: 0.9961
Kappa coefficient: 0.9493	**Accuracy**: 0.9944	**Baseline accuracy**: 0.9421

5.4 Testing the Model on Submissions from 'Suspicious' Accounts

We have trained the model using 'undoubted' events (submissions that we are certain are CAMEO and non-CAMEO). As we explained in Subsection 3.2, some submissions were detected by the step 1 of the algorithm and considered 'suspicious', but did not comply with all the additional filtering described as step 2 in Subsection 3.2. In this experiment, students did not know have any knowledge regarding the availability of a cheating detector. Therefore, we do not expect "advanced cheating" behaviors where they might purposely hide their IP addresses. There are students who share public IP addresses, thus the former method cannot detect them as CAMEO. We expect to be able to detect these CAMEO users with the new machine learning model as the present method does not rely on IP addresses. This is an advantage of the new proposed method. We can assume that those students who shared a public IP address to behave in a similar way than the rest of CAMEO users. Therefore, this method should be able to detect new CAMEO events.

We test now our RF model on the 'suspicious' submissions that were removed before training it, in order to see what portion of those events are classified as CAMEO submissions. We apply the model to the 113045 'suspicious' submissions, obtaining an output indicating that 11157 (9.9%) submissions are detected as CAMEO. Taking into account that the sensitivity (percentage of CAMEO submissions correctly identified as such) of our model is 96.64%, we can be fairly safe saying that most of those were CAMEO submissions as well. These results might have implications regarding our previous approximation of amount of harvesting that we reported in our previous research [8], [10]. More exactly, in our last research study [10] we reported 29788 CAMEO submissions in this same MOOC, if we add this additional set of 11157 CAMEO submissions, it would imply an augment of 37.5% with respect our previous estimate.

6 Variable Importance

Random forest is a good method to address the importance of variables, as analyzed and explained in the original paper by Breiman [25]. Other studies have been successful as well in measuring variable importance within random forest models [26], [27]. We can find some packages in R with the purpose of variable selection and ranking importance of variables using random forest, e.g. *varSelRF* [3] package, importance function within the *randomForest* [4] package, or *VSURF* [5] package. In this paper we use *VSURF* package that provides a robust algorithm and method for ranking the importance of variables using RF; some studies using it can be found in the literature [26], [28].

3. https://cran.r-project.org/web/packages/varSelRF/varSelRF.pdf
4. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf
5. https://cran.r-project.org/web/packages/VSURF/VSURF.pdf

We can find two metrics in the literature to address the importance of variables in a RF model, the first one is the mean decrease node impurity and the second is by permuting out-of-bag (OOB) data. We use the latter one that can be defined as follows. For each tree t belonging to the forest, we take the OOB_t sample (i.e. the cases not included in the boostrap data to construct t) and we denote the misclassification rate of tree t on OOB_t as $errOOB_t$. Next, we randomly permute the values of variable X^j in OOB_t to get a disturbed but realistic sample denoted as $\widetilde{OOB}_t^j$ with an associated $err\widetilde{OOB}_t^j$. Then the variable importance of X^j can be calculated as expressed in Equation 1.

$$VI(X^j) = \frac{1}{ntree} \sum_{t}^{ntree} (err\widetilde{OOB}_t^j - errOOB_t) \quad (1)$$

6.1 Applying *VSURF* Algorithm

The algorithm [28] involves first a preliminary elimination and ranking of the variables, and second an analysis for variable selection. During its computation, the results of *VSURF* algorithm are averaged over many RF runs, which provides more certainty about the results, taking into account the intrinsic random factor of RF due to bagging (bootstrap aggregating). We describe now each one of the three outputs of VSURF algorithm, providing the results and interpretation when applied to our model:

1) Sort the input features by variable importance ($VI(X^j)$) in descending order (averaged over 50 RF runs). It estimates a threshold of minimum VI (based on the VI standard deviation) and removes variables below the threshold, let m be the number of variables left. The m variables selected in descendent order of VI are shown in Figure 2. It is noteworthy to say that no variables were removed in this step, as none of them were below the threshold. A possible explanation is that no variables are redundant and all of them are able to convey some unique information for the prediction of CAMEO submissions, therefore all variables are kept.
2) Constructs a nested collection of RF models involving the k first variables, for $k = 1$ to m. This means that in this collection, the first RF model constructed includes only the most important variable, and the last one includes all the variables. Select the variables which provide the model with the smallest $errOOB$ (averaged over 25 RF runs). This leads to m' variables. The second step reveals that the best model is provided by removing the last three variables *max_attempts*, *attempts_required* and *random* as can be seen in Figure 3, where the red line establishes the cutoff point.
3) The final step takes the m' variables, and constructs a new ascending sequence of RF models by introducing the variables following a stepwise procedure. More specifically, a variable is introduced into the model only if it decreases $errOOB$ more than the average variation provided by noisy variables. Finally, the variables of the last model are selected, Figure 4 shows each model built following the stepwise procedure. After this step, two more variables are removed since these did not improve the model enough. The removed variables in this step are *sum_time_page* and *questions_attempted*. These variables denote an indication of amount of activity and are correlated with other variables measuring student activity e.g. *sum_video_time* [9], thus not improving the model enough to be included.

The last checkup consisted in building a RF model with the final 10 variables selected by the *VSURF* algorithm, and compare it to the model that had the full 15 variables. The test proves that the RF model with only 10 variables performs almost as well as the one with 15 variables.

6.2 Discussion about Variable Importance Results

We originally selected six variables related to student features, six related to problem features and three related to submission features. The *VSURF* algorithm has removed the same ratio of each (1/3) leaving four, four and two features in each category respectively, suggesting that the three categories have influence towards the prediction of CAMEO events. In Section 4 we found that student features have more influence on CAMEO than problem features, and in this section we also want to corroborate if this finding is still true when dividing the influence of student, problem and submissions in several features. In addition, we want to know the specific student or problem features that have a greater importance. Thus, to answer these questions we check the VI order of each feature given by *VSURF* output in previous subsection. The first four variables in terms of VI are *avg_time_correct*, *sum_time_video*, *performance_first* and *attempts_correct_answer*. We can conclude that student features are more important than submission and problem features. We believe this makes sense and it is also in line with the findings of Section 4 regarding the influence of the student and the problem. The fifth and seventh variables in VI order - *time_to_deadline* and *attempt_duration* - are submission features, whereas the sixth, eight, ninth and tenth are *location*, *show_answer*, *type_response* and *type_assignment*, which are problem features. Therefore, it seems that submission features have slightly more importance than problem features, but this hypothesis is not conclusive.

In terms of the order of the variables, we are not surprised to find out that *avg_time_correct*, *performance_first* and *attempts_correct_answer* had some of the highest VI values, since we already found in our research that master accounts had the highest performance in terms of solving questions in the first attempt and doing it very quickly. Additionally, *time_to_deadline* and *attempt_duration* features were also kept, our previous findings already suggested that CAMEO submissions are closer to the deadline and the attempt duration was much shorter (sometimes inhumanly quick) than non-CAMEO submissions. Finally, *location*, *show_answer*, *type_response* and *type_assignment* have been kept as problem features, which is also in line with our previous findings where we saw that students used to apply CAMEO more at the beginning and middle of the course (until they got the certificate). We also found more CAMEO when the show answer button was enabled before

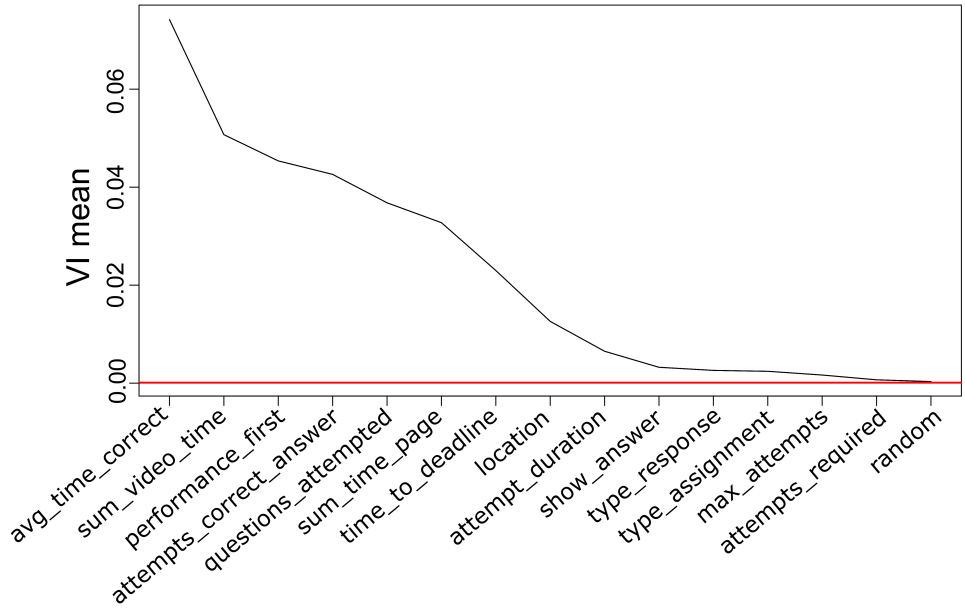

Fig. 2. Descendent ranking of variables in terms of VI.

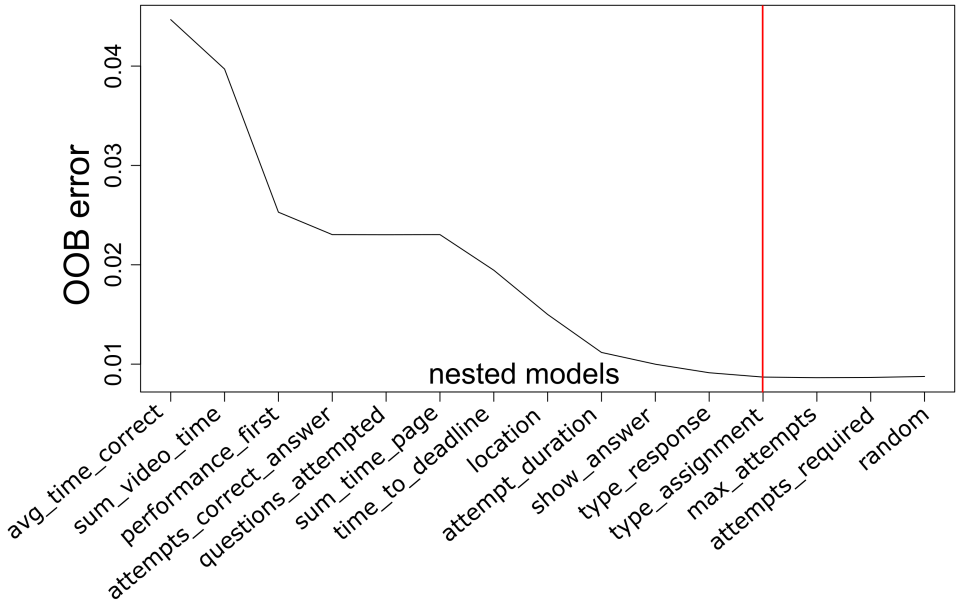

Fig. 3. Nested collection of random forest models.

the assignments' deadline, when the type of response was multiple choice and it was a high stake question in terms of grade (e.g. quizzes instead of checkpoints in our MOOC).

Related to the variables that were removed by the second step of *VSURF* algorithm, we agree that *max_attempts* might not add much information and in the case of *attempts_required* it probably had a high correlation with the student feature *attempts_correct_answer* which was kept as one of the top importance variables of the model. However, we were surprised to see that *random* feature was removed from the model, as we reported in our previous research [8] that CAMEO was found two times less in questions that contain random variables in the statement; we think that the variability provided by *random* feature might be in relationship with other features as well, e.g. *type_response* might provide part of this information as most question with random variables are 'formula' response types and this might be why *random* feature had a low importance. Finally, the two student features *sum_time_page* and *questions_attempted* were removed in the last step of the *VSURF* algorithm despite being variables with a medium importance, we believe that is due to the fact that they denote some indication of the amount of activity of the learner and this information might be provided already by other variables.

7 Conclusions and Future Work

In this study we have proposed a new method for the detection of the CAMEO phenomenon reported in MOOCs

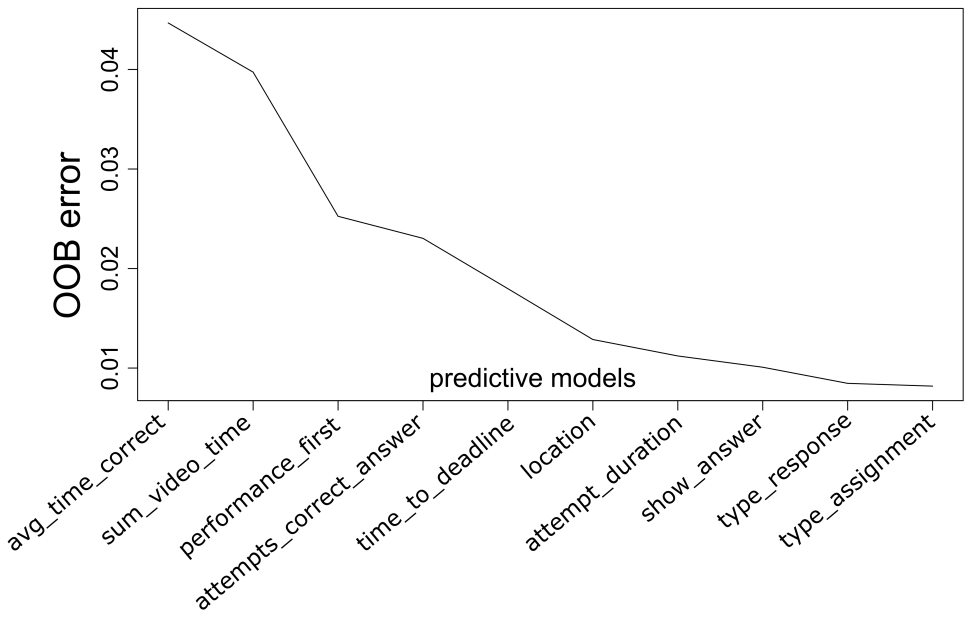

Fig. 4. Sequence of RF models constructed using a stepwise procedure.

[7], [8], [9], [10] based on a machine learning approach that does not rely on the use of IP as detection method. We have designed a RF classification model based on student, submissions and problem features (with a total number of 15 features) without using the IP of the submissions. The model has a high performance offering an AUC close to 1 and a sensitivity and specificity of 0.96 and 0.99 respectively. Since we have achieved such a powerful model, we made a small manual analysis which revealed that indeed some of those false positive submissions are suspicious and could come from academically dishonest behaviors. We trained our RF model and evaluated using 'undoubted' events, but we also test it on the 'suspicious' events, finding that 9.9% of those events were classified as CAMEO, indicating that our previous estimate might need to increment in approximately a 37%.

In addition, we analyzed the dependence of CAMEO on the student and problem. First, we compared the effect of the student and problem on CAMEO as a whole (i.e. involving all the possible features and factors but without defining them). Second, we performed a more in depth analysis by dividing the effect of the student and problem in several features. Finally, we looked into the independent influence of each variable separately. We found that the student has more influence than the problem in CAMEO. The influence of the student is also greater in other studies of gaming the system but there are others where the influence of the problem is greater. The findings back this up both in the importance of the *avg_student* and *avg_problem* variables.We analyzed the importance of the variables included within our RF model using *VSURF* package for feature selection using random forest. Although we found that all the features were related to the predicted variable, student features had the highest VI values followed by submission and problem features; however we should note out that despite the VI is lower, submission and problem features have been kept in the model because these provide additional information. Additionally, we were able to design a model using only 10 features that performs almost as well as the one with 15 features. A future analysis would be to build independent models using only student, submission and problem features separately, and compare which of the models are able to perform better.

One first limitation is that we still need to assess the generalization of the model to other courses. The size of the data sample is small and from a single course, which is a limitation of this study. Additionally, we are training the machine learning model with the tagged sample detected by our previous algorithm [8], [10]. We expect that most of our false negatives (labeled as non-CAMEO by our original algorithm, but that are actual CAMEO) to be within the sample of 'suspicious' submissions which is not used for training, still there might be some false negatives that are influencing the model. The machine learning model is built upon the original algorithm and might inherit some of the limitations. However, as the features that we analyze in both methods are completely different, there is room for the machine learning model to learn the patterns and generalize better than our original algorithm which was based on a fixed set of rules. Additionally, even when the machine learning model might be biased, we believe it to be very valuable since it is a novel approach that shows that it is possible to detect cheating based on machine learning. This can open new machine learning approaches such as anomaly detection for the detection of such behaviors.

The main intrinsic problem of this research is how to establish the ground truth, e.g. we cannot expect students to be honest if we ask them about academic dishonesty. One area of future work would be to develop alternatives to establish a more reliable ground truth, for example, by deploying cookies to uniquely identify when a user is running several accounts using the same device. Furthermore, other possibilities for improvement arise such as training the model with data from several courses to improve gen-

eralization and improve the feature engineering process with new variables such as regarding the complexity of the problem. An important future direction is the detection of cheating by other means such as getting the answer from another student or from an expert outside helper e.g. [29]. One of the possible outcomes from this study would be the implementation of a run-time detector without the use of IP. We can run the model in real time whenever a submission is made. For example sending a warning after 3 submissions detected as CAMEO in a row (which has very little probability of happening randomly).

Another limitation is that this type of method for cheating detection can work when the cheater first gets the correct answer with a harvesting account and next submit it with a master account. For example, the method works well when there are closed questions where the correct answer can be obtained, but the method does not work with peer assessment of essays. Therefore, this type of cheating detection can work in MOOCs that include the use of this type of closed questions but this cheating detection is not valid for MOOCs that are completely evaluated with peer assessment of essays.

Finally, one limitation of this work is that the results might be valid for a specific type of MOOC, this does not affect only to the CAMEO detector based on machine learning (as commented before) but also to which type of variables have more effect on cheating. It would be interesting to test this approach in MOOCs on different topics to see if the conclusions can be valid for other type of courses. Previous work already provided interesting insights regarding students' demographics [7].

ACKNOWLEDGMENTS

The first and second authors want to thank the Madrid Regional Government with grant No. S2013/ICE-2715, the Spanish Ministry of Economy and Competitiveness projects RESET (TIN2014-53199-C3-1-R), the European Erasmus+ projects MOOC Maker (561533-EPP-1-2015-1-ES-EPPKA2-CBHE-JP) and SHEILA (562080-EPP-1-2015-BE-EPPKA3-PI-FORWARD) for partially supporting this work. The authors would like to thank Zhongzhou Chen and Christopher Chudzicki for their help conducting our original research about CAMEO.

REFERENCES

[1] E. G. Lambert, N. L. Hogan, and S. M. Barton, "Collegiate academic dishonesty revisited: What have they done, how often have they done it, who does it, and why did they do it," *Electronic Journal of Sociology*, vol. 7, no. 4, pp. 1–27, 2003.

[2] D. J. Palazzo, "Detection, patterns, consequences, and remediation of electronic homework copying," Ph.D. dissertation, Doctoral Dissertation, Massachusetts Institute of Technology, 2006.

[3] O. R. Harmon and J. Lambrinos, "Are online exams an invitation to cheat?" *The Journal of Economic Education*, vol. 39, no. 2, pp. 116–125, 2008.

[4] M. W. Galbraith and M. S. Jones, "Understanding incivility in online teaching," *Journal of Adult Education*, vol. 39, no. 2, p. 1, 2010.

[5] G. Siemens, "Massive open online courses: Innovation in education," *Open educational resources: Innovation, research and practice*, vol. 5, pp. 5–15, 2013.

[6] A. M. F. Yousef, M. A. Chatti, U. Schroeder, and M. W. andHarald Jakobs, "Moocs: A review of the state-of-the-art," in *Proceedings of the 6th International Conference on Computer Supported Education*, 2014, pp. 9–20.

[7] C. G. Northcutt, A. D. Ho, and I. L. Chuang, "Detecting and preventing "multiple-account" cheating in massive open online courses," *Computers & Education*, vol. 100, pp. 71–80, 2016.

[8] J. A. Ruiperez-Valiente, G. Alexandron, Z. Chen, and D. E. Pritchard, "Using multiple accounts for harvesting solutions in moocs," in *Proceedings of the Third (2016) ACM Conference on Learning@Scale*. ACM, 2016, pp. 63–70.

[9] G. Alexandron, S. Lee, Z. Chen, and D. E. Pritchard, "Detecting cheaters in moocs using item response theory and learning analytics," in *Proceedings of the 6th Workshop on Personalization Approaches in Learning Environments (PALE 2016)*, 2016.

[10] G. Alexandron, J. A. Ruipérez-Valiente, Z. Chen, P. J. Muñoz-Merino, and D. E. Pritchard, "Copying@Scale: Using Harvesting Accounts for Collecting Correct Answers in a MOOC," *Computers & Education*, vol. 108, pp. 96–114, 2017.

[11] R. Baker, J. Walonoski, N. Heffernan, I. Roll, A. Corbett, and K. Koedinger, "Why students engage in "gaming the system" behavior in interactive learning environments," *Journal of Interactive Learning Research*, vol. 19, no. 2, p. 185, 2008.

[12] D. J. Palazzo, Y.-J. Lee, R. Warnakulasooriya, and D. E. Pritchard, "Patterns, correlates, and reduction of homework copying," *Physical Review Special Topics-Physics Education Research*, vol. 6, no. 1, p. 010104, 2010.

[13] M. C. Desmarais and R. S. Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Modeling and User-Adapted Interaction*, vol. 22, no. 1-2, pp. 9–38, 2012.

[14] I. Arroyo and B. P. Woolf, "Inferring learning and attitudes from a bayesian network of log file data." in *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, 2005, pp. 33–40.

[15] R. S. Baker, A. de Carvalho, J. Raspat, V. Aleven, A. T. Corbett, and K. R. Koedinger, "Educational software features that encourage and discourage gaming the system," in *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, 2009, pp. 475–482.

[16] R. S. Baker, A. T. Corbett, and K. R. Koedinger, "Detecting student misuse of intelligent tutoring systems," in *International Conference on Intelligent Tutoring Systems*. Springer, 2004, pp. 531–540.

[17] K. Muldner, W. Burleson, B. Van de Sande, and K. VanLehn, "An analysis of gaming behaviors in an intelligent tutoring system," in *International Conference on Intelligent Tutoring Systems*. Springer, 2010, pp. 184–193.

[18] K. Muldner, W. Burleson, B. Van De Sande, and K. Vanlehn, "An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts," *User Modeling and User-Adapted Interaction*, vol. 21, no. 1-2, pp. 99–135, Apr. 2011.

[19] K. D. Bogle, "Effect of perspective, type of student, and gender on the attribution of cheating," *Proceedings of Oclahoma Academic Science. Oclahoma City*, vol. 80, pp. 91–97, 2000.

[20] S. E. Balbuena and R. A. Lamela, "Prevalence, motives, and views of academic dishonesty in higher education," *Asia Pacific Journal of Multidisciplinary Research*, vol. 3, no. 2, 2015.

[21] K. R. Hamlen, "Academic dishonesty and video game play: Is new media use changing conceptions of cheating?" *Computers & Education*, vol. 59, no. 4, pp. 1145–1152, 2012.

[22] B. A. Wray, A. T. Jones, P. W. Schuhmann, and R. T. Burrus, "Determining the propensity for academic dishonesty using decision tree analysis," *Ethics & Behavior*, vol. 26, no. 6, pp. 470–487, 2016.

[23] G. D. Sideridis, I. Tsaousis, and K. Al Harbi, "Predicting academic dishonesty on national examinations: The roles of gender, previous performance, examination center change, city change, and region change," *Ethics & Behavior*, vol. 26, no. 3, pp. 215–237, 2016.

[24] Y. Kauffman and M. F. Young, "Digital plagiarism: An experimental study of the effect of instructional goals and copy-and-paste affordance," *Computers & Education*, vol. 83, pp. 44–56, 2015.

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

[26] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.

[27] G. Louppe, L. Wehenkel, A. Sutera, and P. Geurts, "Understanding variable importances in forests of randomized trees," in *Advances in neural information processing systems*, 2013, pp. 431–439.
[28] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Vsurf: An r package for variable selection using random forests," *The R Journal*, vol. 7, no. 2, pp. 19–33, 2015.
[29] J. A. Ruipérez-Valiente, S. Joksimović, V. Kovanović, D. Gašević, P. J. Muñoz-Merino, and C. Delgado Kloos, "A Data-driven Method for the Detection of Close Submitters in Online Learning Environments," in *International World Wide Web Conference, WWW'17 Companion*, Perth, 2017.

José A. Ruipérez-Valiente completed his B.Eng. and M.Eng. in Telecommunications at Universidad Católica de San Antonio (UCAM) and Universidad Carlos III of Madrid (UC3M) respectively, graduating in both cases with the best academic transcript of the class. Afterwards, he completed his M.Sc. and P.hD. in Telematics at UC3M while conducting research at Institute IMDEA Networks in the area of learning analytics and educational data mining. During this time, he completed two research stays of three months each, the first one at MIT and the second one at the University of Edinburgh. He has received several academic and research awards and has published more than 25 scientific publications in important journals and conferences of his area of research. Previous to his research appointments he worked in industry at the companies Vocento and Accenture, and currently he is working as data scientist at ExoClick.

Pedro J. Muñoz-Merino is a Visitant Associate Professor at the Universidad Carlos III de Madrid, where he is the Director of the Master in Telematics Engineering. He obtained his acreditation in May 2012 as Associate Professor by the ANECA agency from the Spanish Ministry of Education. Pedro has received several awards for his work on educational technologies. He has done two long research visits: one in Ireland for more than 3 months at the Intel company in 2005, and another in Germany for more than 6 months at the Fraunhofer Institute of Technology in 2009-2010. He is author of around 100 scientific publications and has participated in more than 20 research projects at the national and international level, coordinating some of them with private companies. His present research interests include learning analytics and educational data mining. He has been Programme Committee member and part of the organization team of different conferences and workshops related to educational technologies and learning analytics. In addition, he has been invited to give different talks related to learning analytics topics. He has also coordinated the development and deployment of different learning analytics tools. He is an IEEE Senior member since 2015

Giora Alexandron is the Principal Data Scientist of The Center for Educational Technology, and previously a postdoctoral researcher in the Physics department at MIT. His research centers on educational data science, and he is especially interested in developing algorithms and techniques for adaptivity and analytics in online learning environments. He holds PhD in computer science education from the Weizmann Institute of Science, and MSc in computer science from Tel-Aviv University.

David E. Pritchard educated at Caltech (BSc 1962) and Harvard (PhD 1968), has been with Massachusetts Institute of Technology ever since, where he is now Cecil and Ida Green Professor of Physics. He has won both the Broida and the Schawlow prizes from APS, the Max Born Award from OSA, and the IUPAP Senior Scientist Medal in Fundamental Metrology. He is a member of the National Academy of Sciences, and a fellow of the American Academy for Arts and Sciences, the American Association for the Advancement of Science, the American Physical Society and the Optical Society of America. He has mentored three Nobel prizewinners and four winners of national thesis awards, and has won both a Dean's Teaching and Advising Award and the Earll M. Murman award for advising at MIT. Pritchard has a lifelong interest in teaching problem solving and is the author of A Mechanics Workbook, and the PI of an education group (Research in Learning, Assessing, and Tutoring Electronically). He co-founded Effective Educational Technologies (sold to Pearson Education in 2006, which developed MasteringPhysics.com, MasteringChemistry.com, that were used by – 4M students last year, 400k in physics alone. His current research focus is entirely in education, particularly in developing online and on-land activities to help students improve their strategic thinking and expertise of their learning attitudes.