Prediction of Dilatory Behavior in eLearning: A Comparison of Multiple Machine Learning Models

Christof Imhof, Ioan-Sorin Comsa, Martin Hlosta, Behnam Parsaeifard, Ivan Moser, and Per Bergamin

Abstract—Procrastination, the irrational delay of tasks, is a common occurrence in online learning. Potential negative consequences include a higher risk of drop-outs, increased stress, and reduced mood. Due to the rise of learning management systems (LMS) and learning analytics (LA), indicators of such behavior can be detected, enabling predictions of future procrastination and other dilatory behavior. However, research focusing on such predictions is scarce. Moreover, studies involving different types of predictors and comparisons between the predictive performance of various methods are virtually nonexistent. In this article, we aim to fill these research gaps by analyzing the performance of multiple machine learning algorithms when predicting the delayed or timely submission of online assignments in a higher education setting with two categories of predictors: 1) subjective, questionnaire-based variables and 2) objective, log-data-based indicators extracted from a learning management system. The results show that models with objective predictors consistently outperform models with subjective predictors, and a combination of both variable types performs slightly better with an accuracy of 70%. For each of these three options, a different approach prevailed (gradient boosting machines for the subjective, Bayesian multilevel models for the objective, and Random Forest for the combined predictors). We conclude that careful attention should be paid to the selection of predictors and algorithms before implementing such models in learning management systems.

Index Terms—Dilatory behavior, learning analytics (LA), machine learning (ML), predictive performance, procrastination.

I. INTRODUCTION

The ubiquity of digitization continues to pervade many aspects of life, including learning. While this ongoing trend entails numerous advantages for students, such as more flexible schedules, more engaging distance education, or a greater variety of interactive learning tasks [1], the drawbacks also need to be considered. One potential drawback is the increased likelihood of academic procrastination due to the many distractions our digital world offers. Academic procrastination is a type of dilatory behavior that is commonly defined as the irrational delay of academic tasks such as writing assignments or studying course literature [2]. While seemingly harmless, procrastination may lead to various negative consequences, which include increased stress, lower performance, and reduced mood [3]. Moreover, this behavior is linked to an increased risk of drop-out [4] and is reportedly highly prevalent among students in higher education [2], [5].

However, not all acts of delay are necessarily maladaptive since they can also be used as a deliberate strategy. The positive, productive counterpart to procrastination is known as purposeful delay [6]. Alternative names for this concept include active delay, strategic delay, or active procrastination, the latter of which is considered to be an oxymoron [7]. The term dilatory behavior thus serves as an umbrella term that refers to both positive and negative forms of delay.

A promising approach to measuring indicators of dilatory behavior comes in the form of Learning Analytics (LA). LA is often implemented as a conceptual framework to analyze course characteristics, such as prediction of students’ learning performance, educational data analysis, data collection and measurement, and early intervention [8]. Given the potential consequences of negative dilatory behavior, the ability to detect it early utilizing an LA framework could prove very useful to teachers, lecturers, and students alike. This especially concerns learning management systems (LMS), which are able to provide objective indicators for such behavior. The earlier the detection, the more time remains to intervene and ultimately prevent potential drop-outs.

Predicting dilatory behavior in academic tasks is not trivial for multiple reasons, especially in the field: Various courses must be incorporated to allow for a generalization of such predictions and their timing also needs to be considered. In this context, data collection and the type of variables (be they predictors or outcome variables) play a crucial role. Despite the variety of types of Machine Learning (ML) models that could be employed, their predictive performance may vary as a result of the size of the collected dataset, the number of predictor variables, and the point in time predictions are made. Therefore, the selection of an appropriate prediction model must involve such considerations.

Manuscript received 17 February 2022; revised 13 September 2022; accepted 29 October 2022. Date of publication 17 November 2022; date of current version 16 October 2023. This research was supported by a grant from the Canton of Valais. (Corresponding author: Christof Imhof.)

This work was performed while the authors were with the Institute for Research in Open-, Distance- and e-Learning, Swiss Distance University of Applied Sciences, Ch-3900 Brig, Switzerland (e-mail: christof.imhof@ffhs.ch; ioan-sorin.comsa@ffhs.ch; martin.hlosta@ffhs.ch; behnam.parsaeifard@ffhs.ch; ivan.moser@ffhs.ch). Per Bergamin is with the Institute for Research in Open-, Distance- and e-Learning, Swiss Distance University of Applied Sciences, Ch-3900 Brig, Switzerland, and also with North-West University, Potchefstroom 2531, South Africa (e-mail: per.bergamin@ffhs.ch).

© 2022 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
Previous studies indicate the great potential of ML-based prediction models in LA [9], [10], [11], even when predicting dilatory behavior [12]. To our knowledge, however, comparisons between multiple ML algorithms are rather rare in this context (see [12] and [13]). Moreover, a delay is usually treated as a predictor for other outcome variables rather than being the outcome variable itself. To fill this gap, our article provides the following contributions.

1) We employ eight different ML algorithms to predict dilatory behavior in online assignments.
2) Each of these algorithms was optimized through an innovative method that enables high predictive performance in validation and test datasets.
3) We provide a comparative analysis to determine which algorithms deliver the best predictions for which type of variables and compare the performance with Bayesian models as a baseline.

For our analyses, we repurposed the same dataset as in our prior study, where the prediction of dilatory behavior was investigated, specifically the effect sizes of objective and subjective predictors using Bayesian multilevel regression [14].

The rest of the article is organized as follows: In Section II, we present the related work on dilatory behavior and the ML algorithms employed to predict such tendencies in various tasks. The analytical framework and our research questions are also detailed in this section. In Section III, we then highlight the methods and instruments used to collect the data, followed by an introduction to the prediction framework and a brief illustration of each ML algorithm. A novel cross-validation procedure is also presented to determine the best set of hyperparameters for all of the models. Next, we present our results in Section IV, which includes the results of the cross-validations and the test results of each ML model, which are compared with the performance of the Bayesian multilevel models. This is followed by a discussion of our results in Section V, including the implications and limitations of our findings. Finally, the article concludes with Section VI.

II. RELATED WORK

A. Theoretical Background of Dilatory Behavior

As stated in the introduction, dilatory behavior is an umbrella term that encompasses at least two types of delay, 1) procrastination and 2) purposeful delay. However, another important distinction should be made when investigating delay, namely the one between trait and state delay. State variables are situational (e.g., mood), whereas trait variables are stable with less situation- or time-specific variance (e.g., personality). In the case of procrastination, the correlation between trait and state (or in other words, attitude and actual behavior) is reportedly medium to large at 0.51 [2]. This not only implies that situational factors are important when trying to predict delay, but also that trait procrastination can still serve as a predictor for state procrastination (or vice versa), albeit not a particularly strong one.

This train of thought leads to the question, which additional predictors besides trait procrastination should then be considered when predicting procrastination and other types of dilatory behavior. Two broad categories present themselves: 1) subjective and 2) objective predictors. The former is commonly assessed with questionnaires while the latter can be gained by observation, or in the case of LMS, collection of log data.

When predicting delay with subjective predictors, motivational aspects and learning-related factors suggest themselves. One option is factors in the orbit of self-regulation, given that procrastination is often considered a “self-regulation failure” [2] whereas purposeful delay appears as its successful counterpart [6]. Learning concepts related to self-regulation include academic self-efficacy and self-directed learning (SDL). Academic self-efficacy refers to the belief in one’s ability to succeed at academic tasks such as writing exams [15] and is negatively linked to procrastination [16], forming a vicious circle, but positively connected to purposeful delay, countering that circle [17].

SDL is a process in which students take responsibility for their own learning, which involves self-monitoring, self-management, and self-evaluation of the learning process [18], [19], [20]. Research reveals positive links between SDL and self-efficacy [21] and negative links with procrastination [22].

In the context of LMS, it is self-evident to consider objective predictors based on data extracted from such systems. Popular approaches in LA are to analyze click-based data (e.g., [23], [24]), which provide insights into student activity and engagement with learning content on the platform.

B. Delay-Related Prediction Models in LA

Prediction models that involve delay as a predictor are quite common in the literature. A variety of methods has been implemented to investigate the relationship between procrastination (as the most commonly researched type of delay) and other variables, e.g., academic drop-out. For instance, Cervera-Mercadillo and Lara [25] proposed a log data-based method to potentially predict drop-outs by generating features representing the degree of delay in recommended tasks [25]. Another crucial variable of interest is academic achievement. A first example is [26], who used multiple regression analysis to investigate the effects of procrastination on course achievement. Their regression model predicted achievement at each point in time, with the predictability increasing as time passed.

Levy and Ramim [27] instead implemented data analytics techniques to detect anomalies in their data, corroborating the finding that procrastination negatively predicts performance (grades). The same conclusion was reached by Gareau et al. [28], who used structure equation modeling and discovered a mediating role of task-oriented and disengagement-oriented coping in the relation between trait procrastination and
achievement. Another technique, path analysis, was applied in [29], where Yamada et al. found that positive time management promoted early submissions of paper reports. A different method, association rule mining, was implemented in [13] and [30]. Both research groups identified indicators of procrastination on Moodle, which were used to create association rules in the form of conditionals by implementing either Apriori and/or Predictive Apriori algorithms. The indicators were either based on the sum of clicks on certain activities, the time spent on them, or time intervals until students performed specific actions. In both studies, time-related indicators were closely related to students’ performance.

In contrast, models that explicitly try to predict delay as the outcome variable are rather scarce. Exceptions include studies conducted in [12], [31], [32], and [33], which all intended to classify students based on homework submission data. Ten different ML algorithms [ZeroR, OneR, ID3, J48, Random Forest (RF), decision stump, JRip, PART, NBTree, and Prism] were implemented in [12] to classify students as procrastinators or nonprocrastinators based on feature vectors. The optimal amount of clusters was determined to be three (one nonprocrastinating group and two procrastinating classes that differ in submission scores).

Multiple clustering algorithms were also used in [31] and [33] to detect the optimal amount of clusters, followed by a classification of students into the three resulting clusters (procrastinators, nonprocrastinators, and procrastinator candidates). In both studies, the authors compared eight methods of classification [support vector machines with linear and radial basis kernel functions, Gaussian Processes, decision tree, RF, neural network, AdaBoost, and Naïve Bayes (NB)]. The former implemented a feature vector algorithm involving categorical and continuous features based on spare time (i.e., interval between submission and deadline) and inactive time (i.e., time before the first click on an assignment). They found that neural networks worked best with categorical features and that linear support vector machines outperformed the others in the case of continuous features. The latter reported that the linear support vector machine approach delivered the best predictive performance for their clusters.

Binary classifiers were employed in [32], using four supervised-learning algorithms [logistic regression, decision tree, gradient boosting machine (GBM), and RF] to classify students as procrastinators or nonprocrastinators based on data that was extracted from an intelligent tutoring system (ITS). Among the ML algorithms, gradient boosting had the best performance in terms of classification precision. However, these examples involve a classification of students into clusters, rather than predictions regarding the extent of the delay itself. This research gap was one of the reasons we conducted a previous study to determine if dilatory behavior could be predicted based on a mixture of questionnaire scores and log data [14].

In that study, seven predictors were implemented across six different Bayesian multilevel models to determine which type of predictor (objective versus subjective) and which individual predictors were able to predict dilatory behavior the best. The four subjective predictors were questionnaire scores and the three objective predictors were based on log data. The model fit comparison favored the models that included all seven predictors, but their advantage over the models that only included objective predictors was minimal at best. This implies that the models with objective predictors barely improve when subjective predictors are added. However, this result does not imply that subjective predictors should be discarded completely. First off, the results may not be the same when following other approaches for prediction models, for instance, models based on ML. Second, we focused on individual predictors and comparisons between types of predictors rather than analyzing the actual performance of the predictions in terms of accuracy and other measures.

C. Research Questions

Therefore, the goal of this study was to extend the findings reported in [14] by comparing and contrasting the performance of different approaches to prediction models, determining which ML algorithm delivers the best predictions for delay and for which type of predictor.

Before the predictive performance of these models can be assessed, it needs to be clarified how the hyperparameters for each ML algorithm and type of predictor must be determined in order for them to be optimized. We present a novel cross-validation approach, the $G$-score, to address this and compare the results of the cross-validation with the test results. We then intend to identify the ML model with the best predictive performance for the following three predictor types: 1) subjective (subj), 2) objective (obj), and 3) a combination between the two (comb). Finally, we determine whether models with objective predictors still outperform models with subjective predictors when calculated with various ML algorithms and whether there is still an advantage of models that combine both types of predictors for all of our algorithms. The research questions are thus as follows.

$RQ1$: How well do the results of the cross-validation with the proposed measure ($G$-score) compare to the test results?

$RQ2$: Which ML algorithm improves the predictive performance compared to the baseline models based on the subjective, objective, and combined sets of predictors (intragroup comparison)?

$RQ3$: Which type of predictors (subjective, objective, and combined) allow for the highest predictive performance (intermodel comparison)?

III. METHODS AND PREDICTION MODELS

A. Participants and Courses

We used the same dataset as in a prior study, which included 134 students from a distance university located in Central Europe [14]. The sample included 65 male and 69 female participants (mean age 31.61 years, SD = 7.97, min = 19, max = 61). The students were enrolled in at least one course each during the autumn semester of 2019. The number of involved online courses was 126, each belonging to 1 of 3.
departments (computer science, economics, and health). Every course consisted of several blocks (usually between 5 and 10), each with their own assignment(s). The total number of assignments across all students and courses was 1107. The students participated voluntarily and consented to have their log data extracted from the institution’s LMS by filling in an online survey. As compensation, all participants automatically entered a raffle with a chance of winning cinema vouchers.

B. Instruments, Procedures, and Variables

As described in [14], the procedure started with an e-mail invitation that was sent to all students enrolled at our institution at the end of the semester. The volunteers then followed a link to an online survey, consisting of four questionnaires, whose scores formed the subjective predictors for our models: the General Academic Self-Efficacy (GASE) Scale [34], consisting of 5 items, the SDL Scale (SDLS) [35] with 10 items, the Academic Procrastination Scale—short form (APSS) [36] with 5 items, and the Active Procrastination Scale (APS) [37] with four subscales and a total of 16 items. The questionnaires captured four concepts that are all closely tied to dilatory behavior on a theoretical level, supported by empirical evidence (see [14] for an overview): 1) academic self-efficacy, 2) SDL, 3) procrastination (trait), and 4) purposeful delay (trait), respectively.

The three objective predictors were the number of clicks on an assignment, the interval between the start of a block and the first click on an assignment, and the number of clicks on relevant activities in the course. As explained in [14], these predictors were chosen for their utility as common, readily available indicators of online activity with high face validity and empirical support (see [13], [30]). The number of clicks on an assignment reflects the sum of all clicks on the assignment section of the course made before the deadline had passed. The second predictor, the interval between the start of a block and the first click on an assignment, indicates the time (in days) that passed between when a block had started and when the first click on the assignment was made (i.e., when the task description was first read). The final objective predictor was the number of clicks on relevant activities in the course, which reflected the overall engagement with the course material on the platform (i.e., the sum of all clicks on learning videos, forums, interactive books, etc.). The interval turned out to be the strongest and most consistent individual predictor in the previous study, followed by the number of clicks on the assignment. The to-be-predicted outcome variable was delay, which can assume positive and negative values, with positive values indicating a delayed submission of an assignment (in hours) and negative values meaning a timely submission. In this study, delay was used for regression and classification alike. This required two measures: 1) the extent of delay and 2) the category of delay; the former to determine the error between predicted and actual values, and the latter to determine whether the two classes (delay versus timely submission) would be correctly predicted.

In Fig. 1, we illustrate the architecture of our system. Our dataset, consisting of the subjective and objective predictors detailed above, was processed in multiple steps (involving cross-validation, training, and testing via multiple ML algorithms and corresponding evaluations), leading to several outputs, including the delay measures. The results allow us to draw conclusions about the type of data (objective versus subjective), the type of algorithm (which one performs best for which type of predictor), and further theoretical or practical implications. The figure also highlights the differences between this study and its precursor [14], as explained at the end of Section I. The aspects shared by both papers are indicated by half-circles while those unique to this article are represented by fully-filled circles.

In Table I, we provide an overview of the characteristics of the outcome variable, the four subjective, and the three objective predictors (i.e., their role in the models, their mean, standard deviation, minimum and maximum values, and a brief
description). These characteristics are very useful in adopting different normalization techniques (e.g., standard score or min–max feature scaling). In this article, we adopt the max-absolute normalization technique to have the block-click-interval and delay features in the interval of $[-1, 1]$.

C. ML Models

Before training the models, we first removed all missing values (e.g., assignments that were not handed in at all) and then normalized the data. The processed dataset (1107 rows, each representing one individual assignment) was then split into training and testing sets, the former including 80% of the data (885 rows) and the latter 20% (222 rows). Since the original dataset is rather small, the subsets could differ in their distributions depending on the way the dataset is split. We addressed this concern by repeating the randomized splitting process 10 times, thus creating 10 pairs of training and testing sets. When individually compared, the testing subsets shared 44.7 rows on average ($SD = 5.69$, $min = 31$, $max = 57$), ruling out potential split-related biases. For further processing, we then split all training and testing sets into three subsets each based on the type of predictor:

1) a subset with subjective predictors;
2) one with objective predictors;
3) a subset with both types of predictors combined.

All sets and subsets had an equal number of rows, the only difference being the included predictors.

For every training set, we then determined the hyperparameters of each ML algorithm based on cross-validation. Once the models’ hyperparameters were fixed, we trained each ML algorithm by individually exposing them to all ten training datasets to learn predicting the outcome variable delay based on the subjective, objective, or both sets of predictors combined. Our objective was twofold: 1) predicting real values in terms of delay and 2) assessing how accurate the classifications were. Therefore, the ML algorithms to be analyzed in this article all needed to be able to produce both regression and classification models. We thus chose the following eight algorithms: NB, K-Nearest neighbors (KNN), Radial Basis Function Networks (RBFN), Feed-Forward Neural Networks (FFNN), Regression Trees (RT), GBM, RF, and Support Vector Regression (SVR). Out of this selection, we aimed to determine the algorithms that 1) obtained the best classification performance when predicting the delay and timely-submission classes; and 2) minimized the error between the predicted and actual delay values from the testing datasets. In the following, we introduce each of these techniques, starting with the Bayesian multilevel models that serve as the baseline for our comparisons.

1) Bayesian Multilevel Models: We selected Bayesian multilevel regression models as the baseline for our comparisons since they were already implemented in [14], whose data this study are based on. In that study, Bayesian multilevel models were chosen to match the nested data structure (students being enrolled in multiple courses, each with their own assignments). The authors favored a Bayesian approach over frequentist regression models as they intended to use the study results to inform the priors of the models in a planned follow-up study. In total, six models were calculated: One without any predictors to serve as their baseline (which we do not include in this study), one with questionnaire predictors only, two with log data predictors only (one with random intercepts and the other with additional random slopes), and two with all seven predictors (again with random intercepts and additional random slopes, respectively). Multilevel models allow the relationship between predictors and outcome variables to vary depending on a grouping factor, which is recommended when there is no valid (e.g., theoretical) reason to assume the relationship remains the same for all values of that factor. For this reason, random slope models were calculated. Depending on the predictors, the grouping factors were the student or the course, with the questionnaire scores being associated with the level 2—grouping factor student and the number of clicks on relevant activities being associated with course. The remaining predictors were associated with the assignment and thus located on level 1.

| Variable | Role | Mean | SD  | Min | Max | Description |
|----------|------|------|-----|-----|-----|-------------|
| GASE     | subjective predictor | 19.71 | 3.10 | 8   | 25  | General Academic Self-Efficacy Scale, measures belief in one’s academic abilities |
| SDLS     | subjective predictor | 38.74 | 5.29 | 18  | 50  | Self-Directed Learning Scale, measures how much one feels in charge of one’s learning process |
| APSS     | subjective predictor | 11.04 | 4.16 | 5   | 21  | Academic Procrastination Scale – Short Form, measures tendency to procrastinate in academic tasks |
| Number of clicks on assignment | objective predictor | 72.51 | 11.87 | 43  | 104 | Active Procrastination Scale, measures purposeful delay, i.e., tendency to delay tasks strategically |
| Interval between start of block and first click on assignment (days) | objective predictor | -7.13 | 32.40 | -150.30 | 98.72 | Difference between start of a block (learning unit on Moodle) and the first click on an assignment (days) |
| Number of clicks on relevant activities in the course | objective predictor | 173.83 | 168.44 | 0   | 1237 | Sum of all clicks on activities (quizzes, videos, books, etc.), indicates general activity in the course |
| Delay (days) | outcome variable | -1.66 | 18.26 | -113.51 | 132.43 | Difference between deadline of an assignment and time of submission, positive values indicating delay and negative values meaning early submissions |
2) *Naive Bayes*: NB utilizes Bayes theorem and assumes that features are independent. In this article, we use NB for both classification and regression. When applying NB to regression, we first divide the entire range of the outcome variables into \( N \) parts and consider each part as a new class. The algorithm then classifies them based on the features and finally reassigns the target values to the classes. The only hyperparameter of the NB model is the number of splits, which is determined through cross-validation.

3) *K-Nearest Neighbors*: KNN is a supervised learning algorithm used for both classification and regression. In order to predict the class of a new query point in KNN, we first find its KNNs in the training set and then assign the class of the majority of its neighbors to it. The most common metric for measuring the distances and closeness of the points is the Euclidean distance. We can also have nearer neighbors contributing more by assigning a weight proportional to the inverse of distance to the neighbors. When applying KNN to regression problems, the (weighted) average of the neighbors’ target values is calculated and assigned to the test point. The hyperparameter \( k \) is determined based on cross-validation.

4) *Radial Basis Function Network*: Conceptually, a neural network is a nonlinear function in which the weights must be tuned through training data samples to fit the testing data or unseen observations. RBFN is a simple type of a neural network with the weights structured in only two layers, 1) a hidden layer and 2) an output layer. RBFN differs from other neural networks in the type of activation function used in the hidden layer. Each hidden neuron makes use of a radial activation function that calculates the Euclidean distance between each data point and some data centers which are *a priori* computed through clustering algorithms over the training set. For regression tasks, the activation function of the output neuron is a linear one. When training the RBFN, the weights of hidden and output layers are updated in iterations based on the reinforced error between the predicted and real values. In this sense, the gradient descent algorithm is used to train the weights, where the error rate is an important hyperparameter to be optimized together with the number of hidden nodes and the Gaussian parameter gamma.

5) *Feed-Forward Neural Network*: FFNN differs from RBFN in the sense that the number of hidden layers can be greater than one, the activation function at the level of each hidden node takes simple nonlinear representations (e.g., tangent hyperbolic) and at the output layer, the activation function is linear. The weights are trained similarly to RBFN in iterations, but the error is back-propagated each time through a greater number of layers, and the weights are updated by using the same gradient descent algorithm. Alongside the error rate, the number of hidden layers and nodes must be decided through cross-validation before training and testing the FFNN structure.

Compared to RBFN, training an FFNN is less complex since a clustering approach is no longer needed. Generally, the most notable disadvantage when training any type of neural network is the lack of criteria when to abort the learning process to prevent overfitting. In this article, we propose the following approach to deal with the issue of setting a proper stopping criterion: Out of the training data, a number of validation samples are carefully selected to monitor the performance of the training process. Instead of imposing a fixed stopping criterion, we conducted the training over a large number of iterations and a multiobjective function (to be detailed in Section III-D). The function aimed at balancing the classification performance and minimizing the regression error and was computed over the validation data. The weights of the neural networks are saved each time a new maximum value is found during the training process.

6) *Regression Trees*: The training process of RT is known as a binary recursive partitioning, which splits training data into partitions or branches. The algorithms work iteratively and continue splitting each partition into smaller subpartitions as the training process moves up to each branch. The process of building the tree until each node reaches a specific minimum node size and becomes a terminal node. Once this node is reached, all the responses from all data points are averaged. When testing the RT, each new point follows the split values and variables given by the RT, which was built based on the training data, and the predictions are given based on the response of the terminal node each new sample ends on. When the minimum node size is 1, the RT can overfit the training data. In this article, we determine the minimum node size for each training set based on cross-validation. A very important aspect that concerns the performance of RT is the splitting rule. Standard split criteria (e.g., linear rank statistics, log-rank statistics) cannot detect nonlinear effects in the outcome variable. To overcome this potential drawback, we use the maximally selected rank statistics for selecting the split point in which splitting variables are compared on the \( p \)-value scale [38]. Alongside minimum node size, the lower quantile of the co-variate distribution and the significance level for splitting are all determined through cross-validation.

7) *Gradient Boosting Machines*: Compared to RT, GBM greedily constructs several trees. A new tree is constructed in each round, minimizing the errors made by the previous trees [39]. The commonly tuned parameters include the number of trees, maximum depth, minimum samples to split the data, learning rate, and the type of loss function. Moreover, by manipulating the subsample parameter, a stochastic version of GBM [40], uses bootstrapping averaging similar to RF models, when each iteration is trained only on a fraction of the data.

GBM and its variant XGBoost are used in many existing deployed LA systems [41], [42] and were used in winning solutions for predicting student drop-out in the KDD15 competition [43]. Their high performance is due to their ability to learn from previous mistakes without requiring normalization. As a drawback, the model might suffer from overfitting, which needs to be overcome by cross-validation and exploring a relatively large number of hyperparameters.

8) *Random Forest*: In regression tasks, RF collects the efforts of multiple decision trees and predicts each new data point based on the average responses of all the trees. To
decrease the variance of the model without increasing the bias, the trees must be uncorrelated. In this sense, a bootstrapping procedure is applied, and the training subset of each tree is randomly sampled with replacement in the standard proportion of 63.21% from the overall training set [44]. Moreover, a random subset of features will be considered at each candidate split when training the RTs. This overcomes a high degree of correlation between trees when some variables are very strong predictors for the delay variable. As a splitting rule, in this article, we employ the principle of ExtraTrees in which random cut-points are selected in the top-down splitting process [44]. For each possible splitting variable, a number of random cut-points are generated, and the one with the highest decrease of impurity is selected to split the node. Therefore, the RF algorithm adjusts the following four hyperparameters based on cross-validation:

1) the number of RTs;
2) the minimum node size for all trees;
3) the number of possible variable splits;
4) the number of cut values that are randomly generated in the range of min/max values for each possible split variable.

RF is less prone to overfitting compared to simple RT or GBM. However, the RF algorithm may change considerably by a small change in the training data. Moreover, the RF training time increases considerably in larger datasets, especially when computing the optimal cut-point locally for each feature (i.e., based on log-rank statistics, maximal rank statistics, etc.).

9) Support Vector Regression: Being originally proposed for binary classification problems, support vector machines aim to find the best hyperplane that separates a given set of data points. Training points are mapped in space so that the width of the gap or the soft margin between the two categories would be maximized. Test points are mapped in this space and predicted as being part of one of such categories depending on which side of the gap they belong. In practice, the soft margin that separates the two categories of data points can be controlled based on parameter C. When C is relatively small, a larger separation margin between the classes will be used at the price of a higher rate of misclassifications. When C is higher, the separation gap gets smoother and the rate of misclassifications is lower. A lower misclassification rate increases the risk of overfitting the training data, which can lead to a much higher rate of misclassification for new data points. An optimal value of soft-margin hyperparameters must be found on training data to fit the testing data.

The accuracy of classification depends on the shape of the separation hyperplane that can be set through the kernel function. In this article, we test different types of kernel functions (linear, polynomial, radial basis function, radial basis function with vector subtraction, and hyperbolic tangent), which can be used to best fit the collected data. Moreover, some of these kernel functions must be parameterized before properly training and testing the machine. In regression tasks, the goal of SVR is comparable to classification tasks with the amendment that a certain error is tolerated. In this article, we optimize the soft margin parameter, the kernel function, and error tolerance through cross-validation.

D. Proposed Cross-Validation Measure

In order to optimize the hyperparameters of the implemented ML algorithms (NB, KNN, RBFN, FFNN, RT, RF, GBM, SVR), we employed a cross-validation procedure based on the training sets. Each ML model defines its own grid of hyperparameter configurations that fall in predefined ranges. Each possible configuration of hyperparameters from the grid is then evaluated based on the training data. Finally, the trained ML model with the highest evaluation outcome is employed to predict new data samples from the testing set.

The algorithm splits the training set in K number of subpartitions. Then, the K-fold cross-validation trains the ML model iteratively for each hyperparameter configuration based on K − 1 subpartitions and the results are validated based on the remaining subpartition every time. For each configuration, the validation results are averaged over K number of partitions. Once all hyperparameter configurations are evaluated, the K-fold cross-validation algorithm selects the configuration with the highest validation outcome. The same procedure is repeated for each ML algorithm with its predefined grid search of hyperparameter configurations. The validation outcome can be measured by monitoring the regression error [e.g., mean absolute error (MAE)]. However, in our prediction task, the selected hyperparameters should also improve the classification performance. Moreover, the ratio of early and late submissions in the training data is often unbalanced. Therefore, we propose a novel multiobjective function that measures the validation outcome of each hyperparameter configuration on each split of the dataset. Based on this proposed function, the cross-validation selects the hyperparameter configuration that minimizes the regression error and maximizes the classification performance between the two classes (delay and timely submission).

Given the function f that must be trained by each ML model to best predict delay, we denote by ŷ = f(x) the current estimate of prediction function f, where x is a test data point. In order to evaluate the prediction performance, the ground truth values of delay y should be known in advance. In this study, delay y is measured as the difference between the submission of an assignment and its deadline in days. The deadline is met when y ≤ 0 (timely submission), and the tasks are submitted after the deadline when y > 0 (delay). When a new data sample from the test set is predicted, the decision can take one of the following forms.

1) True positive (TP): ŷ > 0 and y > 0.
2) False positive (FP): ŷ > 0 and y ≤ 0.
3) True negative (TN): ŷ ≤ 0 and y ≤ 0.
4) False negative (FN): ŷ ≤ 0 and y > 0.

Once all test samples are predicted, the classification performance is evaluated based on TP, FP, TN, and FN indicators, which are summed over the number of test data examples. To measure the ML model performance when dilatory behavior is correctly detected, the F1-score function can be employed as follows:

\[
F_1 = \frac{2TP}{2TP + FP + FN}
\]
where a higher $F_{tp}$ value implies a better performance when predicting delay. In unbalanced datasets, $F_{tp}$ does not give any measure of how well the prediction performance is for the timely submission class. In this case, a similar version of $F$-score measure can be computed to monitor the ratio of $TN$ over the number of false predictions:

$$F_{tn} = \frac{2TN}{2TN + FP + FN}.$$  

(2)

By maximizing both $F_{tp}$ and $F_{tn}$ in cross-validation, the hyperparameters will be selected according to the best prediction balance between the two classes. To measure how close the predicted value $\hat{y}$ is from its delay pattern $y$, we measure the MAE $E$ given by

$$E = \frac{1}{T} \sum_{i=1}^{T} \frac{|y_i - \hat{y}_i|}{y_{\text{max}}}$$  

(3)

where $T$ is the number of samples in each dataset split and $y_{\text{max}}$ is the maximum absolute value of delay. Considering (1), (2), and (3), the multiobjective function to be maximized during the cross-validation process becomes

$$G = \frac{(1 - E) + F_{tp} + F_{tn}}{3}.$$  

(4)

The configuration of hyperparameters able to maximize (4) will be selected to train the ML model for the entire training set, and the prediction performance is analyzed in Section IV. Each ML algorithm follows the same cross-validation principle when tuning the corresponding hyperparameters.

E. Analyses

The baseline models (i.e., the Bayesian multilevel models) were calculated with the R package brms [45], and the NB, KNN, and GBM algorithms were employed using the Python scikit-learn package [46]. The remaining ML approaches (FFNN, RBFN, RT, RF, SVR) were deployed in C/C++ using dedicated functions for cross-validation, training, and testing. For the obj and comb predictors, the Bayesian models included random intercepts (BA-RI) and random slopes (BA-RS) due to the nested structure of the data, which was not the case for the subj predictors. The implementation of RT and RF was based on RANGER C++ packages, which are publicly available [44]. The proposed SVR ML algorithm followed the regression model from [47] with five different types of kernel functions: 1) linear (SVR-LIN), 2) polynomial (SVR-POL), 3) tangent hyperbolic (SVR-TAH), 4) radial-basis function (SVR-RBF), and 5) with the vector subtraction function (SVR-VS). All these variants of the SVR model are cross-validated, trained, tested, and compared to the other ML algorithms. The FFNN model we used was previously employed for the optimization of video quality in remote education [48] while the RBFN model was also used to classify high-dimensional vectors in radio communications systems [49].

The complexity of cross-validation processes depends on the grid size, which differs from one ML algorithm to another. A special case in cross-validation is represented by RBFN, where the number of hidden nodes is equivalent to the number of data centers computed for each dataset separately based on the clustering analysis. The clustering process is conducted before cross-validation by employing a heuristic algorithm that iteratively combines a classical k-means algorithm for a more precise calculation of centers with an algorithm that uses the random swapping of data centers from the available dataset to enhance the searching time of globally optimal solutions [49]. Based on the clustering algorithm, we determined the number of optimal clusters that characterize each training set by employing an additional algorithm to calculate the Silhouettes Index (SI) [50]. The SI index interprets and validates the consistency of the clusters for each training set and provides a measure of how well each data point is matched to its own cluster compared to the neighboring clusters. Higher SI values (max value of 1) denote that the data points are very well suited to their clusters while lower values (min value of -1) indicate that the clustering configuration may have too many or too few clusters. In general, when a higher number of clusters is obtained through the computation of SI index, the dataset is much better represented when training the RBFN model, and a higher prediction performance is expected.

We evaluated the performance of the ML algorithms for each type of predictor based on classification metrics and regression errors ($G$-scores for the former and MAE for the latter). As classification measures, we considered the $F_{tp}$ and $F_{tn}$ scores in the computation of the multiobjective $G$ function. When training data are unbalanced, we recommend the use of both scores to find the best configuration of hyperparameters that can balance precision and robustness in both directions. This is the case for our original dataset, which is unbalanced in favor of the timely-submission class, meaning that 67% of assignment data points were classified as timely submissions ($y < 0$), and the remaining 33% as delay ($y > 0$). Therefore, we included the following metrics: Positive predicted value ($PPV = TP/(TP + FP)$), which is the precision of detecting delay; and the true positive rate ($TPR = TP/(TP + FN)$) that indicates how well delay can be predicted without negatively affecting the prediction of timely submissions (sensitivity or recall). However, when data are perfectly balanced, the use of $F_{tp}$ and $F_{tn}$ is not needed. In this case, other classification metrics could be used instead, including accuracy (ACC) as the ratio of the correct and total predictions, and the Matthews Correlation Coefficient (MCC) as a metric of difference between correct and wrong predictions. Generally, the MCC is reportedly more informative compared to other measures [51], [52]. We included ACC and the MCC for the sake of completion.

IV. RESULTS

The aim of this section is to evaluate the performance of ML algorithms in regard to cross-validation and prediction. First, we measured and evaluated the multiobjective function $G$-score for each algorithm and type of predictor in the cross-validation process. This was followed by a calculation of the
same G-scores for the test datasets. These results were then compared to those of the cross-validation (RQ1). RQ2 is then addressed by comparing the ML approaches for each type of predictor in more detail while considering the classification performance indicators and the MAE. Finally, we address RQ3 by evaluating how high the predictive performance is when considering the obj and comb predictors compared to predictions based solely on subj variables.

As explained above, the ML algorithms were cross-validated, trained, and tested separately for 10 randomized subsets to account for the discovered unbalance. The reported results include the mean and SD values over the 10 subsets for each of the 3 types of predictors.

### A. Evaluation in Cross-Validation

In this article, we employed a 4-fold cross-validation scheme for all of the ML algorithms, in which all training sets were divided into 4 subsets with 222 elements each. The same cross-validation procedure was conducted for the 3 types of predictors and each ML algorithm, resulting in a total of 360 different cross-validation processes.

#### 1) Mean G-Score in Cross-Validation

To evaluate the performance of the ML algorithms during cross-validation, we computed the mean G-score and SD over 10 datasets for each of the 3 predictor types. The values exposed in Table II for each ML algorithm are the maximum G-scores of the best configuration of hyperparameters obtained through the cross-validation averaged over 10 training datasets.

We do not report hyperparameters for the Bayesian multi-level models since we selected noninformative priors, as was the case in the previous study [14]. Looking at the subj and obj predictors, RBFN obtained the highest scores of 0.74 and 0.77, respectively. The performance regarding the validation settings was higher compared to the other ML algorithms since the data centers obtained through the proposed clustering approach cover the entire training set. For the comb predictors, RBFN, RF, and GBM obtained comparable G-scores. When using SVR, the LIN and TAH kernel functions are not suitable options to predict delay when employing cross-validation over the training sets. The nonlinear kernels (POL, RBF, VS) can fit the validation sets better in the training \( K - 1 \) sub-partitions. In the next step, we verified if the same prediction trend in cross-validation is followed when predicting new examples from the test sets.

#### 2) Mean G-Score in Test Datasets

The hyperparameters obtained through cross-validation were used to train each ML algorithm by using the entire training sets \( (K \) folds) for each type of predictor. Their performance is then evaluated in the test sets for all ML algorithms based on G-scores. In Table II (right side), we present the mean G-scores and SD values calculated over the 10 test datasets for each type of predictor. In case of the subj predictors, the GBM and SVR-VS models outperformed the others with mean G-scores higher than 0.71. When analyzing the obj predictors, the best performance of \( G = 0.75 \) was achieved by the baseline Bayesian models with random slopes. When combining both types of predictors, RF and SVR-VS remained the best options with a mean G-score higher than 0.76.

#### 3) Mean G-Score Comparison

In order to validate the configuration of hyperparameters selected for each ML algorithm, the G-scores were compared between cross-validation and testing. The most successful ML approach would have the highest G-score among all candidates and all types of predictors with the smallest performance depreciation between cross-validation and testing. NB and KNN both had a rather small difference in G-scores in the validation and testing sets, but the performance level is much lower compared to the other ML approaches. RT and GBM are well known for overfitting in the training data, which can also be observed when looking at their G-scores. SVR is sensitive to the training data, and the performance of delay prediction depends very much on the selected configuration of hyperparameters. This explains the 3% depreciation between validation and test sets for all types of predictors, especially for the case of kernels that use the linear and tangent hyperbolic functions. When analyzing the performance of RBFN and FFNN in the validation and test datasets, we observed the highest degradation in performance. This aspect is explained by the fact that both RBFN and FFNN were trained on 60% of the data while the remaining 20% were used for the stopping criteria. Also, the G-scores reported in cross-validation are the best values that could be

---

**TABLE II**

| ML Alg. | subj | obj | comb | subj | obj | comb |
|---------|------|-----|------|------|-----|------|
| NB      | 0.6457 (0.0084) & 0.6845 (0.013) & 0.6961 (0.0048) | 0.6375 (0.021) & 0.6628 (0.0267) & 0.681 (0.019) |
| KNN     | 0.7071 (0.0068) & 0.7482 (0.0039) & 0.7525 (0.0069) | 0.6758 (0.0222) & 0.7279 (0.0132) & 0.7319 (0.019) |
| RBFN    | 0.7401 (0.0051) & 0.7707 (0.0072) & 0.7688 (0.0054) | 0.6778 (0.0179) & 0.701 (0.0218) & 0.7214 (0.0204) |
| FFNN    | 0.7143 (0.012) & 0.7631 (0.0102) & 0.7673 (0.008) | 0.6695 (0.0368) & 0.6957 (0.0316) & 0.7188 (0.0223) |
| RT      | 0.6772 (0.0106) & 0.7491 (0.0046) & 0.7478 (0.0047) | 0.6811 (0.0251) & 0.7126 (0.025) & 0.7181 (0.035) |
| RF      | 0.7081 (0.008) & 0.755 (0.0057) & 0.7685 (0.0054) | 0.7064 (0.0153) & 0.7457 (0.018) & 0.7626 (0.015) |
| GBM     | 0.70686 (0.0143) & 0.76012 (0.00919) & 0.7713 (0.0071) | 0.71624 (0.023) & 0.73166 (0.021) & 0.7442 (0.0271) |
| SVR-LIN | 0.6785 (0.0071) & 0.7253 (0.0083) & 0.7275 (0.0069) | 0.6591 (0.0422) & 0.6888 (0.0264) & 0.7134 (0.019) |
| SVR-POL | 0.6975 (0.0056) & 0.7348 (0.0041) & 0.7413 (0.0055) | 0.6835 (0.0312) & 0.6934 (0.0273) & 0.715 (0.023) |
| SVR-TAH | 0.6833 (0.01) & 0.7305 (0.0072) & 0.7331 (0.0062) | 0.6544 (0.0318) & 0.6916 (0.0334) & 0.708 (0.024) |
| SVR-RBF | 0.7208 (0.0094) & 0.7406 (0.0059) & 0.7536 (0.0077) | 0.7129 (0.0189) & 0.7288 (0.026) & 0.7538 (0.018) |
| SVR-VS  | 0.7208 (0.0084) & 0.7608 (0.0045) & 0.7631 (0.0048) | 0.7145 (0.0175) & 0.7364 (0.0257) & 0.7589 (0.018) |
found while training in each split. RF was the most stable ML algorithm for all three types of predictors, since the levels of test $G$-scores were very close to those scores obtained in cross-validation.

B. Intramodel Comparisons

In this section, we provide a comprehensive report to compare the employed ML algorithms for each type of predictor (intramodel comparison). Thus, we analyze the mean and SD of the PPV, TPR, $F_{tp}$, $F_{tn}$, MCC, ACC, and MAE performance indicators, where the best values of these indicators are highlighted in green. To answer RQ2, the goal of this section is to determine the most successful ML algorithm for each type of predictor that would maximize the $F_{tp}$ measure and minimize the loss in $F_{tn}$ and regression error (MAE).

1) Subjective Predictors: In Table III, we present the classification and regression performance metrics (mean and SD) for each ML algorithm and calculated over the datasets with subj predictors. When looking at the PPV metric, we obtained the highest amount of correct predictions for the delay class when using SVR with RBF and VS kernels. However, the robustness of these models in predicting the timely submission class is depreciated more than 15% when compared to PPV. The best tradeoff between PPV and TPR was observed with the RF and GBM models. When computing the $F_{tp}$ score, GBM performed slightly better than RF with 45.5%. However, the SD of the RF model is reduced by half when compared to GBM, meaning the RF classifier is less sensitive to the type of dataset, which is used to train the model. The $F_{tn}$ score was also calculated in Table III to measure the tradeoff between the precision for timely submission and the robustness to delay. Since the false predictions of delay have a greater impact than the false predictions of timely submissions due to the unbalance of the datasets, the SVR model with the RBF and VS kernels achieved the highest $F_{tn}$ scores. By monitoring the MCC and ACC classification metrics, it can be concluded that the absolute difference between the false predictions of delay and timely submission was higher for the SVR model (with RBF and VS kernel functions) than for the RF and GBM models. Based on the results collected in Table III, a precision to detect delay of nearly 48% and a classification accuracy of about 67% can be obtained with the GBM model when exclusively considering the subj predictors. When computing the regression error, the SVR model with RBF and VS kernels performed slightly better than GBM. However, we recommend the use of the GBM model when predicting the subj predictors due to the best tradeoff between regression error, $F_{tp}$ and $F_{tn}$.

2) Objective Predictors: In Table IV, we present the classification and regression performance metrics (mean and SD) obtained when predicting delay with obj predictors. When evaluating the predictive performance of the delay class, the Bayesian model with random slopes (BA-RS) provided the best results among all models with the following metrics:

- 1) the precision (PPV = 51.02%) when predicting delay;
- 2) robustness (TPR = 60.88%) to predict timely submission;
- 3) tradeoff between PPV and TPR ($F_{tp}$ = 55.41%).

Also, BA-RS was the best option when measuring the MCC (32.29%) and accuracy (ACC = 69.01%). On average, the GBM model achieved the highest $F_{tn}$ score of 77.53% due to higher robustness to predict delay. Also, its error was lower compared to the other models. When looking at the tradeoff between $F_{tp}$, $F_{tn}$, and MAE, BA-RS was the best option to predict delay with obj predictors.

3) Subjective and Objective Predictors Combined: When looking at the performance of the ML algorithms for the comb predictors in Table V, we observed that the SVR-RBF performed better in precision (with about 2%) but with lower robustness value (with more than 7%) when compared to the RF model. This explains the larger gain obtained by the RF algorithm when measuring the $F_{tp}$ score of about 57%. The same trend can be observed when comparing the SVR-VS and RF models. When measuring the $F_{tn}$ scores, SVR with RBF kernel provided the best results due to the better robustness of the model to predict delay. By comparing the RF, SVR-VS, and SVR-RBF models, we noted that higher $F_{tp}$ scores involve higher MCC levels while higher $F_{tn}$ scores align with higher accuracy values. When calculating the average regression error, GBM and RF models outperformed the other ML models. Although SVR-RBF and SVR-VS provided better precision (PPV), the RF model remained the best option to be employed for the comb predictors due to the best tradeoff.
between precision and robustness to delay and timely submission classes, and a low regression error.

C. Intermodel Comparison

In this section, we compare the predictive performance between each type of predictor to answer RQ3 by taking into account the results displayed in Tables II–V.

When analyzing the performance of the mean $G$-scores in both the cross-validation and testing stages (see Table II), we concluded that the obj predictors are more informative than the subj predictors when predicting delay in assignments. However, a performance gain can be achieved by combining both types of predictors. When analyzing the best $G$-scores in Table II (on the right side), we observe a prediction gain of 4% when comparing the prediction with the obj and subj predictors, and a gain of 1% when predicting with the comb predictors compared to the subj predictors alone. By combining both subj and obj predictors, almost all ML candidates benefit from the perspective of both classification and regression metrics as shown in Table V. For example, the SVR model with all types of kernel functions obtained higher precision (PPV) and robustness (TPR) when involving the comb predictors compared to the subj or obj predictors alone. The same performance gain can be observed when measuring MCC, ACC, or MAE. Another concluding example in this sense is the RF approach that enhanced its $F$-score from 45.43% for subj predictors to 53.48% for obj data, and when combining both, the performance was higher than 57%. An explanation of this gain is given by the importance of the subj predictors that changes when the obj predictors are added. In this sense, we computed the Gini importance index or the mean decrease in impurity that gives the feature importance when RTs are employed [53]. When training the RF model, we make the following observations based on the importance index.

\begin{itemize}
\item[a)] In case of obj predictors, the interval between the start of a block and the first click on an assignment is the most important predictor, followed by the number of clicks on an assignment and the number of clicks on relevant activities.
\item[b)] When training with subj predictors, APS is the most informative, followed by APSS, SDLS, and GASE.
\item[c)] By combining both types of predictors, the most important variables are the obj predictors in the same order as above in a), followed by the subj predictors in the
\end{itemize}
order as before \((b)\), with the only difference that the GASE variable gained a higher importance than APS.

While APS and APSS are more predictive in general, when combined with obj predictors, they do not add any new information. Therefore, the GASE variable could enhance the precision and accuracy above 52% and 70%, respectively, when combining both sets of predictors.

As observed in Tables III and IV, the highest performing ML approach differs from one type of predictor to another when balancing the performance between \(F_{tp}\) and \(F_{tn}\) scores and the regression error. GBM is recommended to be used for subj data due to a better tradeoff between the indicators above. When using the obj predictors only in the form exposed in Table I, the BA-RS approach was the best option to be employed. However, for combined predictors, the RF method outperformed other approaches due to a higher importance of GASE feature among other subj ones. In Table VI, we summarize the results from Tables III and IV and present the intermodel comparison between the highest performing ML approach for each type of predictor. The combined predictors brought a performance gain in terms of MCC of more than 2% when compared to obj datasets and more than 12% when compared to subj predictors. The regression error was the lowest when predicting with the comb datasets while the depreciation in \(F_{tn}\) was negligible between different types of predictors. The highest accuracy value of 70.27% was achieved when combining the predictors, with a slight depreciation of about 1% for obj and 3% for subj predictors.

### V. Discussion

The aim of this study was to expand the findings in [14] by employing multiple ML algorithms to determine which one delivered the best predictions of delay based on objective and subjective variables. First, due to the unbalanced nature of the data (meaning there were twice as many timely submissions as there was delay), we needed to account for it in our approach to optimize each ML algorithm. We achieved this by a cross-validation procedure based on a novel multiobjective function, the \(G\)-score, which involves a tradeoff between positive and negative \(F1\)-scores and regression error. We measured the performance degradation of \(G\)-score between test and cross-validation to verify the authenticity of the selected hyperparameters for each ML algorithm. Except for RBFN and FFNN, all other approaches provided a low degradation in \(G\)-score. We concluded that RBFN and FFNN need more data to cope with such a performance gap. Among the ML algorithms in our pool, RF obtained the lowest degradation in performance, meaning that cross-validation was highly efficient (RQ1).

We then identified the best prediction performances for each of the types of predictors \((\text{subj}, \text{obj}, \text{comb})\) by measuring the trade-off between \(F_{tp}\), \(F_{tn}\), and \(\text{MAE (comb)}\). GBM was the best approach for subjective predictors, BA-RS for the objective predictors, and RF for the combined sets. The highest predictive performance of RF is congruent with the existing literature in LA, where RF consistently ranks among the best models [54], [55]. To the best of our knowledge, our second best approach, Bayesian multilevel models with random slopes, is not mentioned in any LA studies, due to its nature as a statistical approach. Since most of the studies in LA are focused on predictions from objective data and BA-RS was the best algorithm in our case, it might be useful for other researchers to consider multilevel models in their work. After all, repeated measures are common when assessing student data and multilevel models provide the means to reflect nested data structures. Nevertheless, these comparisons should be taken cautiously since the outcomes differ in both studies.

Next, we compared the different types of predictors (RQ3). Predicting delay with objective predictors yielded better results than relying on the subjective variables alone. By mixing both types of predictors, we achieved a slightly higher predictive performance, meaning that some subjective predictors increase their relative predictive power when combined with the objective predictors. These results are unsurprisingly in line with the previous study [14] our data stem from. The superiority of the objective predictors was evident in all the ML models regardless of the underlying algorithm, strengthening the finding of the previous study. Similarly, Pardo et al. in [10] reported that more objective variables correlated with academic outcomes than subjective variables, and in [56], Yu et al. also found a higher accuracy of objective features to predict the course and next semester outcomes. However, in contrast with the previous study, the advantage of combining the two sets of predictors was more apparent here since the exact ranking of the models was rather unclear before.

As expected, the best results across all the performance metrics were achieved when the subjective and objective variables were combined. This is also in line with the work in [10] and [57], despite the different levels of granularity between our study and theirs. While both of these studies focused on the final course outcome, we were interested in a much more fine-grained measure (the delay of individual assignments). In both cases, the authors reported an increase in explained variance when objective variables were added to the subjective variables. Our results demonstrate that this effect remains consistent across different ML algorithms when predicting delay. However, it remains unclear if combined data have a long-term predictive advantage. In [56], for instance, Yu et al. found that combined data only had higher predictive accuracy.
for short-term predictions. For long-term predictions (e.g., the outcome of the following semester), the authors determined that using only objective predictors yielded better results. We suspect this may be due to fluctuations in students’ learning-related dispositions, which cannot be fully captured by a single measurement. Despite their trait-like nature, it would thus be advisable to repeatedly assess subjective factors instead of exclusively relying on once-collected historical student data. This comes with its own risks, e.g., alienating students by a high frequency of reassessments.

Compared to the work in [10], our results show that using ML algorithms might shuffle the order of importance of the subjective variables and the magnitude of their contribution. While APSS is the most predictive feature for subjective data, when combined with objective predictors, it is superseded by the GASE factor, which was the least important factor for subjective-only predictions. This makes it more challenging to determine which variables to favor in a more parsimonious model. In [10], the order of importance remained consistent. We assume that this discrepancy could either be traced back to the covariance structure of the predictors and/or the complex nature of some of the implemented algorithms, particularly the underlying decision trees that empower both RF and GBM. In decision trees, more complex dependencies of the variables are considered when building the model.

A. Limitations

This study has three major limitations that need to be addressed. First, the comparatively low predictive accuracy of the models (some even performing barely above chance) raises the question of why it is not as high as one might expect based on models in other studies. One issue is our relatively small sample size. This could be remedied in a number of ways, for example, by collecting more data across multiple semesters with the same students. The data accumulated over time can then serve as historical data, acting as training data for future predictions. The other contributing factor is the unbalance of our data. Two-thirds of the assignments being handed in on time is not necessarily a representative finding. Given the reported prevalence of procrastination in higher education [2], one could assume that there is more of equilibrium in other populations. Another plausible possibility is a stricter enforcement of deadlines, resulting in an even higher proportion of timely submissions and thus more unbalanced data. The ML approaches involved in this type of prediction thus need to be stable enough to account for such differences in the studied populations.

The second issue concerns the models themselves. Models with objective predictors outperforming models with subjective predictors is not that surprising, considering that the outcome variable delay and our objective predictors are all state variables. When interpreting the results, it is also important to note that the two sets of variables are not directly comparable. This is mainly due to their different granularity: the subjective predictors are based on data assessed at the student level and the objective variables are located on the assignment level in two cases and the course level in one case. This is also reflected in the outcome variable, which shared its granularity only with a few predictors, which we found to be the most important variables, both in this study and the original one [14].

The final limitation is the exclusion of the five covariates featured in the previous study, whose inclusion could potentially improve the predictions. The type of deadline in particular was a strong individual predictor in some of the Bayesian multilevel models, which may well be the case in our current analyses. This could be addressed in a future study.

B. Implications and Future Work

The results imply that future research in procrastination and other types of dilatory behavior should include ML algorithms such as RF, rather than relying on traditional statistical approaches alone. This holds true, even for cases with low numbers of variables. However, as our results show, ML algorithms do not necessarily outperform statistical approaches in every instance. In order to select the best variables to enhance predictions, we thus recommend considering both ML algorithms and statistical models as tools in the LA toolbox and evaluating datasets that include both subjective and objective predictors. This is particularly crucial when applying such models in the field, e.g., for real-time predictions, considering these two types of predictors may not be available simultaneously. While we collected all our data at the end of the semester, the same set of predictors could be split into early and late predictors if they were to be assessed in an LMS with real-time predictions in the future.

For instance, our subjective predictors are trait variables, meaning they are supposedly stable across certain periods and could thus already be assessed at the start of the semester, making them viable as early predictors. Despite not being as strong or consistent as their objective counterparts in our models, the subjective predictors could still potentially have value since they may provide some clues long before the objective predictors become active. Even though a lot of the algorithms performed below chance when operating with subjective variables alone (meaning the performance was lower than simply assuming all assignments would be submitted on time), GBM still managed to perform above chance.

As discussed in [14], the objective predictors we used have the drawback of requiring information about students’ activity across the semester, meaning predictions based on them cannot be made unless that information is accumulated (at least partially). For instance, at the beginning of the semester, virtually no log data are available. Subjective data, however, could fill this gap, e.g., in the form of questionnaires. This again highlights the utility of combining both types of predictors. Moreover, some of these variables change throughout the course (e.g., the number of clicks on an assignment), and so would the predictive effect of that variable when calculating real-time predictions. From this perspective, the optimal points in time should be determined during the semester to achieve the best possible predictive performance that allows for instructional feedback and other interventions to positively impact dilatory behavior and other performance metrics.
These results also imply that selecting a single algorithm and sticking with it for all types of predictors may not be the ideal path to take. Considering that a different approach was favored for each of the three categories, it may be advisable to employ multiple ML algorithms for real-time predictions, depending on the current stage of the semester. One could start with GBM to work with early predictors, which can then be accompanied by a multilevel model for the late predictors or get replaced entirely by RF once all predictors become available.

Combined, our results show promise that these types of models could be used for real-time predictions of delay in the future, which has implications for educational applications. Predictions of dilatory behavior can benefit teachers, allowing them to detect dilatory tendencies well in advance, which in turn may predict academic outcomes and potential drop-outs. If the end goal is to provide timely interventions to reduce maladaptive forms of dilatory behavior (e.g., procrastination), such models would be a necessary step. Teachers could use these predictions to either provide appropriate interventions themselves or have them implemented in adaptive learning systems. On top of that, the classification of students as potential procrastinators (or other types of delayers) can not only help in the prediction of academic underachievement or dropout; it can also simplify the second round of predictions. While procrastinators could benefit from interventions, students classified as nonprocrastinators or strategic delayers may not need further assistance, so the second round of predictions would not need to include them in the same capacity any longer. Crucially, this process requires accurate predictions in order not to miscategorize and thus alienate students. Teachers also need to be able to trust the predictions if they are to be used to inform interventions, which could target maladaptive forms of delay by increasing students’ time management skills or addressing the underlying reasons for such behavior with psychoeducational elements. Potential reasons include fear of failure, self-handicapping, and task aversiveness.

However, the sample size is currently not large enough to provide accurate real-time predictions, meaning more data need to be collected first, involving an expanded array of predictors. This could include task-specific factors (e.g., motivational aspects such as students’ interest in a topic) or predictors related to students’ time management skills. The latter could be achieved by analyzing behavioral patterns, which was successfully incorporated in a mixture models study by Park et al. [58]. Another promising avenue is to assess some of the learning-related factors objectively rather than subjectively, e.g., by replacing the SDL questionnaire with clickstream-based indicators of self-regulated learning (see [59]).

VI. CONCLUSION

In conclusion, when applying ML prediction models in the field, comparisons between various algorithms are needed to determine which ones deliver the highest performance. The suitability of a given ML algorithm when predicting dilatory behavior depends on the type of predictor. While objective predictors work best with a statistical approach (Bayesian multilevel models), subjective predictors are better served with GBMs. When both types of variables are combined, RF was the preferable ML algorithm in our case. Future studies need to increase the predictive performance (e.g., by expanding the roster of predictors) to allow such models to be implemented in learning management systems, ultimately enabling real-time predictions during a semester. This can then serve as the basis for interventions aiming at reducing procrastination and promoting timely submissions.

REFERENCES

[1] A. D. Dumford and A. L. Miller, “Online learning in higher education: Exploring advantages and disadvantages for engagement,” J. Comput. Higher Educ., vol. 30, no. 3, pp. 452–465, Dec. 2018.
[2] P. Steel, “The nature of procrastination: A meta-analytic and theoretical review of quintessential self-regulatory failure,” Psychol. Bull., vol. 133, no. 1, pp. 65–94, 2007.
[3] D. M. Tice and R. F. Baumeister, “Longitudinal study of procrastination, performance, stress, and health: The costs and benefits of dawdling,” Psychol. Sci., vol. 8, no. 6, pp. 454–458, Nov. 1997.
[4] W. Doherty, “An analysis of multiple factors affecting retention in web-based community college courses,” Internet Higher Educ., vol. 9, no. 4, pp. 245–255, Oct. 2006.
[5] A. Ellis and W. J. Knaus, Overcoming Procrastination: Or How to Think and Act Rationally in Spite of Life’s Inevitable Hassles. New York, NY, USA: Inst. Rational Living, 1977.
[6] D. M. Corkin, S. L. Yu, and S. F. Lindt, “Comparing active delay and procrastination from a self-regulated learning perspective,” Learn. Individual Differences, vol. 21, no. 5, pp. 602–606, 2011.
[7] T. A. Pychyl, “Savouring the flavours of delay,” English Stud. Canada, vol. 34, no. 2/3, Jun. 2008, Art. no. 25.
[8] G.-J. Hwang, H.-C. Chuz, and C. Yin, “Objectives, methodologies and research issues of learning analytics,” Interactive Learn. Environ., vol. 25, no. 2, pp. 143–146, Mar. 2017.
[9] S. Azimi, C.-G. Popa, and T. Cucic, “Improving students performance in small-scale online courses - A machine learning-based intervention,” Int. J. Learn. Analytics Artif. Intell. Educ., vol. 2, no. 2, pp. 80–95, Nov. 2020.
[10] A. Pardo, F. Han, and R. A. Ellis, “Combining university student self-regulated learning indicators and engagement with online learning events to predict academic performance,” IEEE Trans. Learn. Technol., vol. 10, no. 1, pp. 82–92, Jan.–Mar. 2017.
[11] R. Conijn, C. Snijders, A. Kleingeld, and U. Matzat, “Predicting student performance from LMS data: A comparison of 17 blended courses using moodle LMS,” IEEE Trans. Learn. Technol., vol. 10, no. 1, pp. 17–29, Jan.–Mar. 2017.
[12] A. Akram et al., “Predicting students’ academic procrastination in blended learning course using homework submission data,” IEEE Access, vol. 7, pp. 102487–102498, 2019.
[13] R. Cerezo, M. Esteban, M. Sánchez-Santillán, and J. C. Núñez, “Procrastinating behavior in computer-based learning environments to predict performance: A case study in Moodle,” Front. Psychol., vol. 8, no. Aug., pp. 1–11, 2017.
[14] C. Imhof, P. Bergamin, and S. McGarrity, “Prediction of dilatory behaviour in online assignments,” Learn. Individual Differences, vol. 88, May 2021, Art. no. 102014.
[15] A. Bandura, Social Foundations of Thought and Action: A Social Cognitive Theory. Englewood Cliffs, NJ, USA: Prentice-Hall, 1986.
[16] K. Wäschle, A. Allgaier, A. Lachner, S. Fink, and M. Ničkles, “Procrastination and self-efficacy: Tracking vicious and virtuous circles in self-regulated learning,” Learn. Instruct., vol. 29, pp. 103–114, 2014.
[17] A. H. C. Chu and J. N. Choi, “Rethinking procrastination: Positive effects of “active” procrastination behavior on attitudes and performance,” J. Social Psychol., vol. 145, no. 3, pp. 245–64, 2005.
[18] S. Birnbaum, “Towards active and self-directed learning: Preparing for lifelong learning, with reference to Dutch secondary education,” in Proc. Annu. Meeting Amer. Educ. Res. Assoc., 1996.
[19] M. S. Knowles, Self-Directed Learning: A Guide for Learners and Teachers. Chicago, IL, USA: Assoc. Press, 1975.
[20] D. R. Garrison, “Self-directed learning: Toward a comprehensive model,” Adult Educ. Quart., vol. 48, no. 1, pp. 18–33, Nov. 1997.
[21] N. Saed and T. Eslaminejad, “Relationship between student’s self-directed-learning readiness and academic self-efficacy and achievement motivation in students,” Int. Educ. Stud., vol. 10, no. 1, pp. 225–232, 2017.

[22] M. Schommer-Aikins and M. Easter, “Cognitive flexibility, procrastination, and need for closure linked to online self-directed learning among students taking online courses,” J. Bus. Educ. Leadership, vol. 8, no. 1, pp. 112–121, 2018.

[23] M. M. Girgiliano, C. D. Guthrie, and M. V. Pusic, “Click-level learning analytics in an online medical education learning platform,” Teach. Learn. Med., vol. 32, no. 4, pp. 410–421, Aug. 2020.

[24] S. Knight, A. F. Wise, and B. Chen, “Time for change: Why learning analytics needs temporal analysis,” J. Learn. Analytics, vol. 4, no. 3, pp. 7–17, Dec. 2017.

[25] F. Cervera-Mercado and J. A. Lara, “A method for generating features representing the students’ degree of anticipation/delay to complete assignments,” in Proc. 13th Int. Conf. Educ. Data Mining, 2020, pp. 574–577.

[26] J. W. You, “Examining the effect of academic procrastination on achievement using LMS in e-learning,” Educ. Technol. Soc., vol. 18, no. 3, pp. 64–74, 2015.

[27] Y. Levy and M. M. Ramim, “A study of online exams procrastination using data analytics techniques,” Interdiscipl. J. E-Skills Lifelong Learn., vol. 8, pp. 97–113, 2012.

[28] A. Gareau, M. Chamandy, K. Kljajic, and P. Gaudreau, “The detrimental effect of academic procrastination on subsequent grades: The mediating role of coping over and above past achievement and working memory capacity,” Anxiety Stress Coping, vol. 32, no. 2, pp. 141–154, 2019.

[29] M. Yamada, Y. Goda, T. Matsuda, Y. Saito, H. Kato, and H. Miyagawa, “How does self-regulated learning relate to active procrastination and other learning behaviors?,” J. Comput. Higher Educ., vol. 28, no. 3, pp. 326–343, 2016.

[30] M. del Puerto Paule-Ruiz, M. Riestra-González, M. Sánchez-Santillán, and J. R. Pérez-Pérez, “The procrastination related indicators in e-learning platforms,” J. Universal Comput. Sci., vol. 21, no. 1, pp. 7–22, 2015.

[31] D. Hooshyar, M. Pedaste, and Y. Yang, “Mining educational data to predict students’ performance through procrastination behavior,” Entropy, vol. 22, no. 1, 2020, Art. no. 12.

[32] S. M. R. Ahidi et al., “Educational sustainability through Big Data assimilation to quantify academic procrastination using ensemble classifiers,” Sustainability, vol. 12, no. 15, 2020, Art. no. 6074.

[33] Y. Yang, D. Hooshyar, M. Pedaste, M. Wang, Y. M. Huang, and H. Lim, “Prediction of students’ procrastination behaviour through their submission behavioural pattern in online learning,” J. Ambient Intell. Humanized Comput., 2020, doi: 10.1007/s12676-020-02041-8.

[34] T. Nielsen, J. Dammeyer, M. L. Yang, and G. Makransky, “Gender fairness in self-efficacy? A Rasch-based validity study of the general academic self-efficacy scale (GASE),” Scand. J. Educ. Res., vol. 62, no. 5, pp. 664–681, 2018.

[35] J. W. Lounsbury, J. J. Levy, S. H. Park, L. W. Gibson, and R. Smith, “An investigation of the construct validity of the personality trait of self-directed learning,” Learn. Individual Differences, vol. 19, no. 4, pp. 411–418, 2009.

[36] J. D. McCloskey, “Finally, my thesis on academic procrastination,” Ph.D. dissertation, Fac. Graduate Sch., Univ. Texas Arlington, Arlington, TX, USA, 2012.

[37] J. N. Choi and S. V. Moran, “Why not procrastinate? Development and validation of a new active procrastination scale,” J. Social Psychol., vol. 149, no. 2, pp. 195–211, 2009.

[38] M. N. Wright, T. Dankowski, and A. Ziegler, “Unbiased split variable selection for random survival forests using maximally selected rank statistics,” Statist. Med., vol. 36, no. 8, pp. 1272–1284, 2017.

[39] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” Ann. Statist., vol. 29, pp. 1189–1232, 2001.

[40] J. H. Friedman, “Stochastic gradient boosting,” Comput. Statist. Data Anal., vol. 38, no. 4, pp. 367–378, 2002.

[41] M. Hlousta, C. Herodotou, V. Bayer, and M. Fernandez, “Impact of predictive learning analytics on course awarding of disadvantaged students in STEM,” in Proc. Int. Conf. Artif. Intell. Educ., 2021, pp. 190–195.

[42] J. A. Ruiz-French and J. A. Lara, “Early prediction and variable importance of certificate accomplishment in a MOOC,” in Proc. Eur. Conf. Massive Open Online Courses, 2017, pp. 263–272.
Martin Hlosta received the Ph.D. degree in computer science from the Technical University Brno, Czech Republic, in 2018. He is currently a Research Fellow with the Institute for Distance Learning and eLearning Research (IFeL), Brig, Switzerland, working on projects for adaptive learning in education. Before, he led OUAnalyse with The Open University, a project improving student retention via machine learning, selected by UNESCO among four of the best projects using AI in Education in 2020. His research mainly focuses on predictive learning analytics and its impact, in particular, to target existing educational inequalities.

Behnam Parsaeifard received the Ph.D. degree in computational physics from the University of Basel, Basel, Switzerland, in 2021. He joined the Institute for Open, Distance and eLearning (IFeL) as a Researcher in November 2021. He was a research intern with the Visual Intelligence for Transportation (VITA) lab, EPFL. His research interest includes applying data mining and machine learning techniques to improve adaptive learning and learning analytics.

Ivan Moser received the Ph.D. degree in psychology from the Institute of Cognitive Psychology, University of Bern, Bern, Switzerland, in 2017. He is currently a Researcher and a Lecturer with the Institute for Research in Open, Distance and eLearning (IFeL), Brig, Switzerland. In the course of his research career, he developed an increasing fascination for the numerous new possibilities that virtual reality (VR) opens as a learning technology. As a leader of several VR-related research projects with FFHS, he investigates the potential and limits of virtual learning environments.

Per Bergamin received the Ph.D. degree in psychology from the University of Bern in 2015. He is currently a Professor in didactics in distance education and e-learning with the Swiss Distance University of Applied Sciences (FFHS), Brig, Switzerland. Since 2006, he has been acting as the Director of the Institute for Research in Open, Distance- and eLearning (IFeL), Brig, and, since 2016, he has been holding the UNESCO Chair on Personalised and Adaptive Distance Education. His research interests include self-regulated and technology-based personalized and adaptive learning; central aspects are instructional design, usability, and application implementation.