2023 Annual Conference & Exposition
Baltimore Convention Center, MD | June 25 - 28, 2023
The Harbor of Engineering Education for 130 Years
Paper ID #37177

On Time-based Exploration of Student Performance Prediction

Dr. Abdulmalek Al-Gahmi, Weber State University

Dr. Abdulmalek Al-Gahmi is an assistant professor at the School of Computing Department of Weber State University. His teaching experience involves courses on object-oriented programming, full-stack web development, computer graphics, algorithms and data structures, and machine learning. He holds a Ph.D. in Computer Science from New Mexico State University, M.S. in Computer Science, M.A. in Extension Education, and B.S. in Electrical Engineering.

Dr. Kyle D. Feuz, Weber State University

Kyle Feuz is an Associate Professor at Weber State University in the School of Computing. He earned his Ph.D from Washington State University under the guidance of Dr. Diane Cook in 2014. He also received his B.S and M.S in Computer Science from Utah Stat

Dr. Yong Zhang, Weber State University

Dr. Yong Zhang is an associate professor in Computer Science at Weber State University. He received the B.E. degree and M.E. degree in Electrical Engineering from Harbin Institute of Technology, China, and the Ph.D degree in Electrical Engineering from West Virginia University, Morgantown, USA. His research interests include digital image and video processing, bioinformatics, and machine learning.

On Time-based Exploration of Student Performance Prediction

Abstract

Predicting student performance early enough to intervene and provide help has been a longstanding topic of interest among the educational research community. Many studies have investigated making these predictions, and two main issues have been pointed out: the portability and robustness of these predictions. Learning Management Systems (LMS) and other tools used by courses today capture extensive amounts of information about student performance that is helpful not only in improving earlier attempts at these predictions, but also in taking them further. This is a work-in-progress study that looks at ways to accurately predict student performance based on the LMS data collected from 274 lower-division Computer Science courses taken by 2,656 students and taught by 37 instructors in various formats (face-to-face, online, virtual, hybrid) over a period of three years in a public four-year university. It uses a time-based approach that looks at the data as a set of sequences or time series, each representing the progress of a student both within a single course and across multiple courses. It tracks the progress of students within the three-year period as they go through the lower-division CS program and get their associate degrees. With courses being different from one another, we explore ways for "normalizing" the data in order to consider the whole learning journey of students across courses. The study explores questions such as: How does the progress of struggling students differ from one course to another across various formats? How early can student performance within these courses be accurately predicted? Can the cumulative progress of students at the end of the program be predicted? Are student journeys through courses unique? Are there patterns that transcend students and courses? How robust and portable are these predictions?

Introduction

The field of higher education has seen a growing interest in using data and analytics to predict student performance. This is motivated by the institutions' need to understand and address factors that contribute to student success or failure. It also provides valuable insights for students and advisors to make informed decisions about their academic activities. With the increasing availability of data, the exploration of student performance prediction has become more sophisticated and has gained momentum in recent years.

All the data used in this study was extracted from the Canvas Learning Management System. Learning Management Systems (LMS's) which are widely used by higher-education institutions are an important source of student-related data. They provide a convenient and important way to deliver learning materials to students. They also are the places where most of the course discussions, student-instructor interactions, and assessments take place. Conveniently they keep an extensive record of all such activities and make that data available using dedicated API services. The LMS used in this study does not provide any information other than activity logs and assessment scores. The latter is what is being used in this study. To our knowledge, none of this data is unique to Canvas or has any specific requirement that other LMS's do not support or have.

A body of research has investigated using such data to predict student performance at the course level. One question that is frequently posed in these studies has been whether student performance can be accurately predicted early enough to intervene and provide needed help to struggling students. Early detection of at-risk students is vital to fostering and promoting student success, which is critical to the mission of any higher-education institution. It allows for planning and providing the appropriate remedial services that students need in a timely manner. It requires the ability to predict student performance several times throughout courses. Many predictive models have been proposed and used to varying degrees of success to make such predictions. Some of these models are at the exam level, some at course level and some at the degree level. These models require the use of data sets that typically come from multiple sources such as student information systems (SIS's) and pre-college information, to name a few.

This study builds upon the work done by a previous paper which focused on a few Computer Science (CS) courses taught by three instructors [1]. The present paper applies and expands upon the insights from that work to a large dataset consisting of 274 lower-division CS courses, taught by 37 instructors in three different modalities (face-to-face, online, and virtual) over a period of three years (Summer 2019 to Spring 2022) in a public four-year university. The LMS data related to assignments and other graded activities are collected and used for each one of these courses. The resulting dataset used in this study consists of many courses in different modalities, taught by different instructors, and spanning multiple semesters. Having such a large data set enables the use of time series classification and forecasting techniques to address key inquiries in the study.

The driving question of this study is whether it is possible to predict student performance reliably and early enough to take corrective action. In order to explore this question, the study applies a time-based data-analytics approach to early student performance prediction. This approach involves representing student progress in courses as sequences or time series, where time steps/points correspond to student activities in those courses, and each sequence represents a single student's progress in a course. In other words, each course is thought of as a collection of time series sequences, each of which represents the progress of a single student in that course.

To summarize, this paper: 1) describes the methods and challenges of cleaning, preprocessing, and handling a large data set with thousands of variable-length time series sequences, 2) replicates and expands upon the results of [1], and 3) applies time series classification and forecasting techniques to investigate whether the performance of students at the end of courses can be accurately and reliably predicted. Such prediction can be framed as both a binary classification problem of whether a student will pass or fail at the end of the course or as a regression problem where the final grade at the end of the course is predicted. Students predicted to fail a course are considered "at-risk", and the timeliness of identifying them is critical to being able to provide them with the help they need.

The rest of this paper is organized as follows. The next section briefly reviews the background and related work of this paper. The section after that discusses the used methodology/approach. The section after that presents and discusses the obtained results. A discussion of future work is presented before this paper concludes.

Background and Related Work

Developing models for predicting student performance in a course has long been a topic of interest. Some of that interest is driven by a desire to evaluate the efficacy of certain teaching methodology [2], [3], while others seek to catch problems early enough in the semester to still have time to intervene [4], [5]. Some of these studies require designing certain randomized experiments [2], [4], [6], [7]. Others, like this study, utilizes the available data on student activities and interactions with the course materials.

For example, Umer et al [6] used several machine learning (ML) algorithms to predict student outcomes in a course by mining the LMS activity log data. They confirm the importance of this data in making such predictions but find out that it does not necessarily lead to improved predictive accuracy. Similarly, Van Goidsenhoven et al [7] analyzed activity log data from an LMS to predict student success. They specifically include courses with blended learning environments and discover that those classes are harder to predict student success based upon activity streams. Both studies use a variety of ML algorithms including random forest and logistic regression. They conclude that counting activities is helpful in making predictions but is not good enough.

Conijn et al [2] studied predicting student performance by comparing 17 blended courses. They focused on studying the portability of predictive models across multiple courses and the timeliness of these predictions. In doing so, they replicated a study by Gašević et al [3] on the effect of instructional conditions on predicting success with a bigger sample size using predictors available for all courses. They pointed out that there is a great diversity in the number of variables being used as predictors. They also pointed out the inconsistency of findings (and non-robustness) when the same or similar predictors are used and claim that there is a need to expand the empirical base of the issue of portability especially as some studies have indicated that prediction accuracy increases over time.

To address the issue of small sample size that previous studies suffer from, Riestra-Gonzalez et al [4] analyzed massive LMS log data for the purpose of achieving early prediction of course-agnostic student performance. They used several ML models in a course-agnostic way to classify students into fail, at-risk, and excellent groups at 10%, 25%, 33%, and 50% of the course. All courses for one year in a single university were used. Furthermore, Dias et al [5] proposed DeepLMS: a deep learning predictive model for supporting online learning, especially in the Covid-19 era. They used deep learning (DL) techniques to forecast the quality of interaction (QoI) with LMS using LSTM networks with RMSE errors. They used online learning as a way of reducing temporal and spatial problems found in traditional courses. They indicated that the QoI of a student is a strong efficacy indicator of the course design.

The use of data mining and machine learning algorithms for predicting student performance has involved various models such as decision trees, random forests, and neural networks. Examples include Sugiharti et al's system based on the Naïve Bayes Classifier algorithm [8], Li et al's use of a deep neural network [9], Chai et al's attrition model utilizing logistic regression, decision trees, and random forests models [10], and Wang et al's two-stage classification framework treating student performance prediction tasks as a short-term sequence prediction problem [11]. Wang et al's framework outperformed others by adaptively extracting the main behavior intention from student sequential behavior information, improving model prediction accuracy.

The use of network analysis in student performance prediction has also been explored in research. It has been used to examine relationships between students and the factors that impact their academic outcomes, such as peer influence, academic support, and institutional policies. These studies have provided valuable insights into the complex relationships that influence student success. Nguyen et al [12] evaluated how spatial-temporal student networks from campus WIFI log data relate to students' demographics, academic performance, and quality of college experience. Their findings reveal co-location of the students similar on ethnic minority identity, family income, and grades; Their regression models demonstrate that academic performance is positively associated with features derived from spatial-temporal networks. Grunspan et al [13] used social network analysis to understand undergraduate classrooms with the aim of predicting and improving learning outcomes. Wise et al [14] applied social network analysis and inductive qualitative analysis to examine the impact of how social interactions as learning support may impact massive open online courses.

In addition to use of LMS data, a growing number of studies have leveraged data from a range of other sources, including student records, institutional surveys, and external data sources, to provide a more comprehensive understanding of student performance. These studies have highlighted the importance of considering temporal relationships when developing models for student performance prediction. Liu et al [15] investigated how clickstream data can be used to predict students' learning behaviors, identify at-risk students, and inspire potential teaching and learning improvements. Gamulin et al [16] explored the use of student access time series to predict student final learning outcome in blended learning courses. They applied discrete Fourier Transforms and principal component analysis to improve and compress time series data, then built classification models based on naïve Bayes and support vector machines to predict student performance. Liu et al [17] proposed a student course result prediction model based on historical course results and course basic information. Their model is based on numeric and non-numeric feature vector embedding and model optimization with data augmentation and integration. Mitrovic et al [18] used a feed-forward neural network to predict the number of errors a student will make in database courses based on all actions a student performs in the class.

In summary, there is great interest in developing methods and models that help educators and institutions understand the journeys students go through during course enrollments. The driving question in many of these studies has been whether student performance can be accurately predicted early enough to intervene and provide needed help. Different data sources have been used to make such predictions, and many of the previous studies make use of fine-grained interaction and activity logs, which suffer from a lack of portability and robustness. This study makes predictions based on assessment data (quizzes, exams, assignments, and discussions) that are kept and extracted from the LMS.

Approach

This study looks at a student's journey throughout a course as a sequence of activities captured as a time series rather than as a single data point. Such a view leads to a finer-grained understanding of this journey and a better utilization of the readily available data. As mentioned above, the driving question here is whether student performance can be reliably predicted early enough to intervene. This paper explores a data set of such sequences for the purpose of developing robust and reliable models for early student performance prediction. Other questions explored by this study include:

- How does the progress of struggling students differ from one course to another across various formats and instructors?
- How early can student performance within these courses be accurately predicted?
- Can the cumulative progress of students at the end of the program be predicted?
- Are there patterns that transcend students and courses?
- How robust and portable are these predictions?

This paper approaches these questions in two ways:

- By devising certain important features that are used as predictors using traditional machine learning models.
- Using time series sequences to build forecasting models to help predict the performance of the student at the end of the course from any time point during the course.

Data Acquisition

The first step is to identify the courses suited for this study and collect the data. Such a process involves 274 lower-division CS courses required for students to get their associate degrees over a period of three years. These courses are selected for many reasons. First, as required CS courses, they tend to have more students. Secondly, they are also offered in more modalities than other courses. Many of them are offered multiple times in different formats by different instructors within the same semester. Thirdly, the quantity and quality of the data obtained from the LMS about a course depend on how much the LMS is used in that course. Most of these courses use LMS as the primary place of instruction where learning materials are posted, discussions ensue, and assignments and other graded activities are submitted. Finally, the period of three years allows for tracking students from the time they enter the program to the time they graduate. Although such analysis is not included in this paper, it is a future portion of this work-in-progress study.

The next step is to select which data to obtain from the LMS. The LMS keeps an extensive record of all the activities and events that take place in it. In addition to the basic information about students, it has data about assignments, quizzes, and other graded activities including submission attempts, scores, and due dates to name a few. There are also activity logs that it keeps about what, when, and how many times a student accesses a certain resource like a page, a module, or an assignment. This paper focuses on the LMS data related to assignments and other graded activities. This data consists of four sets pulled separately and then joined.

- Data about the course: name, start and end dates.

- Data about students' final scores.

- Data about the assignments and other graded activities: names, groups, weights, rules, and total possible points.

- Data about student interactions with these assignments and graded activities: submission attempts, lateness, and grades.

Two types of APIs (Application Programming Interfaces) were used to obtain these sets of data: RESTful and GraphQL. This is due to the nature of our LMS in which the GraphQL APIs, although more convenient, are still being developed. The data is obtained in JSON format and is converted to a tabular format that is more fitting for data analysis.

Data Preparation

The data underwent two preparatory steps: anonymization and standardization/normalization of all graded activity scores. Anonymization involved using ids instead of names for courses, instructors, students, assignments, and assignment groups. Standardization involved ensuring that the possible total score at the end of each course added up to 100%. This is also necessary to make sure that a score of 90% on a quiz that is worth 5% of the final score, for example, is not the same as a score of 90% on an exam worth 30% of the final score. This was a complex task as courses had to be grouped into categories and normalized differently based on their setup in the LMS. Calculations were verified by comparing the cumulative scores at the end of the course to the final score obtained from the LMS.

There were also some data cleanup steps. All the courses without any LMS student activity were removed. There were also students whose cumulative normalized scores did not match their final scores. Courses with such students were excluded from the final data set. The data set resulting from these preparatory steps contains time series sequences indexed by student ids, course ids, and timestamps. The data set contains columns for normalized and possible scores and cumulative normalized and possible scores. The next step is to analyze the data to gain insights.

Data Exploration and Analysis

The prepared data is then used for a variety of data exploration and analysis tasks. First, a few exploratory tasks are carried out to ensure that the data is correct. Then, the performance of struggling and passing students is compared using cumulative scores and possible scores. Student performance is typically measured by the final score at the end of the course, which is the result of accumulating the student's scores from all assignments and graded activities in the course. It makes sense then to use cumulative scores in these time series. In other words, the score of a student at any given time point is the sum of all their scores from the beginning of the course up to that time point. The possible score can also be added cumulatively. This is the score of a hypothetical student getting 100% on every assignment and graded activity at any given time point. Using the possible scores, we can compare the actual performance of a student to what is possible to achieve

Secondly, a similar analysis to the one described in a previous paper [1] was replicated using this large new data set, which includes far more courses, students, and instructors. This will help to determine how reliable and robust the results of that analysis are.

Thirdly, time series prediction and forecasting methods are applied to this data set with the goal of determining how early and accurately student performance can be predicted. The time series sequences in the data set vary in length, ranging from 39 time points in some courses to 376 time points in others. Time series with longer sequences contain long periods of inactivity, sometimes lasting for months. To standardize these sequences, we experimented with two approaches:

- Compressing the time series by removing the periods of inactivity. This works well for long time series, but it has the side effect of reducing short time series to just a few data points. While this approach does not affect the order of events, it destroys the information about how far apart these events are.

- Giving all time series sequences a fixed length. This involves choosing a value for that length, and 100 seems to be a logical choice because it allows us to think of each data point as the status of the student at that percentage point of the course. For longer time series, two or more data points may have to be combined. When this happens, a new time point is created whose values are the sum of normalized and possible scores. Doing this keeps the order of the events and therefore the shape of the time series (except for those combined time points). It also maintains the relative distance between data points.

The latter approach is what we report here. In addition to compressing and fixing the length of these time series sequences, two additional quantities were calculated: missed opportunity and relative achievement. The missed opportunity at a given point in time, t, is calculated as:

$$\text{missed\_opportunity}_t = \text{possible\_score}_t - \text{actual\_score}_t$$

It represents the amount of coursework that the student has missed so far. The relative achievement on the other hand is calculated as:

$$\text{relative\_achievement}_t = (\text{actual\_score}_t / \text{possible\_score}_t) \times 100$$

It represents how much of what is possible for the student to achieve is achieved.

For the time series analysis, and because the data set is by nature unbalanced (5471 passed students vs. 1497 failed students), a balanced sub-data set with the same number of passed and failed students was used. The following section presents and discusses the results of these steps.

Results

Data Exploration

To get a sense of what the collected data set tells us, one can compare the performance of two students in an arbitrary course: one who struggled and ended up failing the course and another who passed it. Figure 1 shows the journeys of these two students side by side. Both actual scores and possible scores allow us to visualize how students progress through courses using upward stair-like curves such as the ones shown in Figure 1. To label this data, the student's numeric final score is converted to a binary pass-fail value using a cutoff of 74%. A final score < 74% is considered a failing grade (i.e., they would need to retake the course to continue in the program), and a final score >= 74% is considered a passing grade. Furthermore, the horizontal steps are determined by the number of graded activities and how they are distributed throughout the course, while the vertical steps are determined by the weights of these assignments and activities.

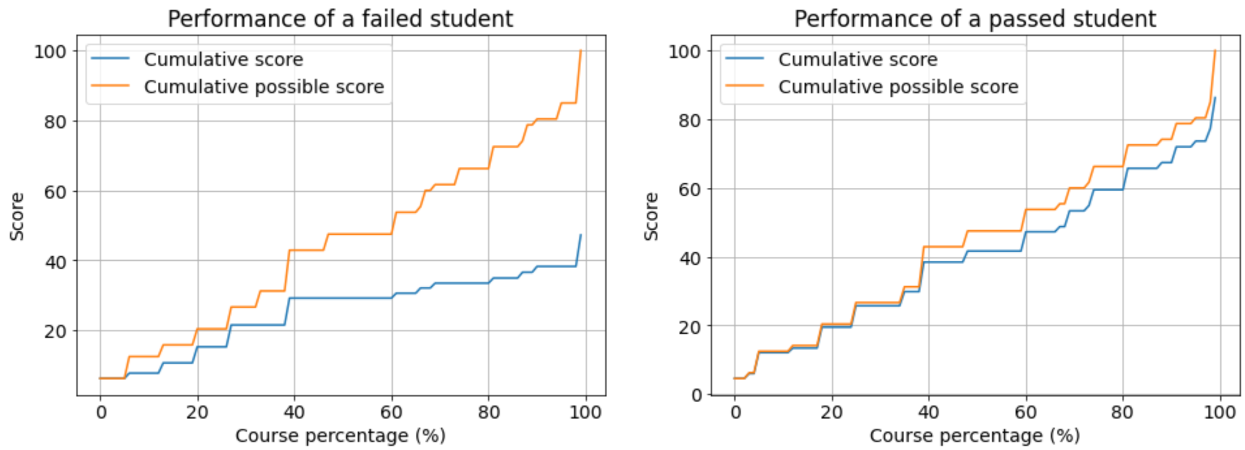

Figure 1: The progress of two students throughout a course

The student's struggle in a course can be visualized by how big or how small the difference is between their cumulative score and possible score curves. One can also use the data to see what progress looks like for an average at-risk (failed) student compared to that of an average passed student. Figure 2 shows such progressions side by side. As can be seen, averaging results in smooth (almost linear) curves instead of stair-like ones. The gap between actual scores and possible scores remains as before.

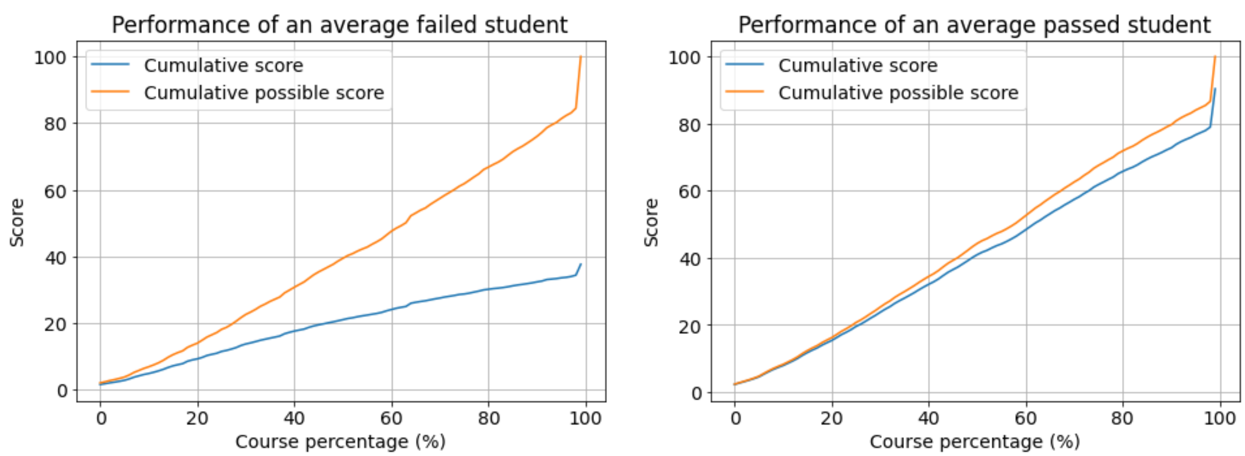

Figure 2: The progress of an average failed student compared to an average passed student.

Making Predictions

To predict whether a student will pass the course based on their performance up until a certain point of time in the course, we first need to determine what features will be used in the classification problem. In this paper, we build on the previous work of [1]. That paper identifies the four key features (see Table 1) which can be calculated at any point in time as students progress through the course, and we will use those same features here as well.

Table 1: Features used to classify a student as at-risk of failing.

Feature	**Description**
Normalized score intercept	The intercept of the line fitted to the students cumulative normalized scores for all submissions up to this point in time.
Normalized score slope	The slope of the line fitted to the students cumulative normalized scores for all submissions up to this point in time.
Missed opportunity	As defined above.
Relative achievement	As defined above.

To formulate this as a standard classification problem, we need to determine at what point in time within the course's progress we will compute the features and perform the classification. On such a diverse data set, picking a time point that is meaningful across all courses is challenging. A single point in time, such as February 10, 2020, will not work since some courses have long since concluded and others have not yet even begun. Similarly, picking a point in time relative to the semester (e.g., the third Thursday of the semester) is also problematic because not all the courses included in this study follow the traditional semester schedule. Some courses operate outside of the bounds of a semester. Instead, we define points in time relative to the course itself on a fixed scale from 0 to 100, essentially defining a point in time as a percentage of the course completion (by time).

Table 2: Used hyper-parameters values per algorithm

Algorithm	**Parameters**
Decision Tree	max_depth=2
Random Forest	max_depth=2
Logistic Regression	C=1e5
Multilayer Perceptron	max_iter=1000

By treating the data as a standard supervised learning binary classification problem, we can apply any of the many different binary classification algorithms to this problem. In this paper, we have chosen to compare the accuracy of four different algorithms: Decision Tree, Random Forest, Logistic Regression, and Multi-layer Perceptrons. For each algorithm, we use the Sci-kit Learn library [19] with the default parameters, except as noted in Table 2. We split the data set into a training data set and a testing data set by randomly shuffling the data and then selecting

25% of the instances for testing. Each algorithm is trained using the training data set, and then the accuracy is measured using the test data set. We repeat this process at 10% increments of progress in the course. The results are shown in Figure 3.

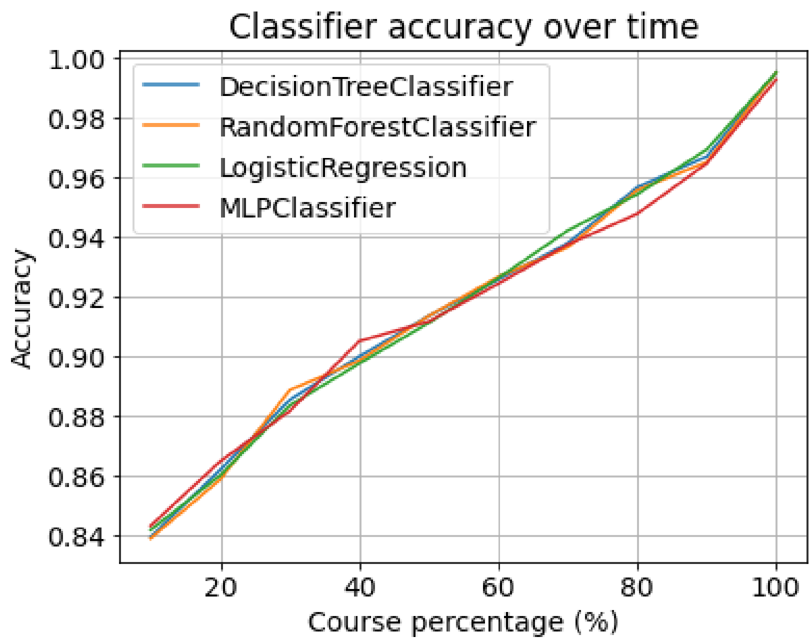

Figure 3: Accuracy over time for each binary classification algorithm.

As shown in Figure 3, the accuracy of the classification is similar, regardless of the classification algorithm used. In each case, the algorithms can correctly identify which students will pass or fail the course with a high degree of accuracy. With only 10% of the course completed, the algorithms hover around 84% accuracy, and by the time the course is about 40% complete, the algorithms can predict which students will pass or fail with an accuracy of about 90%. These results mean that we can identify students likely to struggle to complete the course while they still have time to change their trajectory and achieve success. The next step will be to determine what interventions can successfully alter a student's trajectory and integrate the two processes.

Time Series Prediction and Forecasting

Having time series sequences of cumulative scores with 100 time points each allows us to interpret each time point as a course percentage. The 10th time point represents the work done during the first tenth of the course semester. With that in mind, multiple ways of predicting student performance early in the course were tried. The following is a summary of such attempts.

First, we investigate how accurate relative achievement (and missed opportunity) would be as predictors. The idea here is to count, for every time point (course percentage), how many actual passed students would be predicted as such by these features and how many actual failed students would be predicted as such at that time point. Figure 4 shows how relative achievement can be used as a good predictor.

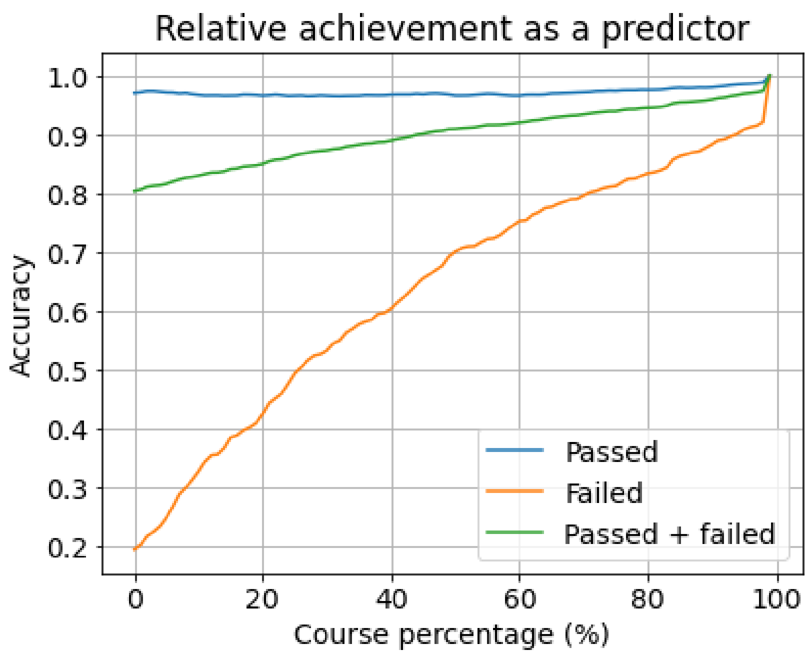

Figure 4: Relative achievement as a predictor

As can be seen from this figure, relative achievement is an excellent predictor of passed students (> 96% throughout the semester; blue curve). Its accuracy of predicting failed students, however, improves with time (the orange curve). At 40% of the course, it is able to predict failed students with 60% accuracy. Combining both curves results in the green curve which suggests >88% overall accuracy at 40% of the course. This is similar to the above results of Figure 3. Breaking the data by modality to see if some modality behaves differently results in Figure 5. This figure shows that the curves remain effectively the same for each modality. However, the virtual curves are less smooth due to the small number of virtual courses compared to face-to-face and online courses.

Using missed opportunity as a predictor did not lead to good results, and its curves were not included in this paper for brevity.

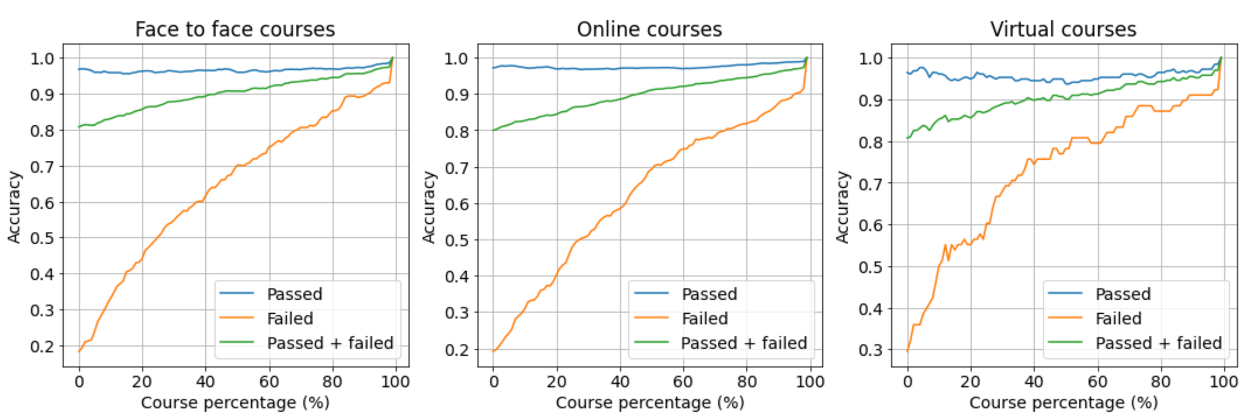

Figure 5: Relative achievement as a predictor broken down by modality

Next, we evaluate whether a time series classification model can accurately classify these sequences as either passed or failed. This is a multivariate model using four features: cumulative scores, cumulative possible score, relative achievement, and missed opportunity. Using a K-Nearest Neighbor time series classifier (K=3) results in 94.6% accuracy when given the whole sequence. We can use this classifier to predict (forecast) whether the student will pass or fail the course given a partial sequence. Figure 6 shows the accuracy of such a classifier at different course percentages.

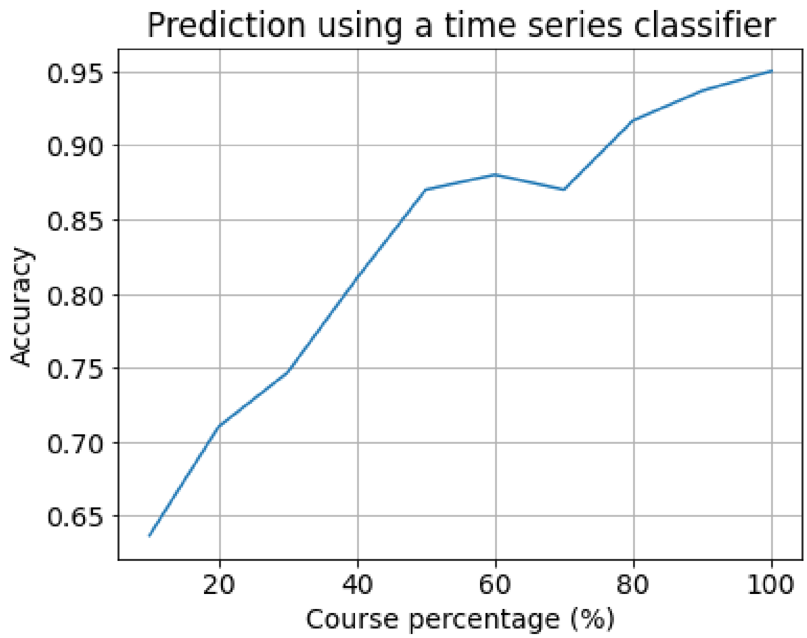

Figure 6: Student performance prediction using a time series classifier

The accuracy, as shown in this figure, improves over time, and at 30% of the course, at-risk students can be predicted with more than 75% accuracy.

Multivariate time series forecasting using deep neural networks (DNNs) implemented using Tensorflow [20] and Keras [21] was also performed. Three models were trained and evaluated:

- A one-dimensional Convolutional Neural Network (CNN) with two convolutional layers (64 units each)
- A Recurrent Neural Network with a single Long Short-Term Memory (LSTM) layer of 64 units
- A Recurrent Neural Network with a single Gated Recurrent Unit (GRU) of 64 units

Figure 7 shows the performance of these three models at increments of 10% of the course percentage. At a given percentage, these models will try to predict the final grade at the end of the course, given a sequence length equal to that percentage. For instance, at 30%, the sequence length will be 30 time steps. The 1D convolutional neural network, as shown in this figure, did not perform well. The accuracies of the RNN models (both using LSTM and GRU layers) increment with time for the most part, and their accuracies are slightly less than the results of both figures 3, 4, and 6.

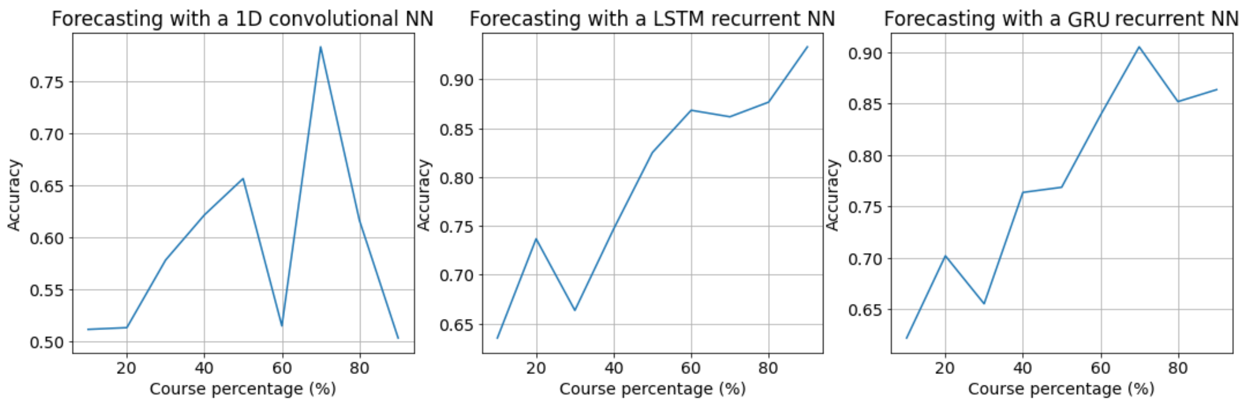

Figure 7: Performance of multiple DNN forecasting models

Concluding Remarks and Future Work

As can be seen from the above section, at-risk students can be predicted early on in the semester with sufficient accuracy. The fact that different models and methods are producing similar results speaks to the robustness of these results. Furthermore, these results are derived from data containing multiple courses with different modalities taught by different instructors, which speaks to their portability, as noted in [2].

These results also suggest that course modality does not seem to be a significant factor, at least not in the models used in this paper. Students' journeys throughout courses appear to be similar regardless of the course or instructor. This suggests the existence of some global patterns that repeat across multiple courses, sections, instructors, and modalities. Investigating the local patterns that are unique to students remains an open question of interest to this work-in-progress study.

The models presented in the paper need further improvement before they can be deployed to production. One way to improve these models is to augment the data they use with relevant student data from sources other than the LMS. As the background section discussed, student performance in courses is affected by multiple factors, some of which can be used to enhance the models presented in this paper.

Further work is needed to analyze the journeys of students across multiple courses. The current data set enables tracking students as they progress through the associate degree, making it possible to consider the course as the seasonality of these multi-course time series sequences. There is also a need to investigate whether incorporating activity counts into the models described in this paper can improve their performance, as previous research suggests that activity counts alone are not by themselves good predictors of student performance [7].

In summary, this paper demonstrated how time series sequences of graded activities can provide insights into student progression through courses. It presented and evaluated several approaches to using these sequences to develop models for early prediction of student performance, which can be crucial in providing timely support to struggling students, improving both learning outcomes and student retention.

Bibliography

[1] A. Al-Gahmi, K. Feuz, and Y. Zhang. "On Time-based Exploration of LMS Data and Prediction of Student Performance", 2022 ASEE Annual Conference & Exposition, Minneapolis, MN, 2022, August. ASEE Conferences, 2022.

[2] 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 Transactions on Learning Technologies, vol. 10, no. 01, pp. 17-29, 2017.

[3] D. Gašević, S. Dawson, T. Rogers and D. Gasevic, "Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success," The Internet and Higher Education, vol. 28, pp. 68-84, 2016.

[4] M. Riestra-González, M. d. P. Paule-Ruíz and F. Ortin, "Massive LMS log data analysis for the early prediction of course-agnostic student performance," Comput. Educ., vol. 163, pp. 104-108, 2021.

[5] S. B. Dias, S. J. Hadjileontiadou, J. Diniz and L. J. Hadjileontiadis, "DeepLMS: a deep learning predictive model for supporting online learning in the Covid-19 era," Scientific Reports, vol. 10, no. 1, p. 19888, 2020.

[6] R. Umer, A. Mathrani, T. Susnjak and S. Lim, "Mining Activity Log Data to Predict Student's Outcome in a Course," in Proceedings of the 2019 International Conference on Big Data and Education, New York, NY, USA, 2019.

[7] S. V. Goidsenhoven, D. Bogdanova, G. Deeva, S. v. Broucke, J. D. Weerdt and M. Snoeck, Predicting Student Success in a Blended Learning Environment, New York, NY, USA: Association for Computing Machinery, 2020.

[8] E. Sugiharti, S. Firmansyah and F. R. Devi, "Predictive evaluation of performance of computer science students of unnes using data mining based on naÏve bayes classifier (NBC) algorithm". Journal of Theoretical and Applied Information Technology, 95, 4 (2017), 902.

[9] S. Li and T. Liu, "Performance prediction for higher education students using deep learning. Complexity", 2021 (2021), 1-10.

[10] K. Chai and D. Gibson, "Predicting the Risk of Attrition for Undergraduate Students with Time Based Modelling", International Association for Development of the Information Society (2015).

[11] X. Wang, X. Yu, L. Guo, F. Liu and L. Xu, "Student performance prediction with short-term sequential campus behaviors", Information, 11, 4 (2020), 201.

[12] Q. Nguyen, O. Poquet, C. Brooks and W. Li, "Exploring Homophily in Demographics and Academic Performance Using Spatial-Temporal Student Networks", International Educational Data Mining Society (2020).

[13] D. Grunspan, B. Wiggins, and S. Goodreau, "Understanding classrooms through social network analysis: A primer for social network analysis in education research", CBE—Life Sciences Education, 13, 2 (2014), 167-178.

[14] A. Wise and Y. Cui, "Learning communities in the crowd: Characteristics of content related interactions and social relationships in MOOC discussion forums", Computers & education, 122 (2018), 221-242.

[15] Y. Liu, S. Fan, S. Xu, A. Sajjanhar, S. Yeom and Y. Wei, "Predicting Student Performance Using Clickstream Data and Machine Learning", Education Sciences, 13, 1 (2022), 17.

[16] J. Gamulin, O. Gamulin and D. Kermek, "Using Fourier coefficients in time series analysis for student performance prediction in blended learning environments", Expert systems, 33, 2 (2016), 189-200.

[17] J. Liu, C. Yin, K. Wang, M. Guan, X. Wang and H. Zhou, "Students' Course Results Prediction Based on Data Processing and Machine Learning Methods", Journal of Signal Processing Systems, 94, 11 (2022), 1199-1211.

[18] A. Mitrovic and T. Wang, "Using Neural Networks to Predict Student's Performance," in Computers in Education, International Conference on, Auckland, New Zealand, 2002 pp. 969.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg and J. Vanderplas , "Scikit-learn: Machine Learning in Python," The Journal of Machine Learning Research, vol. 12, p. 2825–2830, 2011.

[20] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015, Software available from tensorflow.org.

[21] F. Chollet, "Keras", 2015, Software available from https://keras.io.