JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Volume 7(2), 1-17. http://dx.doi.org/10.18608/jla.2020.72.1

# Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance

Fu Chen[1], Ying Cui[2]

**Abstract**

Predictive analytics in higher education has become increasingly popular in recent years with the growing availability of educational big data. Particularly, a wealth of student activity data is available from learning management systems (LMSs) in most academic institutions. However, previous investigations into predictive analytics in higher education using LMS activity data did not adequatelyaccommodate student behaviours in the form of time series. In this study, we have applied a deep learning approach — long short-term memory (LSTM) networks — to analyze student online temporal behaviours using their LMS data for the early prediction of course performance. To reveal the potential of the deep learning approach in predictive analytics, we compared LSTM networks with eight conventional machine-learning classifiers in terms of the prediction performance as measured by the area under the ROC (receiver operating characteristic) curve (AUC) scores. Results indicate that using the deep learning approach, time series information about click frequencies successfully provided early detection of at-risk students with moderate prediction accuracy. In addition, the deep learning approach showed higher prediction performance and stronger generalizability than the machine learning classifiers.

Notes for Practice

- Machine learning classifiers have been widely used in predictive learning analytics (PLAs) in higher education, which requires extensive work on feature engineering and a large course with many failing students.
- This study finds that, compared with conventional machine learning models, LSTM networks can be used to predict student course performance with higher accuracy and generalizability using time-series dependencies between student daily click frequencies in the learning management system.
- The LSTM approach uses a simple feature for prediction, which is more likely to be successfully applied in a wide range of courses.
- The LSTM approach can be an effective screening tool for detecting at-risk students regardless of course type, which improves the efficiency and affordability of in-process course evaluations.

**Keywords**

Learning analytics, predictive analytics, learning management system, long short-term memory network, LSTM, machine learning

**Submitted:** 18.09.2019 — **Accepted:** 22.05.2020 — **Published:** 19.09.2020

*Corresponding author[1] Email: fu4@ualberta.ca Address: 6–110 Education Centre North, Department of Educational Psychology, University of Alberta, Edmonton, Alberta, Canada, T6G 2G5*

*[2] Email: yc@ualberta.ca Address: 6–110 Education Centre North, Department of Educational Psychology, University of Alberta, Edmonton, Alberta, Canada, T6G 2G5*

## 1. Introduction

Thanks to the increasing availability of and access to educational big data in this era, learning analytics is playing a growing role in addressing many contemporary challenges in education (Daniel, 2015; Siemens & Long, 2011). At the first International Conference on Learning Analytics and Knowledge, learning analytics was defined as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimising learning and the environments in which it occurs" (Long, Siemens, Conole, & Gašević, 2011). Particularly, learning analytics in higher education has received increasing attention from both the education and computer science communities in recent years.

According to a review by Cui, Chen, Shiri, and Fan (2019), the number of publications on predictive learning analytics (PLAs) in higher education has dramatically increased in the past ten years. Given the ever-expanding developments in data science techniques, more researchers, senior administrators, and educators have realized the potential of data available in different educational systems for improving educational outcomes. For instance, predictive analytics, an important strand of research in learning analytics, can help institutions identify at-risk students who may drop out of a program or fail a course, make better decisions, and derive actionable insights (Daniel, 2015).

One of the earliest applications of predictive analytics in higher education is the Course Signals system at Purdue University (Arnold & Pistilli, 2012), which used student grades, demographic information, and behaviours in the learning management system (LMS) to predict their course performance. Subsequently, more and more institutions applied predictive analytics in their educational systems to explore the relationships between student data from different sources and their academic performance (Sclater, Peasgood, & Mullan, 2016). However, how institutions built the models, incorporated the models in LMSs, and how well these models performed have not been well documented. Fortunately, a large body of empirical studies from researchers in education and computer science can be found in the literature. For example, in Romero, Espejo, Zafra, Romero, and Ventura's (2013) study, final course marks could be predicted by students' click frequencies and time in the LMS with moderate accuracy. In their study using student online learning behaviours and demographic information extracted from the massive open online courses (MOOCs), Al-Shabandar et al. (2017) found that student click stream actions were strongly correlated with their academic success. In addition, Luo, Sorour, Goda, and Mine (2015) found that student grades could be successfully predicted by their free-style comments after lessons with an accuracy over 80%. These studies have demonstrated the beneficial use of predictive analytics in higher education with different sources of information utilized in predictive models.

In this study, we use students' daily click frequencies in the LMS to predict their course performance in large undergraduate courses at a western Canadian university using a deep learning model — long short-term memory (LSTM) networks. We trained the LSTM networks with data within different time frames (i.e., four, six, eight, and ten weeks) to examine how early a good prediction can be achieved. We evaluated the performance of the LSTM networks in comparison to conventional machine learning classifiers in terms of their prediction accuracy. In the following sections, we provide the rationales for using LMS data and the details of our methodologies.

## 2. LMS for Learning

Nowadays, using LMSs for teaching and learning has become increasingly popular in higher education (Cole & Foster, 2007). LMSs are web applications providing a variety of tools facilitating teaching and learning, which include sharing course materials, online forums, online quizzes, collecting and evaluating assignments, and recording grades (Cole & Foster, 2007). By using these tools, instructors are able to deliver timely feedback to students and design appropriate interventions. Moreover, student actions in LMSs are recorded and stored by the system, which could be analyzed by institutions, administrators, and instructors to gain insights from student online behaviours.

In terms of constructivist learning theory (Hein, 1991), LMSs are important for learning success in higher education. The theory stresses that learning is an active process; learners are not passive recipients of knowledge inputs, but should actively engage in learning and construct meaning from learning resources (Duffy & Cunningham, 1996). In this regard, LMSs are designed for integrating multiple learning resources and delivering them in a flexible way, which have the potential to provide a platform that facilitates students to construct knowledge by themselves. More specifically, LMSs allow students to get access to all the course materials in a convenient and contextual manner, facilitate them to integrate what they have learned in the course, and help construct meaning from the new course materials. In addition, because learners are able to collectively create cognitive strategies or construct common knowledge through the social context provided by the learning community (Edwards & Mercer, 2013), the connections between learners and their instructors, peers, and others are important for successful learning. In this sense, LMSs are capable of offering students a social tool for communicating with their peers and instructors about difficulties in learning, allowing instructors to provide timely and constructive feedback and support.

Theoretically, in what ways do LMSs affect student learning? A possible answer is that LMSs might play a role in influencing self-regulated learning processes. According to Winne and Hadwin's (1998) model, student self-regulated learning is a four-stage event consisting of 1) clarifying tasks, 2) setting goals and making plans, 3) adopting tactics and strategies, and 4) examining the outcomes from previous stages. Each of these learning stages is initialized by task conditions (e.g., learning time, learning resources, and social context) and cognitive conditions (student interest, previous knowledge, and motivation), followed by student cognitive processes faced with tasks and then their products (e.g., student behaviours and performance), which are finally evaluated by internal or external feedback and standards. These elements of self-regulated learning are directly or indirectly associated with the features of LMSs. For example, to improve task conditions of self-regulated learning, instructors can provide a variety of learning resources through LMSs and students are able to be socially connected with instructors and other students. Moreover, LMSs are capable of recording learning products, explicating internal and external

ISSN 1929-7750 (online). 

evaluation standards, and stimulating student interest and motivation by establishing a more engaging and intriguing learning context. To sum up, self-regulated learning can be facilitated by a well-designed LMS, and thus, student interaction with the LMS might be indicative of their learning outcomes.

Given the common use of the LMS in higher education and its importance for teaching and learning, a large body of studies on predictive analytics have used LMS data together with student information from other sources to predict academic performance. In their review, Cui et al. (2019) identified several major student-level data sources used in predictive analytics literature, including intermediate course performance, like quiz grades (e.g., Luo, Koprinska, & Liu, 2015), student behaviours in LMSs (e.g., Romero et al., 2013), survey data on socio-emotional variables (e.g., Guarín, Guzmán, & González, 2015), demographic information (e.g., Evale, 2016), and academic history (e.g., Ochoa, 2016). Studies that utilized multiple data sources often showed a high classification accuracy rate. For example, using students' academic history, social behaviours, demographic information, and educational background to predict dropout, the prediction accuracy could reach over 80% (Meedech, Iam-On, & Boongoen, 2016).

This study focuses on the use of LMS data for the prediction of student course performance based on two considerations. First, the use of multiple sources of information requires a data warehouse that integrates data from various information systems, which is not available for all institutions. In most academic institutions, instructors often have access to student data from the LMS only. Second, LMS data are particularly useful in generating actionable information that helps design interventions. Based on the analytics of LMS data, for example, feedback regarding how students can change their behaviours to increase their chances of learning success can be provided (e.g., participate in group discussions or submit assignment on time). Therefore, this study focuses on exploring how to make best use of LMS data without other student information for prediction, which might be a challenging but valuable topic.

Moreover, early prediction and intervention is a key goal of predictive analytics (Macfadyen & Dawson, 2010). However, some previous studies in this area used student LMS data from the entire course (e.g., Conijn, Snijders, Kleingeld, & Matzat, 2017), which might reveal the influential online behaviours contributing to academic success, but has limited practical implications for timely intervention. Other studies focusing on early detection of at-risk students have shown that data from the early stages of a course can be successfully used to predict student learning outcomes. For example, Casey and Azcona (2017) used student online behaviours in the LMS to predict their pass/fail results, finding that using the first-four-week data could lead to an accuracy rate above 75%. Milne, Jeffrey, Suddaby, and Higgins (2012) found that success in a course was positively related to student LMS usage during the first week. In addition, some studies have shown that the accuracy of learning outcomes predicted by student LMS behaviours increased over time (e.g., Casey & Azcona, 2017; Hu, Lo, & Shih, 2014; Schell, Lukoff, & Alvarado, 2014).

## 3. Methods for Predictive Analytics

Examining the methods for predictive analytics in higher education is another common theme in the literature. For example, most review papers in this area (e.g., Cui et al., 2019; Shahiri & Husain, 2015) have identified the most commonly used machine learning classifiers, such as logistic regression, decision tree, naïve Bayes, support vector machine, neural networks, and k-nearest neighbours. In addition, many studies on predictive analytics compared the prediction performance between different machine learning classifiers. Despite the success of these methods and their variants, using machine learning techniques for predictive modelling often requires many attempts at data preprocessing and feature engineering since structured datasets are often required for most machine learning classifiers. Not many studies have explicitly reported how they pre-processed data and extracted features (Cui et al., 2019), and there is no explicit guidance on how to do data pre-processing and feature engineering. This may be because different datasets require different pre-processing procedures, and the selection of features is often arbitrarily decided by how researchers think of the potential factors influencing learning outcomes.

One way to avoid intensive data pre-processing and feature engineering is to use the time-series data in the LMS, such as daily student usage. According to the self-regulated learning theory mentioned above, student learning involves several consecutive stages. Aggregating student learning behaviours as single components might lead to the loss of information revealing how students progressively achieve their learning goals. This might in turn decrease the accuracy of predictive models based on machine learning classifiers. One way to analyze student time series data is to use the deep learning approach in data science. Deep learning is a subset of machine learning based on artificial neural networks, and it is capable of solving complex problems given extremely diverse and unstructured datasets. Among various deep learning models, recurrent neural networks, such as LSTM networks, have become increasingly used in recent years to analyze time series or sequential data. However, the application of deep learning models in predictive analytics is still rare, with very few studies reported in the literature (Coelho & Silveira, 2017). For example, Okubo, Yamashita, Shimada, and Ogata (2017) used 108 student course behaviours (e.g., attendance, course views, report submission, etc.) to predict their final course marks using LSTM networks. Their results showed that final grades could be predicted by student behaviours during the first four weeks with an accuracy

rate over 80%. The potential of their approach for early detection of at-risk students was validated by their other study (later the same year) using log data from 937 students in six courses (Okubo, Yamashita, Shimada, & Konomi, 2017). In addition, in a study using student event streams in the LMS for graduation prediction, Kim, Vizitei, and Ganapathi (2018) proposed a deep learning approach based on the bidirectional LSTM network, which was capable of improving the prediction accuracy substantially in the first few course weeks. Generally, these studies have shed light on the potential for using student time series behaviours in the LMS to predict course performance by deep learning. Despite their promising findings, however, it is still unclear the extent to which the deep learning approach improves the predictive model in contrast to the widely used machine learning classifiers with aggregated LMS features. In addition, some of these approaches exploit a variety of student information, which sometimes is not fully available to course instructors. Therefore, a simple, generalizable deep learning framework is needed and its prediction performance would be more convincing if compared with other conventional machine learning models.

In our study, we use students' daily click frequencies without any other auxiliary information to predict their final course performance. Particularly, our approach is devised to make early detection of at-risk students based on their online activities in the LMS, which facilitates timely warning for enhanced course learning. Given the time-series nature of LMS log file data, we adopt LSTM networks to model long-term dependencies of LMS activities. To further demonstrate the effectiveness of our approach, we conduct a comprehensive examination of conventional machine learning classifiers as baselines. In addition, given that machine learning classifiers require multiple aggregated features for prediction, in our approach, we use the most influential features to cross validate the choice of daily click frequencies as a representation of LMS behaviour. Our specific research questions are as follows:

1. Do LSTM networks outperform conventional machine learning classifiers in predicting course performance?
2. Are LSTM networks capable of detecting at-risk students early?
3. Do machine learning classifiers suggest that click frequencies are predictive of course performance?

## 4. Methods

### 4.1. Data

This study used LMS data from a mandatory undergraduate course offered at a large Canadian university. The course was administered through Moodle (https://moodle.org/), an open-source, free LMS designed to facilitate learning and teaching in the educational context. With Moodle, instructors can flexibly design modules according to their syllabus, providing students with online access to activities, course materials, communication tools, and assessments. Students therefore can benefit their learning by having more interactions with the instructor, their peers, and the course content. The research team had access to anonymized data and all activities were conducted in accordance with the ethical and scientific requirements of the university research ethics board.

In this study, we used log file data and grade books from a mandatory undergraduate course in the Faculty of Education. Pre-service teachers take this course to prepare for becoming a primary or high school teacher. Specifically, they learn the important concepts and issues regarding how to develop evaluation instruments and evaluate student performance. The course is offered every fall and winter semester, and the course structure has been kept consistent in recent years. Students' final grades are determined by their performance in two assignments — a midterm exam and a final exam. To facilitate student learning, some practice quizzes and practice assignments are provided on Moodle.

Data from two semesters were used for training, validating, and testing predictive models. A total of 141 and 527 students attended this course in the two semesters, respectively. Considering that using a small training sample with a large number of features for learning may lead to overfitting, we used the data from semester 2 (the larger sample) to build the predictive model. Seventy-two percent of the data (407 students) in semester 2 was randomly selected for training and validation; the remaining 28% of the sample (120 students) was used for testing. Although the test dataset from semester 2 was absolutely unseen for training each classifier, it shares the same course structure and components with the training dataset. It is therefore expected that each classifier should have a reasonable prediction performance on the test dataset from semester 2. Moreover, we also tested the models on semester 1 to evaluate the model generalizability. Despite having similar course structures and requirements across different academic years, different instructors taught this course across the two semesters. Therefore, the datasets from these two semesters were not completely homogeneous. As such, if the predictive models performed well on the test dataset from semester 1, their generalizability would be validated to some extent.

### 4.2. Class Imbalance Handling

In this study, we used final grades as the indicator of course performance. Very few students (two in semester 2 and none in semester 1) failed the course (see Table 1). Therefore, the at-risk students in this study were defined as those who might get a final mark of C+ or below. This is an arbitrary cut-off point due to the limited number of failing students. For courses with

more failing students, it is suggested to use pass/fail as the cut-off point. On average, over 70% of students got a course mark above B-, indicating good course performance. The target with fewer than 25% of students as one group (poor performance) was imbalanced, which should be handled prior to training. We used a well-known sampling method, synthetic minority oversampling technique (SMOTE; Chawla, Bowyer, Hall, & Kegelmeyer, 2002) to mitigate class imbalance. SMOTE first finds a data point and its *k* nearest neighbours in the feature space, and then randomly selects one vector between this data point and one of its *k* nearest neighbours. This vector is then multiplied by a random value between 0 and 1 and added to the original data point to generate a synthetic data point. It is worth noting that oversampling by SMOTE is typically used for the training dataset but not the validation and test datasets, because the true performance of a classifier can only be validated by the original and authentic data, although it is unlikely that the test datasets are balanced in reality.

**Table 1.** Final Mark Distribution for Each Dataset

| Target | Final Mark | Semester 1 | Semester 2 |
|---|---|---|---|
| Good | A+ | 2 | 16 |
| | A | 7 | 41 |
| | A- | 20 | 85 |
| | B+ | 39 | 99 |
| | B | 25 | 82 |
| | B- | 24 | 73 |
| Poor | C+ | 13 | 71 |
| | C | 2 | 35 |
| | C- | 6 | 18 |
| | D+ | 3 | 4 |
| | D | 0 | 1 |
| | F | 0 | 2 |
| Total | Good | 117 | 396 |
| | Poor | 24 | 131 |

### 4.3. LSTM Networks

We used LSTM networks as a deep learning approach for predicting course performance by student time series LMS behaviours. We first subset student click frequency data during the first 28 days of the semester and then calculated each student's click frequency for each of the 28 days. The time frame was chosen given the consideration of early prediction. This resulted in a dataset of click frequency with 28 time slots for the LSTM network. The outputs of this first-28-days network were early predictions of course performance. However, we also implemented the 42-days, 56-days, and 70-days LSTM networks to reveal how early the model would be capable of making a good prediction by deep learning. The LSTM networks were implemented by *keras* in python (Chollet, 2015).

#### 4.3.1. Introduction to LSTM networks

The LSTM network is a subset of recurrent neural networks (RNNs), which are neural networks (NNs) including temporal information. As shown in Figure 1, loops allow information to be transmitted from one time slot to its subsequent time slot in RNNs. When unrolled, an RNN looks similar to a normal NN but involves multiple copies of the same NN with information passing from one network to a successor. Specifically, the output of a network at one time slot would become its input at a subsequent time slot, which makes RNNs very successful in modelling time series data. However, RNNs suffer the vanishing or exploding gradient problem, especially when learning over data of long sequences (see Bengio, Simard, & Frasconi, 1994). This is because with multiple time slots, successive multiplication with the recurrent weight matrix is needed to update the weights, which might lead to either disappearing or explosive gradients during backpropagation. As such, RNNs are only good at learning over data of short sequences. In our study, student time series behaviours are in the form of long sequences because students might log into the LMS multiple times every day during the semester. In addition, we assume that student learning outcomes are attributable to their online behaviours over a long period rather than over several consecutive days. For example, given two students with the same aggregated LMS login frequency, one student might check the LMS every day over the semester while the other might visit very frequently before assignment due dates or exams. The discrepancy in these long-term behaviours might reflect student motivation towards and engagement in learning. As such, long sequences of online behaviours rather than aggregated indicators is needed to represent student differences in learning. We therefore use a variant of RNNs — LSTM networks (Hochreiter & Schmidhuber, 1997) — to learn over student time-series behaviours in the LMS, given its advantage of exploiting long-term memory with long sequences. Specifically, LSTM networks address the problem of vanishing and exploding gradients by introducing a forget gate, an input gate, and an output gate in an LSTM cell. These three

gates bring in a large degree of flexibility for LSTM networks to model time series dependencies over long sequences by a fine-grained control over the information of inputs, the information to be remembered or forgotten in the internal cell state, and the information of outputs. LSTM networks demonstrate impressive power in many application domains, such as robot control (e.g., Mayer et al., 2008), speech recognition (e.g., Graves & Schmidhuber, 2005), and medicine (e.g., Choi, Bahadori, Schuetz, Stewart, & Sun, 2016). In education, applications with LSTM networks are relatively rare, but they have been successfully used to address some emerging educational issues of interest in learning analytics (e.g., Coelho & Silveira, 2017) and personalized learning recommendations (e.g., Zhou, Huang, Hu, Zhu, & Tang, 2018).

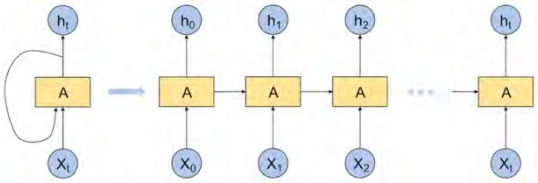


**Figure 1.** Recurrent neural network and its unrolled form.

Figure 2 demonstrates the architecture of an example LSTM network used in this study, which is built on the data of the first-28-days click frequencies. The input data are of three dimensions: a batch size of 64 samples, 28 time slots, and a feature size of one (click frequency). On top of the input nodes, an LSTM layer is stacked in the network, which models the time dependencies between click frequencies. We conducted a hyperparameter search on five candidate LSTM output node sizes — 10, 20, 30, 40, and 50 — and selected 20 as the final one. The outputs of the LSTM layer are also of three dimensions: the batch size, the number of time slots, and the output node size. Only the last output in the sequence is fed into the next layer, leading to a two-dimensional output of batch size and node size. Thereafter, a dropout layer is stacked on the LSTM layer to mask its outputs by dropping 50% of them. The dropout layer is used as a regularization technique to prevent overfitting (Gal & Ghahramani, 2016). Given that the shape of LSTM outputs does not accord with the target shape, a dense layer with the Softmax activation function is added to produce the final predictions. The dense layer changes the LSTM output shape from three dimensions to two, dropping the time slots. As such, the final outputs of the model align with the targets in terms of data shape. The Adam algorithm (Kingma & Ba, 2014) was used to optimize each LSTM model. An epoch size of 64 samples was used for training; the binary cross entropy was used as the cost function. According to previous recommendations (e.g., Goodfellow, Bengio, & Courville, 2016, p. 192; Lippmann, 1987), one or two hidden layers are typically sufficient for neural networks to classify samples; we therefore adopted one hidden layer. For the number of nodes within each hidden layer, however, there are no rules, so configuring the number of nodes by trial and error is considered feasible. In this study, the final numbers of nodes used for the 28-day, 42-day, 56-day, and 70-day LSTM networks are 20, 40, 50, and 60 respectively. Moreover, given that LSTM networks introduce randomness into learning (e.g., randomly initializing weights, randomly shuffling data in optimization), we repeated each model 10 times to reduce the stochastic influence.

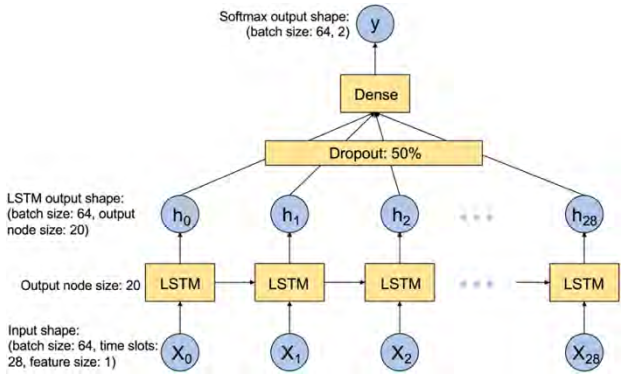


**Figure 2.** Our approach: Modelling time-series behaviours with LSTM networks.

#### 4.3.2. Evaluation Metric

In this study, we used AUC as the evaluation metric for both LSTM networks and conventional machine learning classifiers. AUC indicates the area under the receiver operating characteristic (ROC) curve (Ling, Huang, & Zhang, 2003). In a ROC curve, sensitivity is plotted as a function of the false positive rate (1 minus specificity) for different cut-off values of predicted probabilities of a class. The AUC can be interpreted as the probability that a random low-performing student is ranked as more

likely to be low-performing than a random high-performing student. AUC rates range from 0 to 1. An AUC rate of 1 indicates that the model is perfect in discriminating high- and low-performing students, and an AUC rate of 0 indicates that the model incorrectly classified all high-performing students as low-performing and all low-performing students as high-performing, which is extremely unlikely. AUC has a chance rate of 0.5, indicating a completely random model. The acceptable AUC range for a predictive model depends on the context. Typically, in most research areas, an AUC rate above 0.7 is preferred (e.g., Mandrekar, 2010; Rice & Harris, 2005).

In our study, we used AUC as the evaluation metric because it is insensitive to class imbalance when evaluating models. Given a highly imbalanced dataset with many positive samples and very few negative samples, a classifier can reach a very high accuracy rate by simply predicting all samples as positive. However, in this case, the classifier should be considered useless despite such a high accuracy rate, and AUC should be used as an alternative evaluation metric. Technically, most machine learning models use predicted probabilities to produce predicted class labels. Unlike accuracy and other evaluation metrics typically assuming the probability of 0.5 as a cut-off value for predicting a sample as positive or negative, AUC evaluates a model using different cut-off values. This feature of AUC is especially advantageous for PLAs given that in reality only a limited number of students might fail or struggle with a regular university course and a large percentage of students might drop an online course. In addition, in terms of model comparison of different machine learning classifiers, it is evident that AUC is more discriminating than the metric of accuracy for revealing a model's predictive capacity. For example, naïve Bayes demonstrated higher prediction performance than decision tree in terms of AUC despite their performance being similar in terms of accuracy (Ling et al., 2003). In addition, when comparing different predictive models, it is desirable to compare the entire ROC curve of different models rather than their performance at a particular point (e.g., accuracy rates), since they provide more information on the overall capability of a model discriminating high- and low-performing students. In this sense, as a summative indicator of the ROC curve, AUC is preferred in our study.

### 4.4. Machine Learning Classifiers

We implemented eight of the most widely used classifiers including NN, logistic regression (LR), naïve Bayes (NB), support vector machine (SVM), decision tree (DT), k-nearest neighbours (kNN), Random forest (RF), and gradient boosting machine (GBM) for the predictive modelling using the *caret* package in R (Kuhn, 2008). All data pre-processing tasks for machine learning were conducted in R (R Core Team, 2018).

#### 4.4.1. Feature Extraction

For traditional machine learning classifiers, student online activities in the LMS over the same first 28 days as the LSTM network approach were used for predictive modelling. We built LSTM network models over different periods to examine whether the early prediction was as accurate as that over the full duration of the course. However, we did not extract features over longer periods of time for machine learning classifiers because machine learning models were used as baselines in comparison with the LSTM network approach in terms of the accuracy of early prediction. In accordance with most previous studies (e.g., Casey & Azcona, 2017; Conijn et al., 2017; Romero et al., 2013), we extracted two major features from the log data: time-related and frequency-related features. The time-related features measure the time students spent in the LMS as well as their time-use habits (e.g., consistency, measured by the standard deviation of time durations). The online time was only calculated for modules "Quiz" and "File" because student interactions with the other modules during the first four weeks were rare. In addition, it is also unlikely that students spent long in viewing their grades, and most students used the "Assignment" module only for the assignment submission. The frequency-related features are simple counts of student clicks in the LMS. In addition, click frequencies on/off campus, and during weekdays/weekends were also calculated as features.

There are several features related to online sessions. In this study, an online session is defined as a series of LMS events occurring within a continuous period of time. Unlike click frequency, which only counts all the online events without considering their continuity, an online session includes a login action and a logoff or exit action with other online actions in between. As such, the number of online sessions can be considered another measure of how frequently students use the LMS. However, there are no explicit login/logoff options in the LMS, and likewise there are no login/logoff actions recorded in the log files. We therefore use a chunk of consecutive actions in the log files to indicate an online session, which, however, is not accurate enough for reflecting students' real online sessions. In addition, we cannot directly observe what students did in each online session, so it is therefore possible that an online session is not a real collection of continuous events on the LMS. For example, a student might click on the lecture notes, then leave to visit other websites, then return to read the lecture notes later. In this case, although the system records this student's activities as consecutive events, they can actually be divided into two different online sessions. In this study, a total of 21 features are extracted from student log file data with no categorical features (see Table 2).

**Table 2.** Descriptive Summary of Each Feature for Training Data for the Education Course

| No. | Feature | *M* | *SD* | Median |
|---|---|---|---|---|
| 1 | Number of total clicks | 188.00 | 95.37 | 180.00 |
| 2 | Number of clicks on campus | 87.27 | 63.55 | 73.00 |
| 3 | Ratio of on-campus to off-campus clicks | 2.79 | 11.50 | 0.78 |
| 4 | Number of online sessions | 1208.47 | 1285.01 | 778.00 |
| 5 | Total time for all online sessions | 143.94 | 127.18 | 80.87 |
| 6 | Standard deviation of online session time | 13.51 | 5.22 | 13.00 |
| 7 | Mean time of online sessions | 85.53 | 75.99 | 58.57 |
| 8 | Standard deviation of time between online sessions | 3064.32 | 1722.16 | 2697.03 |
| 9 | Number of clicks during weekdays | 158.57 | 80.44 | 148.00 |
| 10 | Number of clicks during weekends | 29.44 | 37.39 | 16.00 |
| 11 | Ratio of weekend to weekday clicks | 0.24 | 0.53 | 0.11 |
| 12 | Number of clicks for module “Assignment” | 7.84 | 6.08 | 8.00 |
| 13 | Number of clicks for module “File” | 38.56 | 19.10 | 35.00 |
| 14 | Number of clicks for module “Forum” | 3.01 | 9.83 | 0.00 |
| 15 | Number of clicks for module “Overview report” | 0.06 | 0.40 | 0.00 |
| 16 | Number of clicks for module “Quiz” | 42.69 | 42.06 | 37.00 |
| 17 | Number of clicks for module “System” | 90.63 | 44.71 | 83.50 |
| 18 | Number of clicks for module “User report” | 2.51 | 4.06 | 1.00 |
| 19 | Total time on module “Quiz” | 108.88 | 294.91 | 17.00 |
| 20 | Total time on module “File” | 576.36 | 700.29 | 292.50 |
| 21 | Standard deviation of time on module “File” | 99.14 | 117.69 | 41.96 |

*Note*: The descriptive summary of each feature was calculated over the first 28 days.

The first ten features relate to the overall student usage of Moodle during the first four weeks. Features 11–21 are related to seven Moodle modules utilized by the course instructors, as described below.

- **System**: The system provides the bare bones for instructors to construct the course and build modules. Student activities here are mainly recorded as viewing the course in the log data.
- **Assignment**: Instructors can assign tasks to students, collect student work, and provide grades and feedback.
- **Forum**: Course participants can engage in online discussions and other forms of networking. For example, students can use the forum as a social place to get to know each other, instructors can deliver course announcements, and students can give feedback to their peers and instructor anonymously.
- **Quiz**: Instructors may create online quizzes for evaluation purposes, or students may use them for self-assessment. Except for essay questions, each quiz question is marked automatically. Students can view their quiz grades in the gradebook.
- **File**: The file module is for instructors to upload course resources such as lecture notes, in-class presentations, or even course-related mini websites.
- **Overview report**: The overview allows students to view all the courses in which they are enrolled and their grade for each.
- **User report**: The user report includes a student’s grades for each assignment, the grading weights of each component, the feedback given by instructors or teaching assistants, the grade ranking in comparison with peers, and the overall grade for the course.

#### 4.4.2. Missing Data Handling

As we analyzed the data, we found that some extracted features had missing values. For example, for the ratio of on-campus to off-campus clicks, if all of a student’s LMS clicks were on campus — for example, if they lived in residence — the ratio would be infinite, which was recoded as a missing value for imputation. In addition, for the features related to standard deviations, if there was only one sample of data, the estimated standard deviation was undefined, which was recoded as a missing value for imputation. The k-nearest neighbour imputation (kNN) was carried out for the missing values. kNN is a powerful algorithm for handling all kinds of missing data. For an arbitrary missing value, kNN finds its closest neighbouring values in the training dataset (e.g., using the mean of these neighbouring values). In addition, each feature was normalized by standardizing the data with a mean of 0 and a standard deviation of 1.

#### 4.4.3. Feature Selection

Given a total of 21 features extracted from the log file data, reducing the number of features is helpful for improving the model performance, reducing the computation time, and making the prediction more interpretable (Chandrashekar & Sahin, 2014). In addition, our approach uses students' daily click frequencies rather than other LMS features to represent their behaviours. Using feature selection to identify those most influential is thus helpful in validating the choice of click frequencies for our approach. If the most influential features identified by machine learning classifiers are frequency-related, it is more convincing that LMS use patterns over the semester provide an implicit indicator of final course performance.

In this study, the well-known feature selection algorithm, Recursive Feature Elimination (RFE), was used to remove the weakest features. Specifically, the RFE algorithm first uses all predictors to fit the model on the training dataset and calculate the model performance. Each feature is then ranked by the feature importance. Thereafter, for a total of $n$ features, we can define a value of $k$ indicating the feature subset size ($k = 1, \ldots n$). The corresponding top $k$ features are used to fit the model and calculate the model performance. As such, we can find the optimal number of features by comparing the model performance with different feature subset sizes. In this study, AUC is used as the evaluation metric for RFE. In RFE, a machine learning model should be specified for tuning the model with different feature subset sizes. For a particular machine learning classifier, the model can be tuned and finalized directly with RFE, which can be in turn evaluated by the test dataset. To remove the confounding effect of different features selected by different classifiers, we used a particular machine learning model rather than different models in RFE. Among the eight models, we used RFE with random forest for feature selection due to the following considerations. Random forest is an extremely popular and powerful ensemble model, which is capable of reducing the variance in predictions while maintaining low bias. Moreover, the bootstrapping sampling technique used by random forest is a desirable feature for the small dataset used in our study. It is also very friendly with high-dimensional data given that it uses a subset of features in training. In addition, it is robust to outliers and non-linear data, and works well with imbalanced data.

Although the optimal feature subset size with the highest prediction accuracy was much larger than five, using more features does not improve the model significantly given the negligible difference in prediction accuracy. Therefore, we only selected the top five features of the optimal feature subset. Furthermore, in addition to RFE with random forest, the other seven machine learning models with all features also provided a ranking of feature importance. Therefore, aside from the top five features selected by RFE, we also selected two additional top features with high feature importance identified by most other models, resulting in a total of seven selected features for machine learning.

#### 4.4.4. Hyperparameter Tuning and Model Evaluation

Consistent with most previous studies (e.g., Casey & Azcona, 2017; Hu et al., 2014; Romero et al., 2013), we used the 10-fold cross-validation to tune the hyperparameters and evaluate conventional machine-learning classifiers. The 10-fold cross-validation is a resampling procedure for evaluating models on unseen data, which is widely used for limited data samples. In 10-fold cross-validation, the dataset is first shuffled randomly and then split into 10 folds. Each fold in turn is taken as the test dataset with the remaining nine folds used for training. A model can then be trained by the training dataset and evaluated on the test dataset. This process repeats 10 times and the model performance can be summarized by using the 10 evaluation scores obtained. This 10-fold cross-validation is popular because it minimizes both the testing bias and the variance of small datasets. Compared with the hold-out method (e.g., holding out half the data for testing and training on the rest), a 10-fold cross-validation uses almost all available samples for training, which greatly reduces bias. As well, the test performance is summarized across all samples, again reducing variance (i.e., with performance stable across different test datasets). In our study, each fold included 41 samples, meaning that 366 samples were used for training in each cross-validation iteration. To enhance the reliability of model evaluation, our 10-fold cross-validation was repeated five times.

Regarding hyperparameter tuning for the eight machine-learning classifiers, we evaluated each hypothesized hyperparameter value using 10-fold cross-validation. The final values were determined by the model with the highest score on AUC. In terms of candidate hyperparameter values, although grid search and manual search are widely used, we used the random search for optimization. A random search is more efficient than a grid or manual search, especially when many hyperparameters must be tuned, because it is capable of finding models that are just as good with much less computation time (Bergstra & Bengio, 2012). For example, we used a single hidden layer neural network with L2 regularization for the algorithm considering two hyperparameters: size (the number of nodes in the hidden layer) and decay (the regularization weight). For each combination of randomly selected values for these two hyperparameters, the 10-fold cross-validation was applied to select the optimal size and decay using the largest score on AUC. We specified 10 combinations of random hyperparameters for each model.

## 5. Results

### 5.1. LSTM Network Results

Figure 3 plots the trend of AUC scores of each LSTM network at each experimental repeat. Generally, all LSTM networks were estimated stably, given the low standard deviations shown by each network (see Table 3). Specifically, all networks showed an average validation AUC above 70%, with the highest for the first-56-days network (75.2%) and the lowest for the first-28-days network (71.3%) (see Table 3). For the generalizability of each network, the semester 1 test AUC scores were close to those for semester 2 for each LSTM model. Notably, the average AUC score achieved 68.2% when the click events of the first 42 days were used for modelling.

Despite the fact that the 70-days network included almost all the time series information, it did not provide optimal predictive performance in terms of either validation or test AUC. However, the discrepancies in AUC scores between models were not substantial. Considering the demand for early prediction of course performance, the above results suggest that student clicks on Moodle during the early weeks could be successfully used for the timely prediction of their final course performance.

**Table 3.** Average AUC (%) and Standard Deviation of Each LSTM Network for the Education Course

| *M* (*SD*) | First 28 days | 42 days | 56 days | 70 days |
|---|---|---|---|---|
| Validation AUC | 71.3 (2.15) | 73.4 (0.59) | 75.2 (0.97) | 73.8 (0.44) |
| Test: Semester 1 | 59.6 (1.27) | 68.2 (1.45) | 61.1 (3.02) | 65.6 (1.47) |
| Test: Semester 2 | 61.6 (1.06) | 68.2 (1.82) | 64.4 (2.79) | 65.0 (2.74) |

### 5.2. Machine Learning Results

Subsequently, we utilized the extracted features to predict student course performance with the eight commonly used machine-learning classifiers. The performance by AUC for each classifier with all features was approximately 60% on average (see Table 4). Specifically, logistic regression and random forest showed the highest (62.8%) and lowest (57.9%) validation AUC scores, respectively. However, the generalizability performance of each classifier was lower.

Aside from modelling with all features, we were also interested in which LMS predictors were more influential for student course performance. To this end, using the feature selection method mentioned above, the following seven features were selected as the most important for further predictions:

- Number of total clicks
- Number of clicks during weekdays
- Number of clicks for module “File”
- Number of clicks for module “System”
- Number of clicks for module “Forum”
- Number of online sessions
- Number of clicks on campus

**Table 4.** AUC (%) of Each Classifier Using All and Selected Features for the Education Course

| Data | Features | NN | LR | NB | GBM | SVM | DT | kNN | RF |
|---|---|---|---|---|---|---|---|---|---|
| Validation | All | 58.4 | 62.8 | 61.7 | 59.6 | 59.2 | 58.0 | 58.1 | 57.9 |
| | Selected | 63.6 | 64.2 | 66.2 | 64.2 | 60.4 | 57.7 | 59.3 | 59.4 |
| Test: Semester 1 | All | 57.9 | 53.5 | 50.3 | 55.7 | 64.6 | 56.8 | 60.6 | 59.5 |
| | Selected | 51.3 | 54.3 | 54.8 | 58.3 | 60.4 | 51.2 | 56.6 | 57.8 |
| Test: Semester 2 | All | 64.0 | 56.6 | 52.7 | 53.5 | 60.5 | 58.7 | 63.1 | 56.8 |
| | Selected | 52.9 | 60.0 | 63.2 | 58.8 | 54.9 | 56.3 | 62.2 | 52.0 |

Using fewer features, all classifiers demonstrated higher validation performance (see Table 4). The best validation performance and the best test performance on the dataset from semester 2 were found for naïve Bayes (66.2%). In addition, some classifiers increased while some decreased in generalizability regarding the change in AUC scores. Generally, support vector machine, gradient boosting machine, and k-nearest neighbours showed relatively better performance than other classifiers with respect to both validation and test AUC scores.

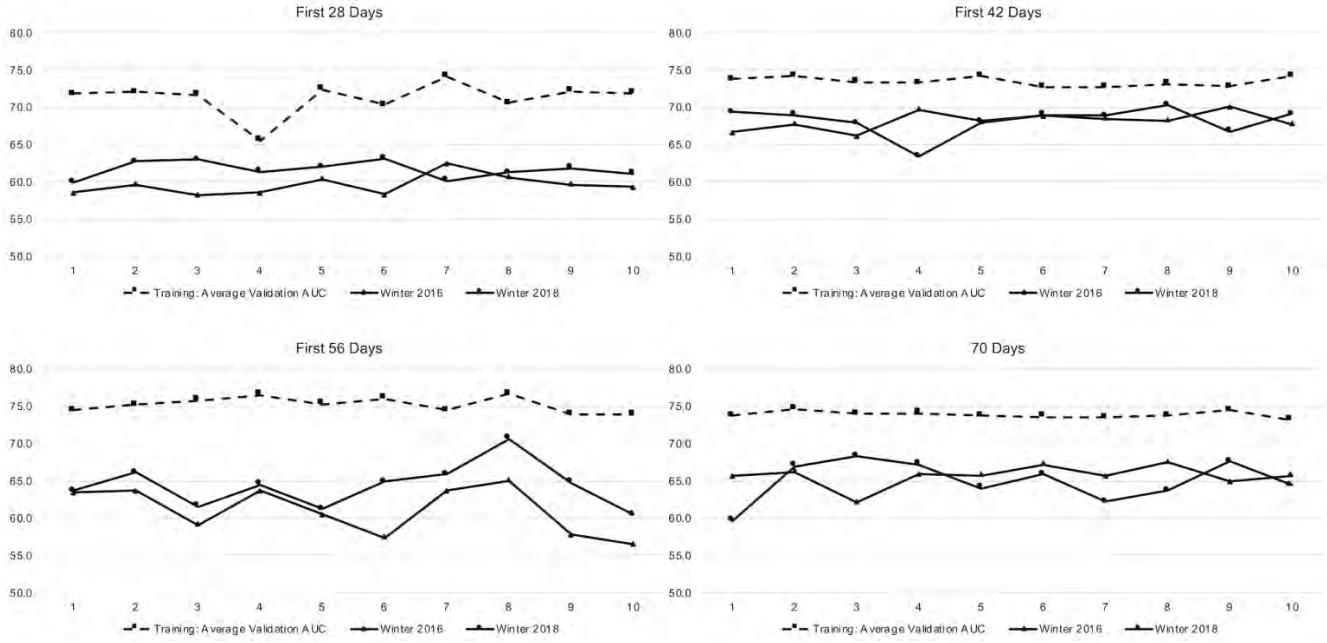


**Figure 3.** Frequency trend of student online actions on Moodle for the education course.
*Note:* the x-axis indicates the number of repeats.

## 6. A Follow-up Study for Generalizability

The above results indicate the potential of our approach for building a predictive model of student course performance based on their behaviours in the LMS. This model can perform well for different types of courses given that it utilizes daily click behaviour. A follow-up analysis with a new dataset further demonstrates the generalizability of our approach.

### 6.1. Dataset Description

The dataset used for the follow-up analysis was from a mandatory introductory biology course for undergraduates. The course is offered every fall and winter semester. We used the course data from two semesters for analysis. The data of semester 1 was partitioned into a training dataset, a validation dataset, and a test dataset; the data of semester 2 was only used for testing. There were 290 students in semester 1 and 311 students in semester 2. The analytical procedures for the biology course were exactly the same as those for the education course mentioned above.

### 6.2. Results

Table 5 presents the average AUC rates for each LSTM network with respect to different lengths of data for training. In general, all LSTM networks were estimated stably, given their low standard deviations. All networks showed a validation AUC above or close to 80% with the highest for the 70-days network (83%) and the lowest for the first-42-days network (76.9%). For the generalizability of each network, the test AUC scores for semester 1 were around 75%, which was higher than the scores for semester 2. This result is not surprising given that the training dataset was also from the course in semester 1 and the dataset of semester 2 might differ in content and design features.

**Table 5.** Average AUC (%) and Standard Deviation of Each LSTM Network for the Biology Course

| *M* (*SD*) | First 28 days | 42 days | 56 days | 70 days |
|---|---|---|---|---|
| Validation AUC | 80.1 (0.00) | 76.9 (0.02) | 82.6 (0.01) | 83.0 (0.01) |
| Test: Semester 1 | 74.6 (0.01) | 78.7 (0.01) | 73.4 (0.01) | 78.2 (0.01) |
| Test: Semester 2 | 63.1 (0.01) | 63.5 (0.01) | 65.7 (0.00) | 64.5 (0.00) |

Similar to the results for the education course, the discrepancies in AUC scores between models with different time frames were not substantial. Particularly, the training model using the first-28-days data was as competitive as those with more data, indicating its potential for early detection of at-risk students.

Table 6 presents the AUC rates for each machine learning classifier for the biology course. Most showed a testing AUC rate below 60% and performed worse on the test dataset of semester 2. Compared with the results of LSTM networks, it is evident that the machine learning classifiers were not advantageous over our approach in terms of predictive capacity.

**Table 6.** AUC (%) of Each Classifier Using All and Selected Features for the Biology Course

| Data | Features | NN | LR | NB | GBM | SVM | DT | kNN | RF |
|---|---|---|---|---|---|---|---|---|---|
| Validation | All | 61.0 | 55.2 | 61.0 | 58.3 | 60.4 | 60.9 | 59.4 | 56.0 |
| | Selected | 63.0 | 62.2 | 67.3 | 63.5 | 66.2 | 62.3 | 63.1 | 67.0 |
| Test: Semester 1 | All | 55.7 | 53.2 | 51.9 | 56.0 | 51.1 | 59.6 | 55.7 | 54.4 |
| | Selected | 59.3 | 56.7 | 50.2 | 60.0 | 51.4 | 57.4 | 55.3 | 60.3 |
| Test: Semester 2 | All | 50.8 | 54.2 | 55.2 | 59.9 | 50.0 | 54.7 | 52.1 | 57.1 |
| | Selected | 54.4 | 54.6 | 50.4 | 60.9 | 53.2 | 51.5 | 52.1 | 54.5 |

## 7. Discussion

In this study, we examine the potential of a deep learning approach — LSTM networks — in predictive analytics of LMS log data. In addition, given the limited research comparing deep learning with conventional machine-learning classifiers for predictive analytics (e.g., Jiang et al., 2018; Le, Pardos, Meyer, & Thorp, 2018), our study provides further evidence for the potential of deep learning to predict learning outcomes. Our results suggest that deep learning could be successful in predicting course performance using daily click frequencies in the LMS. Generally, the LSTM networks demonstrated better prediction performance than the machine learning classifiers.

The performance of the eight machine learning classifiers used as baselines for our LSTM approach can be discussed separately for using all features versus using selected features. When all features were used, as expected, the AUC scores on the test dataset (i.e., semester 2) were close to their validation AUC scores for most classifiers. Some classifiers, such as naïve Bayes, however, showed much lower AUC scores on the test dataset than on the validation dataset. Regarding generalizability, most classifiers showed poor test performance on the test dataset from the external course (i.e., semester 1) in terms of AUC scores, but the performance of NN, SVM, kNN, and RF were relatively better than the others. When the selected top features were used, all models improved in terms of validation AUC scores. This finding echoes the idea that using *the right* features may be more important than using *more* features since feature selection is of great importance for machine learning (Hall & Smith, 1998). From the perspective of bias–variance trade-offs, using more features may reduce the bias in training but also increases the variance. In other words, more features lead to better model fit to the training data, represented by lower training errors. However, when unseen data are used for testing, models often do not perform well because they are built on the features important for training but not for testing.

Another key finding is that the most influential features for machine learning are all frequency-related, suggesting that time-related features may not be very useful for prediction. Unlike in-class learning activities, we cannot directly observe how much time students actually spend in the LMS. Students might head off in search of coffee, leaving the system open. Other researchers have pointed out that time-related LMS features should be treated with caution (e.g., Casey & Azcona, 2017). All features related to online sessions were also found to be of limited use in this study. Similarly, we cannot directly observe what students actually do in each online session. Therefore, an online session might not be a real collection of consecutive events in the LMS. The importance of click frequency for prediction has been observed in other studies as well. For example, in Amrieh, Hamtini, and Aljarah's (2016) study, three of the top four indicators were frequency-related LMS features, with number of course resources visited ranked as the most important. This finding also supports using daily click frequencies rather than daily time spent on the LMS for time series modelling.

Regarding the eight machine learning classifiers, predictive models based on aggregated features might be of limited predictive capacity for different types of courses. Given the results from the biology course, the testing AUC scores for most machine learning classifiers were also not satisfactory. Particularly, test performance did not differ much between the two datasets, indicating that the aggregated features extracted from log files for the biology course were not good indicators of course performance. Furthermore, the best-performing classifiers were different in the two courses, suggesting that predictive models should be customized for different types of courses.

In general, the LSTM networks used in the education course showed slightly higher test AUC scores than the best-performing machine learning classifiers (i.e., SVM and kNN). Specifically, the first-28-days network demonstrated a similar test performance compared with the best-performing machine learning classifier. The advantages of LSTM networks were much more compelling for the biology course. Specifically, the test AUC scores for each were much higher than all the machine learning classifiers, irrespective of test datasets. Notably, the first-28-days data showed an AUC rate around 75% on the test

dataset from the same semester as the training dataset. These results are promising given that, in contrast to machine learning classifiers using multiple aggregated LMS features, LSTM networks modelled a relatively simpler feature — daily click frequencies— which required much less feature engineering. Generally, models using data over a longer period had better predictive performance than the first-28-days model, but the test performance did not consistently increase as number of weeks increased.

Often overlooked in previous studies, we examined the generalizability evaluation of predictive models. Our approach was evaluated by both the test performance on a dataset from a different semester and the comparison between the two types of courses. On average, in contrast with the chance level of AUC (i.e., 0.5), our test AUC scores were more than 20% above for both the education and biology courses. However, most LSTM networks showed an absolute AUC score above 60% for the different semester, which could be considered as moderate generalizability. Since the use of LSTM networks might be restricted by sample size, when training on a small dataset, a discrepancy in model performance is expected between training and testing. In terms of comparing training and test performance, our approach performed well for both courses, with a slightly higher test performance for the biology course, further validating the generalizability of our approach.

Compared with the majority of machine learning classifiers, our approach demonstrates stronger generalizability in terms of higher test AUC scores for the different semesters and courses. When all LMS log features were used, most classifiers showed a validation or test AUC score below 60%, consistent with previous studies (e.g., Romero et al., 2013). A possible explanation is that the machine learning classifiers used multiple LMS log features for prediction, which introduced more variance into the model, while our approach only used daily click frequencies in the LMS — a finding of great practical importance.

One major goal of the institutional application of predictive analytics is to build a flexible model that performs well for a wide range of courses. This can be challenging since different courses have distinct structures, teaching and learning activities, and requirements. Deriving a set of common aggregated features, which in turn limits the generalizability of conventional machine learning classifiers, can be difficult. However, using student time series data might provide a solution, as shown by our findings. Some previous studies demonstrated the potential of learners' time series data in a variety of contexts. The 2018 special issue of the *Journal of Learning Analytics* on temporal analyses of learning data featured the results of several studies along these lines:

1. Learners' temporal sequences of group talk in solving an algebra problem were analyzed with regression analyses to predict their group learning outcomes (Chiu, 2018)
2. Student discourse transcripts were analyzed over time with a combination of socio-semantic network analysis and dialogical discourse analysis to characterize and interpret their group discourse and collaboration practises (Oshima, Oshima, & Fujita, 2018)
3. Two novel measures were proposed to identify learners' timing behaviours in a self-paced online course and examine the time effects on learners' post-course self-efficacy (Riel, Lawless, & Brown, 2018)
4. A generalizable multi-step approach was developed to analyze multimodal data to interpret learning trajectories in intelligent tutoring systems (Liu, Stamper, & Davenport, 2018)
5. A sequence data model was proposed to analyze student learning records and LMS activities for both a within-semester prediction of final course grades and a between-semester prediction of program completion (Mahzoon, Maher, Eltayeby, Dou, & Grace, 2018)

Despite these promising uses of time-series data, differences in data types, analytical approaches, and application contexts mean that generalizable intervention pathways for learners are unlikely to be developed. Temporal analyses of learning data applicable in a wide range of contexts, however, are still worth further exploration.

The simplicity of our approach would be beneficial for building predictive models for educational institutions in practice. Typically, feature engineering is a burdensome and challenging task in conventional machine learning applications. The features used in previous models can help to build new learning analytics models (Berland, Baker, & Blikstein, 2014); however, extracted features that work well in one scenario might be ineffective in another. For a course in education requiring extensive online readings, for example, the frequency of visits to class reading materials might be very indicative of final course performance, while the same feature might be useless for predicting performance in a computer science course that emphasizes programming skills.

With conventional machine learning approaches, model designers and developers must collaborate with course instructors to identify unique LMS features tailored to each course, which is unlikely feasible in practice. In this sense, given the importance of generalizability and simplicity, it is desirable to use simple, generalizable features in learning analytics applications. Our results suggest that daily click frequency demonstrates better predictive performance than the combination of other aggregated features, and works well for two very different types of courses, indicating good potential in practice. Although our approach is simpler and more generalizable, there are still many technical considerations in designing an LSTM

framework for building a predictive analytics model. For example, model developers still need to obtain adequate data for training, tuning hyperparameters, and deciding the optimal time to send warnings to students.

Compared to a set of aggregated features, daily LMS usage is relatively simple for instructors to understand and use. Course instructors can simply monitor student usage to flag at-risk students, review their other course activities and assignment performance, and send them timely warnings through the LMS. This is especially beneficial for large classes (e.g., first-year introductory courses) because instructors are less able to pay close attention to every student while frequent in-process evaluations of student performance would be costly. In addition, the model built in one school year can be refined and used repeatedly in future school years, which reduces the development cost. In general, our approach is not meant to replace other evaluation and intervention systems in higher education. Rather it serves as an effective screening system for flagging at-risk students regardless of course type, which improves efficiency and saves the cost of in-process course evaluations.

In terms of limitations, our sample sizes were quite small for predictive modelling, which possibly undermines the power of the deep learning approach. Future studies analyzing data from courses with more students are encouraged to validate our findings. Another concern relates to the class imbalance for which we used the SMOTE oversampling approach to compensate. Despite this, oversampling still introduced bias in training given that many synthetic samples were generated. In addition, most machine learning models require that the training and test data come from the same feature and target distributions. In our analysis, the test datasets were largely imbalanced. As such, the difference in target class distribution between training and test data also undermined the model generalizability. Future studies are needed to examine how to better address class imbalance for LSTM analysis.

## Declaration of Conflicting Interest

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## Funding

The publication of this article received financial support from the University of Alberta Teaching and Learning Research Fund (RES0035131).

## References

Al-Shabandar, R., Hussain, A., Laws, A., Keight, R., Lunn, J., & Radi, N. (2017). Machine learning approaches to predict learning outcomes in massive open online courses. *Proceedings of the 2017 International Joint Conference on Neural Networks* (IJCNN 2017), 14–19 May 2017, Anchorage, Alaska, USA (pp. 713–720). Washington, DC: IEEE Computer Society. https://dx.doi.org/10.1109/IJCNN.2017.7965922922

Amrieh, E. A., Hamtini, T., & Aljarah, I. (2016). Mining educational data to predict student's academic performance using ensemble methods. *International Journal of Database Theory and Application*, *9*(8), 119–136. https://dx.doi.org/10.14257/ijdta.2016.9.8.13

Arnold, K. E., & Pistilli, M. D. (2012). Course Signals at Purdue: Using learning analytics to increase student success. In S. Buckingham Shum, D. Gašević, & R. Ferguson (Eds.), *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge* (LAK '12), 29 April–2 May 2012, Vancouver, BC, Canada (pp. 267–270). New York: ACM. https://dx.doi.org/10.1145/2330601.2330666

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166. https://dx.doi.org/10.1109/72.279181

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*(10), 281–305. https://dl.acm.org/doi/10.5555/2188385.2188395

Berland, M., Baker, R. S., & Blikstein, P. (2014). Educational data mining and learning analytics: Applications to constructionist research. *Technology, Knowledge and Learning*, *19*, 205–220. https://dx.doi.org/10.1007/s10758-014-9223-7

Casey, K., & Azcona, D. (2017). Utilizing student activity patterns to predict performance. *International Journal of Educational Technology in Higher Education*, *14*, 4. https://dx.doi.org/10.1186/s41239-017-0044-3

Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering, 40*(1), 16–28. https://dx.doi.org/10.1016/j.compeleceng.2013.11.024

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://dx.doi.org/10.1613/jair.953

Chiu, M. M. (2018). Statistically modelling effects of dynamic processes on outcomes: An example of discourse sequences and group solutions. *Journal of Learning Analytics*, *5*(1), 75–91. https://doi.org/10.18608/jla.2018.51.6

Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (2016). Doctor AI: Predicting clinical events via recurrent neural networks. *Proceedings of Machine Learning Research, vol. 56: Machine Learning for Healthcare Conference* (MLHC 2016), 19–20 August 2016, Los Angeles, CA, USA (pp. 301–318). Retreived from https://proceedings.mlr.press/v56/Choi16.pdf

Chollet, F. (2015). *keras*. GitHub repository. Retrieved from http://github.com/fchollet/keras

Coelho, O. B., & Silveira, I. (2017). Deep learning applied to learning analytics and educational data mining: A systematic literature review. *Proceedings of the Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 28, No. 1, p. 143). http://dx.doi.org/10.5753/cbie.sbie.2017.143

Cole, J., & Foster, H. (2007). *Using Moodle: Teaching with the popular open source course management system.* Sebastopol CA: O'Reilly Media.

Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017). Predicting student performance from LMS data: A comparison of 17 blended courses using Moodle LMS. *IEEE Transactions on Learning Technologies*, *10*(1), 17–29. http://dx.doi.org/10.1109/TLT.2016.2616312

Cui, Y., Chen, F., Shiri, A., & Fan, Y. (2019). Predictive analytic models of student success in higher education: A review of methodology. *Information and Learning Sciences, 120*(3/4), 208–227. http://dx.doi.org/10.1108/ILS-10-2018-0104

Daniel, B. (2015). Big data and analytics in higher education: Opportunities and challenges. *British Journal of Educational Technology*, *46*(5), 904–920. http://dx.doi.org/10.1111/bjet.12230

Duffy, T. M., & Cunningham, D. J. (1996). Constructivism: Implications for the design and delivery of instruction. In D. Jonassen (Ed.), *Handbook of research for educational communications and technology*. New York: Macmillan.

Edwards, D., & Mercer, N. (2013). *Common knowledge: The development of understanding in the classroom*. London: Routledge.

Evale, D. (2016). Learning management system with prediction model and course-content recommendation module. *Journal of Information Technology Education: Research, 16*, 437–457. http://dx.doi.org/10.28945/3883

Gal, Y., & Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, & R. Garnett (Eds.), *Proceedings of the 30th International Conference on Neural Information Processing Systems* (pp. 1027–1035). Red Hook, NY: Curran Associates Inc. http://papers.nips.cc/paper/6241-a-theoreticallygrounded-application-of-dropout-in-recurrent-neural-networks.pdf

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks, 18*(5–6), 602–610. http://dx.doi.org/10.1016/j.neunet.2005.06.042

Guarín, C. E. L., Guzmán, E. L., & González, F. A. (2015). A model to predict low academic performance at a specific enrollment using data mining. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, *10*(3), 119–125. http://dx.doi.org/10.1109/RITA.2015.2452632

Hall, M. A., & Smith, L. A. (1998). Practical feature subset selection for machine learning. In C. McDonald (Ed.), *Computer Science '98: Proceedings of the 21st Australasian Computer Science Conference* (ACSC'98), 4–6 February 1998, Perth, WA, Australia (pp. 181–191). Berlin: Springer. Retrieved from http://hdl.handle.net/10289/1512

Hein, G. E. (1991). *Constructivist learning theory*. Paper presented at The Museum and the Needs of People: International Committee of Museum Educators Conference (CECA), 15–22 October 1991, Jerusalem, Israel. Retreived from http://www.exploratorium.edu/IFI/resources/constructivistlearning.html

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735

Hu, Y. H., Lo, C. L., & Shih, S. P. (2014). Developing early warning systems to predict students' online learning performance. *Computers in Human Behavior*, *36*, 469–478. http://dx.doi.org/10.1016/j.chb.2014.04.002

Jiang, Y., Bosch, N., Baker, R., Paquette, L., Ocumpaugh, J., Andres, J. M. A. L., Moore, A. L., & Biswas, G. (2018). Expert feature-engineering vs. deep neural networks: Which is better for sensor-free affect detection? In C. P. Rosé et al. (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence in Education* (AIED 2018), 27–30 June 2018, London, UK (pp. 198–211). *Lecture Notes in Computer Science*, vol. 10947. Springer. http://dx.doi.org/10.1007/978-3-319-93843-1_15

Kim, B. H., Vizitei, E., & Ganapathi, V. (2018). GritNet: Student performance prediction with deep learning. arXiv preprint. Retreived from http://arxiv.org/abs/1804.07405

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint. Retreived from http://arxiv.org/abs/1412.6980

Kuhn, M. (2008). Building predictive models in R using the caret package. *Journal of Statistical Software, 28*(5), 1–26. http://dx.doi.org/10.18637/jss.v028.i05

Le, C. V., Pardos, Z. A., Meyer, S. D., & Thorp, R. (2018). Communication at scale in a MOOC using predictive engagement analytics. In C. P. Rosé et al. (Eds.), *Proceedings of the 19th International Conference on Artificial Intelligence in Education* (AIED 2018), 27–30 June 2018, London, UK (pp. 239–252). *Lecture Notes in Computer Science*, vol. 10947. Springer. http://dx.doi.org/10.1007/978-3-319-93843-1_18

Ling, C. X., Huang, J., & Zhang, H. (2003). AUC: A better measure than accuracy in comparing learning algorithms. In Y. Xiang & B. Chaib-draa (Eds.), *Advances in Artificial Intelligence: Proceedings of the 16th Conference of the Canadian Society for Computational Studies of Intelligence* (Canadian AI'03), 11–13 June 2003, Halifax, NS, Canada (pp. 329–341). *Lecture Notes in Computer Science*, vol. 2671. Springer. http://dx.doi.org/10.1007/3-540-44886-1_25

Lippmann, R. P. (1987). An introduction to computing with neural nets. *IEEE ASSP Magazine*, *4*(2), 4–22. http://dx.doi.org/10.1109/MASSP.1987.1165576

Liu, R., Stamper, J. C., & Davenport, J. (2018). A novel method for the in-depth multimodal analysis of student learning trajectories in intelligent tutoring systems. *Journal of Learning Analytics*, *5*(1), 41–54. https://doi.org/10.18608/jla.2018.51.4

Long, P., Siemens, G., Conole, G., & Gašević, D. (Eds.). (2011). *Proceedings of the 1st International Conference on Learning Analytics and Knowledge* (LAK '11), 27 February–1 March 2011, Banff, AB, Canada. New York: ACM.

Luo, J., Sorour, S. E., Goda, K., & Mine, T. (2015). Predicting student grade based on free-style comments using Word2Vec and ANN by considering prediction results obtained in consecutive lessons. In O. C. Santos et al. (Eds.), *Proceedings of the 8th International Conference on Educational Data Mining* (EDM2015), 26–29 June 2015, Madrid, Spain (pp. 396–399). International Educational Data Mining Society.

Luo, L., Koprinska, I., & Liu, W. (2015). Discrimination-aware classifiers for student performance prediction. In O. C. Santos et al. (Eds.), *Proceedings of the 8th International Conference on Educational Data Mining* (EDM2015), 26–29 June 2015, Madrid, Spain (pp. 384–387). International Educational Data Mining Society.

Macfadyen, L. P., & Dawson, S. (2010). Mining LMS data to develop an "early warning system" for educators: A proof of concept. *Computers & Education, 54*(2), 588–599. http://dx.doi.org/10.1016/j.compedu.2009.09.008

Mahzoon, M. J., Maher, M. L., Eltayeby, O., Dou, W., & Grace, K. (2018). A sequence data model for analyzing temporal patterns of student data. *Journal of Learning Analytics*, *5*(1), 55–74. https://doi.org/10.18608/jla.2018.51.5

Mandrekar, J. N. (2010). Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*, *5*(9), 1315–1316. http://dx.doi.org/10.1097/JTO.0b013e3181ec173d

Mayer, H., Gomez, F., Wierstra, D., Nagy, I., Knoll, A., & Schmidhuber, J. (2008). A system for robotic heart surgery that learns to tie knots using recurrent neural networks. *Advanced Robotics, 22*(13–14), 1521–1537. http://dx.doi.org/10.1163/156855308X360604

Meedech, P., Iam-On, N., & Boongoen, T. (2016). Prediction of student dropout using personal profile and data mining approach. In K. Lavangnananda, S. Phon-Amnuaisuk, W. Engchuan, & J. Chan (Eds.), *Intelligent and Evolutionary Systems: Proceedings of the 19th Asia Pacific Symposium on Intelligent and Evolutionary Systems* (IES 2015), 22–25 November 2015, Bangkok, Thailand. *Proceedings in Adaptation, Learning and Optimization*, vol. 5 (pp. 143–155). Springer. http://dx.doi.org/10.1007/978-3-319-27000-5_12

Milne, J., Jeffrey, L. M., Suddaby, G., & Higgins, A. (2012). Early identification of students at risk of failing. In M. Brown, M. Hartnett, & T. Stewart (Eds.), *Proceedings of the 29th Annual Conference of the Australasian Society for Computers in Learning in Tertiary Education* (ASCILITE 2012), 25–28 November 2012, Wellington, New Zealand (pp. 657–661). Australasian Society for Computers in Learning in Tertiary Education. Retrieved from https://www.learntechlib.org/p/42660/

Ochoa, X. (2016). Adaptive multilevel clustering model for the prediction of academic risk. *Proceedings of the 11th Latin American Conference on Learning Objects and Technology* (LACLO 2016), 3–7 October 2016, San Carlos, Costa Rica (pp. 1–8). IEEE. http://dx.doi.org/10.1109/LACLO.2016.7751800

Okubo, F., Yamashita, T., Shimada, A., & Konomi, S. (2017). Students' performance prediction using data of multiple courses by recurrent neural network. In W. Chen et al. (Eds.), *Proceedings of the 25th International Conference on Computers in Education* (ICCE 2017), 4–8 December 2017, Christchurch, New Zealand (pp. 439–444). Asia-Pacific Society for Computers in Education.

Okubo, F., Yamashita, T., Shimada, A., & Ogata, H. (2017). A neural network approach for students' performance prediction. *Proceedings of the 7th International Conference on Learning Analytics and Knowledge* (LAK '17), 13–17 March 2017, Vancouver, BC, Canada (pp. 598–599). New York: ACM. http://dx.doi.org/10.1145/3027385.3029479

Oshima, J., Oshima, R., & Fujita, W. (2018). A mixed-methods approach to analyze shared epistemic agency in jigsaw instruction at multiple scales of temporality. *Journal of Learning Analytics*, *5*(1), 10–24. https://doi.org/10.18608/jla.2018.51.2

R Core Team (2018). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. Retreived from https://www.r-project.org/

Rice, M. E., & Harris, G. T. (2005). Comparing effect sizes in follow-up studies: ROC area, Cohen's D, and R. *Law and Human Behavior*, *29*(5), 615–620. https://dx.doi.org/10.1007/s10979-005-6832-7

Riel, J., Lawless, K. A., & Brown, S. W. (2018). Timing matters: Approaches for measuring and visualizing behaviours of timing and spacing of work in self-paced online teacher professional development courses. *Journal of Learning Analytics*, *5*(1), 25–40. https://doi.org/10.18608/jla.2018.51.3

Romero, C., Espejo, P. G., Zafra, A., Romero, J. R., & Ventura, S. (2013). Web usage mining for predicting final marks of students that use Moodle courses. *Computer Applications in Engineering Education*, *21*(1), 135–146. https://dx.doi.org/10.1002/cae.20456

Schell, J., Lukoff, B., & Alvarado, C. (2014). Using early warning signs to predict academic risk in interactive, blended teaching environments. *Internet Learning*, *3*(2), 6. Retreived from https://pdfs.semanticscholar.org/b064/c434033ebc2240782886bd10c1a3813ca809.pdf

Sclater, N., Peasgood, A., & Mullan, J. (2016). *Learning analytics in higher education*. Bristol, UK: JISC. Retreived from https://www.jisc.ac.uk/sites/default/files/learning-analytics-in-he-v3.pdf

Shahiri, A. M., & Husain, W. (2015). A review on predicting student's performance using data mining techniques. *Procedia Computer Science*, *72*, 414–422. https://dx.doi.org/10.1016/j.procs.2015.12.157

Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education. *EDUCAUSE Review, 46*(5), 30–40. Retreived from https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education

Winne, P. H., & Hadwin, A. F. (1998). Studying as self-regulated learning. In D. J. Hacker, J. Dunlosky, & A. C. Graesser (Eds.), *Metacognition in educational theory and practice* (pp. 277–304). Mahwah, NJ: Lawrence Erlbaum Associates.

Zhou, Y., Huang, C., Hu, Q., Zhu, J., & Tang, Y. (2018). Personalized learning full-path recommendation model based on LSTM neural networks. *Information Sciences, 444*, 135–152. https://dx.doi.org/10.1016/j.ins.2018.02.053