C.M.P. Theunissen
I6090664

Maastricht University

Predicting Student Performance: A Deep Learning Approach

C.M.P. Theunissen
Business Intelligence and Smart Services
Maastricht University

January 2019

MSc Thesis

Abstract – Even though predicting student performance is a hot topic in educational data mining, little has been written on the application of state-of-the art deep learning techniques in this context. This paper compares the performance of long-short term memory (LSTM) networks, an advanced type of recurrent neural network, to that of traditional machine learning methods when it comes to predicting academic performance. The analysis is based on student data originating from the UK's Open University, which is one of the largest distance learning institutions in Europe. The number of submitted assessments per day, the students' average assessment score and clickstream data from the virtual learning environment are used as input data for the models. In contrast to traditional classifiers, LSTMs are capable of accounting for the time dimension in sequential course data. Therefore, it is hypothesized that LSTMs can predict student performance better than traditional classifiers. The results demonstrate that the performance of LSTMs is competitive and more consistent across three distinct courses than that of the popular conventional models. The revealed potential of LSTMs suggests that deep learning has merit in an educational data mining context and warrants further research.

Table of contents

List of figures

List of tables

1. Introduction

The last decade has seen a tremendous growth in volume, variety and velocity of data generated by humankind. Many organizations in numerous different areas have capitalized on this phenomenon by extracting valuable knowledge from data through processing and analysis. One such area is the field of educational data mining (EDM). EDM concerns the application and development of computerized methods to analyze the unique kinds of data found in educational settings to help study educational research questions (Baker, 2010). Romero and Ventura (2010) conducted a comprehensive literature review in which they identified the task of predicting student performance to be one of the most popular research lines in the field of EDM. In essence, this research line is all about utilizing algorithms that can automatically identify patterns in the input data such as past level of education or test scores obtained allowing them to predict how well students will perform.

Being able to accurately predict students' academic performance can be beneficial for students, teachers and educational institutions alike. Systems equipped with the ability to predict student performance early on can warn students at risk before it is too late to change their behavior, while prospective exceptional students can be offered additional stimulating material. Moreover, the systems can provide teachers with additional insights on which particular students need extra assistance. Furthermore, academic institutions could potentially design more effective admission policies as a result of these type of systems. Researchers have already conducted pilot tests with such intervention solutions at Purdue University in Indiana (Arnold & Pistilli, 2012). Their system called Course Signals incorporates predictive models to provide students with real-time feedback on their chance of success in their respective courses. According to their research, the Course Signals system does not only substantially increase the number of satisfactory grades students obtain, but it also manages to decrease the amount of unsatisfactory grades and student withdrawals. They also found that the earlier a student encounters Course Signals, the greater the positive effects are. Moreover, teachers reported that it provides them with better insights into which students require help. Jayaprakash, Moody, Lauría, Regan & Baron (2014) build upon this success and similarly found that such preemptive intervention strategies can positively impact student learning outcomes. Furthermore, they suggest that predictive models may be scalable to other institutions with differing characteristics.

Accurate algorithms that can predict academic performance as early as possible are necessary before it is even possible to reap these kind of benefits. Traditional machine learning models are most commonly used for this purpose. These models include but are not limited to decision trees, logistic regression models, naive Bayes classifiers, k-nearest neighbor classifiers (k-nn), random forest models and artificial neural networks (ANNs) in particular the multi-layered perceptron (MLP). These tried and tested methods are able to make predictions with great accuracy depending on the application and the corresponding data. They all make use of supervised learning to perform their classification task, which means that they learn through examples with a labeled target variable. For example, when one has a dataset with student information including a target variable indicating whether the students passed or failed the course, one can used supervised learning algorithms to find a pattern between the independent variables and the target variable. One major shortcoming of these traditional classifiers mentioned is that they are not able to take into account the sequential time-dimension that data may possess, which is often the case in educational settings where data is collected at various points in time during a course or program. For example, these classifiers will generally not be able to comprehend that the data collected in week eight follows data collected in week one, which might render the classifier unable to recognize the complete pattern present in the data.

Fortunately, a deep-learning classifier called the recurrent neural network (RNN) addresses this shortcoming of traditional machine learning classifiers. Long short-term memory (LSTM) networks are a type of RNN that can take into account the time dimension present in data and can automatically choose to remember or forget past sequences in order to predict the outcome with greater accuracy. Another advantage of deep learning approaches over traditional machine learning models is that the structure of deep learning models allows them to recognize complex non-linear patterns. Therefore, this state-of-the-art approach has the potential to be valuable for the purpose of predicting academic performance.

Currently, the amount of research in which deep learning approaches are applied in EDM is limited. Coelho and Silveira (2017) conducted a systematic literature review in which they identified recent applications of deep learning in educational data mining and learning analytics. The authors discovered 39 artificial neural network-related papers, but only six deep learning related papers.

Merely three out of those six were related to predicting student performance even though the results of the deep learning approaches were deemed promising.

The purpose of this paper is to contribute to the limited amount of existing research on the application of deep learning to predict student performance by comparing the performance of LSTMs to that of traditional classification methods. The data analyzed for this purpose originates from the Open University Learning Analytics Dataset (OULAD), which is a dataset publicized by the United Kingdom's Open University, one of the largest distance-learning institutions in Europe (Kuzilek, Hlosta, Zdrahal, 2017).

First, the relevant literature related to this research is outlined and the study's hypothesis is formulated. Second, the research design is examined. Third, the data preparation and pre-processing stage is described. Fourth, the methodology is explained. Finally, the results of the study are discussed.

2. Literature Review

There has been an abundance of scientific research on predicting student performance. This section sheds light on the relevant studies and findings in the field. The first half examines the relevant literature corresponding to the traditional classifiers, while the second half sheds more light on the application of deep learning in this field and the development of LSTMs.

2.1. Traditional data mining approaches for predicting student performance

2.1.1. Decision tree

Decision trees and classification trees in particular have been widely used for the purpose of predicting academic performance. Classification trees comprise a subfamily of tree-based models with the aim to subdivide the data into smaller, more homogeneous groups (Kuhn & Johnson, 2013). The two most common measures used to make the nodes of the split as pure as possible are the Gini index and cross entropy. Because classification trees consist of nested if-then statements, they are relatively straightforward to understand and implement.

Yadav and Pal (2012) conducted research to determine the effectiveness of decision trees when it comes to predicting student performance. Three different decision tree algorithms (ID3, C4.5 and CART) were applied to the data of 90 engineering students, after which their results were compared. The ID3 and C4.5 algorithms both use information gain as their metric to base the splits of the trees on, while the CART algorithm uses the Gini-index. Both past performance indicators and demographic attributes were included as predictors in the models. The target variable is comprised of three classes and indicates whether a student (1) passed all classes, (2) was promoted while failing some courses or (3) failed the first year altogether. The results obtained after a 10-fold cross validation show that the C4.5 algorithm achieved the highest accuracy with 68% correct predictions, while the other two decision tree algorithms both achieved 62% accuracy. In a similar study, Adhatrao et al. (2013) compared the capability of the C4.5 and ID3 decision tree algorithms to predict the academic performance of first-year engineering students. The training data consisted of features such as gender, board examination marks, scores in entrance examinations and admission type from students admitted to the second year of the engineering course. Both models were capable of predicting whether 182 current first-year engineering students passed or failed the course with approximately 75% accuracy. The predictive performance of the ID3 decision tree algorithm was also examined by Ogunde and Abjibade (2014). The model used attributes such as gender, student entry grades and entrance examination scores from Nigerian university students as input data. The decision tree predicted the grades of the bachelor students correctly with approximately 80% accuracy.

Compared to other classifiers, the decision tree models appear to have merit as well. Cortez and Silva (2008) found that tree-based models outperformed both multilayer perceptrons with one hidden layer and support vector machines in terms of predicting grades of Portuguese secondary school students. Many attributes were retrieved from mark reports and questionnaires including grades, number of absences and a broad collection of demographic, social and school related attributes. When including the second period grade, decision tree models achieved an accuracy of 93% while predicting whether Portuguese language course students will either pass or fail the course and an accuracy of 76% when predicting with five target labels instead. Both of these scores were higher than the accuracy levels of the multilayer perceptrons, support vector machines and random forest models when the second period grade was included.

2.1.2. Random Forest

A random forest (RF) can be regarded as an ensemble of decision trees mainly used for classification or regression tasks. For classification purposes, each decision tree in the forest casts a vote for the classification of a new sample. The proportion of those votes determines the eventual probability with which the new example is assigned to a specific class (Kuhn & Johnson, 2013). Every tree in the forest trains on a random set of predictors, which serves to de-correlate the trees. Therefore, random forests do not tend to overfit nearly as much as single decision trees do. The downside is that random forests are more intensive to compute. Moreover, the way in which random forest models make predictions is significantly less transparent than is the case with decision trees.

The random forest models applied by Cortez and Silva (2008) that predict whether secondary school students partaking in a Portuguese language course would pass or fail proved to be superior to multi-layer perceptrons, support vector machines and decision trees when the second period grade was not yet known. Intuitively, the overall performance of the models increases when more grades are available to use as input variable. The random forest classifier reached an accuracy level of 85% when no student grades were used, 90% when the first period score was used and close to 93% when the second period score was used. One possible explanation for the superiority of the tree-based models in this context could be that the large number of variables benefit tree-based learners as they in fact perform an internal feature selection. While comparing different classifiers to predict undergraduate students' grades, Asif, Merceron, Ali and Haider (2017) found that random forests based on the Gini index outperform their counterparts based on information gain or accuracy with 71% to 69% and 63% respectively. The model also proved to be more accurate than a rule induction algorithm, various decision trees and an artificial neural network.

2.1.3. Logistic regression

Logistic regression is a popular statistical model that is commonly used to predict a dependent binary variable given the models independent variables. Instead of fitting a linear function to the data, a logistic model fits a logistic function that ranges from 0 to 1. Consequently, logistic regression models can be valuable for classification tasks. The advantages of logistic regression

are its simplicity and the ability to make inferential statements about model terms. (Kuhn & Johnson, 2013)

Ayán and García (2008) demonstrated that logistic regression models are better suited to predict academic achievement than linear regression models. One advantage of logistic regression over linear regression is that the former does not require the variables to meet normality and homoscedasticity assumptions. Attributes such as prior grades and key demographic variables were used to predict the academic performance of university students. Students who had withdrawn from the programme were not included. The results obtained illustrated that the logistic regression models fit the data better than linear regression models. The fact that the logistic regression was able to predict 75% of the cases correctly while the linear regression approach only explained 50% of the dependent variable's variance speaks in favor of the logistic regression model. Numerous studies included logistic regression models when attempting to determine to best model to predict student performance, but they hardly ever establish themselves as the most accurate classifier in this context (Gray, McGuinness & Owende, 2014; Hlosta, Zdrahal & Zendulka, 2017; Kotsiantis, Pierrakeas & Pintelas, 2004).

2.1.4. Support vector machines

Another type of supervised learning model that has been used to predict student performance is the support vector machine (SVM). An SVM acts as a boundary to separate the training data points based on the class of their target variable. The boundary that separates the classes is designed to be the hyperplane that leaves the largest margin between the data points from opposing classes that are closest to the hyperplane. Once the hyperplane is established, the model assigns the new examples to the class that corresponds to the side of the boundary on which the examples are located. According to Kuhn and Johnson (2013), SVM models are very competitive for most classification tasks. However, SVMs can be relatively resource intensive.

Gray et al. (2014) observe the applicability of SVMs when it comes to predicting college student performance. Three student cohorts from 2010 through 2012 with a total of 1072 first year students at the Institute of Technology Blanchardstown in Ireland were analyzed. Besides the general demographic and prior academic performance attributes, the researchers also incorporated

psychometric factors indicators such as motivation, personality and learning strategies in their models. The researchers found that SVMs were the most accurate among neural nets, k-NN classifiers, decision trees, NB classifiers and logistic regression models when testing on the entire dataset using cross-validation. SVMs were capable of identifying strong and poor academic achievers correctly with 73% accuracy. None of the other algorithms exceed an average accuracy of 70%. However, when using the 2010 and 2011 student cohorts as training set and the 2012 cohort as testing set the performance of all models decreased considerably except for that of the k-NN classifier. Interestingly, the researchers found that the models explicitly trained on younger students achieved good results, while the algorithms trained on students above an age of 21 were less accurate.

2.1.5. Naive Bayes

The naive Bayes (NB) classifier relies on Bayes' rule to classify examples. Bayes' rule centers on combining observed predictors in order to determine the probability that the outcome is of a certain class (Kuhn & Johnson, 2013). The computational complexity is reduced considerably by adding the stringent assumption that all of the predictors are independent of the others. As a result, the model can be computed quickly and even though the assumption tends to be unrealistic in many cases, the model's performance remains competitive relative to other common classifiers.

Kotsiantis et al. (2004) compared multiple machine learning algorithms including classification trees, NB classifiers, 3-NN classifiers, logistic regression, neural networks and SVMs. The research showed that the NB classifier was the most appropriate to predict a student's performance due to its superior accuracy, overall sensitivity and ease of implementation. The training set constituted of key demographic characteristics of 354 Hellenic Open University's students and attributes from tutors' records such as written assignment grades and absence. In total, fifteen features were used as input variables, while the binary dependent variable indicated whether a student passed or failed the final examination. The training phase was divided in 9 consecutive stages. Every stage adding either a student's presence in the corresponding face-to-face meeting or a score for the corresponding written assignment. For example, in the first stage only demographic attributers were used as input, while in the ninth stage all attributes from tutors' records were also included. Intuitively, the accuracy of the models improved as more grades and absence notices were

incorporated in the model. In other words, later in the course it is easier to predict how students are going to perform as opposed to early in the course. The accuracy of the NB classifier ranged from 63% with only demographic predictors to 82% with the entire selection of predictors. The average of 72.48% was the highest among all other considered models. However, the average accuracy values of the top four classifiers were not significantly different. The decision tree and 3-NN classifiers on the other hand were significantly less accurate in their predictions. Similarly, Asif et al. (2017) concluded that the naive Bayes classifier predicts graduation marks most accurately among decision trees, random forests, neural nets, 1-nearest neighbor and rule-based classifiers, with an accuracy measure of 83.65%. Mainly variables such as pre-university marks and yearly marks for the first two years of the four-year programme were used as model input. In addition, Osmanbegović and Suljić (2012) also found that the Bayesian classifier performed better than the decision trees and MLPs when using input data collected from surveys. The Bayesian classifier managed to classify whether students passed or failed correctly with 76.65% accuracy, while the MLP and decision tree did not surpass the baseline of 75.88% of students that passed.

2.1.6. Multilayer perceptron

Multilayer perceptrons (MLPs) have been widely adopted for the purpose of predicting student performance. Generally, these MLPs are typical feed-forward neural networks with one or two hidden layers. These network architectures typically consist of (1) an input layer through which the input data enters the network, (2) an output layer, which outputs the corresponding class of the example at the end of the network and (3) a predefined number hidden layers in between the input and output layer meant to process the data and find relevant patterns. These layers are comprised of a predefined number of nodes that each hold a value between 0 and 1. Each node in a layer is connected to all nodes in the two adjacent layers. As a result, combinations of nodes in one layer can activate certain nodes in the next layer based on their values and the connection between the nodes. The parameters involved tend to be estimated through supervised numerical routines such as back-propagation (Kuhn & Johnson, 2013). MLPs proved to be powerful classifiers as they allow for extensive customization and are able to recognize non-linear patterns. The main drawback is that the structure of artificial neural networks (ANNs) does not provide any insights on the approximated relationship.

Oladokun, Adebanjo and Charles-Owaba (2008) used a MLP with two hidden layers to predict whether students will achieve a good, average or poor level of performance. The input variables that are obtained from students' applications and transcripts of records include matriculation exam scores, results in various subjects and student background information. The performance of the MLP was validated through cross-validation and by the use of a separate test set. A total of 112 student records of Nigerian university students were analyzed. The MLP is capable of predicting 82% of the good performance class correctly, 53% of the average performance class and 88% of the poor performance class. The average performance of the model in the study is reported to be 74% by taking the average of the three previous values. In a similar study, Delgado, Gibaja, Pegalajar and Pérez (2006) used ANNs to predict which students will pass and which will fail the considered course based on features extracted from Moodle logs with an 80% success rate. These ANNs encompass a hidden layer with a non-linear radial basis activation function and a linear output layer. Similar to the VLE used in this paper, Moodle logs contain information on the students' activities and accessed resources. No comparisons are made between ANNs and other data mining methods in these studies, which makes it difficult to assess whether ANNs are comparatively superior to other models.

Kabakchieva (2012) did in fact find that an ANN classifier with a single hidden layer predicted whether students belong to the weak or strong performing class more accurately than other models including a rule-learner, a decision tree and a k-NN classifier. The ANN produced accuracy and recall percentages of 74%. This study also made use of a much larger dataset with a total sample size of 10.067 students in the Bulgarian education sector. Besides student background information, the author used pre-university data and data corresponding to the current semester. Bendangnuksung and Prabu (2018) demonstrated that a more complex neural network with two hidden layers performed better than such a single layer ANN as well as decision trees, and naïve Bayes classifiers.

2.1.7. Other approaches

Minaei-Bidgoli, Kashy, Kortemeyer and Punch (2003) combined multiple classifiers to predict student performance and compared the performance with that of traditional classifiers. The authors used student and course data of an introductory physics course, which included 12 homework sets

that the students had to complete online. The educational web-based system LON-CAPA collected a range of detailed features, which were used as input for the models, including total number of correct answers to the homework sets, the amount of attempts a student needed, the time spent on a particular problem and so on. The researchers used 10-fold cross validation to measure the models' average performance The combination of multiple classifiers achieved the highest accuracy levels, namely 86.8% for two class labels (passed and failed), 70.9% for 3 classes (high, middle and low) and 51.0% for nine classes representing the students' grades. The results proved to be superior overall to those of the other contestant models, which included various tree classifiers, a NB classifier, an MLP and a k-NN classifier. Furthermore, the authors applied a genetic algorithm as a means to optimize the parameters of the models, which resulted in a 10 to 15% mean individual performance improvement. Additionally, they identified that the number of correct answers and the number of attempts were the most predictive features.

In a similar effort to build a generalizable classifier, Pandey and Taruna (2016) propose an integrated classifier that consists of three complementary algorithms, namely decision trees, k-nn classifiers and Aggregating One-Dependence Estimators (AODE). The results of this integrated approach are promising as it achieves an accuracy score ranging from 87.03 to 98.86% depending on the dataset thereby beating the other examined classifiers. More details on the testing and validation approach would prove insightful to assess the practical implications of these findings. Nevertheless, the LSTM network laid out in this study could potentially be a valuable asset to such an integrated approach.

Hlosta et al. (2017) propose a novel approach to predict student performance as early as possible without using training data from previous periods in which the particular courses were taught. Instead, they train their machine learning models only on the data generated during the current course. They find that there is a particularly strong correlation between failing or not submitting the first assignment in a course and not finishing the course. Therefore, they attempt to identify the students at risk of not submitting the first assignment by comparing patterns of their behavior with that of students who already submitted the assignment. Thereby, making the assumption that students who do not submit the first assignment exhibit different behavior before the assignment is due than students who did in fact submit the assignment. The average performance of the used

models varies over time. Relatively, the XGBoost, random forest and SVM models achieved the best performance beating the logistic regression and naive Bayes classifier. The XGBoost model achieves F1-scores of 0.67, 0.55 and 0.71 for courses B, D and F respectively, precision scores of 0.58, 0.40 and 0.67 and recall scores of 0.81, 0.90, 0.76 for those same courses. The relatively high recall scores illustrate that the approach is able to identify a relatively large proportion of students who are actually at risk. However, the lower precision scores imply that a relatively high proportion of students that where not at risk were predicted to be at risk. The F1-score exemplifies the harmonic mean between the precision and recall metrics. The reported F1-scores were higher compared to those of the benchmark model that accounted for legacy data. The fact that their study bases their results on the same dataset as this study does provides an attractive benchmark to compare the models developed in this paper to. However, their research is more focused on identifying at-risk students including students that withdraw from the course, whereas this study is more concerned with predicting students' final performance not taking into account dropouts. Moreover, instead of using the submission of the first assignment as dependent variable, this study predicts whether a student passes or fails directly.

As identified in the literature review conducted by Del Río and Insuasti (2016), decision trees, ANNs and NB classifiers are the most popular techniques for academic performance prediction. Even though SVMs, logistic regression and random forest models are not as popular, their performance will still be examined, as they have proven to be competitive classifiers in many data mining contexts. In addition, the literature review found that the most popular predictor variables are comprised of academic data either with or without other categories of data. Moreover, course grade, some form of GPA or binary pass/fail target classes are among the most frequently used target variables. Similarly, the models in this study predict whether students will pass or fail their corresponding course. This section's biggest take away is that the performance of classifiers seems to depend heavily on format and quality of the data as well as the context in which the research takes place.

2.2. Deep learning approaches for predicting student performance

In recent years, deep learning has emerged as one of the state-of-the-art methods in machine learning and it has proven its worth in many applications ranging from speech recognition to drug discovery. Goodfellow, Bengio and Courville (2016) define deep learning the following way:

> Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones. (p. 8)

Generally, this definition applies to neural networks with complex architectures. However, there is no universally agreed upon point, which separates shallow learning from deep learning. This paper employs a common rule of thumb suggesting that feed-forward neural networks with two or fewer hidden layers are considered to be shallow, while feed-forward neural networks with more hidden layers and neural networks with more complex architectures such as RNNs or CNNs are considered to be deep learners. Guo, Zhang, Xu, Shi and Yang (2015) conducted one of the few studies in which a deep learning approach was compared to traditional classifiers designed to predict student performance. The deep learning network was comprised of four hidden layers and managed to predict students' grades (5 class labels) with an average of accuracy of 77%. The accuracy level of the other classifiers including a NB classifier, a MLP and a SVM did not exceed 49%. The inferior performance of the MLP was caused by its tendency to overfit, while the SVM and NB classifiers ended up being too shallow to compete with the deep learning network.

Apart from deep feed forward neural networks, recurrent neural networks (RNNs) also have the potential to distance themselves from traditional classifiers in this context. RNNs in particular have brought about considerable advancements in sequential data analysis (LeCun, Bengio & Hinton, 2015). This is mainly enabled by the way in which RNNs are designed, making use of loops that allow information to persist over time. As a result, RNNs have the capability to recognize the evolution of data patterns over time as opposed to feed-forward neural networks. Due to this particular architecture, RNNs can be considered as the deepest of all neural networks. (Schmidhuber, 2015). RNNs do not just feed the input at the current time step to the network. Instead, they feed both the input at the current time step as well as the hidden state of the previous time step. Since the hidden state gets updated every iteration, it represents the network's memory

of the entire sequence that came before the current time step. Figure 1 (Olah, 2015) below illustrates this process, as not only the input at the current time step, x_t, determines the current hidden state of the network, *A*, but also the hidden state from the previous sequence. Therefore, the effect of all previous sequences is being remembered to predict output h_t.

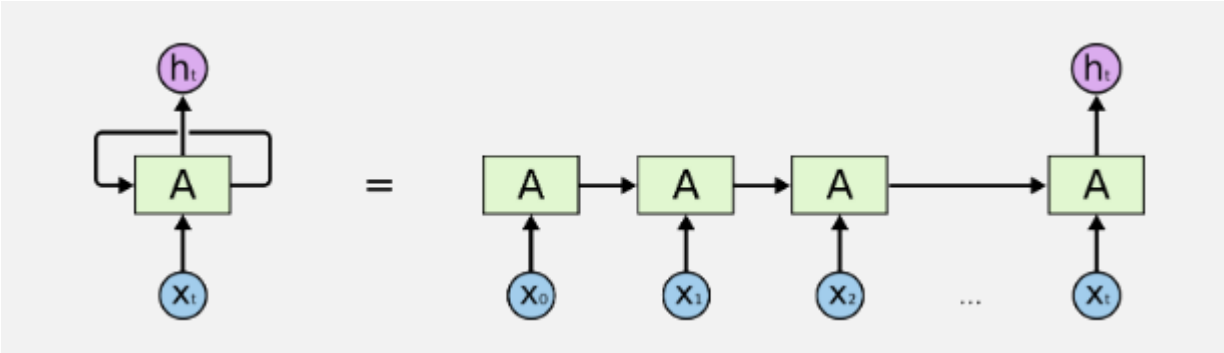


FIGURE 1 RNN LAYER

Note. Adapted from "Understanding LSTM Networks" by Olah, C. (2015, August 27). Retrieved from: http://colah.github.io/posts/2015-08-Understanding-LSTMs

Even though many educational institutions collect large volumes of sequential data on students progressing through courses, research that attempts to measure the potential benefits of applying RNNs to this educational data is still scarce. Okubo, Yamashita, Shimada and Konomi (2017) made an effort to predict student performance by using RNNs with gated recurrent units and compared its performance with multiple regression analysis. The models are supposed to predict the student's final course grade out of five class labels with data originating from logs stored in Kyushu University's learning support system including for example quiz scores, report submissions and slide views. Instead of using cross validation to test the performance of the models, the authors train the data on four courses taught in 2015 while testing the data on two courses taught in 2016 to provide a more realistic picture for the application in mind. The RNN achieves an accuracy ranging from 71% correct predictions in the first week to around 85% in weeks six until eight. Intuitively, the performance of the model increases as the course progresses and more data becomes available. In week one, the RNN is more than 20 percentage points more accurate than the multiple regression analysis. However, the performance of the regression analysis converges towards that of the RNN until they reach a similar level of accuracy after week five. Nonetheless, the RNN

appears to predict students with a low grade (D or F) with more accuracy overall than the regression analysis, which suggests that the RNN would be more appropriate to identify students at risk of failing a course. Similarly, Mondal and Mukherjee (2018) make use of RNNs to predict student academic performance. The designed RNN achieved an accuracy score of 85.4%, which is higher than the accuracy achieved by the one- and two-layer ANNs proposed by Bendangnuksung and Prabu (2018). Since both studies utilize the same dataset this result could be seen as an indication that RNNs are able to predict whether students will perform well in their courses with more accuracy.

This paper follows a similar approach in which long short-term memory networks are utilized in order to predict student success while using only three distinct features. Subsequently, their performance is compared to that of other widely used data mining models. LSTMs are a specific RNN architecture introduced by Hochreiter and Schmidhuber (1997). Where traditional RNNs typically face problems when trying to connect present information with information that happened many time steps ago, LSTMs are designed to tackle this problem of vanishing or exploding gradient descent more efficiently thereby allowing longer sequences to be modeled. LSTM cells are designed to replace the RNN cells and perform a more extensive series of matrix operations making it possible for the network to remember what happened a long time ago. Specifically, LSTMs make use of memory cells and gate units to be able to remember only relevant information. Over the years, there have been multiple contributions to the LSTM architecture. One of the most notable is the addition of the forget gate by Gers, Schmidhuber and Cummins (2000). The introduction of forget gates made it possible for LSTM cells to forget irrelevant information by resetting itself at appropriate times thereby allowing LSTMs to effectively solve continual problems. In contrast, the most widely used classifiers identified previously are not able to account for a separate time dimension nor the complicated nonlinear interrelationships that exist between variables and factors for predicting academic performance (Guo et al., 2015). Therefore, LSTMs could prove valuable when predicting student performance, especially considering the availability of sequential data such as clickstreams in educational institutions. Even though this paper is among the first to apply LSTMs to predict student performance, LSTMs have been applied to predict student drop out with an average accuracy of 90% (Liu, Xiong, Zou & Wang, 2018). The data collected from Chinese MOOCs provides similar clickstream data as the OULAD dataset that this study uses. On top of

that, this study uses a similar approach wherein the performance is estimated multiple times over the course duration.

This paper's hypothesis stems from the notion that LSTMs are capable of modeling complex nonlinear relationships over relatively lengthy periods of time whereas traditional machine learning classification methods are not.

Hypothesis: Long short-term memory networks perform better when it comes to predicting student performance than traditional machine learning methods including decision trees, support vector machines, naive Bayes classifiers, random forests, logistic regression models and feed-forward neural networks.

The following subquestions are of vital importance to make a statement about the performance of LSTM networks relative to traditional data mining methods when it comes to predicting student performance:

1. How well can the LSTM network predict student performance?
2. How well can the traditional data mining methods predict student performance?
3. Does the LSTM network outperform the traditional data mining methods in terms of predicting student performance?

These questions act as guidelines for the research approach laid out in the upcoming section.

3. Research design

The following comparative analysis is conducted following the cross-industry standard process for data mining (CRISP-DM). The CRISP-DM framework consists of six phases that act as a guideline for data mining projects (Chapman et al., 2000). The first of the six major phases is the business understanding phase. The importance of predicting academic performance and identifying at-risk students has already been established in the prior sections. The ultimate goal is to develop an algorithm that can predict student performance as accurately as possible. The next section will focus on the data understanding and data preparation phase. In this phase, the Open University's

dataset that is used for this study is examined and prepared for analysis. After the pre-processing of the data, the modeling phase will commence. In this section, LSTMs and widely used traditional machine learning models are adopted to predict academic performance. Finally, the effectiveness of the models is compared during the evaluation phase following the three research questions specified in the previous section. The deployment phase is beyond the scope of this research.

4. Data and pre-processing

This paper makes use of the Open University Learning Analytics Dataset (OULAD), which is a public dataset containing data on roughly 30.000 students over seven distinct courses (also called *modules*) taught at one of Europe's largest distance-learning institutions (Kuzilek, Hlosta & Zdrahal, 2017). The data covers a timespan of two years divided into a maximum of 4 periods termed *presentations*. These presentations include courses that started in February 2013 (*2013B*) and February 2014 (*2014B*) as well as in October 2013 (*2013J*) and October 2014 (*2014J*). The distribution of students over the modules in each presentation is illustrated in figure 2 below. The figure indicates that courses BBB, DDD and FFF are the only ones that span over all four presentations. Therefore, the analysis is limited to those three courses as they represent the largest group of students over the longest period of time. The models will be developed for each course seperately as opposed to blending all three courses together in one big dataset.

All course presentations have differing durations. For example, the presentation of course BBB that started in February 2013 ended after 240 days, while the presentation of the same course starting in October 2014 took 262 days. In the appendix the exact structure of the examinations together with the durations of each course presentation are depicted. Only the data recorded before the exam submission day is used for the prediction, as it will be of no use to predict student performance after the students completed their final examination.

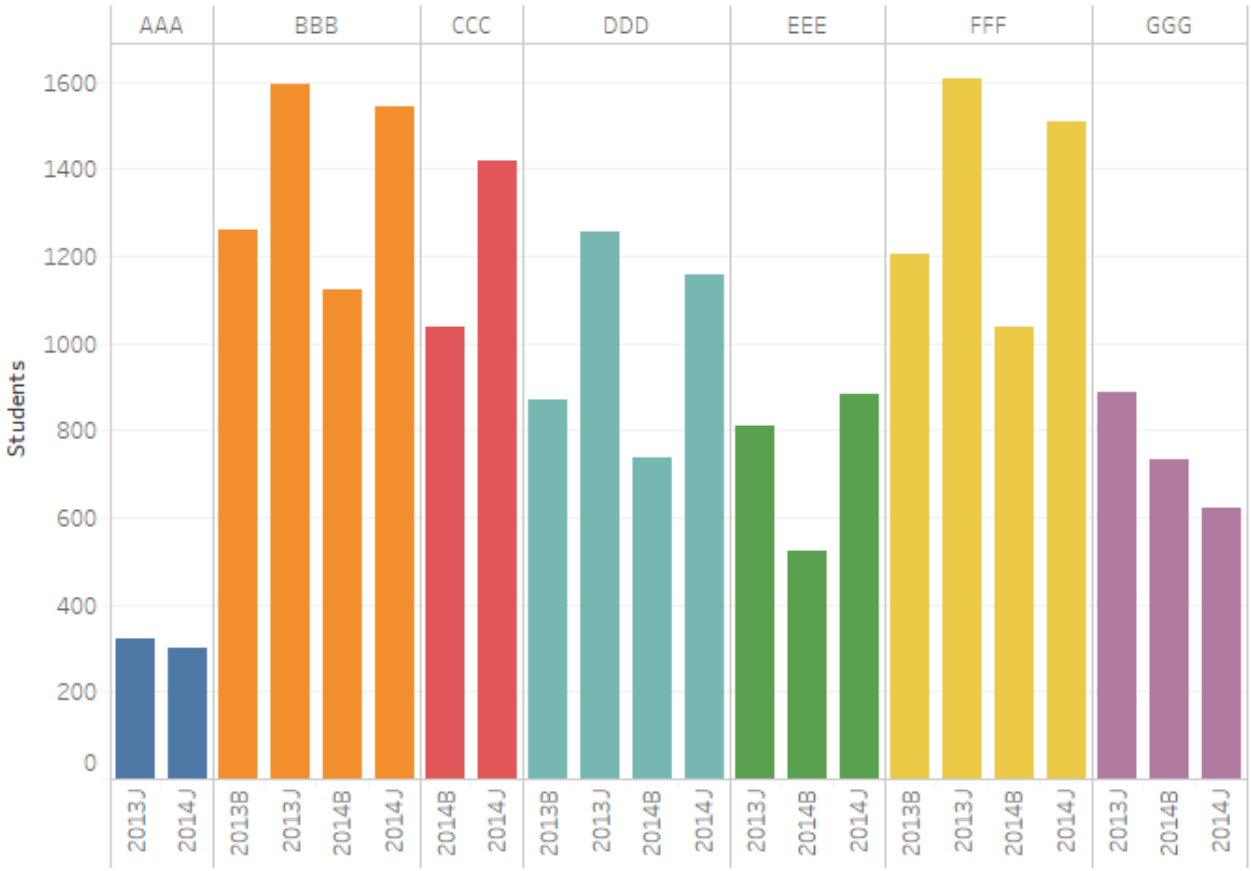


FIGURE 2: STUDENT DISTRIBUTION

The grades of the final examinations are not present in the dataset and for courses BBB and FFF the examination date is missing. In this case, it is assumed that the final exam took place on the last day of the course presentation based on the following quote in the official documentation related to the OULAD dataset: "If the information about the final exam cut-off day is missing, it takes place during the last week of the module-presentation" (Kuzilek, Hlosta, Zdrahal, 2017, p. 6). On the other hand, for course DDD students appear to be able to submit their examination attempts before the submission deadline. Therefore, the cut-off date varies for each student enrolled in course DDD. Similarly, students have different registration dates. The LSTM architecture makes it possible to account for these differing sequence lengths by padding the shorter sequences with a value that does not occur in the original dataset. In this case, these courses with shorter presentations are padded with '-1' values. The so-called masking layer ensures that that these values are skipped when training the network. The architecture of traditional machine learning methods does not allow for this. Therefore, the missing values in the features that are caused by

this are replaced by ‘0’ values. The same goes for students who registered on a later date than the first registered student in the course. The VLE also records student behavior before the course starts, as the students are usually already able to access the VLE prior to the course. Course durations are not the only aspect of the courses that change over subsequent periods. Changes to the examination structure also occur on occasion. For example, course BBB maintains the same amount of examinations from February 2013 up and until February 2014. However, in the October 2014 presentation the number of assessments are reduced from twelve to six assessments.

In terms of evolution over the four course presentations, the three courses are all distinctly different. Course FFF is the ideal benchmark, as it experienced no structural changes over the four course presentations as can be seen in appendix A. On the other hand, courses BBB and DDD did face changes in their assessment structure. As already mentioned, course BBB maintained the same number of exams for the first three presentations, but halved the amount for the last period. Course DDD experienced a reduction in the number of assessments immediately after the first course presentation in February 2013. The three presentations that followed remained the same. Due to these differences, the opportunity arises to examine how well each model deals with structural changes.

The information present in the dataset includes student background information such as gender and their age group, but also data that is recorded in the VLE such how often and when students clicked on specific resources, their assessment results and their (de-)registration date. Because there is no data on the students’ activity except during those seven courses in 2013 and 2014, the students’ results in earlier courses are not taken into account. This comes with the benefit that the results are generalizable to first-period courses in which the educational institution does not have any prior information on the learning behavior of particular students. Except for the missing examination dates and some missing data with respect to the demographic variables, the dataset is relatively clean and well structured. One exception being that a few students deregistered from the course before the course was over, while still being classified as having failed the course instead of having withdrawn. The status of these few students has manually been converted to “withdrawn”.

As to not introduce unnecessary complexity to the model, only three variables are taken into account. First, the amount of times a student clicks in the VLE each day. Second, the amount of assessments a student submits each day. Third, the student's average assessment score that is updated daily. The three variables are rescaled to lie between 0 and 1 for each course presentation separately in order to account for structural changes between the course presentations such as the differing durations and examination structure. This changes the absolute nature of the variables to a relative one. The intuition behind this standardization can be illustrated by recalling the fact that the amount of assignments was cut from twelve to six in the October 2014 presentation of the BBB course. If the models are trained on the first three course BBB presentations, they might identify that when a student submits all twelve assignments this has a substantial positive influence on the student's final performance. However, when the models are subsequently tested on the fourth course BBB presentation and a student submitted the new maximum of six assignments, they could erroneously interpret this as a mediocre effort. Rescaling the variables for each presentation separately accounts for this issue as both the maximum of twelve and of six would be rescaled to 1.

As stated, the first feature is the number of times a student clicks on something in the VLE per day. Since the learning process at the Open University predominantly takes place online, the daily number of clicks is expected to be a reasonable proxy for student effort. One issue with rescaling the daily number of clicks separately for each course presentation arises from the presence of outliers. There are days in the dataset on which a student clicked an exorbitant number of times in the VLE, which inflates all other values in the course presentation after rescaling. This issue is tackled by transforming the rescaled number of clicks per day that are larger than 0 into percentiles. For instance, if the number of clicks a student performs in a day is higher than that of the bottom 5% of students on that day but lower than that of the bottom 10% cut-off point, the value is transformed into 0.10, which represents the 0.05 – 0.10 bin. The resulting twenty bins represent the ranking of the number of clicks that a student conducted on a certain day relative to that of the other students within the course presentation on that same day. The second feature is the number of assignments a student submitted on a particular day. The number of assignments a student submitted and the day on which this happened might have an influence on the student's final performance. The advantage of the LSTM's architecture in this case would be its ability to identify

whether the student submitted his or her assignment at an early or late stage. Because the traditional machine learning models do not take the sequential time order into account they are not suited extract this type of information. The third and final feature is the student's average assignment score. The student's assignment score is updated daily and calculated as the cumulative average score of all the assignment that the student submitted in the course so far. Even though every assessment has a specific weight tied to it, this updated score is not weighted in any way. The reason being that assessments with a very low weight might still capture information relevant to the student's final result. Exams have a weight of 100% and are counted separately from the other assessments. Therefore, it is hypothetically possible that a student achieves the full score on every assessment but still fails the course if the student does not pass the exam. None of the demographic information is used as input data for the models in an attempt to reduce model complexity. Therefore, the research is even applicable to educational institutions that do not collect demographic information on their students.

In addition to the features mentioned, the dataset holds information on the final result of each student in their respective courses. Students are classified as either having failed the course, passed the course, passed the course with distinction or withdrawn from the course. Similar to Minaei-Bidgoli et al. (2003) the students that dropped out of the course are not taken into consideration as nothing can be said about their final result in the course. As already mentioned, a target variable with two labels will intuitively result in more accurate predictions than a target variable with three labels. However, the downside is that relevant information might be lost. Therefore, this study focuses on a binary target variable with two class labels, "pass" and "fail". Consequently, the students that passed with distinction are classified as a regular "pass". The distribution of student performance over the three courses is illustrated in figure 3 below. It is interesting to note that for all three courses, the last presentation has the lowest proportion of students that failed the course. Additionally, the October course presentations seem to include more students than the February presentations. The "pass" class is the majority class in every course presentation.

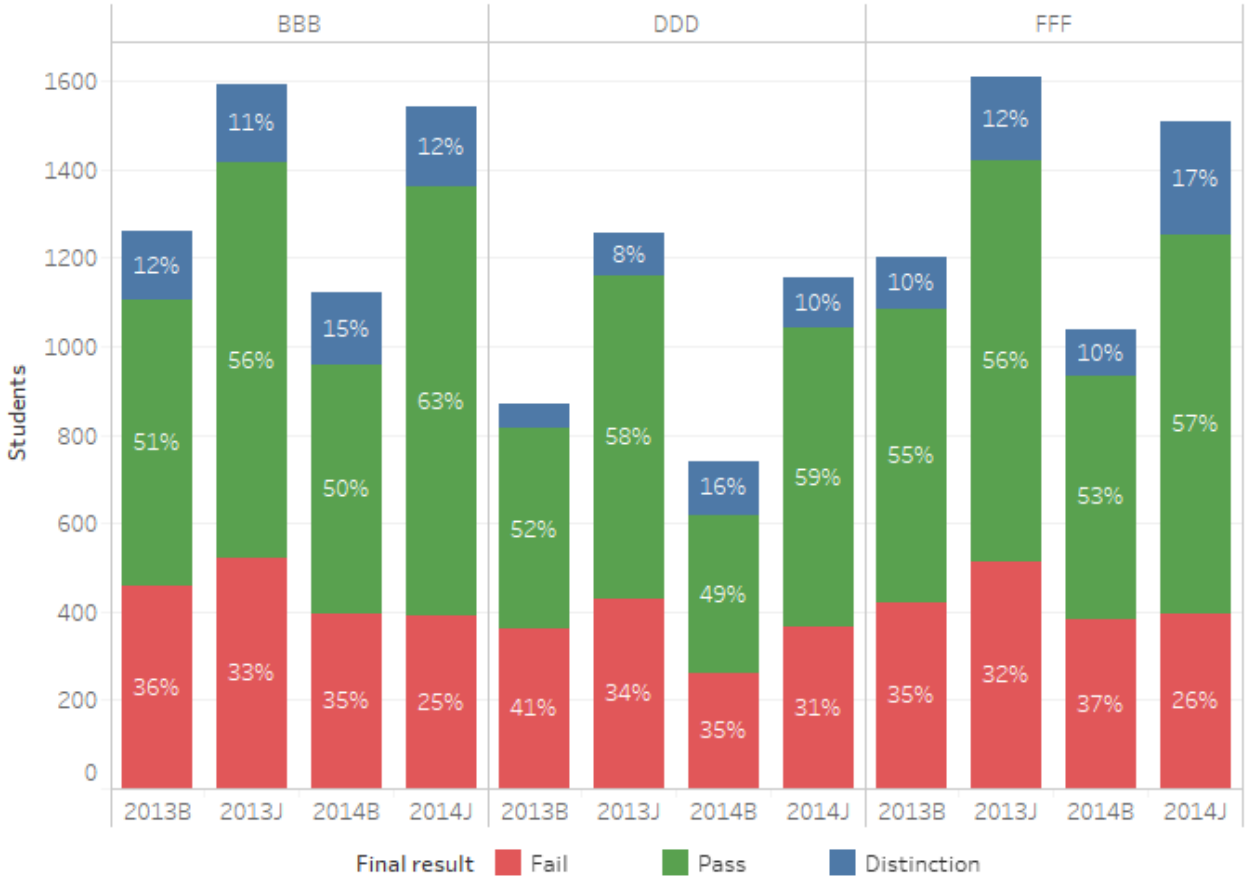


FIGURE 3: STUDENT PERFORMANCE DISTRIBUTION

Due to the difference in architecture between the LSTM and the traditional machine learning models, the input data requires a different format. The LSTM requires three-dimensional input data consisting of samples, features and sequences, while the traditional classifiers can only process a two-dimensional input comprised of samples and features. The samples are in this case the individual students. The sequences are the days between the day that the students get access to the course on the VLE and the day before final examination. It is important to note that the traditional machine learning models do not accommodate this time dimension. The features in this case are the considered variables. Even though the dimensions are different, it is vital to use the same input data for comparison's sake. Because the LSTM already has the dimensions representing the students and the course days, only three predictor variables (clicks, assessments and score) are required to import the relevant data. In contrast, the traditional classifiers need to contain a feature for every predictor on a certain day in the course to maintain the same granularity as the LSTM has. Essentially, both the features and time dimension are accommodated in the feature dimension.

For example in the case of course BBB, which counts a maximum of 291 days, there is a feature that stores the amount of clicks for each day resulting in 291 features. Because the same goes for the other two variables, the amount of predictors totals 873 features for the traditional machine learning algorithms. An attempt to reduce the amount of variables has been made using principal component analysis. However, the prediction results suffered considerably. Therefore, the choice has been made to keep the number of features as is. Finally, all models use the binary performance target variable, which represents either a pass or a fail for the corresponding course.

Finally, the datasets for each course are split into training and test sets. The models are trained on the first three course presentations after which they are tested on the remaining fourth course presentation. This approach provides a more realistic view of the performance of the models than a random train-test split. The fact that two of the three courses change in structure over the presentations makes it possible to identify how well the different models perform as the courses undergo a restructuring. Considering the notion that educational institutions may change the structure of their courses regularly, this approach offers a relatively realistic insight into the models' predictive accuracy.

5. Methodology

In order to test the hypothesis and answer the research questions it is necessary to train the LSTM and the traditional machine learning models on the training data and testing its effectiveness on the test dataset. The LSTM model is developed with Python's Keras package and the hyperparameters are tuned with the help of the Hyperas package.

It is always challenging to determine when the performance of the models' should be measured. The earlier the identification of at-risk students the better. However, the performance of the models is most likely higher towards the end of the course as more clickstream data and assignment scores become available. To tackle this dilemma, the performance of the models' is measured at eleven different points during the course starting at the beginning of the course and ending a day before

the final examination. The fact that the VLE is already accessible for students several days before the course starts makes it possible to carry out a prediction the day the course starts without any kind of demographic data. Whether this prediction will prove accurate is questionable however. The remainder of the course is split into ten deciles of equal length unless the number of days is not divisible by ten. In that case, the last decile is slightly shorter than the other ones. The number of days for each course depends on the course presentation with the longest duration. The decile lengths for Course BBB, DDD and FFF are 27 days, 26 days and 24 days respectively. For example, the performance of the models' is measured every 27 days since the start of course BBB except for the last prediction which takes place 18 days after the tenth prediction. The time between the last and second last presentation is shorter due to the fact that the maximum duration of a BBB course presentation is 269, which is not divisible by ten and is a longer duration than the last course presentation, which the models are tested on.

As per the comparative nature of this study, the LSTMs and the traditional machine learning models are all developed separately. The LSTM consists of (1) an input layer, (2) an output layer with a single node and a sigmoid activation function, and (3) a hidden LSTM layer with 16 nodes and a hyperbolic tangent activation function, which allows the model to learn non-linear functions. Between the input and the LSTM layer the masking layer mentioned earlier is located as well as a batch normalization layer causing the activation of the input layer to have a mean of 0 and a standard deviation of 1. A dropout layer is added behind the LSTM layer to prevent overfitting by setting 50% of the input units to 0 at each update during the training phase. Figure 4 provides a high level overview of the LSTM network. The LSTMs are trained in batches of 32 samples at a time. The learning rate of the Adam optimizer as well as the number of training epochs are determined by hyperparameter optimization. A predefined grid of possible parameters is developed and the performance of the model with 10 different combinations of those parameters is tested on a validation set comprised of 20% of the samples from the training data set. Subsequently, the combination of parameters with the best performance is used as the definitive combination of parameters for the model. After which, the model is tested on the separate test set corresponding to the fourth and final course presentation of each course. The more computing power one has, the more extensive one can make the parameter grid and the higher the number of iterations that can

be evaluated in order to get closer to the optimal combination of hyperparameters. The hyperparameters are optimized for every decile in every course separately.

The traditional machine learning models are all created with Python's popular Scikit-learn package. To reiterate, these models include decision trees, random forests, support vector machines, logistic regression models, naive Bayes classifiers and multilayered perceptrons with a single hidden layer. Whenever possible, the parameters of the traditional classifiers were tuned by implementing a parameter grid search with cross-validation. Due to the parameter tuning, the decision trees and random forest can be based on both cross-entropy and the Gini index. Additionally, it allows the SVM to either use a linear kernel or a radial basis function. The number of neurons in the hidden layer of the MLP is tuned as well as its activation function. The definitive settings of these parameters depends on the performance they reach on the training set by means of 5-fold cross validation.

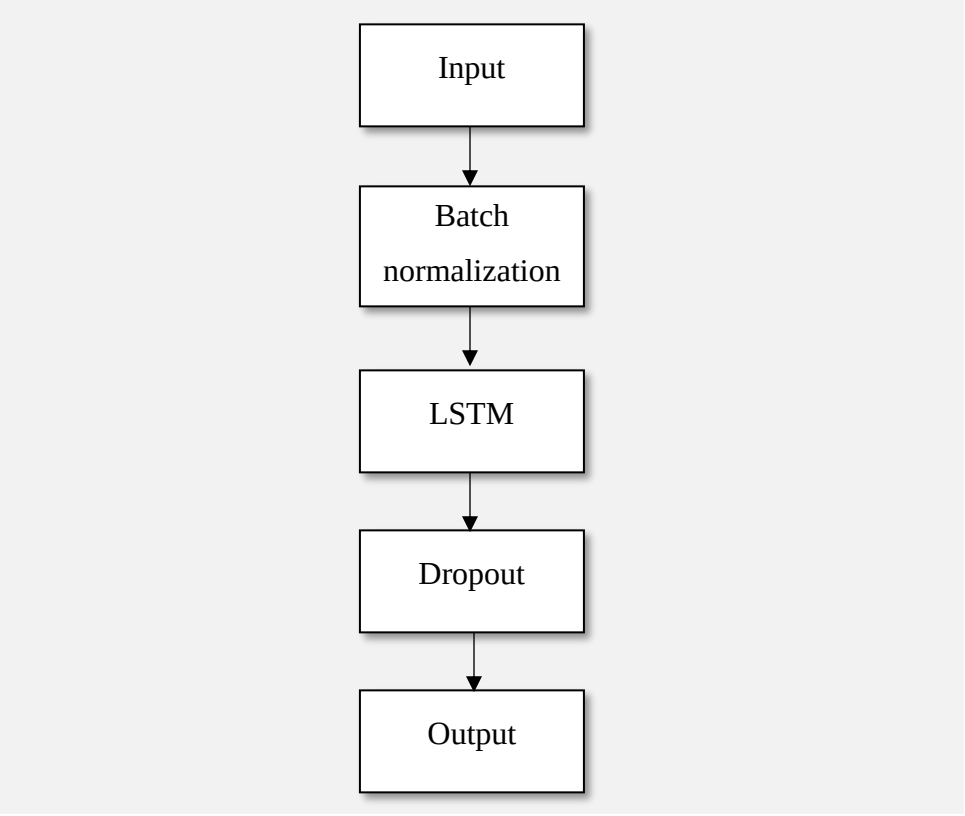


FIGURE 4: LSTM NETWORK LAYERS - HIGH LEVEL OVERVIEW

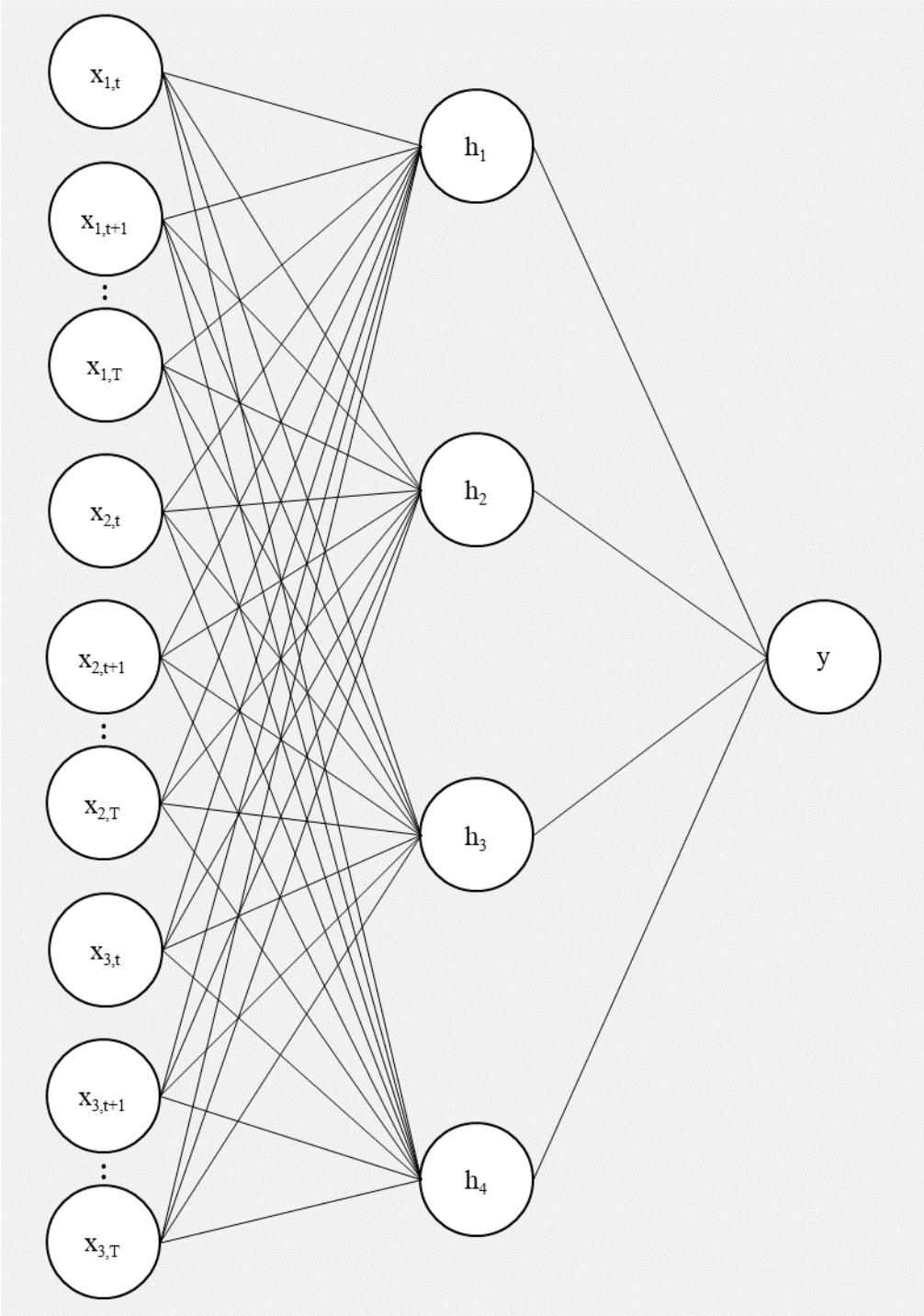


FIGURE 5: FEED FORWARD NEURAL NETWORK

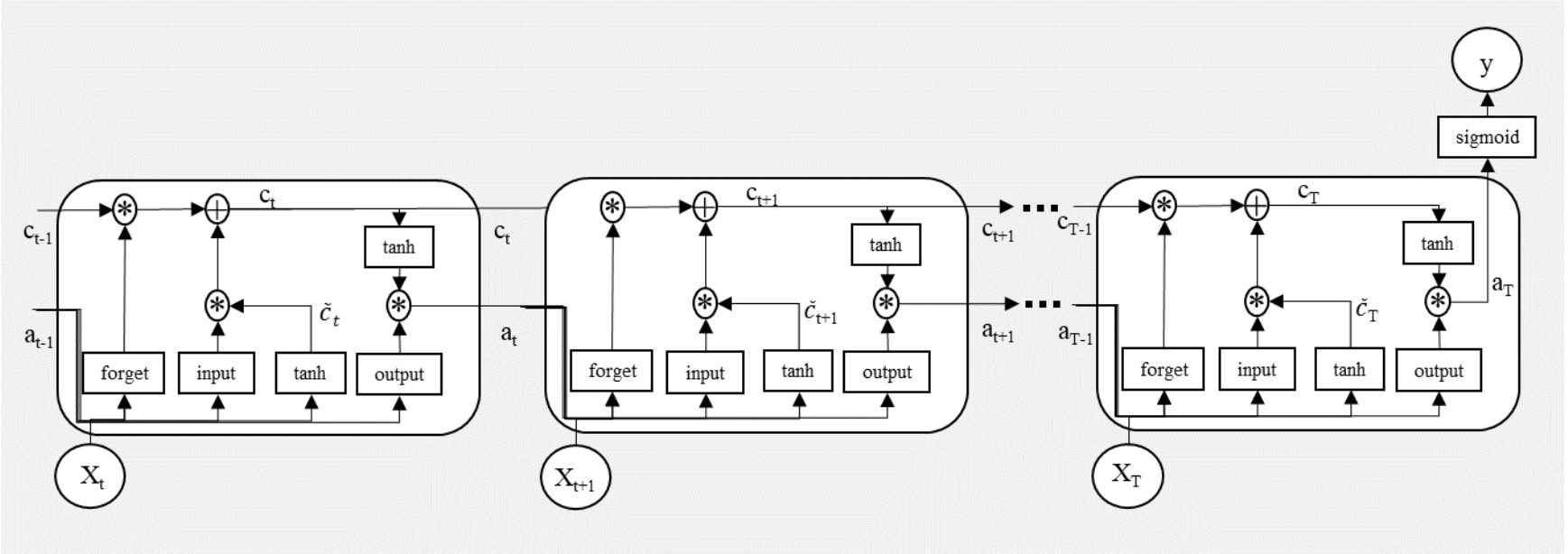


FIGURE 6: LSTM

Figure 5 and 6 illustrate the main differences between the traditional classifiers, in this case a feed forward neural network with one hidden layer, and the LSTM network. The first main difference is that the ANN does not explicitly account for the time dimension of the data in contrast to the LSTM. The ANN's leftmost layer represents the input layer where each node symbolizes an input feature, x, at a certain day, t. For the current application, $x_{1,t}$ can be seen as the 'daily clicks' feature at day t, $x_{2,t}$ as the 'number of assessments completed' feature at day t and $x_{3,t}$ the 'average assessment score' feature at day t. The input nodes between day $t+1$ and the final day of the course presentation, T, are not depicted in the figure, but their presence is indicated by the dotted line. It is important to note that none of the input nodes are directly connected to one another. Each input node is only connected to all the nodes in the subsequent hidden layer. Consequently, the ordering and the sequence of the input nodes are irrelevant to the model. The model is not capable of recognizing that the student's number of clicks on the first day should be taken into account before the number of clicks on the second day. The LSTM on the other hand, does possess this ability as illustrated in figure 6. In contrast to the ANN, the sequential ordering of the LSTM cells is dependent on the day at which the specific input data was registered. It can be seen that the vector of inputs, X, on a certain day, t, is imported in the corresponding LSTM cell, while its impact carries over to the subsequent LSTM cells. For this application, the one-dimensional input vectors consists of the three features mentioned earlier on the corresponding day. The main element that makes the LSTM better at modeling long-term dependencies than RNNs is the addition of the cell state, c_t, which is meant to tackle the issue of vanishing gradient descent present in RNNs (Hochreiter & Schmidhuber, 1997). The cell state resembles the long-term memory of the cell, as it can store information that happened very early on in the sequence. In order to control and protect the cell state, the forget- and input gates were introduced (Gers, Schmidhuber & Cummins, 2000). These are two sigmoidal functions that help decide whether information in the cell state should be remembered or forgotten and updated with information that is more current. The candidate cell state, $\tilde{c}_t$, is a hyperbolic tangent function of the input vector, X_t, the hidden state of the previous LSTM cell, a_{t-1}, and their corresponding weight matrices and is meant to replace the current cell state if the model decides it is necessary for the cell state to be updated. The cell state can flow unchanged through many LSTM cells before ever being updated. The elementwise operations taking place in the cell are indicated with an '*'-symbol for multiplication and a '+'-symbol for addition. The hidden state of the network, a_t, represents the essence of what the model has learned

up to time *t*. The hidden state is a one-dimensional vector that consists of a predetermined amount of hidden units. The LSTM cells in this application possess sixteen hidden units. After passing the sigmoidal output gate, the hidden state of the previous timestep, a_{t-1}, and the inputs of the current timestep, X_t, multiplied with their corresponding weight matrices, are combined with the cell state to compute the new hidden state of the network, a_t, which also serves as one of the outputs of the LSTM cell. Subsequently, the newly computed hidden state is passed through to the next LSTM cell, which corresponds to the subsequent timestep. In the last LSTM cell of the sequence, the final hidden state of the network is passed through a sigmoidal function to predict whether the student passed or failed. Similar to ANNs, the LSTM model is trained by means of backpropagation (Hochreiter & Schmidhuber, 1997). The input and output samples are fed into the model, after which the parameters of the models are iteratively updated with the goal of minimizing the error between the actual output labels given by the data and the likelihood of the output labels predicted by the model. Ultimately, the architecture of the LSTM model is responsible for its ability to take into account sequential data in contrast to the widely used feed forward neural networks.

An additional advantage of LSTM models over traditional classifiers when modeling sequential data is the fact that LSTMs can easily handle differing sequence lengths. Through so-called masking layers, one can embed sequences with a specific value that causes the LSTM to skip over those sequences. For this application, some students registered at a later date for the courses than other students. Before the input data was passed through the LSTM, the shorter sequences were padded by replacing the missing values resulting from this late registration by a '-1' value. The masking layer recognizes these values and makes sure the LSTM does not take these values into account while training. Traditional machine learning models cannot account for these shorter sequence lengths and often are incapable of processing missing values. Therefore, the least damaging option for these traditional classifiers for this application is to impute the missing values with '0' values, as students that were not registered yet did not perform any clicks in the VLE and did not make any assessments yet. The capability of LSTMs to account for differing sequence lengths could prove to be beneficial for student performance prediction.

Which performance measure is the most appropriate to assess the models' performance depends to a large extent on the interpretation of the problem and context. Accuracy is an important and

intuitive metric as it measures the proportion of correct predictions to the total number of predictions. However, since there are disproportionally more students that passed the courses than failed, accuracy might not reflect the models' effectiveness properly. For example, if the model would predict that every student would pass the course, the accuracy would be high whereas the model would be virtually useless. In this context, the recall performance metric is therefore at least equally as important as accuracy. A model's recall measure computes the proportion of students that are correctly predicted to fail the course to the total number of students that actually fail. In other words, a model's recall measure illustrates how well the models recognize students at risk. Recall is also known as the true positive rate or sensitivity. Formally,

$$Accuracy = \frac{\sum True\ positives + \sum True\ negatives}{\sum Total\ population}$$

$$Recall = \frac{\sum True\ positives}{\sum True\ positives + \sum False\ negatives}$$

where the term positives refers to the students who actually fail, negatives refers to the students who actually pass, while true means the prediction was correct and false indicates that the prediction was incorrect. Both of these metrics are used to provide a balanced picture of the models' performance. Table 1 below illustrates the benefits and costs associated with various combinations of recall and accuracy.

TABLE 1: RECALL AND ACCURACY IMPLICATIONS

	Low accuracy	High accuracy
Low recall	**Benefits**: None **Costs**: Costs from misclassifying both well-performing and at-risk students	**Benefits**: Gains from identifying well-performing students correctly **Costs**: Costs from misclassifying at-risk students
High recall	**Benefits**: Gains from identifying at-risk students **Costs**: Costs from misclassifying well-performing students	**Benefits**: Gains from identifying both well-performing and at-risk students correctly **Costs**: None

6. Results

This section evaluates the results of the developed model on a course by course basis and will be guided be the three research questions developed previously:

1. How well can the LSTM network predict student performance?
2. How well can the traditional data mining methods predict student performance?
3. Does the LSTM network outperform the traditional data mining methods in terms of predicting student performance?

These questions serve as a means to assess the validity of the paper's hypothesis which states that LSTM networks perform better when it comes to predicting student performance than traditional machine learning methods. As previously mentioned, course BBB's structure changed in its test set, while course DDD's course structure changed in its training set and course FFF's structure remained the same. The predictions produced by the models will therefore not only indicate which model performs the best, but also what the effect of structural changes in a course will be on model performance. The LSTM's confusion matrices for every prediction made are displayed in appendix C.

Figure 5 as well as table 2 illustrate the accuracy and recall levels of the models for course BBB. A general increase in the performance of most models over time can be identified as expected. When the model is trained with only pre-course data, the LSTM, the logistic regression, the RF, the SVM and the MLP predict that all the students will pass since this is the majority class. Because 76% of the students in the test set pass the course, the accuracy of the mentioned models at decile 0 is 76%. However, the models cannot identify the students that fail at this early stage. Therefore, the recall ends up being 0 and the predictions are not valuable even though the accuracy is solid. The decision tree does not perform much better with a recall score of 0.03. Surprisingly, the NB classifier correctly identifies 98% of the students that end up failing the course before the course has even started. However, the accuracy of the NB classifier is merely 26% at this point, as the classifier predicts that almost all students will fail the course, while the majority ends up passing. The accuracy of the LSTM models drops to around 60% in subsequent periods as a larger proportion of at-risk students is correctly identified causing the recall score to increase. It is not until the halfway mark of the course until the accuracy of the LSTM model starts to improve. The highest performance of the LSTM model is reached at decile 8 and 9 when 86% of all its predictions are correct. The LSTM's recall score rises to 88% by the end of the course. Even though the random forest and decision tree are very good at identifying at-risk in the second half of the course, the accuracy remains around a relatively mediocre 60%. The discrepancy can be attributed to the fact that the RF and DT predict that an overly large group of students is going to fail the course, while they pass in reality. The logistic regression model befalls a similar fate with an accuracy of approximately 70% in the last three deciles and a recall score of close to 80%. The opposite is true for the SVM, the NB and the MLP. The accuracy of those models is competitive with that of the LSTM as they reach accuracy scores around 85% for the last three deciles. Their recall on the other hand is not. The MLP's recall is closest, but still approximately 10 percentage points lower than that of the LSTM. Both the SVM and the NB get stuck with a recall of approximately 50%, meaning that only half of the students that are going to fail the course are actually predicted to fail by these two models. As a result, the LSTM seems to be the most consistent model thanks to having both a strong recall and a strong accuracy score in the second half of the course. The higher prediction accuracy some other models possess comes at the cost of a significantly lower recall measure and

vice versa. The performance of the LSTM is promising considering the fact that the structure of course BBB changed in the fourth course presentation which was used as test set.

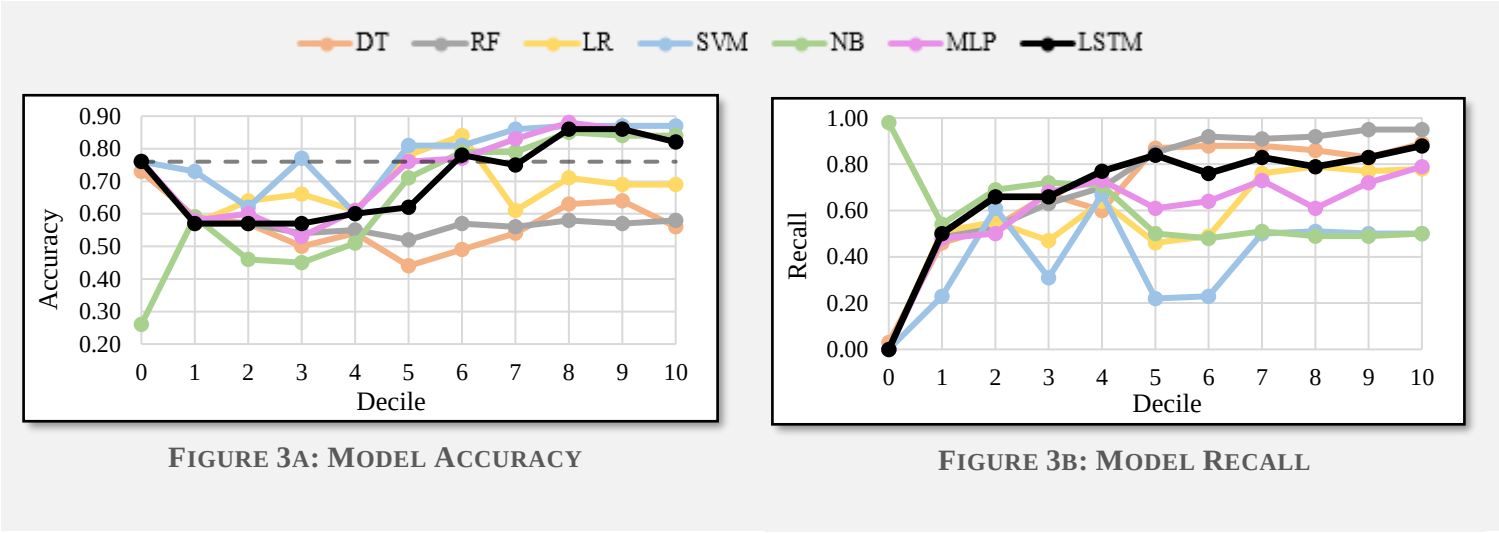


FIGURE 3A: MODEL ACCURACY

FIGURE 3B: MODEL RECALL

FIGURE 7: COURSE BBB MODEL PERFORMANCE

TABLE 2: COURSE BBB MODEL PERFORMANCE

	Accuracy							Recall						
	LSTM	**DT**	**RF**	**LR**	**SVM**	**NB**	**MLP**	**LSTM**	**DT**	**RF**	**LR**	**SVM**	**NB**	**MLP**
Decile 0 (Pre-course)	0.76	0.73	0.76	0.76	0.76	0.26	0.76	0.00	0.03	0.00	0.00	0.00	0.98	0.00
Decile 1	0.57	0.59	0.57	0.57	0.73	0.59	0.58	0.50	0.46	0.50	0.51	0.23	0.54	0.48
Decile 2	0.57	0.57	0.57	0.64	0.62	0.46	0.60	0.66	0.53	0.53	0.55	0.61	0.69	0.50
Decile 3	0.57	0.50	0.54	0.66	0.77	0.45	0.53	0.66	0.67	0.63	0.47	0.31	0.72	0.68
Decile 4	0.60	0.54	0.55	0.61	0.60	0.51	0.61	0.77	0.60	0.70	0.64	0.67	0.71	0.73
Decile 5	0.62	0.44	0.52	0.78	0.81	0.71	0.76	0.84	0.87	0.85	0.46	0.22	0.50	0.61
Decile 6	0.78	0.49	0.57	0.84	0.81	0.79	0.77	0.76	0.88	0.92	0.49	0.23	0.48	0.64
Decile 7	0.75	0.54	0.56	0.61	0.86	0.79	0.83	0.83	0.88	0.91	0.76	0.50	0.51	0.73
Decile 8	0.86	0.63	0.58	0.71	0.87	0.85	0.88	0.79	0.86	0.92	0.79	0.51	0.49	0.61
Decile 9	0.86	0.64	0.57	0.69	0.87	0.84	0.86	0.83	0.83	0.95	0.77	0.50	0.49	0.72
Decile 10	0.82	0.56	0.58	0.69	0.87	0.84	0.82	0.88	0.89	0.95	0.78	0.50	0.50	0.79

The performance of the models in course DDD is displayed in figure 6 and table 3 below. Overall, the models' accuracy scores appear to be relatively competitive as opposed to the models' accuracy percentages in course BBB. Every model predicted student performance with increasingly more

accuracy over time. The LSTM starts out with an accuracy of 71% before the course starts, but the ability of the model to identify failing students is not up to par. During the second half of the course, accuracy rises to 87% while the recall oscillates between 60 and 75%. The LSTM became better at identifying at-risk students as the course progressed, but not as well as in course BBB. The recall value of 73% approximately 27 days before the final examination deadline is still a relatively good result. However, the fact that the performance one decile before was more than 10 percentage points lower is discouraging. The accuracy remained high in the second half of the course however, with a steady increase from 85% in decile 7, almost 80 days before the final exam, to 87% one day before the exam. On its own, the LSTM definitely appears to be an effective measure to predict student performance. However, the results of other models appear to be more competitive than in course BBB. The RF and logistic regression accuracies seem to be closest to that of the LSTM network. While the logistic regression performs relatively worse in terms of recall, the RF and DT have a recall that is similar to that of the LSTM in the second half of the course. Therefore, it LSTM and RF could be considered as the best performing models for course DDD. Even though course DDD's structure changed after the first course presentation, the LSTM is still able to predict student performance with 87% by the end of the course. Its recall is considerably lower than for course BBB however. The performance of the DT, NB and logistic regression models do not compare favorably to that of the other models either because of lower accuracy or recall levels. Even though the LSTM turns out to be one of the most effective models, the results for this course do not necessarily confirm that the model is unequivocally superior to the traditional classifiers.

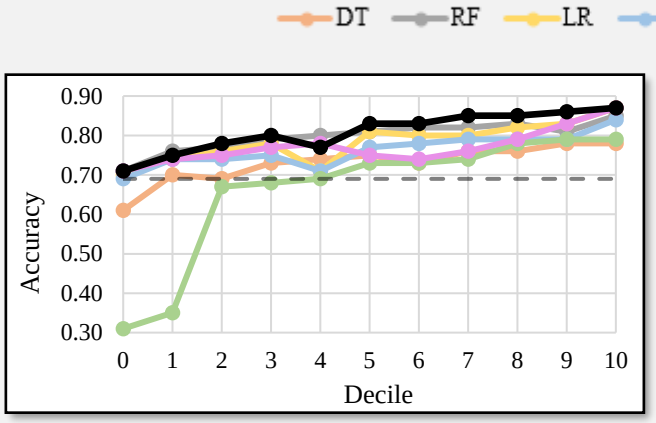


FIGURE 4A: MODEL ACCURACY

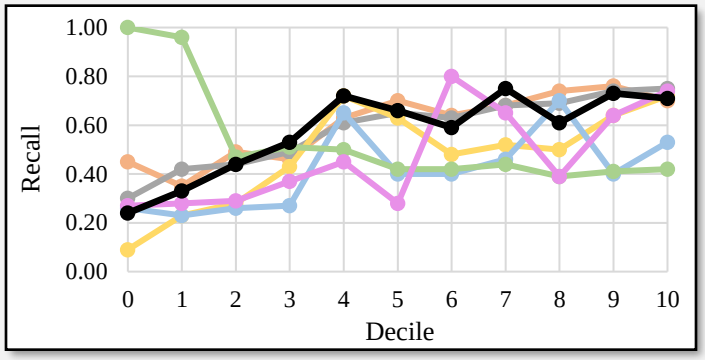


FIGURE 4B: MODEL RECALL

FIGURE 8: COURSE DDD MODEL PERFORMANCE

TABLE 3: COURSE DDD MODEL PERFORMANCE

	Accuracy							Recall						
	LSTM	**DT**	**RF**	**LR**	**SVM**	**NB**	**MLP**	**LSTM**	**DT**	**RF**	**LR**	**SVM**	**NB**	**MLP**
Decile 0 (Pre-course)	0.71	0.61	0.71	0.70	0.69	0.31	0.71	0.24	0.45	0.30	0.09	0.26	1.00	0.27
Decile 1	0.75	0.70	0.76	0.74	0.74	0.35	0.74	0.33	0.35	0.42	0.23	0.23	0.96	0.28
Decile 2	0.78	0.69	0.77	0.76	0.74	0.67	0.75	0.44	0.49	0.44	0.28	0.26	0.47	0.29
Decile 3	0.80	0.73	0.79	0.78	0.75	0.68	0.77	0.53	0.46	0.49	0.43	0.27	0.51	0.37
Decile 4	0.77	0.74	0.80	0.71	0.71	0.69	0.78	0.72	0.63	0.61	0.72	0.65	0.50	0.45
Decile 5	0.83	0.75	0.81	0.81	0.77	0.73	0.75	0.66	0.70	0.65	0.63	0.40	0.42	0.28
Decile 6	0.83	0.73	0.82	0.80	0.78	0.73	0.74	0.59	0.64	0.63	0.48	0.40	0.42	0.80
Decile 7	0.85	0.76	0.82	0.80	0.79	0.74	0.76	0.75	0.68	0.68	0.52	0.46	0.44	0.65
Decile 8	0.85	0.76	0.83	0.82	0.79	0.78	0.79	0.61	0.74	0.69	0.50	0.70	0.39	0.39
Decile 9	0.86	0.78	0.81	0.83	0.79	0.79	0.83	0.73	0.76	0.74	0.64	0.40	0.41	0.64
Decile 10	0.87	0.78	0.85	0.87	0.84	0.79	0.87	0.71	0.70	0.75	0.72	0.53	0.42	0.74

Course FFF can be considered as the benchmark course, as it did not experience any structural changes over the four course presentations. Therefore, it is not unusual that figure 7 and table 4 illustrate that the overall accuracy levels are relatively higher for course FFF than for the other two courses. The LSTM manages to achieve the highest accuracy in the last two periods with 92% and 94% correct predictions in the final two deciles of the course. Combined with a recall measure of 81% and 83% respectively, the LSTM appears to perform remarkably well in terms of predicting student performance and identifying at-risk students. Most other models achieve a competitive

accuracy level in the second half of the course as well. In fact, after decile 5 not a single model scores below 80% accuracy. It is not until decile 7 or 8 that most models start to consistently identify at-risk students correctly more than 80% of the time. Besides the LSTM, the MLP, RF, and SVM attained a solid balance between the accuracy and recall scores as well. In the final two deciles, the MLP outperforms the LSTM in terms of recall with percentages of 84% and 89%. Nevertheless, the accuracy of the LSTM remains higher. Based on these results, the MLP identifies more at-risk students correctly, while the LSTM identifies more well performing students which results in less misclassifications of well performing students as at-risk students. Consequently, the LSTM cannot simply be regarded as the optimal model for this course. Instead, the optimal model depends on the benefit/cost ratio that the educational institution places on correctly classifying at-risk students versus misclassifying well performing students as at-risk. Nonetheless, the LSTM is again among the most effective models.

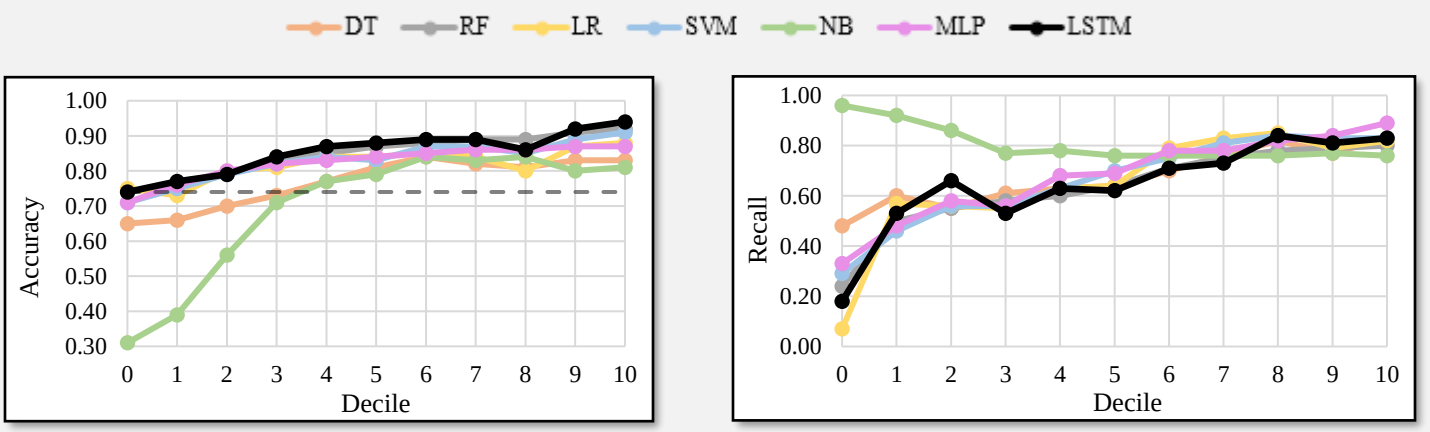


FIGURE 5A: MODEL ACCURACY

FIGURE 5B: MODEL RECALL

FIGURE 9: COURSE FFF MODEL PERFORMANCE

TABLE 4: COURSE FFF MODEL PERFORMANCE

	Accuracy							Recall						
	LSTM	**DT**	**RF**	**LR**	**SVM**	**NB**	**MLP**	**LSTM**	**DT**	**RF**	**LR**	**SVM**	**NB**	**MLP**
Decile 0 (Pre-course)	0.74	0.65	0.74	0.75	0.71	0.31	0.71	0.18	0.48	0.24	0.07	0.29	0.96	0.33
Decile 1	0.77	0.66	0.76	0.73	0.75	0.39	0.76	0.53	0.60	0.50	0.57	0.46	0.92	0.48
Decile 2	0.79	0.70	0.79	0.80	0.79	0.56	0.80	0.66	0.55	0.55	0.56	0.56	0.86	0.58
Decile 3	0.84	0.73	0.83	0.81	0.82	0.71	0.82	0.53	0.61	0.58	0.55	0.56	0.77	0.56
Decile 4	0.87	0.77	0.85	0.84	0.84	0.77	0.83	0.63	0.63	0.60	0.63	0.63	0.78	0.68
Decile 5	0.88	0.81	0.87	0.84	0.83	0.79	0.84	0.62	0.64	0.64	0.64	0.70	0.76	0.69
Decile 6	0.89	0.84	0.88	0.85	0.87	0.84	0.85	0.71	0.70	0.71	0.79	0.75	0.76	0.78
Decile 7	0.89	0.82	0.89	0.84	0.87	0.83	0.86	0.73	0.76	0.75	0.83	0.81	0.76	0.78
Decile 8	0.86	0.81	0.89	0.80	0.85	0.84	0.86	0.84	0.82	0.78	0.85	0.84	0.76	0.82
Decile 9	0.92	0.83	0.91	0.87	0.89	0.80	0.87	0.81	0.77	0.79	0.79	0.83	0.77	0.84
Decile 10	0.94	0.83	0.92	0.88	0.91	0.81	0.87	0.83	0.83	0.80	0.82	0.83	0.76	0.89

The fact that the LSTM performs well in all three courses compared to the other models speaks to its effectiveness when it comes to predicting student performance. In course BBB, the LSTM has the strongest balance between accuracy and recall and can therefore be regarded as the most effective. In course DDD and FFF some of the other models caught up to the LSTM in terms of performance. However, by performing well in all three courses, the LSTM has shown that it can predict student performance adequately regardless of the structural changes that take place in courses. Therefore, the hypothesis that long short-term memory networks perform better when it comes to predicting student performance than traditional machine learning methods is partly supported by the results. Even though the LSTMs are not unequivocally superior in every occasion, the LSTM always delivers a competitive performance and is more consistent across courses than the traditional classifiers.

7. Discussion

The fact that the LSTM is more consistent across courses and has competitive performance relative to the most widely used machine learning algorithms as identified in the literature review, makes it clear that it is relevant to study the merit of deep learning in an educational context. Moreover,

finding that the LSTM is the best performing model in the course that endured structural changes in the latest course presentation also points to the potential benefits of LSTMs. In fact, it could indicate that the LSTM is capable of identifying a more fundamental relationship between the input and output variables, which is not as affected by structural changes, by taking the sequential nature of the data into account. This paper corroborates efforts by Guo et al. (2015), Okubo et al. (2017) and Liu et al. (2018) that show the potential of deep learning methods in educational data mining. Literature on student performance prediction using deep learning is still scarce however. This paper is an attempt to fill that gap by conducting one of the first comparisons between LSTMs and multiple popular machine learning algorithms for academic performance prediction. The promising results are encouraging for future research in the field as well as implementations of LSTMs in educational software systems with the purpose of predicting student performance.

Based on the results, LSTMs could prove to be a viable alternative to traditional machine learning models for systems dependent on student performance prediction such as the one at Purdue University (Arnold & Pistilli, 2012). The models' balance between accuracy and recall measures should be one of the elements educational institutions should consider when implementing such a system. As stated in the previous section, the LSTM tends to have a higher accuracy than recall score, especially in courses DDD and FFF. Meanwhile, some traditional models showed a higher recall score than the LSTMs but a lower overall accuracy. The educational institution should determine the expected costs of misclassifying well-performing and at-risk students as well as the expected benefits of correctly classifying well-performing and at-risk students as summarized in table 1. For example, detecting the at-risk students can be valuable for pre-emptive intervention and providing assistance to those particular students. Identifying these at-risk student correctly could raise the number of passing students and increase the retention rate (Arnold & Pistilli, 2012). Meanwhile, misclassifying well-performing students as at-risk students could very well cost the institution additional resources better spent elsewhere. Determining the expected value of each classification is therefore vital in order to make a well informed decision on which model to implement in the system. More research is necessary in order to assess whether LSTMs always tend to favor accuracy over recall to a higher extent than other models, because the literature review demonstrated that the model results strongly depend on the data used and the context. Educational institutions should also keep in mind that deep learning methods are widely regarded as black box

algorithms. In other words, it is difficult to dissect the relationship between the particular input and output variables through deep learning models. If this is important to educational institutions, more transparent models such as decision trees are recommended. Additionally, the results show that it is important to consider the impact of structural course changes on the performance of the models. The majority of the models' perform better in the context of course FFF, which experienced no structural changes as opposed to courses BBB and DDD, which both experienced changes in terms of their assessments. Therefore, educational institutions should take into account that the accuracy of their predictive models is likely to suffer when changing a course's structure because of the discrepancy that arises between the datasets. The LSTMs consistent performance across courses suggests that the LSTM might be well suited to deal with these type of structural changes.

One limitation could be found in the notion that only four course presentations were available per course. Deep learning methods are well known for their capability to process large volumes of data. Therefore, more periods to train the model on could paint a more comprehensive picture of the LSTM's potential. In addition, the results of the study might not be generalizable to non-distance learning organizations, as those are less likely to have a strong emphasis on learning through their online platform. Clickstreams might therefore not be as good of a proxy for student effort or they might not even be collected at all. Consequently, it is important to conduct further research on the generalizability of the LSTM's promising results in a non-distance learning educational institution. Moreover, the anonymization of the course titles could be considered as a limitation as the results might not be generalizable across different fields of education. However, the fact that the analysis has been performed on three different courses mitigates this concern to a reasonable degree. The limited computational power available made it infeasible to conduct an exhaustive grid search during the hyperparameter optimization. There exists a chance that more optimal parameter settings are available, which could increase model performance. Furthermore, the binary target variable limits the options available to detect very high performing students such as those who passed with distinction. Additional research could shed light on the potential of LSTMs when predicting a target variable with multiple labels. LSTMs can even be used for regression tasks, as such they can be applied to predict grades. In terms of further research, researchers could also analyze whether performance can be increased by adding other variables such as demographic or psychometric factors. The addition of results in prior courses could also possess predictive power valuable for

performance prediction. Additional features that are time invariant can be added to the LSTM by means of an auxiliary input layer. Because the three analyzed course presentations have relatively long durations, future research is necessary to determine whether the results also hold for shorter courses. The RNN in Okubo et al. (2017) also experiences a substantial rise in accuracy from 71% to almost 85% even though the course only takes eight weeks. These results suggests that the course duration does not have to be a limiting factor for performance prediction. However, Okubo et al. (2017) made use of more input features such as attendance and more granular clickstream data, which might have contributed to this rise in accuracy. As demonstrated, there still are a lot of opportunities for future research considering the novelty of deep learning in an educational data mining context.

8. Conclusion

In conclusion, the massive rise in data mining has had big implications in all walks of life, education included. In the fields of learning analytics and educational data mining that originated from this phenomenon, a lot of time and effort has been spent on predicting student performance by applying machine learning techniques. A myriad of studies have been conducted in an attempt to find out which machine learning algorithm is the most suited to predict student performance. On top of that, educational institutions have implemented systems that depend on machine learning models to predict academic performance. Even though deep learning has been widely regarded as the state-of-the art in the field of machine learning for numerous years now, it has rarely been applied to predict student performance. This study showed that deep learning in the context of academic performance prediction has merit by comparing LSTMs to traditional machine learning methods. Data mining models were used to predict whether students enrolled in three different courses of the Open University would pass or fail the course. Since the Open University is a distance learning institution, clickstream data was included as input in the models as well as the amount of assessment submitted and the average assessment score. The LSTMs showed to have competitive performance consistent across three courses. The paper hypothesized that LSTMs perform better when it comes to predicting student performance than traditional machine learning methods due to the fact that LSTMs can account for the time dimension in sequential data as opposed to traditional

machine learning algorithms that only take features and samples as inputs. Even though LSTMs did not appear unequivocally superior to popular machine learning methods, the results illustrated that LSTMs have the potential to accurately predict student performance and detect at-risk students. Additional research is necessary to fully understand and capture the benefits that deep learning can bring in an educational context.

Reference List

Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R., & Honrao, V. (2013). Predicting Students' Performance Using ID3 and C4.5 Classification Algorithms. *International Journal of Data Mining & Knowledge Management Process, 3*(5), 39 – 52.

Arnold, K.E., & Pistilli, M.D. (2012, April 29 – May 02). *Course signals at Purdue: Using learning analytics to increase student success*. Paper presented at the 2nd International Conference on Learning Analytics and Knowledge, Vancouver, BC, Canada. doi: 10.1145/2330601.2330666

Asif, R., Merceron, A., Ali, S.A., & Haider, N.G. (2017). Analyzing undergraduate students' performance using educational data mining. *Computers & Education, 113*, 177-194. doi: 10.1016/j.compedu.2017.05.007

Ayán, M. N. R., & García, M.T.C. (2008). Prediction of University Students' Academic Achievement by Linear and Logistic Models. *The Spanish Journal of Psychology, 11*(1), 275 - 288. doi: 10.1017/S1138741600004315

Baker, R.S.J.D. (2010). Data Mining for Education. To appear in McGaw, B., Peterson, P., Baker, E. (Eds.) *International Encyclopedia of Education (3rd edition).* Oxford, UK: Elsevier.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (2000). *CRISP-DM 1.0 Step-by-step data mining guide*. The CRISP-DM consortium.

Coelho, O.B., & Silveira, I.F. (2017, October). *Deep Learning applied to Learning Analytics and Educational Data Mining: A Systematic Literature Review*. Paper presented at the 28th Brazilian Symposium on Computers in Education. doi: 10.5753/cbie.sbie.2017.143

Cortez, P., & Silva, A. M. G. (2008). Using data mining to predict secondary school student performance. In Brito, A.C. & Teixeira, J. M. F (Eds.), *Proceedings of 5th Future Business Technology Conference*. (pp. 5-12). Porto: Eurosis.

Del Río, C. A., & Insuasti, J. A. P. (2016). Predicting academic performance in traditional environments at higher-education institutions using data mining: A review. *Ecos de la Academia, 4*, 185 – 201.

Delgado Calvo-Flores, M., Gibaja Galindo, E., Pegalajar Jiménez, M.C., & Pérez Piñeiro, O. (2006) Predicting students' marks from Moodle logs using neural network models. *Current Developments in Technology-Assisted Education*, 586-590.

Gers, F.A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation, 12*(10), 2451-2471. doi: 10.1162/089976600300015015

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. Retrieved from: http://www.deeplearningbook.org/

Gray, G., McGuinness, C., & Owende, P. (2014, February 21-22). *An application of classification models to predict learner progression in tertiary education*. Presented at the 2014 IEEE International Advance Computing Conference, Gurgaon, India. doi:10.1109/IAdCC.2014.6779384

Guo, B., Zhang, R., Xu, G., Shi, C., & Yang, L. (2015, July 27-29). *Predicting Students Performance in Educational Data Mining*. Paper presented at 2015 International Symposium on Educational Technology (ISET), Wuhan, China. doi:10.1109/iset.2015.33

Hlosta, M., Zdrahal, Z., & Zendulka, J. (2017, March 13-17). *Ouroboros: Early identification of at-risk students without models based on legacy data*. Presented at the Seventh International Learning Analytics & Knowledge Conference, Vancouver, BC, Canada. doi: 10.1145/3027385.3027449

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Jayaprakash, S.M., Moody, E.W., Lauría, E.J.M., Regan, J.R., & Baron, J.D. (2014). Early Alert of Academically At-Risk Students: An Open Source Analytics Initiative. *Journal of Learning Analytics, 1*(1), 6-47. doi: 10.18608/jla.2014.11.3

Kabakchieva, D. (2012). Student Performance Prediction by Using Data Mining Classification Algorithms. *International Journal of Computer Science and Management Research, 1*(4), 686-690.

Kotsiantis S., Pierrakeas C., and Pintelas P. (2004). Predicting Students' Performance in Distance Learning Using Machine Learning Techniques. *Applied Artificial Intelligence, 18*(5), 411-426. doi: 10.1080/08839510490442058

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. New York, NY: Springer

Kuzilek, J., Hlosta, M., & Zdrahal, Z. (2017). Open University Learning Analytics dataset. *Sci. Data*, 4:170171. doi: 10.1038/sdata.2017.171

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. *Nature, 521*(7553), 436-444 doi: 10.1038/nature14539

Liu, Z., Xiong, F., Zou, K., & Wang, H. (2018). *Predicting Learning Status in MOOCs using LSTM*. Retrieved from: https://arxiv.org/abs/1808.01616

Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G., & Punch, W.F. (2003, November 5-8). *Predicting student performance: An application of data mining methods with an educational web-based system*. Paper presented at the 33rd Annual Frontiers in Education, Westminster, United States. doi: 10.1109/FIE.2003.1263284

Mondal, A., & Mukherjee, J. (2018). An Approach to Predict a Student's Academic Performance using Recurrent Neural Network (RNN). *International Journal of Computer Applications, 181*(6), 1-5.

Ogunde, A. O., & Abjibade, D. A. (2014). A Data Mining System for Predicting University Students' Graduation Grades Using ID3 Decision Tree Algorithm. *Computer Science and Information Technology*, 2(1), 1-26.

Okubo, F., Yamashita, T., Shimada, A., & Konomi, S. (2017). *Students' Performance Prediction Using Data of Multiple Courses by Recurrent Neural Network*. Presented at the 25th International Conference on Computers in Education, New Zealand.

Oladokun, V.O., Adebanjo, A.T., & Charles-Owaba, O.E. (2008). Predicting Students' Academic Performance using Artificial Neural Network: A Case Study of an Engineering Course. The Pacific Journal of Science and Technology, 9(1), 72-79.

Olah, C. (2015). Understanding LSTM Networks [Online image]. Retrieved from: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Osmanbegović, E., & Suljić, M. (2012). Data Mining Approach for Predicting Student Performance. *Journal of Economics and Business, 10*(1), 3-12.

Pandey, M., & Taruna, S. (2016). Towards the integration of multiple classifier pertaining to the Student's performance prediction. *Perspectives in Science, 8*, 364-366. doi: 10.1016/j.pisc.2016.04.076

Romero, C., & Ventura, S. (2010). Education Data Mining: A Review of the State-of-the-Art. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40*(6), 601-618. doi: 10.1109/TSMCC.2010.2053532

Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. *Neural Networks, 61*, 85-117. doi: 10.1016/j.neunet.2014.09.003

Yadav, S. K., & Pal, S. (2012). Data Mining: A Prediction for Performance Improvement of Engineering Students using Classification. *World of Computer Science and Information Technology Journal, 2*(2), 51 – 56.

Appendix A: Course structures

Course BBB Assignments:

2013B Asgmts	Date	Weight	2013J Asgmts	Date	Weight	2014B Asgmts	Date	Weight	2014J Asgmts	Date	Weight
TMA 1	19	5	**TMA 1**	19	5	**TMA 1**	12	5	**TMA 1**	19	0
TMA 2	47	18	**TMA 2**	47	18	**TMA 2**	40	18	**TMA 2**	54	10
CMA 1	54	1	**CMA 1**	54	1	**CMA 1**	47	1	**TMA 3**	110	20
TMA 3	89	18	**TMA 3**	96	18	**TMA 3**	82	18	**TMA 4**	152	35
CMA 2	89	1	**CMA 2**	96	1	**CMA 2**	82	1	**TMA 5**	201	35
TMA 4	124	18	**TMA 4**	131	18	**TMA 4**	117	18	**Exam**	NA	100
CMA 3	124	1	**CMA 3**	131	1	**CMA 3**	117	1	**Duration**	262 days	-
TMA 5	159	18	**TMA 5**	166	18	**TMA 5**	152	18			
CMA 4	159	1	**CMA 4**	166	1	**CMA 4**	152	1			
TMA 6	187	18	**TMA 6**	208	18	**TMA 6**	194	18			
CMA 5	187	1	**CMA 5**	208	1	**CMA 5**	194	1			
Exam	NA	100	**Exam**	NA	100	**Exam**	NA	100			
Duration	240 days	-	**Duration**	268 days	-	**Duration**	234 days	-			

CMA = Computer Marked Assessment
TMA = Teacher Marked Assessment

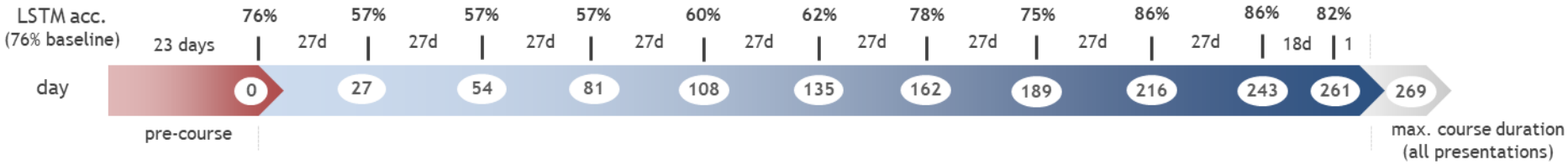
Course DDD Assignments:

2013B Asgmts	Date	Weight	2013J Asgmts	Date	Weight	2014B Asgmts	Date	Weight	2014J Asgmts	Date	Weight
CMA 1	23	2	**TMA 1**	25	10	**TMA 1**	25	10	**TMA 1**	20	5
TMA 1	25	7.5	**TMA 2**	53	12.5	**TMA 2**	53	12.5	**TMA 2**	41	10
CMA 2	51	3	**TMA 3**	88	17.5	**TMA 3**	74	17.5	**TMA 3**	62	10
TMA 2	53	10	**TMA 4**	123	20	**TMA 4**	116	20	**TMA 4**	111	25
CMA 3	79	3	**TMA 5**	165	20	**TMA 5**	158	20	**TMA 5**	146	25
TMA 3	81	12.5	**TMA 6**	207	20	**TMA 6**	200	20	**TMA 6**	195	25
CMA 4	114	4	**Exam**	261	100	**Exam**	241	100	**Exam**	NA	100
TMA 4	116	15	**Duration**	261 days	-	**Duration**	241 days	-	**Duration**	262 days	-
CMA 5	149	4									
TMA 5	151	15									
CMA 6	170	3									
TMA 6	200	15									
CMA 7	206	6									
Exam	240	100									
Duration	240 days	-									

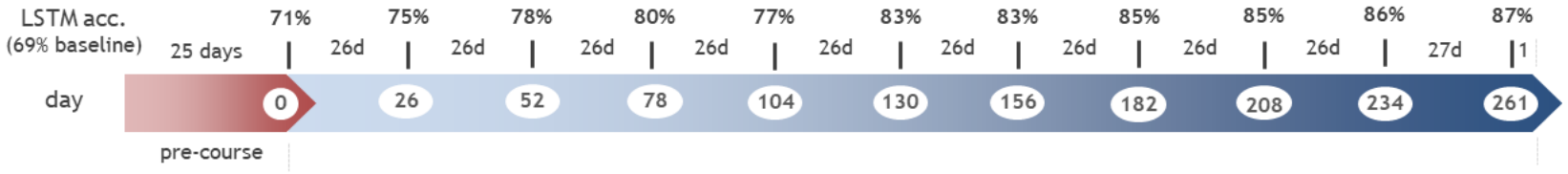
Course FFF Assignments

2013B Asgmts	Date	Weight	2013J Asgmts	Date	Weight	2014B Asgmts	Date	Weight	2014J Asgmts	Date	Weight
TMA 1	19	12.5	**TMA 1**	19	12.5	**TMA 1**	24	12.5	**TMA 1**	24	12.5
TMA 2	47	12.5	**TMA 2**	47	12.5	**TMA 2**	52	12.5	**TMA 2**	52	12.5
TMA 3	89	25	**TMA 3**	96	25	**TMA 3**	87	25	**TMA 3**	94	25
TMA 4	131	25	**TMA 4**	131	25	**TMA 4**	129	25	**TMA 4**	136	25
TMA 5	166	25	**TMA 5**	173	25	**TMA 5**	171	25	**TMA 5**	199	25
CMA 1	222	0	**CMA 1**	236	0	**CMA 1**	227	0	**CMA 1**	241	0
CMA 2	222	0	**CMA 2**	236	0	**CMA 2**	227	0	**CMA 2**	241	0
CMA 3	222	0	**CMA 3**	236	0	**CMA 3**	227	0	**CMA 3**	241	0
CMA 4	222	0	**CMA 4**	236	0	**CMA 4**	227	0	**CMA 4**	241	0
CMA 5	222	0	**CMA 5**	236	0	**CMA 5**	227	0	**CMA 5**	241	0
CMA 6	222	0	**CMA 6**	236	0	**CMA 6**	227	0	**CMA 6**	241	0
CMA 7	222	0	**CMA 7**	236	0	**CMA 7**	227	0	**CMA 7**	241	0
Exam	222	100	**Exam**	236	100	**Exam**	227	100	**Exam**	241	100
Durat-ion	240 days	-	**Durat-ion**	268 days	-	**Durat-ion**	241 days	-	**Durat-ion**	269 days	-

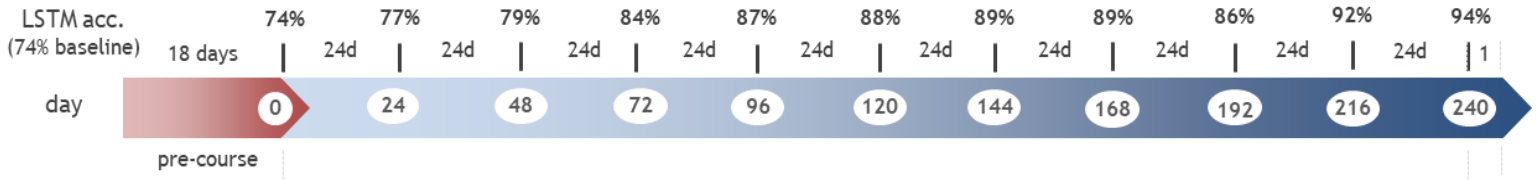
Appendix B: Decile breakdown with LSTM accuracy



Course BBB presentation 2014J
262 days



Course DDD presentation 2014J
262 days



Course FFF presentation 2014J
241 days

Appendix C: LSTM confusion matrices

Course BBB Decile 0	Predicted: Fail	Predicted: Pass
Actual: Fail	1	368
Actual : Pass	0	1152

Course BBB Decile1	Predicted: Fail	Predicted: Pass
Actual: Fail	185	184
Actual : Pass	471	681

Course BBB Decile 2	Predicted: Fail	Predicted: Pass
Actual: Fail	242	127
Actual : Pass	522	630

Course BBB Decile 3	Predicted: Fail	Predicted: Pass
Actual: Fail	243	126
Actual : Pass	525	627

Course BBB Decile 4	Predicted: Fail	Predicted: Pass
Actual: Fail	285	84
Actual : Pass	531	621

Course BBB Decile 5	Predicted: Fail	Predicted: Pass
Actual: Fail	311	58
Actual : Pass	520	632

Course BBB Decile 6	Predicted: Fail	Predicted: Pass
Actual: Fail	282	87
Actual : Pass	253	899

Course BBB Decile 7	Predicted: Fail	Predicted: Pass
Actual: Fail	308	61
Actual : Pass	320	832

Course BBB Decile 8	Predicted: Fail	Predicted: Pass
Actual: Fail	293	76
Actual : Pass	134	1018

Course BBB Decile 9	Predicted: Fail	Predicted: Pass
Actual: Fail	307	62
Actual : Pass	149	1003

Course BBB Decile 10	Predicted: Fail	Predicted: Pass
Actual: Fail	324	45
Actual : Pass	236	916

Course DDD Decile 0	Predicted: Fail	Predicted: Pass
Actual: Fail	85	273
Actual : Pass	60	732

Course DDD Decile 1	Predicted: Fail	Predicted: Pass
Actual: Fail	119	239
Actual : Pass	48	744

Course DDD Decile 2	Predicted: Fail	Predicted: Pass
Actual: Fail	156	202
Actual : Pass	48	744

Course DDD Decile 3	Predicted: Fail	Predicted: Pass
Actual: Fail	190	168
Actual : Pass	59	733

Course DDD Decile 4	Predicted: Fail	Predicted: Pass
Actual: Fail	256	102
Actual : Pass	164	628

Course DDD Decile 5	Predicted: Fail	Predicted: Pass
Actual: Fail	237	121
Actual : Pass	70	722

Course DDD Decile 6	Predicted: Fail	Predicted: Pass
Actual: Fail	213	145
Actual : Pass	46	746

Course DDD Decile 7	Predicted: Fail	Predicted: Pass
Actual: Fail	270	88
Actual : Pass	90	702

Course DDD Decile 8	Predicted: Fail	Predicted: Pass
Actual: Fail	218	140
Actual : Pass	36	756

Course DDD Decile 9	Predicted: Fail	Predicted: Pass
Actual: Fail	261	97
Actual : Pass	68	724

Course DDD Decile 10	Predicted: Fail	Predicted: Pass
Actual: Fail	255	103
Actual : Pass	51	741

Course FFF Decile 0	Predicted: Fail	Predicted: Pass
Actual: Fail	68	319
Actual : Pass	77	1040

Course FFF Decile 1	Predicted: Fail	Predicted: Pass
Actual: Fail	206	181
Actual : Pass	172	945

Course FFF Decile 2	Predicted: Fail	Predicted: Pass
Actual: Fail	254	133
Actual : Pass	179	938

Course FFF Decile 3	Predicted: Fail	Predicted: Pass
Actual: Fail	206	181
Actual : Pass	55	1062

Course FFF Decile 4	Predicted: Fail	Predicted: Pass
Actual: Fail	243	144
Actual : Pass	59	1058

Course FFF Decile 5	Predicted: Fail	Predicted: Pass
Actual: Fail	240	147
Actual : Pass	29	1088

Course FFF Decile 6	Predicted: Fail	Predicted: Pass
Actual: Fail	274	113
Actual : Pass	52	1065

Course FFF Decile 7	Predicted: Fail	Predicted: Pass
Actual: Fail	282	105
Actual : Pass	53	1064

Course FFF Decile 8	Predicted: Fail	Predicted: Pass
Actual: Fail	324	63
Actual : Pass	141	976

Course FFF Decile 9	Predicted: Fail	Predicted: Pass
Actual: Fail	315	72
Actual : Pass	44	1073

Course FFF Decile 10	Predicted: Fail	Predicted: Pass
Actual: Fail	323	64
Actual : Pass	27	1090