Heliyon 9 (2023) e15382

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

ANN-LSTM: A deep learning model for early student performance prediction in MOOC

Fatima Ahmed Al-azazi [a,*], Mossa Ghurab [b]

[a] Information Technology Department, University of Science and Technology, Sana'a, Yemen
[b] Computer Science Department, Sana'a University, Sana'a, Yemen

ARTICLE INFO

Keywords:
Multi-class classification
Student performance prediction
Deep learning
Virtual learning environments
MOOC

ABSTRACT

Learning Analytics aims to discover the class of students' performance over time. This helps instructors make in-time interventions but, discovering the students' performance class in virtual learning environments consider a challenge due to distance constraints. Many studies, which applied to Massive Open Online Courses (MOOC) datasets, built predictive models but, these models were applied to specific courses and students and classify students into binary classes. Moreover, their results were obtained at the end of the course period thus delaying making in-time interventions. To bridge this gap, this study proposes a day-wise multi-class model to predict students' performance using Artificial Neural Network and Long Short-Term Memory, named ANN-LSTM. To check the validity of this model, two baseline models, the Recurrent Neural Network (RNN) and Gated Recurrent Unit (GRU), were conducted and compared with ANN-LSTM in this context. Additionally, the results of ANN-LSTM were compared with the state-of-the-art models in terms of accuracy. The results show that the ANN-LSTM model obtained the best results among baseline models. The accuracy obtained by ANN-LSTM was about 70% at the end of the third month of the course and outperforms RNN and GRU models which obtained 53% and 57%, respectively. Also, the ANN-LSTM model obtained the best accuracy results with enhancement rates of about 6–14% when compared with state-of-the-art models. This highlights the ability of LSTM as a time series model to make early predictions for student performance in MOOC taking benefit of its architecture and ability to keep latent dependencies.

1. Introduction

Discovering the class of student's performance in the first days of the course duration in the virtual learning environments (VLE) facilitates accomplishing Learning Analytics goals. Consequently, many researchers proposed predictive models to predict student performance earlier in Massive Open Online Courses (MOOC) courses in binary classes (pass or fail) or (dropout or not), but few studies have developed models that predict student performance in multi-classification form like [1–7]. Hence, the multi-class models need more studies to improve prediction performance [8]. As techniques used in the prediction of student performance in online higher education, traditional artificial intelligence techniques are commonly applied while more advanced techniques like deep learning are rarely applied [9]. Since the majority of studies have taken the approach of developing predictive models that target specific courses like [10–13], but overfitting can take place if new courses are devised [14]. Thus, course-agnostic predictive models which train and

* Corresponding author.
E-mail address: f.alazazi@ust.edu.ye (F.A. Al-azazi).

https://doi.org/10.1016/j.heliyon.2023.e15382
Received 22 December 2022; Received in revised form 4 April 2023; Accepted 5 April 2023
Available online 7 April 2023

evaluate with different courses are required to overcome this problem. As stated in Ref. [15], significant contributions of clickstream data in the identification of student performance. So, institutions can use clickstream data to develop real-time analytic reports of online students. As a result, they can make decisions more timely and informed. Also, clickstream data can be combined with demographic and course data and get promising results in studying student performance [16]. The aforementioned gaps motivated us to propose a developed model that overcomes them. To facilitate higher education decision-making processes towards sustainable education in the MOOC environments as well as enable instructors to take suitable in-time actions, the main aim of this study is to build a multi-class course-agnostic day-wise predictive model to identify the class of students' performance in MOOC environments as early as possible with satisfied accuracy using demographic and activity clickstream data. To achieve this study aim, the following objectives rise.

1. To develop a new day-wise deep learning model to predict students' performance in multi-class form.
2. To determine which recurrent deep learning model among RNN, LSTM, and GRU can get better students' performance prediction accuracy.
3. To compare the proposed model with similar state-of-the-art models in terms of accuracy.

The rest of this study is organized as follows: Section 2 discusses the previous works in the scope of student performance predictive models implemented in MOOC environments. The research methodology and the description of the dataset are presented in Section 3. In addition, the settings of experiments and the stages of the proposed model are described with the details of each stage. Section 4 highlights the results of the experiments. Moreover, the result of monthly prediction and comparison between the proposed model and the baseline models are shown in this Section. A discussion of the obtained results is presented in Section 5. Section 6 displays the conclusion and the future work.

2. Related works

Massive Open Online Course (MOOC) is an online course that is open to anyone with no restrictions, usually organized with a set of learning objectives in an area of study, often provided over a specific period in a virtual learning environment that allows interaction between peers or students and instructors. MOOCs facilitate creating a learning community [17]. Open University Learning Analytics Dataset (OULAD) dataset is a popular MOOC dataset used by several studies which applied performance predictive models on MOOC courses. Several predictive models were trained on the OULAD dataset and published in 2019, 2020, 2021, and 2022.

In 2019, Hassan et al. used clickstream data to predict withdrawn students every five weeks with an LSTM deep model [18]. The accuracy obtained in the 5th week was around 80% and in the 25th week was around 97%. Different machine and deep learning models (Random Forest (RF), Multiple Layered Perceptron with multiple activation functions, and Gaussian NB) were proposed in Ref. [19] to predict student performance pass and fail using demographic information, clickstream data, generated features, and Total Features.

Earlier in 2020, Yanbai et al. proposed Recurrent Neural Network (RNN)- Gated Recurrent Unit (GRU) joint neural network to predict whether students will fail or pass using demographic, clickstream, and assessment data [20]. The prediction was at the course level. So, the last courses were used as a test set. The average accuracy obtained in all courses was between 60 and 90% from the 5th week till the 39th week. The class of student performance was treated as a binary classification in Ref. [21] The prediction was at the course level. So, the last courses were used as a test set. The average accuracy obtained in all courses was between 60 and 90% from the 5th week till the 39th week. The class of student performance was treated as a binary classification and four deep ANN models were deployed to predict failed, withdrawn, and distinct students using demographic and clickstream data. The sparse reduction was used to select 30 features over a total of 54 features and the "MinMax" scalar was used to transform the values of the selected features. The prediction was performed in four quarters of the semester. The accuracy obtained in the "fail" prediction model was around 77%, 81%, 86%, and 88% in quarter1, quarter2, quarter3, and quarter4, respectively. In the "fail" prediction model withdrawal student's data was ignored and the distinction merged with passed data. The accuracy obtained in the withdrawn prediction model was around 78%, 86%, 90%, and 93% in quarter 1, quarter 2, quarter 3, and quarter 4, respectively. In the withdrawn prediction model fail student's data was ignored and the distinction merged with passed data. The accuracy obtained in the two distinction prediction models was between 80 and 81% when withdrawn and fail data was ignored and between 80 and 85% when withdrawn and pass data was ignored. A multi-class predictive model was proposed in Ref. [7]. The target classes were "pass", "fail", and "withdrawn". Assessment and clickstream features were used as input for the predictive model. Also, it developed a regression model to predict the final assessment grades.

In 2021, Adnan et al. used a Deep Feed Forward Neural Network (DFFNN) to predict a student's final result (distinct, pass, withdrawn, and fail) with the best average accuracy obtained at the last time of courses was 43% when input data were demographic data, 63% when input data was demographic and clickstream, 71% when input data was demographic, clickstream and assessment and 72% when input data was all features available in OULAD dataset [4]. In Addition, multi-classification was converted to binary classification. After this conversion, the accuracy obtained was 90% at the end of the courses. Another research published is [12]. It used demographic and aggregated clickstream data of two social science courses (AAA_2013J and AAA_2014J) and two STEM courses (CCC_2014B and CCC_2014J) as input. It developed a supervised machine learning model with an expectation maximum algorithm to predict whether a student will stay or drop out in the previous week then used the resulting probability as input to improve prediction accuracy for the current week. A Synthetic Minority Over-Sampling (SMOTE) technique was used in Ref. [12]. The average accuracy obtained for selected courses in all weeks was approximately 88%. A semi-supervised learning ensemble model of Artificial neural

network (ANN) was proposed in Ref. [13] to predict student performance (pass or fail) before midterm and at the end of the course. Five specific courses of the OULAD data set were chosen as input to the proposed model. Each chosen course was available in two different semesters. The first course was used for training while the second was used for testing. The average accuracy obtained for selected courses was 87.47% in the middle of the semester. Distinction and withdrawal of students' data were ignored in Ref. [13], which represents around 40% of the OULAD dataset. Prediction limited to the trained course. As [12], Hlioui et al. [22] predicted withdrawal students with different models (Decision tree(J48), Random Forest, Bayesian classifier (TAN), SVM classifier, and MLP) classifiers using clickstream, assessment, and demographic data. Student performance was divided into two values: withdrawal and completion ("Distinction" or "Pass" or "Fail"). The importance of assignment information for students' performance prediction was explored in Ref. [23]. It developed a Multiple Instance Learning predictive model to predict passed and failed students. Features used were assessment data. Predictive models with different classification algorithms (NB, RF, KNN SVM, and ANN) were proposed in Ref. [24] and compared the performance in predicting final exam grades. Assessment, Demographic, and Clickstream data were used as input for the prediction models. Three predictive models were proposed and compared in Ref. [25]. The performance of ANN, SVM, and ANN in binary classification (pass or fail) was compared at the end of the course using demographic, assessment, and clickstream data. In Ref. [26], predictive models were developed to classify students as withdrawal or non-withdrawal and at-risk or not at-risk using RFDT, FFNN, MLP, Gradient Boosting Machine, and LR. Two multi-class predictive models (RF and ANN) were constructed in Ref. [6] to classify students to distinction, pass and fail. Withdrawn students' data were excluded. A three-layer LSTM model was proposed in Ref. [1] to classify student performance with multi and binary classification using clickstream and demographic data. The attention technique was used in this model. The clickstream data was aggregated weekly. A binary classification target, pass or fail, was the output of the predictive models proposed in Ref. [27]. KNN, ANN, SVM, RF, and Naive Bayes (NB) were compared. Total clickstreams in all activities were calculated and input to these models besides assessments and demographic data. Multi-classification predictive models proposed in Ref. [5]. ANN, SVM, RF, and Naive Bayes were compared. Feature selection algorithms were applied to demographic and clickstream data before inputting them into the selected models.

In 2022, a prediction framework based on six traditional machine learning algorithms (SVC (R), SVC (L), Naïve Bayes , KNN (U), KNN (D), and Sofmax) was proposed in Ref. [11]. The expected output value of the predictor was" qualified" or" unqualified". The proposed framework in Ref. [11] used the "DDD" module (course) with clickstream data in 12 activities. In Ref. [2] researchers developed a Bayesian Network (BN) base prediction model for the final performance and compared it with Gradient Boosting Decision Tree, Multi-Layer Perception (MLP), and Naïve BN ensemble models. The value of calculated performance was divided, according to the student's scores in the assessments, into fail, pass, good, and distinction. As a multi-class classification model, Adnan et al. [3] developed different ML models to predict student performance with the four classes Distinct, pass, fail, and withdrawn. The compared models were Random Forest with two different criteria 'gini' and 'entropy', AdaBoost, Extra Tree classifier, K-Nearest Neighbour (KNN), Decision Tree, Support Vector Machine, Gradient Boosting, Logistic Regression, Gaussian NB, Bernoulli NB classifier, and only one DL classifier (Feed Forward Neural Network (FFNN)).

Multi-class course-agnostic student performance predictive models still need to increase the accuracy at the early time of MOOC courses in case of using demographic and clickstream data. Although the predictive models proposed in Refs. [3,4] are multi-class course-agnostic models, they are not applied in the early time during the course period. while a periodical predictive model proposed in Ref. [1] which starts the prediction for the class of student performance from the fifth week, the accuracy obtained in the fifth week was 53%. Generally, the accuracy obtained at the end of the course was 66% and 63%, for models proposed in Refs. [3,4], respectively which consider low accuracy.

Based on the discussed related works the gap of this study raised. First, models in Refs. [10–13] are not general. They are applied to specific courses. So, overfitting can take place if new courses are devised [14]. Second, most researchers proposed predictive models to predict student performance in binary classes (pass or fail) or (dropout or not), but few studies have developed models that predict student performance in multi-classification forms like [1–7]. The multi-class prediction models lead to the ability to use the model for several objectives. For example, identify students expected to drop out of the course (withdrawn students) or at risk of failing the course (fail students, and help instructors in providing personalized content for students based on their performance. Third, models in Refs. [13,18,20,28] discard some parts of the dataset. For example, discarding all withdrawal students' data can lead to overfitting. Finally, although instructors want to predict students' performance on any day through the course duration based on the current and previous clickstream behavior, most of the related works predict the class of student's performance on the last day of the course duration.

3. Research methodology

This work is in the context of predicting student performance who studies in Massive Open Online Courses provided by virtual learning environments using demographic and clickstreams data which generate during the interaction between students and virtual learning environments. The proposed model will use recurrent neural network models because they are commonly applied to time series data. This section discusses the experimental research methodology. It demonstrates the applied method in this study, the description of the used dataset, the evaluation metrics, and the proposed model architecture.

3.1. Method

This experimental study will be implemented in four phases. Each phase contains several stages. Firstly, in the dataset preparation phase, the OULAD dataset will be downloaded. Then the dataset will be prepared and aggregated in a suitable form. The data relating

to specific course duration will be selected starting from the first day of the course until reaching the last day of the course. After that the dataset will be split. In the model construction phase, the proposed model will be built, trained, and evaluated. Then, in the baseline models construction phase, two baseline models will be built, trained, and evaluated with the same dataset. Finally, in the results comparison phase, the result obtained by the proposed model and the baseline models will be compared and the next experiment with data from the next day of the course will start till reaching the last day of the course. Fig. 1 illustrates the experiment's flowchart.

3.2. Dataset description

Open University Learning Analytics Dataset (OULAD) is one of the public Learning Analytics datasets containing data for 32,593 students studied in open online courses [3]. This data was collected from the Open University, one of the largest distance-learning institutions in the United Kingdom [22]. It contains more than 10 million student interaction records The OULAD dataset contains demographic, assessment, course, and clickstream data. Seven online courses are offered in four different semesters for students from different education levels and ages. Unlike the Harvard University dataset which did not provide a timely record of student activity behavior [26], in OULAD each interaction "click" on any activity in a virtual learning environment was recorded in a separate table called "StudentVle". The OULAD dataset can be downloaded from (https://analyse.kmi.open.ac.uk/open_dataset). Table 1 describes the OULAD dataset.

3.3. The evaluation metrics

The evaluation methods used in most previous research are precision, recall, F1-score, and accuracy. This study will use the same evaluated methods. All equations of the evaluation metrics were taken from Ref. [29]. The Precision is calculated by dividing the percentage of True Positive elements by the total of positively predicted units. To calculate precision, Equation (1) is used.

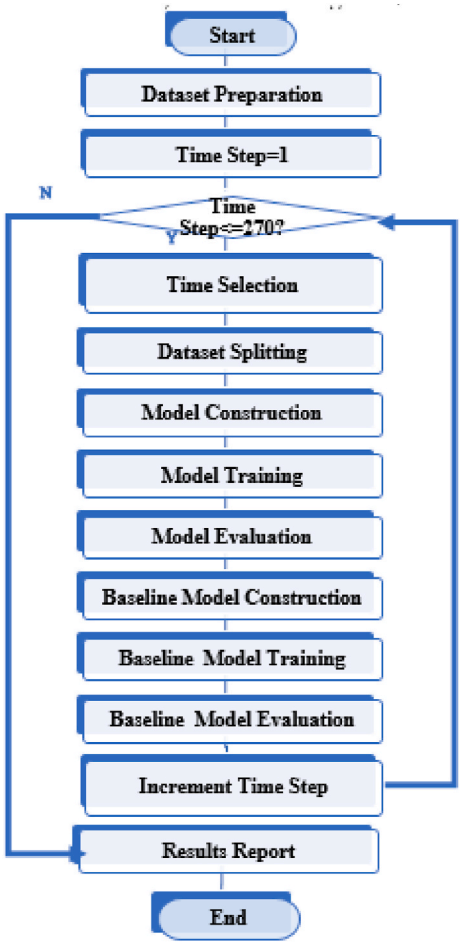

Fig. 1. The experiments' flowchart.

Table 1
The OULAD dataset description.

Years	2013,2014
Courses	7 courses
Number of courses introduced	22
Min – max course period	235–270 days
No. of unique students	28785
No. of students in all courses	32593
No. of assessments	206
No. of activities	6364

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

where TP is the number of true Positive elements and FP is the number of false positive elements. The model's capacity to find all Positive elements in the dataset is measured by the recall recall. Recall can be calculated by Equation 2

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

where FN is the number of those elements that the model has labeled as negative but are positive. Accuracy is a popular metric in multi-class classification that can be calculated by Equation 3

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP and TN are the elements that the model correctly classifies and FP and FN are the elements that the model incorrectly classifies. F1-score is interpreted as a weighted average of Precision and Recall. Equation (4) can be used to calculate it.

$$\text{F1} - \text{Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

To monitor the performance of the proposed model when treating all classes equally macro F1-score will be recorded. Macro F1-score is a harmonic mean of Macro Average Precision and Macro Average Recall [29] computed with Equation (5):

$$\text{Macro F1} - \text{Score} = 2 * \frac{(\text{Macro Average Precision} * \text{Macro Average Recall})}{\text{Macro Average Precision}^{-1} + \text{Macro Average Recall}^{-1}} \tag{5}$$

where the Macro Average Precision is the arithmetic mean of all classes' precision and can be calculated with Equation (6):

$$\text{Macro Average Precision} = \frac{\sum_{k=1}^{k} \text{Precision}_k}{k} \tag{6}$$

The Macro Average recall is the arithmetic mean of all classes recall and can be calculated with Equation (7):

$$\text{Macro Average Recall} = \frac{\sum_{k=1}^{k} \text{Recall}_k}{k} \tag{7}$$

Also, to monitor the performance of the proposed model when treating all instances (students) equally micro F1-score will be recorded. Micro F1-score is computed with Equation (8):

$$\text{Micro Average Recall} = \frac{\sum_{k=1}^{k} \text{TruePositive}_k}{GrandTotal} \tag{8}$$

where True Positive is the number of true predicted instances and Grand Total is the total number of instances.

3.4. ANN-LSTM model architecture

The features which will be used to predict the class of student performance will be demographic and clickstream features. These features are arranged in comma-separated values text files (CSV) which will be merged into one text file and fed to the ANN-LSTM model. The ANN-LSTM model will contain three layers, input, hidden, and output layers. As the input layer LSTM neural network will be used because LSTM is widely used in sequential data analysis [30] while the rest layers will use ANN which is the simplest neural network. The output of the proposed model will be the four categories of students' performance distinction (D), pass (P), fail (F), and withdrawn (W). Fig. 2 illustrates the proposed model architecture.

4. Results and implementation

To conduct the proposed model, environmental settings are needed. In addition, the dataset requires pre-processing which is depicted in subsection 2. The proposed model and its baseline models have been constructed as presented in subsections 3 and 4. Finally, the results of the experiments are shown in later subsections.

4.1. Environmental settings

The Jupyter Notebook IDE was used to implement the ANN-LSTM model in Python. Numpy, Pandas, Scikit-learn, TensorFlow and Keras open-source libraries were imported to implement the ANN-LSTM model. The source code of implemented experiments is uploaded on https://github.com/FatimaAlazazi/ANN-LSTM-model. The experiments were run on a machine with Core i7 and 16 GB RAM. The operating system was Windows 10 64-bit. Due to deep learning models taking a long time in training, a group of experiments was run on Windows 10 64-bit virtual machine workstation on a server machine to accelerate obtaining the results. All baseline experiments were run on a desktop PC with Core i7 and 32 GB RAM.

4.2. Dataset preparation

A SQL server database was created then all data inside these files were imported to the created database. Because the data of students' interaction with the activities are recorded as rows, a query was built to extract the sum of students' clicks with all the activities in one row for each day with columns representing each activity Fig. 3(a) describes the organization of raw data related to student activity interaction while Fig. 3(b) describes the organization of resulted query.

The data set CSV files were loaded into Pandas Dataframes Then, a sequence of merge operations was done till all demographic and clickstreams data were collected in the final Dataframe. Dataframes were merged based on unique identifiers like id_student, and

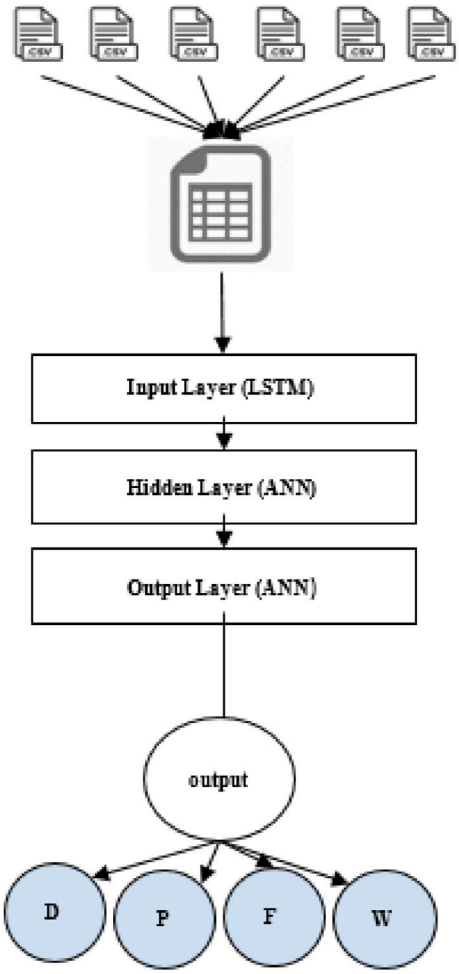

Fig. 2. The proposed model architecture.

id_site and arranged in Pandas DataFrame. The final dataframe was stored in a Numpy array in preparation for being entered into the ANN-LSTM model. As stated in Ref. [19], student interaction with the virtual learning environments (VLE) before the course stating does not importantly affect their performance, before the course started clickstreams were ignored.

The data set did not keep records for days with no interaction. So new records were added to represent which day each student did not interact. For example, if record r represented that students interacted with activity "a" in day 6 and did not interact with any activity in the previous days, then 5 records will be added related to this student with zero value interaction.

To improve the results of the ANN-LSTM model, feature value scaling was applied for numerical features with the "MinMax" scalar method.

One-hot encoder using the "get_dummies" method was applied to categorical demographic features to transform them into a binary vector while the "LabelEncoder" method was applied to the final result column for the same purpose. The number of features after the encoder was 71. The final prepared dataset can be downloaded from https://drive.google.com/file/d/1S5_k1VPA_U5mX1B7ac_ZxSjw18N1QA-p/view?usp=sharing.

For each time step (a day of the course duration), the ANN-LSTM model was trained and evaluated using demographic and clickstream data in this day and all previous days. So, the records related to this time interval were selected from the entire dataset. The total implemented experiments were 270 experiments which represent an experiment for each day in the course period. For each experiment that was implemented on a specific day of the course duration, the clickstream data related to that day and all days before it was used as input features to the ANN-LSTM model. A numerical variable named "Time Step "was declared to represent the current day in the course duration starting from one which represents the first day. When the time step equals one, the clickstream data of the first day is selected from the entire dataset and was fed to the ANN-LSTM model and baseline models as input features. After ANN-LSTM and baseline models have been constructed, trained, and evaluated using the selected clickstream data, the "Time Step" variable incremented by one to start the second-day experiment. The clickstream data of the first and second day of the course duration were selected and fed to the ANN-LSTM model and baseline models as input features and so on until reached the last day of the course duration which is day 270.

Holdout validation was applied. Prepared data was split into approximately 30% for testing, and 70% for training, while 20% of training data was used as validation data.

4.3. Model construction and configuration

From Keras API a sequential model was constructed. The input layer was an LSTM layer with an input shape of a 3-dimensional Numpy array and 200 units as output. The next layer was the Dropout layer with an output of 200 units. This Dropout layer was added to reduce the overfitting of data. A hidden Dense layer with an output of 100 units and an activation function of "Relu" was added before the output layer. Finally, another Dense layer was added as an output layer with an activation function of "Softmax" and with 4 output units representing the four classes of student performance categories Distinction, Pass, Fail, and Withdrawn. After adding layers, the constructed model was compiled. The optimizer hyperparameter was 'Adam '. The loss function was set to 'Categorical_Crossentropy '. As a metric of evaluation, 'Categorical_Accuracy' was used.

After the ANN-LSTM model was constructed, the training stage started. In the training stage the training data was fed to the ANN-LSTM model with the number of epochs set to 100 as reference [4] and batch size set to 100." Fit" method provided by Keras API was used for training the ANN-LSTM model.

code module	code presentation	student id	activity id	date	click_sum
1	2013	111	act 1	1	4
1	2013	111	act 2	1	20
1	2013	111	act 3	1	5
1	2013	111	act 4	1	3

(a)

code_mod	code_presentatio	student_id	date	act 1	act 2	act 3	act 4
1	2013	111	1	4	20	5	3

(b)

Fig. 3. (a) The organization of raw data related to student activity interaction (b) the organization of the resulting query.

Two evaluation stages were applied, validation and testing. From the validation stage precision, recall, and F1-score measurements were recorded. Then, the ANN-LSTM model was evaluated by the "evaluate" method provided by Keras API with pre-determined testing data. The batch size during the evaluation stage was 100. In each time step experiment training and validation accuracy were compared. Finally, accuracy was recorded to be compared with other experiments' accuracy.

4.4. Baseline models construction

RNN and GRU models were constructed to check the validity of the ANN-LSTM model. The RNN model had the same sitting as the ANN-LSTM model except for the input layer. So, the input layer of the RNN model is an RNN network consisting of 200 "simple RNN cells". Likewise, the GRU model had the same sitting as the ANN-LSTM model except for the input layer. So, the input layer of the GRU model is a GRU network.

Also, DFFN proposed in Ref. [4], AML model proposed in Ref. [1], and ML models in Ref. [3] were used as baseline models because they implemented multi-class classification which are closed to this work.

Two stages of evaluation were applied, validation and testing. From the validation stage precision, recall, and F1-score measurements were recorded. Then, RNN and GRU models were evaluated by the "evaluate" method provided by Keras API with pre-determined testing data. The batch size during the evaluation stage was 100. In each time step experiment training and validation accuracy were compared. Finally, accuracy was recorded to be compared with ANN-LSTM results.

5. Results

This subsection shows the experimental results which are divided into: the accuracy of the ANN-LSTM, the comparison between the results of ANN-LSTM model with baseline models results and the comparison with the state of the art results. The chosen models in the state of the art are DFFNN, AML and RF.

5.1. ANN-LSTM daily accuracy

The accuracy obtained on the first day of the course was 43% in the testing data. The accuracy of the ANN-LSTM model in the training data was higher than the accuracy in the validation data on the first day of the module. From day 15, the accuracy of the ANN-LSTM in the validation data was close to the accuracy of the ANN-LSTM in the training data. The accuracy was approximately 56%. After day 15 the accuracy of the ANN-LSTM in the validation data was higher than the accuracy of the ANN-LSTM in the training data. By day 41 the accuracy reached 65%. Per day 270 which represents the last day in the longest module (course), the accuracy reached 72%. Generally, the accuracy of the ANN-LSTM model in the testing data started from 43% on the first day of the course and reached 72% on the last day of the course duration. Fig. 4 shows the accuracy for all course days.

5.1.1. Comparison of ANN-LSTM model with RNN and GRU baseline models

To check the validity of the ANN-LSTM model, the same prepared data of demographic and clickstream for the first three months of course duration was input to RNN and GRU models. GRU model got 43% accuracy on the first day and 54% in day 30. In the RNN model, the accuracy on the first day was about 45% and on day 30 the accuracy was about 52%. Although the RNN model got better accuracy than ANN-LSTM in the first eight days, the ANN-LSTM model outperformed RNN and GRU models in the rest course days. Generally, the ANN-LSTM model got better accuracy than RNN and GRU models. Fig. 5 shows the accuracy of the ANN-LSTM, RNN, and GRU models in the first, second, and third months.

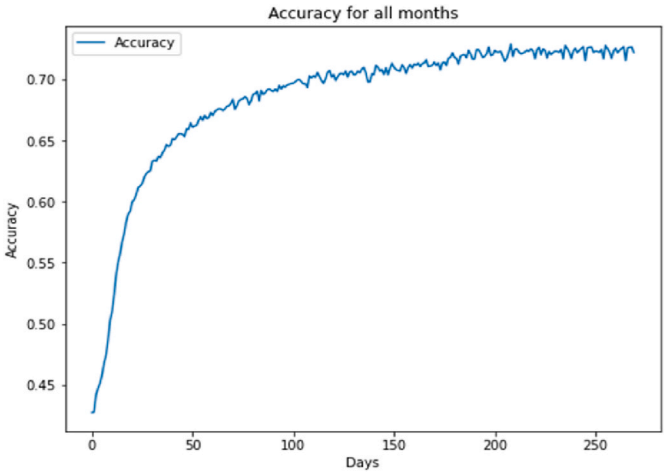

Fig. 4. Accuracy for all course days.

The arithmetic average of the accuracy enhancement rate between the ANN-LSTM and GRU model was about 15% while the arithmetic average of the accuracy enhancement rate between the ANN-LSTM and RNN model was about 21%. Figs. 6 and 7 show the accuracy enhancement rates in the first three months of the ANN-LSTM model compared with GRU and RNN models, respectively.

Using macro F1-score measurement, which treats all classes equally regardless of the class size, the best result was obtained by the ANN-LSTM model. The value of the macro F1-score obtained by the RNN model in the first three months was between (20% and 30%). GRU model got a macro F1-score value between (23% and 40%), while the ANN-LSTM got a value between (24% and 61%). Fig. 8 shows the macro F1-score of the first three months.

Using micro F1-score measurement, which treats all instances equally so the weight of each class will differ based on the number of instances belonging to it [29], the best result was obtained by the ANN-LSTM model. The value of the micro F1-score obtained by the RNN model in the first three months was between (28% and 41%). GRU model got a micro F1-score value between (29% and 47%), while the ANN-LSTM got a value between (31% and 63%). Fig. 9 shows the micro F1-score of the first three months.

To give the majority class higher weight, a weighted F1-score was calculated. The value of the weighted F1-score obtained by the RNN model in the first three months was between (26% and 38%). GRU model got a weighted F1-score value between (27% and 46%), while the ANN-LSTM got a value between (29% and 63%). Fig. 10 shows the weighted F1-score for the first three months.

Comparing macro, micro, and weighted F1-score for each model separately, the ANN-LSTM model got very close values of macro, micro, and weighted measurements. This phenomenon highlights that the ANN-LSTM model behaves equally when treating all classes equally or when giving a majority class higher weight. In RNN and GRU models micro F1-score values were higher than macro and weighted F1-score values. This implies that RNN and GRU models do well in the case of treating all instances equally. Fig. 11, Fig. 12, and Fig. 13 show the macro, micro, and weighted F1-score of ANN-LSTM, GRU, and RNN Models respectively.

5.2. *Comparison of ANN-LSTM model with DFFNN, AML and RF models*

ANN-LSTM model got better accuracy than the DFFNN model proposed in Ref. [4] in the case of using demographic and clickstream data on the last day of the courses. DFFNN got an accuracy of 63% while ANN-LSTM got 72%. This rate of improvement is due to the ability of LSTM to remember long dependencies between parameters. Table 2 shows a comparison between ANN-LSTM and DFFNN models when using demographic and clickstream data.

Other similar models to compare are RF models proposed in Ref. [3] which got the best results among GB, DFFNN, Gaussian NB, Bernoulli NB, KNN, DT, LR, ada boost, SVM, and Extra tree classifier. As it is clear in Table 3 ANN-LSTM model got the best score in most cases and it got the best accuracy, macro, and weighted F1-score. This may be justified by ANN-LSTM being a time series model and keeping latent dependencies between futures while RF is not a time series. The accuracy obtained was 72%, 66%, and 66% for ANN-LSTM, RF "gini" and RF "entropy" models respectively. In the F1-score metric, the ANN-LSTM got a better rate in the Distinction, Fail, and Withdrawn classes while RF got a better F1-score rate in the Pass class. It's possible that having a memory cell in the LSTM that can track the learner's weekly behavior [31] is one of the underlying factors causing the ANN-LSTM model getting best results than the DFFNN and RF models.

When comparing the ANN-LSTM model with the AML model proposed in Ref. [1], as is seen in Table 4, ANN-LSTM outperforms the AML model by about 10% in the 5th week and all other weeks. This may be because ANN-LSTM was trained with day-wise data while AML was trained with aggregated weekly data. Another possible reason is that in the AML model, all activity interaction data was added to one variable which represents the sum of all clicks in all activities while ANN-LSTM treats each activity separately.

Finally, Fig. 14 summarizes the comparison between baseline models proposed in similar studies in terms of accuracy, macro F1-score, and weighted F1-score. ANN-LSTM got the best result with 72% accuracy, 66% macro F1-score, and 68% weighted F1-score. Fig. 15 shows the enhancement rate of ANN-LSTM Compared with DFFNN, RF, and AML in terms of accuracy.

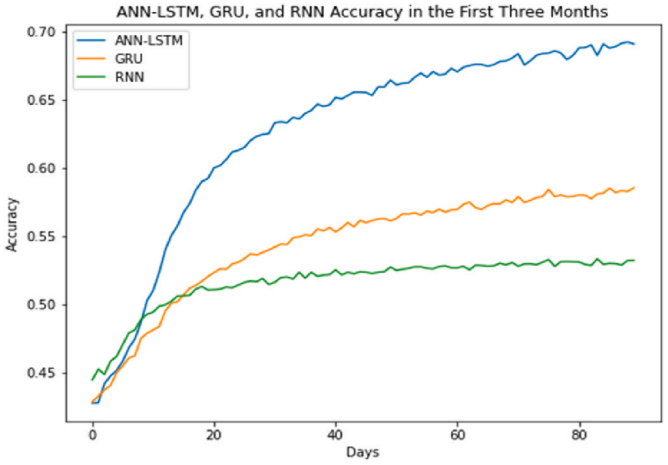

Fig. 5. Accuracy of the ANN-LSTM, RNN, and GRU models in the first three months.

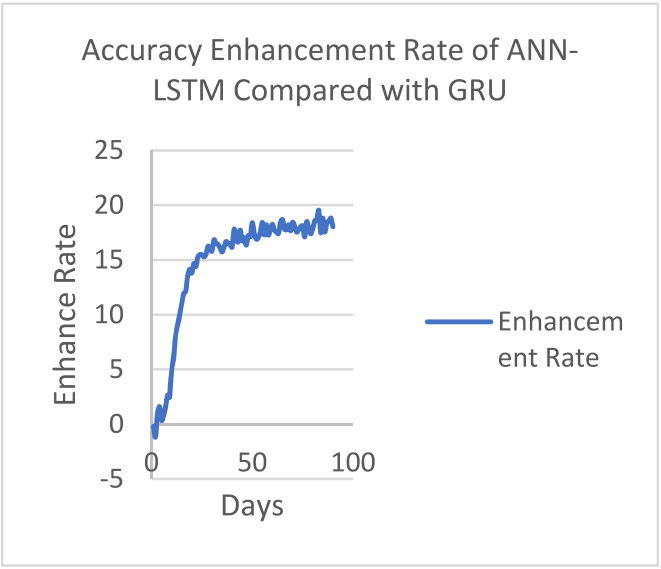

Fig. 6. Accuracy enhancement rate of ANN-LSTM compared with GRU.

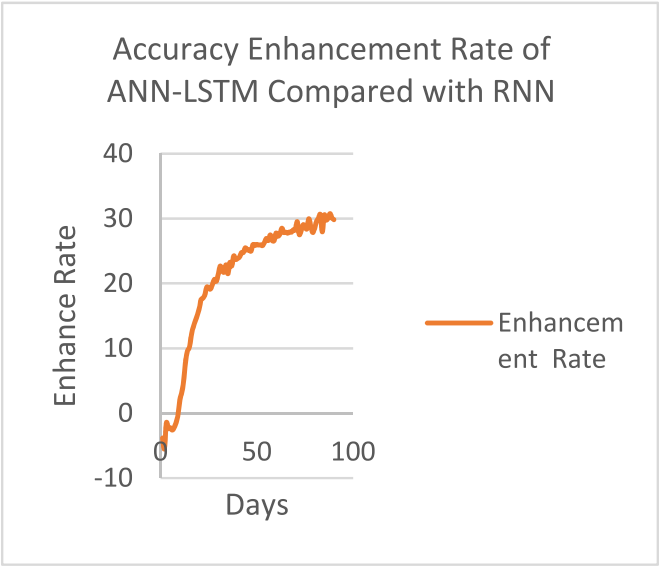

Fig. 7. Accuracy enhancement rate of ANN-LSTM compared with RNN.

6. Discussion

The ANN-LSTM model was developed to predict the class of students' performance in the multi-class form to achieve the first objective of this study which was to develop a day-wise deep learning model to predict students' performance in multi-class form. Generally, the ANN-LSTM got low accuracy for the first 15th days with an accuracy of 43% on the first day of the course. This may be due to a variety of students' behaviors and a lack of click stream data in the early days of the course. From day 15 data seems to be more representative. Hence, the accuracy of the ANN-LSTM in the validation data was close to the accuracy of the ANN-LSTM in the training data and the accuracy reached 56%.

The results show that the ANN-LSTM model obtained the best students' performance prediction accuracy. The average enhancements' rates of accuracy obtained in the first three months between the ANN-LSTM model and GRU and RNN were 15% and 21%, respectively. Although the RNN model got better accuracy than ANN-LSTM in the first eight days, the ANN-LSTM model outperformed RNN and GRU models in the rest course days. This phenomenon may be justified by the ability of LSTM to remember the previous student's behavior based on the architecture of the LSTM network. The answer to the second research objective, which was to

Fig. 8. Macro F1-Score of the first three months.

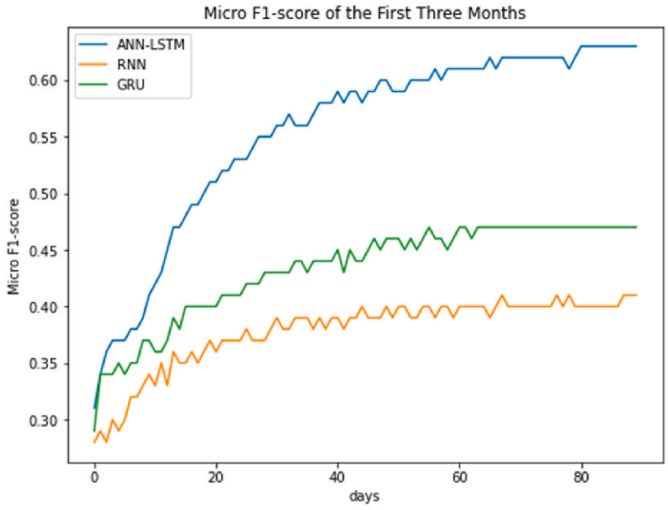

Fig. 9. Micro F1-Score of the first three months.

determine which deep learning model among RNN, LSTM, and GRU can get better students' performance prediction accuracy, is that the LSTM model obtained the best students' performance prediction accuracy.

The result of comparing the ANN-LSTM model with similar state-of-the-art models, which was the third objective of this research, is that ANN-LSTM obtained the best accuracy. ANN-LSTM model got better accuracy than the DFFNN model proposed in Ref. [4] in the case of using demographic and clickstream data on the last day of the courses. DFFNN got an accuracy of 63% while ANN-LSTM got 72%. This rate of improvement is due to the ability of LSTM to remember long dependencies between parameters. Compared with RF models proposed in Ref. [3], the ANN-LSTM model got the best score in most cases and it got the best accuracy, macro, and weighted F1-score. This may be justified by ANN-LSTM being a time series model and keeping latent dependencies between futures while RF is not a time series. Compared with the AML model proposed in Ref. [1], as is seen in Table 4, ANN-LSTM outperforms the AML model by about 10% in week 5 and all other weeks. This may be because ANN-LSTM was trained with day-wise data while AML was trained with aggregated weekly data. Another possible reason is that in the AML model, all activity interaction data was added to one variable which represents the sum of all clicks in all activities while ANN-LSTM treats each activity separately. So, treating behavioral features in day-wise form increases prediction accuracy.

7. Conclusion and future work

A multi-class course-agnostic day-wise deep learning predictive model named Artificial Neural Network-Long Short-Term Memory (ANN-LSTM) was constructed to periodically predict the class of student performance in MOOC Massive Open Online Courses (MOOC).

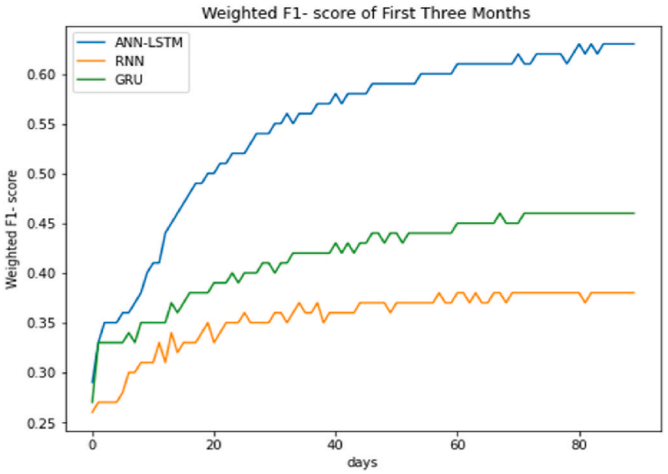

Fig. 10. Weighted F1-Score of The First Three Months.

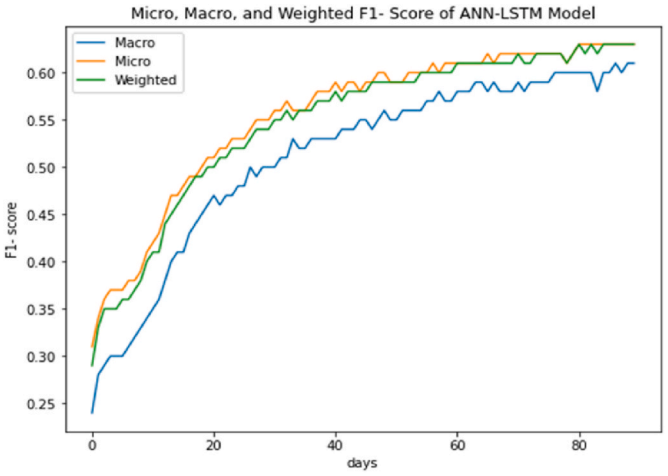

Fig. 11. Macro, micro, and weighted F1-Score of ANN-LSTM model.

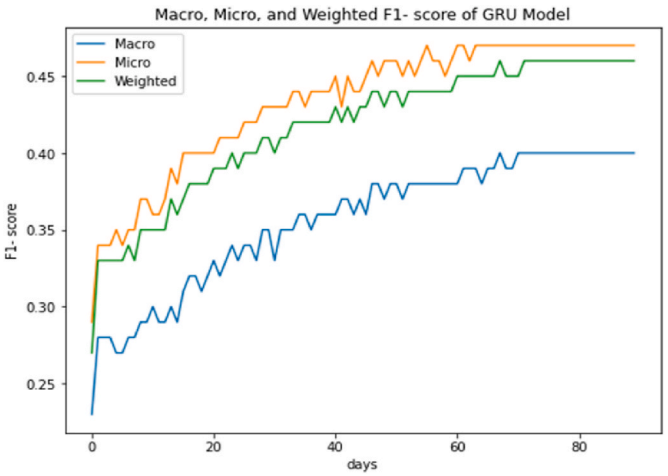

Fig. 12. Macro, micro, and weighted F1-Score of GRU model.

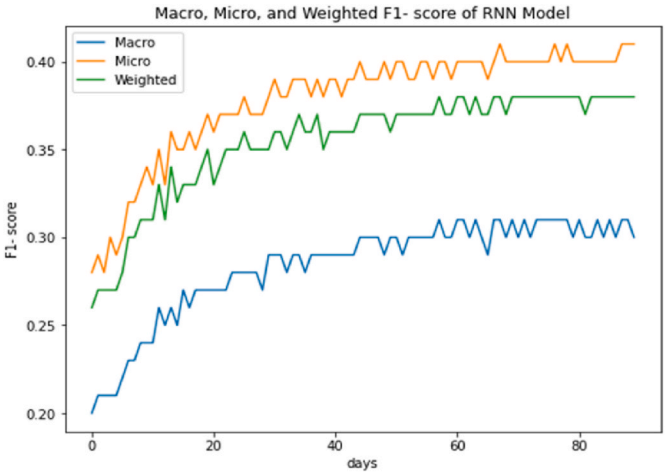

Fig. 13. Macro, micro, and weighted F1-Score of RNN model.

Table 2
Comparison between ANN-LSTM and DFFNN models when using demographic and clickstream data.

	ANN-LSTM			DFFNN		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Distinction	**0.82**	**0.47**	**0.59**	0.77	0.02	0.04
Fail	**0.79**	**0.55**	**0.65**	0.52	0.20	0.29
Pass	**0.84**	0.6	0.7	0.62	**0.92**	**0.74**
Withdrawn	**0.82**	0.62	0.7	0.67	**0.76**	**0.71**
Accuracy	0.72			0.63		
Macro average	**0.82**	**0.56**	**0.66**	0.64	0.48	0.45
Weighted average	**0.82**	0.58	**0.68**	0.63	**0.63**	0.56

Table 3
Comparison between ANN-LSTM and RF models.

	ANN-LSTM			RF 'gini'			RF 'entropy'		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
Distinction	**0.82**	**0.47**	**0.59**	0.66	**0.48**	0.56	0.66	0.48	0.56
Fail	**0.79**	**0.55**	**0.65**	0.56	0.45	0.50	0.55	0.44	0.49
Pass	**0.84**	0.6	0.70	0.69	0.86	**0.77**	0.69	**0.87**	**0.77**
Withdrawn	**0.82**	**0.62**	**0.70**	0.64	0.50	0.57	0.65	0.48	0.56
Accuracy	**0.72**			0.66			0.66		
Macro average	**0.82**	**0.56**	**0.66**	0.64	**0.57**	0.60	0.64	**0.57**	0.59
Weighted average	**0.82**	0.58	**0.68**	0.65	**0.66**	0.65	0.65	**0.66**	0.64

Table 4
Comparison between ANN-LSTM and AML.

ANN-LSTM					AML			
Week	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
Week 5 (Day 35)	**63.58**	**75**	**41**	**52**	53.51	43.29	39.89	37.20
Week 10 (Day 70)	**68.04**	**80**	**47**	**58**	57.79	45.71	43.73	41.70
Week 15 (Day 105)	**69.80**	**81**	**50**	**61**	61.68	49.05	46.49	44.43
Week 20 (Day 140)	**70.46**	**80**	**53**	**64**	65.00	54.98	49.30	46.66
Week 25 (Day 175)	**71.35**	**81**	**57**	**67**	67.40	58.00	51.15	48.43

The ANN-LSTM model was developed to predict the class of students' performance in multi-class form. In addition, RNN and GRU models were developed, trained, and evaluated to determine which deep learning model among Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), and Gated Recurrent Unit (GRU) can get better students' performance prediction accuracy. Then, the results of RNN, GRU, and ANN-LSTM models were compared in terms of accuracy, precision, recall, and F1-score. The results show

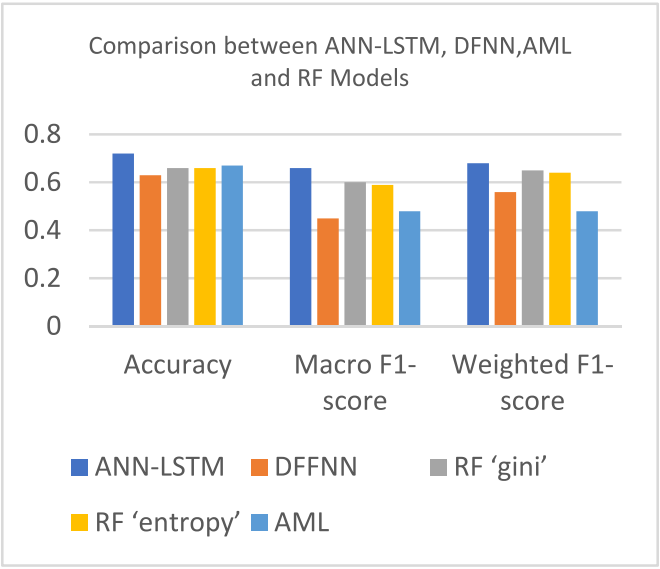

Fig. 14. Summary comparison between ANN-LSTM, DFNN, AML and RF models.

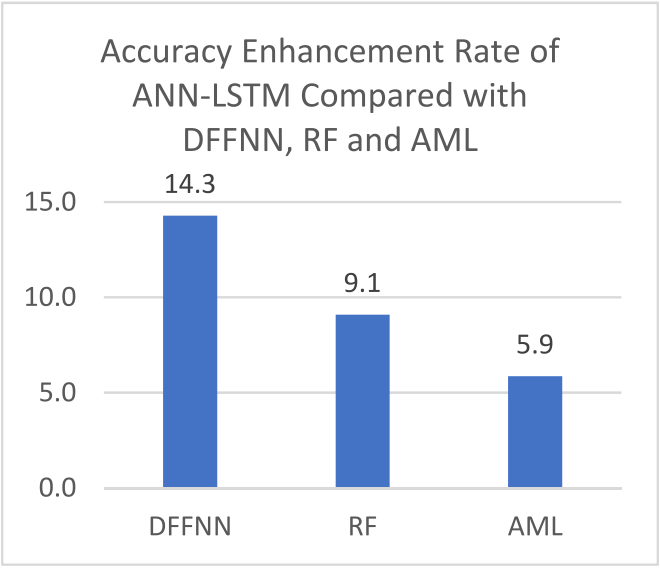

Fig. 15. Accuracy enhancement rate of ANN-LSTM compared with DFFNN, RF and AML.

that the ANN-LSTM model obtained the best students' performance prediction accuracy. The average enhancements' rates of accuracy obtained in the first three months between the ANN-LSTM model and GRU and RNN were 15% and 21%, respectively. Furthermore, for more validation of ANN-LSTM efficiency, the ANN-LSTM has been compared with the state-of-the-art models in terms of accuracy, recall, precision, and F1-score regardless of the implementation environments. ANN-LSTM model obtained the best accuracy results with enhancement rates of about 6% when compared with the AML model, 9% when compared with the RF model, and 14% when compared with the DFFNN model at the end of the course duration. In addition, the conclusions are as follows: Students' data (demographic and behavioral data) in the first days of course duration (especially the first three months) shows good results in the case of MOOC students' performance prediction. Hence, there are opportunities to predict the class of students in MOOC courses during the first months.

As a limitation of this study, researchers may need to consider that MOOC students generate massive clickstream interaction records and deep Learning techniques consume a long time in training computations. This may cause delay data processing, and consequently delay evaluating research results.

In future works, more studies and experiments are needed to increase the accuracy of prediction in the first days of course duration

with additional preprocessing methods like trying to make the OULAD dataset balanced using under-sampling or oversampling techniques. Furthermore, study the effect of the functional model instead of the sequential model design, using assessment data on predicting student performance, assigning weights for each class based on the class importance, and aggregating clickstream data periodically on the accuracy of predicting student's performance.

Author contribution statement

Fatima Ahmed Al-azazi: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Mossa Ghurab: Conceived and designed the experiments; Analyzed and interpreted the data.

Data availability statement

Data associated with this study has been deposited at original dataset - https://analyse.kmi.open.ac.uk/open_dataset prepared dataset - https://drive.google.com/file/d/1S5_k1VPA_U5mX1B7ac_ZxSjw18N1QA-p/view?usp=sharing ANN-LSTM model - https://github.com/FatimaAlazazi/ANN-LSTM-model.

Declaration of interest's statement

The authors declare no competing interests.

References

[1] Y. Xie, "Student performance prediction via attention-based multi-layer long-short term memory, J. Comput. Commun. 9 (8) (2021) 61–79, https://doi.org/10.4236/jcc.2021.98005.
[2] J. Hao, J. Gan, L. Zhu, "MOOC performance prediction and personal performance improvement via Bayesian network, Educ. Inf. Technol. (2022), 0123456789, https://doi.org/10.1007/s10639-022-10926-8.
[3] M. Adnan, A.A.S. Alarood, M.I. Uddin, I. ur Rehman, "Utilizing grid search cross-validation with adaptive boosting for augmenting performance of machine learning models, PeerJ Comput. Sci. 8 (2022) e803, https://doi.org/10.7717/peerj-cs.803.
[4] M. Adnan, et al., "Predicting at-risk students at different percentages of course length for early intervention using machine learning models, IEEE Access 9 (2021) 7519–7539.
[5] S. O.-A. of the R. S. for Cell and undefined 2021, Feature engineering, mining for predicting student success based on interaction with the virtual learning environment using artificial neural network, Annalsofrscb.Ro 25 (6) (2021) 12734–12746. https://www.annalsofrscb.ro/index.php/journal/article/view/8002.
[6] M.S. Ahmad, A.H. Asad, A. Mohammed, "A machine learning based approach for student performance evaluation in educational data mining, 2021 Int. Mobile, Intelligent, Ubiquitous Comput. Conf. MIUCC 2021 (2021) 187–192, https://doi.org/10.1109/MIUCC52538.2021.9447602.
[7] R. Alshabandar, A. Hussain, R. Keight, W. Khan, "Students performance prediction in online courses using machine learning algorithms,", Proc. Int. Jt. Conf. Neural Networks (2020) https://doi.org/10.1109/IJCNN48605.2020.9207196.
[8] H. Hassan, N.B. Ahmad, S. Anuar, "Improved students' performance prediction for multi-class imbalanced problems using hybrid and ensemble approach in educational data mining, J. Phys. Conf. Ser. 1529 (5) (2020), https://doi.org/10.1088/1742-6596/1529/5/052041.
[9] F. Ouyang, L. Zheng, P. Jiao, Artificial Intelligence in Online Higher Education: A Systematic Review of Empirical Research from 2011 to 2020, Springer US, 2022, https://doi.org/10.1007/s10639-022-10925-9.
[10] H. Karimi, J. Huang, T. Derr, "A deep model for predicting online course performance,", Assoc. Adv. Artif. Intell. (2020). https://www.semanticscholar.org/paper/A-Deep-Model-for-Predicting-Online-Course-Karimi-Huang/a2b68e9d01819de465beb8fbcd8d3b358c737f44#related-papers.
[11] F. Qiu, et al., "Predicting students' performance in e-learning using learning process and behaviour data, Sci. Rep. 12 (1) (2022) 1–15, https://doi.org/10.1038/s41598-021-03867-8.
[12] B. Pei, W. Xing, "An interpretable pipeline for identifying at-risk students, J. Educ. Comput. Res. (2021), https://doi.org/10.1177/07356331211038168.
[13] C. Vo, P. Nguyen, "ST_OS: an effective semisupervised learning method for course-level early predictions, IEEE Trans. Learn. Technol. 14 (2) (2021) 238–256, https://doi.org/10.1109/TLT.2021.3072995.
[14] G. Ramaswami, T. Susnjak, A. Mathrani, "On developing generic models for predicting student outcomes in educational data mining, Big Data Cogn. Comput. 6 (1) (2022), https://doi.org/10.3390/bdcc6010006.
[15] Q. Li, R. Baker, M. Warschauer, "Using clickstream data to measure, understand, and support self-regulated learning in online courses, Internet High Educ. 45 (2020) 100727, https://doi.org/10.1016/j.iheduc.2020.100727.
[16] R. Baker, et al., "The benefits and caveats of using clickstream data to understand student self-regulatory behaviors: opening the black box of learning processes, Int. J. Educ. Technol. High. Educ. 17 (1) (2020) 1–24, https://doi.org/10.1186/s41239-020-00187-1.
[17] T. Liyanagunawardena, "Massive open online courses, Humanities 4 (1) (2015) 35–41, https://doi.org/10.3390/h4010035.
[18] S.U. Hassan, H. Waheed, N.R. Aljohani, M. Ali, S. Ventura, F. Herrera, "Virtual learning environment to predict withdrawal by leveraging deep learning, Int. J. Intell. Syst. 34 (8) (2019) 1935–1952, https://doi.org/10.1002/int.22129.
[19] M. Wasif, H. Waheed, N.R. Aljohani, S.-U. Hassan, Understanding Student Learning Behavior and Predicting Their Performance, 2019, pp. 1–28, https://doi.org/10.4018/978-1-5225-9031-6.ch001.
[20] Y. He, et al., "Online at-risk student identification using RNN-GRU joint neural networks, OR Inf. 11 (10) (2020) 1–11, https://doi.org/10.3390/info11100474.
[21] H. Waheed, S. Hassan, N.R. Aljohani, J. Hardman, R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models,", Comput. Hum. Behav. (2019) https://doi.org/10.1016/j.chb.2019.106189.
[22] F. Hlioui, N. Aloui, F. Gargouri, "A withdrawal prediction model of at-risk learners based on behavioural indicators, Int. J. Web Base. Learn. Teach. Technol. 16 (2) (2021) 32–53, https://doi.org/10.4018/IJWLTT.2021030103.
[23] A. Esteban, C. Romero, Applied Sciences Assignments as Influential Factor to Improve the Prediction of Student Performance in Online Courses, 2021.
[24] B. Kumar Verma, D.N. Srivastava, H. Kumar Singh, "Prediction of students' performance in e-learning environment using data mining/machine learning techniques, J. Univ. Shanghai Sci. Technol. 23 (5) (2021) 569–593, https://doi.org/10.51201/jusst/21/05179.
[25] F. Alnassar, T. Blackwell, E. Homayounvala, M. Yee-King, "How well a student performed? A machine learning approach to classify students' performance on virtual learning environment,", Proc. 2021 2nd Int. Conf. Intell. Eng. Manag. ICIEM 2021 (2021) 1–6, https://doi.org/10.1109/ICIEM51511.2021.9445286.
[26] R. Al-Shabandar, "The application of machine learning for early detection of at-risk learners in massive open online courses, Diss. Abstr. Int. Sect. A Humanit. Soc. Sci. 82 (11-A) (2021). https://search.ebscohost.com/login.aspx?direct=true&db=psyh&AN=2021-61325-162&lang=it&site=ehost-live.

[27] A. Paramita, "Implementing machine learning techniques for predicting student performance in an E-learning environment, IJIIS Int. J. Informatics Inf. Syst. 4 (2) (2021) 149–156, https://doi.org/10.47738/ijiis.v4i2.112.
[28] N.R. Aljohani, A. Fayoumi, S.U. Hassan, "Predicting at-risk students using clickstream data in the virtual learning environment, Sustain. Times 11 (24) (2019) 1–12, https://doi.org/10.3390/su11247238.
[29] M. Grandini, E. Bagli, G. Visani, Metrics for Multi-Class Classification: an Overview," Pp. 1–17, 2020. http://arxiv.org/abs/2008.05756.
[30] Z. Han, J. Zhao, H. Leung, K.F. Ma, W. Wang, "A review of deep learning models for time series prediction, IEEE Sensor. J. 21 (6) (2021) 7833–7848, https://doi.org/10.1109/JSEN.2019.2923982.
[31] A.A. Mubarak, H. Cao, S.A.M. Ahmed, "Predictive learning analytics using deep learning model in MOOCs' courses videos, Educ. Inf. Technol. 26 (1) (2021) 371–392, https://doi.org/10.1007/s10639-020-10273-6.