JOURNAL OF LEARNING ANALYTICS

SOLAR SOCIETY for LEARNING ANALYTICS RESEARCH

Volume 11(2), 23–41. https://doi.org/10.18608/jla.2024.7985

A Novel Deep Learning Model for Student Performance Prediction Using Engagement Data

Mohd Fazil[1]*, Angélica Rísquez[2], Claire Halpin[3]

Abstract

Technology-enhanced learning supported by virtual learning environments (VLEs) facilitates tutors and students. VLE platforms contain a wealth of information that can be used to mine insight regarding students' learning behaviour and relationships between behaviour and academic performance, as well as to model data-driven decision-making. This study introduces a system that we termed ASIST: a novel **A**ttention-aware convolutional **S**tacked Bi**LSTM** network for student representation learning to predict their performance. ASIST exploits student academic registry, VLE click stream, and midterm continuous assessment information for their behaviour representation learning. ASIST jointly learns the student representation using five behaviour vectors. It processes the four sequential behaviour vectors using a separate stacked bidirectional long short term memory (LSTM) network. A deep convolutional neural network models the *diurnal weekly interaction* behaviour. It also employs the attention mechanism to assign weight to features based on their importance. Next, five encoded feature vectors are concatenated with the assessment information, and, finally, a softmax layer predicts the high-performer (H), moderate-performer (M), and at-risk (F) categories of students. We evaluate ASIST over three datasets from an Irish university, considering five evaluation metrics. ASIST achieves an area under the curve (AUC) score of 0.86 to 0.90 over the three datasets. It outperforms three baseline deep learning models and four traditional classification models. We also found that the attention mechanism has a slight impact on ASIST's performance. The ablation analysis reveals that *weekly event count* has the greatest impact on ASIST, whereas *diurnal weekly interaction* has the least impact. The early prediction using the first seven weeks of data achieves an AUC of 0.83 up to 0.89 over the three datasets. In yearly analysis, ASIST performs best over the 2018/19 dataset and worst over the 2020/21 dataset.

Notes for Practice

- Existing literature based on prediction models relies on simple neural network or statistical analysis, ignoring different aspects of virtual learning environment (VLE) interaction data and missing the potential of advanced neural network components like the attention mechanism.
- This study presents a novel attention-aware convolutional bidirectional long short term memory (LSTM) model, which we termed ASIST (**A**ttention-aware convolutional **S**tacked Bi**LSTM**), for student representation learning. It jointly models data based on student *demographic data*, previous *academic performance*, results from *continuous assessment*, and different aspects of *student interaction with the VLE*.
- The ASIST model was extended into ASIST_{early} for early prediction of student progression, which is effective using only the first seven weeks of VLE interaction data.
- The results of this study demonstrate that deep learning models can be effectively deployed to implement early intervention protocols and guide pedagogical design to promote student success.

Keywords

Learning analytics, educational data mining, predictive models, deep representation learning

Submitted: 09/03/2023 — **Accepted:** 27/03/2024 — **Published:** 12/05/2024

Corresponding author[1] Email: mfazil@imamu.edu.sa Department of Information Technology, College of Computer and Information Sciences, IMSIU, KSA; Centre for Transformative Learning, University of Limerick, Ireland. ORCID iD: https://orcid.org/0000-0002-8936-848X

[2] Email: angelica.risquez@ul.ie Centre for Transformative Learning, University of Limerick, Ireland. ORCID iD: https://orcid.org/0000-0002-2619-6297

[3] Email: halps1988@gmail.com Centre for Transformative Learning, University of Limerick, Ireland.

1. Introduction

Virtual learning environments (VLEs) generate abundant log-file data because they leave "learning traces" (Gasevic et al., 2015), providing insight into student activities and learning processes. As higher education institutions leverage these online learning platforms to facilitate content delivery, as well as communicative and assessment aspects of the pedagogical process, learner interaction with a VLE platform generates a wealth of information, providing a rich source of insight, such as student learning behaviour and its relationship with student performance. Researchers use this information for different predictive analytic tasks. The "collection, analysis, and examination of student information to understand the learning environment and student interaction behaviour, and using these insights to optimize the learning environment" is called learning analytics (LA) (Long & Siemens, 2011). The challenge with LA often has to do with two main issues: first, deploying robust and practically sustainable methods for educational data mining (EDM) (Shafiq et al., 2022), and second, using that data to inform student supports in useful ways. Selwyn (2020) criticizes the propensity of many educational institutions to use LA for institutional surveillance rather than individual support. Therefore, it is crucial that LA and the educational data be used as an instrument for empowering the most disadvantaged student cohorts (Essa, 2019). This paper tries to strike that balance—between methodological progress with EDM and on-the-ground LA application—in ways that are both useful and practicable.

1.1 Predictive Models for LA

The rapid development of tools and generated datasets has fuelled the growth and evolution of EDM over the years (Dowah et al., 2019; Hernandez-Blanco et al., 2019; Ang et al., 2020). The literature broadly classifies existing LA studies using EDM into statistical or exploratory analytics and predictive analytics categories (Shafiq et al., 2022). The first group of approaches collects and analyzes student information to gain insight into their learning behaviour, interaction with the learning environment, socio-demographic impact, and cognition level. The second category of approaches presents models to predict at-risk, drop-out, and high-performing students (Ang et al., 2020). The use of platform-generated learning information has been a common way to conduct statistical and predictive models to predict student performance (Waheed et al., 2020), at-risk students (Chui et al., 2020), and drop-out students (Santos et al., 2014). Researchers also used socio-demographic, academic, and admission information to predict student performance (Christian & Ayub, 2014; Mishra et al., 2014; Marbouti et al., 2016). Also, early performance prediction is vital to designing effective intervention measures and recommending remedial content and strategies.

Within the predictive analysis category, existing models can be classified into two types: (i) classical machine learning (ML) methods and (ii) deep learning (DL) methods. The classical ML models use hand-crafted features extracted from student information for predicting student performance. Examples of these early approaches, often based on training classical ML classifications and regression models to predict student academic success, are multiple in the literature (Huang & Fang, 2010; Romero et al., 2013; Strecht et al., 2015; Migueis et al., 2018; Rizvi et al., 2019; Wasif et al., 2019; Priya et al., 2021; Palacios et al., 2021; Jeslet et al., 2021; Yagci, 2022). While it yields interesting results, the main issue with classical ML is that hand-crafting features is time-consuming, domain specific, and infeasible in a scalable sense. To find more efficient methods of analysis of big datasets, recent advances in artificial neural networks (ANNs) have been applied to different problem domains (Young et al., 2018; Zhang et al., 2019; Fazil et al., 2021) in order to make it possible to automatically extract the features from raw data (Qiu et al., 2018). The existing literature has various DL models with promising results in diverse domains like socialbot prediction (Fazil et al., 2021; Fazil & Abulaish, 2018) and rumour prediction (Abulaish et al., 2019). Researchers are also employing neural networks in LA to predict at-risk students (Waheed et al., 2020; Sharada et al., 2018) and early drop out students (Wang et al., 2017; Xing & Du, 2018), for example.

The development of ANN and DL methods has eased the problem of manual feature engineering to some extent, and while some use manually designed features as input to deep models, some approaches completely avoid feature engineering. However, these advanced DL models, completely free from feature engineering, are like "black-box" models and face interpretability issues (Waheed et al., 2020). Therefore, although some recent approaches for student performance prediction in LA rely on DL-based models, these studies also train classical ML models due to the use of various categories of features. Examples of such approaches are multiple: Alam and colleagues (2018) defined features from student activity logs to train a deep belief network and five classical ML models to classify the students; Raga and Raga (2019) characterized each student using 18 features and created a simple feed-forward neural network for early performance prediction; Hu and Rangwala (2019) trained course-specific multilayer perceptron and recurrent neural network (RNN) models; Waheed and colleagues (2020) used singular value decomposition and trained a simple ANN for student performance prediction; Karimi and colleagues (2020) applied a graph convolution network for student and course representation learning and encoded student behaviour using a long short term memory (LSTM) network to predict student performance; Hai-tao and colleagues (2021) also used the graph convolution network for performance prediction; Ramanathan and Thangavel (2021) used a stacked LSTM-based DL model for student performance prediction; Waheed and colleagues (2022) used socio-economic information and arranged various VLE interaction information by week and trained four conventional and one DL models (LSTM) for early prediction; and Li and colleagues

(2022) modelled library use and web-browsing behaviours of students employing LSTM and convolutional neural networks (CNNs).

All of these approaches are simple deep neural networks with multiple hidden layers, and the application of complex deep models integrating several neural network components remains under-studied. The existing DL approaches suffer from two main limitations. First, they are simple RNN and ANN models, ignoring important neural network components like the attention mechanism, so they generally do not model different behavioural aspects of students using attentional deep neural networks. Second, they still use complex hand-crafted features to characterize students and suffer the pitfalls of manual feature engineering. This study bridges this gap and presents a novel attention-aware convolutional bidirectional RNN-based model (BiLSTM) employing simple feature engineering for student modelling to predict student performance. In doing so, we are aiming to propose a model that carefully balances the need for engaging with complex deep models integrating various neural network components while keeping the amount of feature engineering to a minimum to make it practicable in a real-case scenario.

1.2 Our Contributions

The novelty of the predictive model proposed in this study, which we termed ASIST (**A**ttention-aware convolutional **S**tacked B**iLST**M), resides in three aspects: (i) existing approaches model all the information as a single sequence, whereas ASIST models each category of information as a separate component and passes them to an attention-aware BiLSTM/CNN network; (ii) our model avoids complex feature engineering and uses a straightforward feature design; and (iii) it integrates the strength of various neural network components into a unified model to uncover fine-grained regularities in student behaviour. ASIST models students using demographic, continuous assessment, and interaction information, collected from the institutional VLE at the University of Limerick to predict the high-performing, moderate-performing, and at-risk students in a cohort. The VLE has various tools to facilitate and manage teaching, assessment and feedback, and collaboration and communication with and between the students. ASIST models the diurnal interaction behaviour of a student with the VLE as a tensor of order two. ASIST applies deep CNN on the tensor to find high-level feature maps representing students' *inter-day* and *inter-week* interaction behaviour. It also investigates the VLE interaction information to observe students' weekly interaction and events-related behaviour by modelling these behaviours as sequential information. An attention-aware stacked BiLSTM network is applied to these behaviour vectors to learn encoded representation. Demographic and academic information is added as a 15-dimensional auxiliary vector and passed through a BiLSTM network to encode students' background information. ASIST processes the behaviour vectors using either CNN or BiLSTM but does not assign any priority score to features. To this end, it applies the attention mechanism to determine the importance score for each feature. The concatenation layer concatenates the encoded representations from CNN and BiLSTM networks. Next, the continuous assessment information is added to the concatenated vector and passed through a dense layer followed by a softmax layer to predict student performance. In summary, the main contributions of this study are as follows:

- The study presents a novel attention-aware convolutional bidirectional LSTM model, ASIST, for student representation learning by joint modelling of student *demographic and academic* information, *continuous assessment* information, and different dimensions of *VLE interaction* information.
- Unlike existing studies, which model all the VLE interaction information as a single sequence, the presented architecture models different aspects of VLE interaction information using multiple sequence.
- We perform a rigorous evaluation over a real-world dataset from the University of Limerick to observe the efficacy of ASIST in predicting at-risk, moderate-performing, and high-performing students. We also perform behaviour ablation analysis to investigate the impact of each behavioural component on the performance of ASIST.
- We extend the proposed model as an early performance prediction model and evaluate it over the given dataset. Also, we train and evaluate the model over the dataset of each academic year.

2. Method

This study was conducted at an Irish university, in the context of an LA project aimed to analyze the factors leading to student success. The authors completed a baseline analysis to identify the personal, educational, and learning engagement factors that contribute to student success. The institution uses a VLE to facilitate all teaching, learning, and assessment interactions with students. This study obtained student VLE interaction datasets for three large (350–600 students) first-year modules from business, science, and humanities spanning four academic years from 2018/19 to 2021/22. All of the modules were offered in the fall semester, and their module leaders volunteered to participate in this study. We sought research ethics approval from

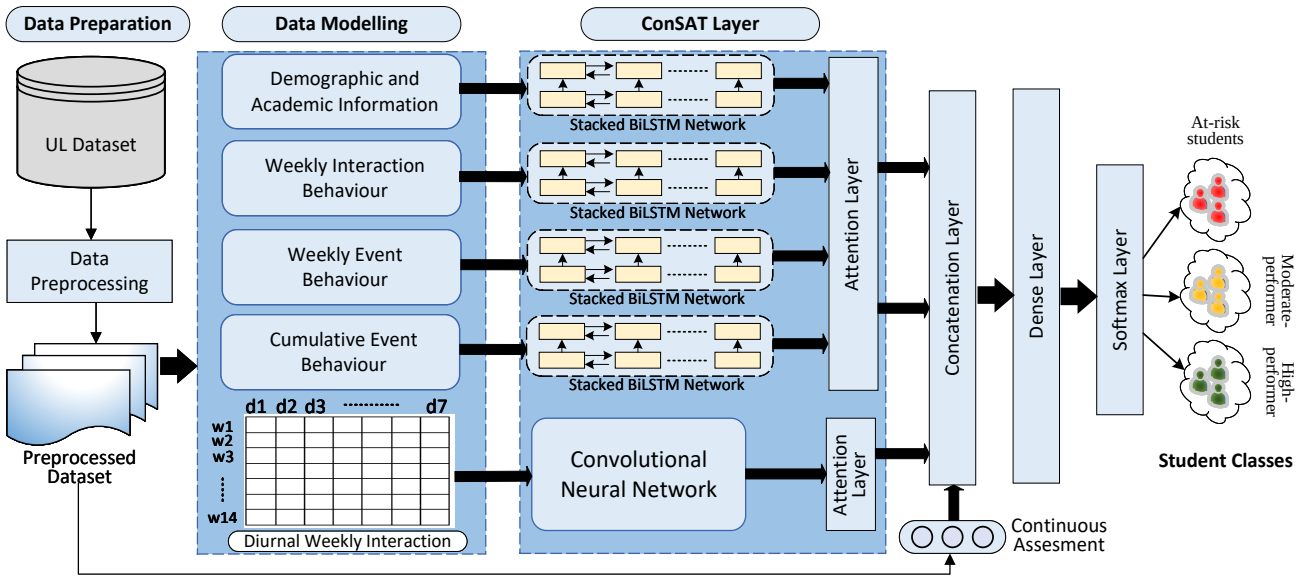


Figure 1. Architecture of the proposed ASIST model.

the concerned authorities of the respective faculties. The institution's information technology division provided the dataset, including the student activity log and academic registry data.

2.1 Dataset

We evaluated ASIST over three large cohort datasets from an Irish university (University of Limerick). The dataset has demographic, academic, and VLE interaction information for three modules—module-1, module-2, and module-3—spanning four academic years from 2018/19 to 2021/22. It contains information for 1,557, 2,078, and 2,433 students for module-1, module-2, and module-3, respectively. In the university, students are assigned to one of the 11 grades based on aggregate performance in continuous assessments and a final exam. However, we convert the university grades into high-performer (H), moderate-performer (M), and at-risk (F) categories, as discussed in Section 2.2.1. Table 1 presents the distribution of the three categories of students in the modules.

Table 1. Dataset statistics.

Module	Student Category			Total Students
	High-performer (H)	Moderate-performer (M)	At-risk (F)	
Module-1	373	966	218	1,557
Module-2	209	1,693	176	2,078
Module-3	401	1,478	554	2,433

2.2 ASIST Model

Figure 1 depicts the architectural representation of the proposed model. The following subsections introduce each layer of it.

2.2.1 Data Preparation

This layer processes the raw data to filter the noisy content and convert it to the required format. The information technology division of the university provided the academic registry and VLE interaction information in two files: AcademicRegistry.csv and VLE.csv. The academic registry is a single file including student demographic and academic information for all three modules. However, the institutional VLE data has a separate file for each academic year from 2018/19 to 2021/22 for each module. We created a single file for each module, combining the data from 2018/19 to 2021/22. In the university, students are assigned to one of the 11 grades A1, A2, B1, B2, B3, C1, C2, C3, D1, D2, and F, depending on their performance. We filtered the students with other grades. We adjusted the 11 grades into three, as given in Table 2. We assigned A2 students as high-performers and D1 and D2 students as at-risk because they are adjacent to the highest-performing (A1) and at-risk (F) categories of students, respectively. We assigned the remaining categories of students to the moderate category. For example, Figure 2 shows the original and adjusted grade distribution for module-1. It shows that most students fall under the moderate-performer category, whereas approximately 10% are at risk of failure. We also filtered the duplicate rows and

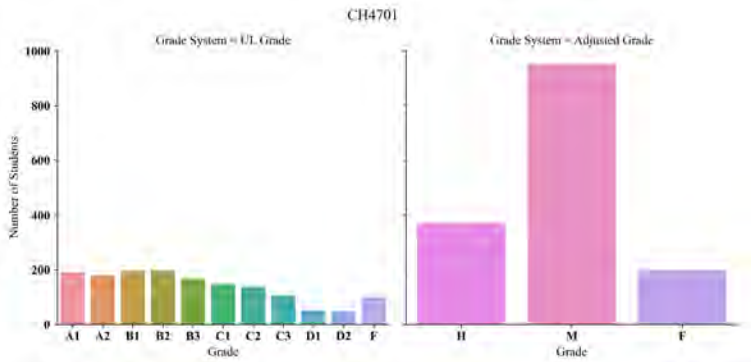


Figure 2. Original grade and adjusted grade distribution for module-1.

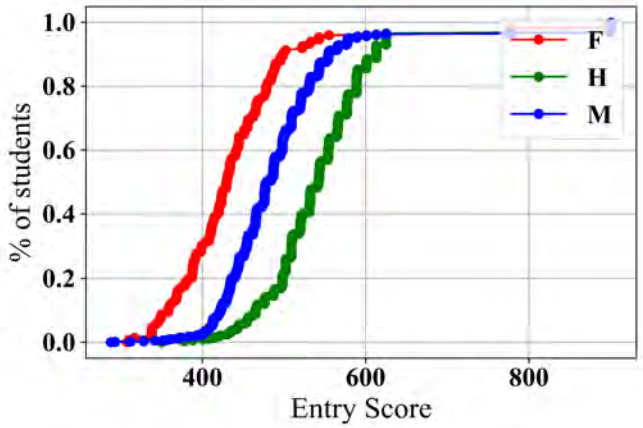


Figure 3. Grade and adjusted grade distribution for module-1.

extracted the date component from the date-time value. We also filtered the users with missing grade information in the academic registry file.

Table 2. UL grades and adjusted grades.

UL Grade	Adjusted Grade
A1, A2	High-performer (H)
B1, B2, B3, C1, C2, C3	Moderate-performer (M)
D1, D2 and F	At-risk (F)

2.2.2 Data Modelling

This layer transforms the preprocessed data to model the five behaviour representations of students. We model the VLE interaction information using four behaviour vectors rather than a single vector to find patterns from different student behaviours. To this end, the VLE interaction information is used to model the *weekly interaction*, *weekly event*, *cumulative event*, and *diurnal weekly interaction* behaviours of students. Although it is manual feature engineering, it is simple and straightforward. The following subsections provide a brief description of all of these representations.

2.2.3 Demographic and Academic Behaviour

The existing literature has studies wherein researchers have found a relation between students' demographic and academic records and their performance in upcoming courses (Migueis et al., 2018; Waheed et al., 2020). Therefore, this study uses this information from the university's academic registry (AR) and extracts 15 features, including three demographic and 12 academic pieces of information. Table 3 lists these 15 attributes and their brief description. We analyze the historical academic performance of three categories of students using entry score distribution, as shown in Figure 3 for module-1. It shows that 90% of at-risk students got less than 500, whereas only 30% of high-performing students did. The figure shows a significant difference among the three categories of students considering entry score distribution. This 15-dimensional vector, represented using $\mathscr{D}$, models the students based on their demographic and academic information.

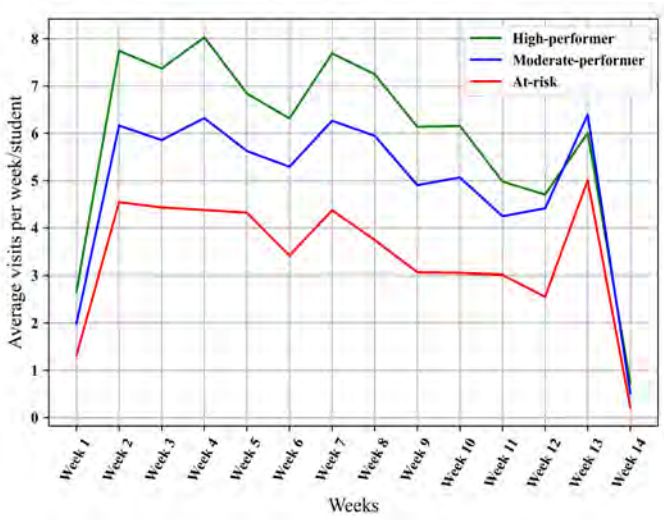


Figure 4. Students' *Interactions per week* for module-1.

Table 3. Demographic and academic attributes and their descriptions.

Attribute	Description
Age	Calculated from student's date of birth
Gender	Student's gender
FDN Class	Student's geographical location
Entry score	Sum of student performance in all of the subjects at leaving cert level
Access route	Student route, such as HEAR, DARE, LC, to enrollment in higher education in the Irish education system
SUSI recipient	A binary variable representing whether the student has been supported by Student Universal Support Ireland (SUSI) or not
SEM1 Grade	Student's grade in the first semester
SEM1 #Module	Number of courses studied by student in the first semester, representing student's academic load
LC_English	Student's performance in the English module at leaving cert level
LC_Irish	Student's performance in the Irish module at leaving cert level
LC_Math	Student's performance in the mathematics module at leaving cert level
LC_Chemistry	Student's performance in the chemistry module at leaving cert level
LC_Physics	Student's performance in the physics module at leaving cert level
LC_Economics	Student's performance in the economics module at leaving cert level
LC_Physics_with_Chemistry	Student's performance in the physics_with_chemistry module at leaving cert level

2.2.4 VLE interaction Behaviour

The university in this study uses a VLE to help teachers with such academic tasks as uploading resources (e.g., video tutorials), creating forums, and evaluating assignments. Students use the university's VLE to access course materials, submit assignments, track midterm results, chat on forums, and perform other academic activities. This study obtains the VLE interaction dataset of first-year students for three large cohorts—chemistry (module-1), management (module-2), and economics (module-3)—spanning four academic years from 2018/19 to 2021/22. All of the modules were offered in the fall semester. The information technology division at the University of Limerick maintains the student activity log. A session is created every time a student logs in to the VLE. Also, each activity on the VLE generates a particular event, depending on the underlying action. A log file is maintained to log all of the activities in the VLE, where each row/instance represents an activity using a set of 10 fields, briefly described in Table 4. This study uses session, visit, login, and interaction interchangeably. Using the log file, we model student behaviour using the following four one-dimensional vectors.

Table 4. Log file attributes and their descriptions.

Field	Description
Event date	The date and time when this event happened
Username	The unique ID of the student who performed the action
Event	The particular action performed by the student, e.g., *content.read*
Reference	The relative path of the accessed resource
Translation	The actual resource
IP	The IP address of the device used to access the VLE resource
Iporg	The Internet service provider used by the student while accessing the resource
Iplocation	The student's location while accessing the resource
useragent	A string containing basic information regarding the accessing device, such as Web browser and OS
platform	The operating system installed in the accessing device
browser	The browser used to access the resource
sessionid	The unique number assigned to a student every time they log in to the VLE

Weekly Interaction Behaviour Student interaction with the VLE may be an important indicator of student performance. Therefore, to analyze the frequency of student access to the VLE, we compute the total VLE logins for a student in a week. High access frequency represents a student's consistent effort and persistence. On the contrary, a low value may mean that the student is at risk and needs immediate intervention, although in some cases, there might be other reasons. To this end, we first compute the total number of visits (interactions) in a week. To observe the weekly access pattern over the semester, we find the visit count for each of the 14 weeks to create the weekly interaction behaviour vector $\mathscr{I}_W$. We further investigated the weekly interaction behaviour of three categories of students and found a notable difference among the three categories. For example, Figure 4 shows the weekly interaction of students for module-1. It depicts that the average access frequency of high-performing students is seven per week, except for a few weeks. On the other hand, the frequency is only four for at-risk students. Interestingly, the interaction frequency gap between H, M, and F is significantly reduced in the 13th week (exam week).

Weekly Event Behaviour Every interaction with the VLE generates an event based on the underlying activity. Once a student logs in to the VLE, it is important to model the number of activities performed within the environment, to distinguish from students who may just log out immediately. We tracked this behaviour by computing the event frequency. The frequency and type of events may be another good indicator to predict student grades. In the log file, each row represents an event. A student can do multiple events in a session/interaction/visit. We first computed a student's total event count in a week. We further found the event count for each of the 14 weeks to find the weekly event behaviour vector, E_W. Therefore, the weekly event behaviour is represented using a 14-dimensional vector. We also investigated the weekly event count of three categories of students, as shown using a line graph in Figure 5. The figure shows that just like interactions per week, the difference among three categories of students considering the weekly event count is noteworthy. On average, at-risk students perform approximately half the weekly events of the high-performer.

Cumulative Event Performance As discussed earlier, every interaction with the VLE generates an event, depending on the underlying activity. In the VLE, students perform events like *content.read* and *ancc.read*. The frequency and type of events performed may be another indicator of student performance. Some events may be more important than others, so analyzing different event counts is also vital. To this end, we modelled the cumulative count of each event to observe the student event performance behaviour. We investigated the cumulative event count of the three categories of students. For example, Figure 6 exhibits the distribution of event count for module-1. It exhibits that events like *annc.read*, *content.read*, and *calendar.read* show trivial differences in performance frequency among the three categories of students. However, this difference is considerable for assessment-related events like *sam.assessment.submit.checked* and *sam.assessment.take*, where high-performers perform these events frequently. The cumulative distribution reveals less activity by at-risk students. The cumulative event count, a 24-dimensional vector, is represented using E_C.

Diurnal Weekly Interaction The activity log representing student interaction with various VLE resources has rich information. Every time a student logs in to the VLE, it creates a session with a unique ID. Further, it assigns every activity to the underlying session ID. The active session ends once the student logs out. We used the students' daily interaction frequency in the VLE to model regularity in users' diurnal interaction. We modelled the diurnal interaction frequency grouped by week. To this end, we first counted the total interactions on a particular date. To observe the temporal evolution of the student interaction, we

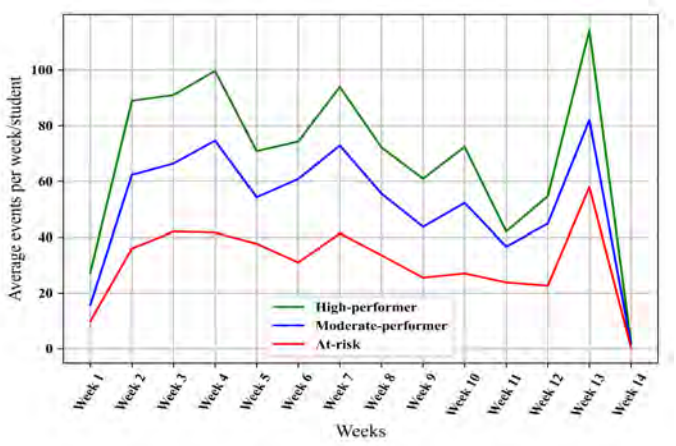


Figure 5. Students' *Event count per week* for module-1.

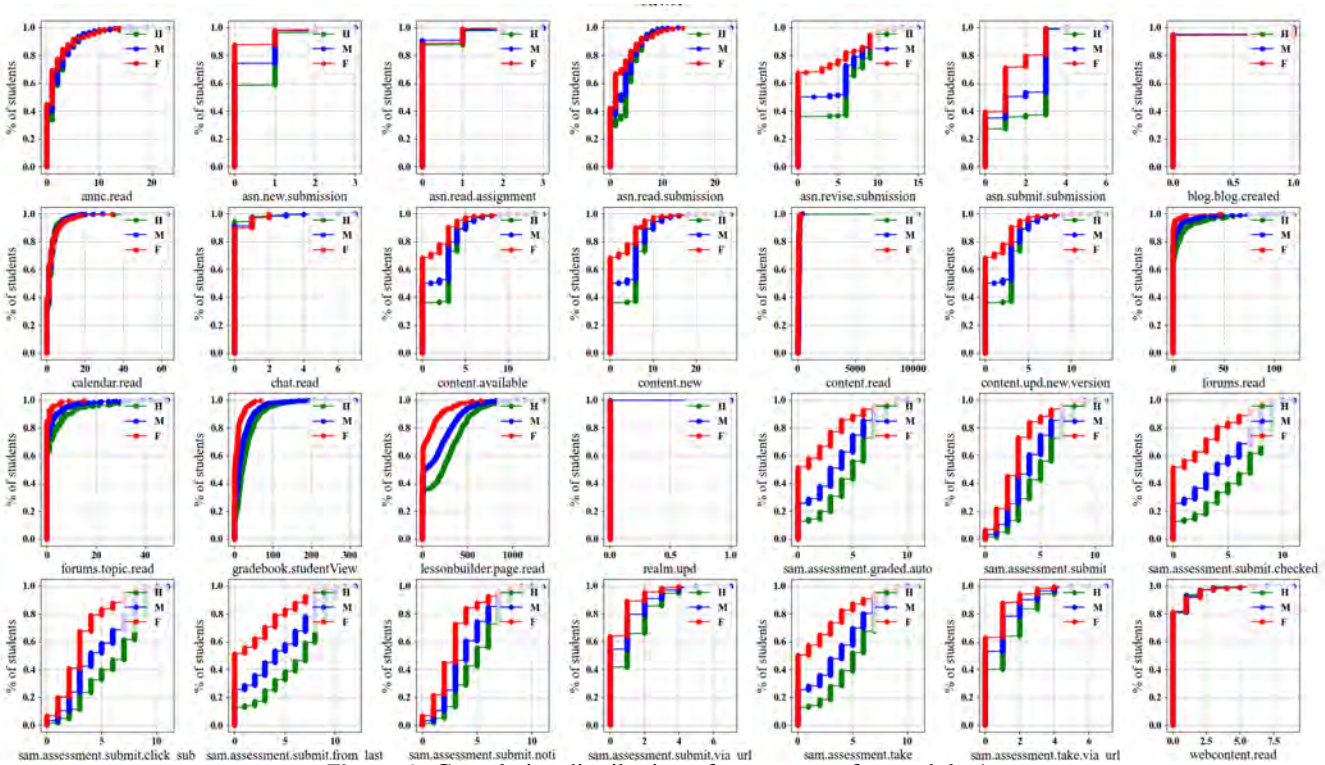


Figure 6. Cumulative distribution of event count for module-1.

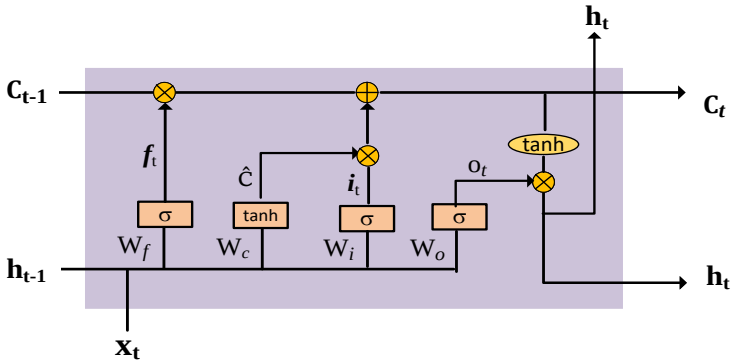


Figure 7. Architecture of an LSTM cell redrawn from Arbel (2018).

computed it for each day of the week, creating a seven-dimensional vector. At the University of Limerick, an academic semester runs for 14 weeks, so we computed the interaction count for every day of each week for 14 weeks, creating a seven-dimensional vector for each week. We model the diurnal weekly VLE interaction, represented by $\mathscr{I}_D$, by organizing the 14 constructed seven-dimensional vectors as a two-dimensional tensor $M^{N_w \times N_d}$, where N_w and N_d represent the number of weeks (i.e., 14) and the number of days in a week (i.e., seven), respectively. This representation encodes the inter-day and inter-week interaction patterns.

Continuous Assessment In a module, students' progress is tracked based on their performance in continuous assessment, either midterm exams or assignments. The student's performance in these assessments is a good indicator of their progress and final grade. We considered student performance in the first four assessments, represented using $\mathscr{A}$, and used them as the auxiliary features at the concatenation layer. In this regard, assessment scores are, first, sorted from highest weight to lowest, and then the student results in the first four are selected. We chose only four assessments from each module so that the length of the assessment vector is the same for all modules.

2.3 ConSAT Layer

This section describes neural network components used in the ASIST model to process the identified behaviour representations. The proposed model employs stacked bidirectional LSTM (BiLSTM) to process the weekly interaction behaviour, weekly event count, cumulative event count, and academic registry information. It also includes a two-layer CNN to process the diurnal VLE interaction behaviour. Additionally, the attention layer assigns an importance score to each value of behaviour vectors. The following subsection presents a detailed description of each of the discussed components.

2.3.1 Stacked BiLSTM Network

The proposed model uses a two-layer stacked RNN to process and encode the four sequential behavioural vectors. We use BiLSTM, an RNN, to process the four sequential pieces of behaviour information to model temporal dynamic behaviour (Hochreiter & Schmidhuber, 1997). The BiLSTM network handles the *vanishing gradient* problem using forget gate and the additive property. The BiLSTM takes two LSTMs, forward and backward, to capture both the historical and future context in a temporal sequence vector. In this study, student VLE interaction is modelled using four temporal sequential vectors; therefore, we use BiLSTM to learn encoded representations from these vectors. The memory block in LSTM uses a memory cell to decide what to forget and what to remember, enabling it to learn long-range dependencies. Figure 7 shows the internal architecture of an LSTM cell. It has information in the form of hidden state $h(t)$ and cell $c(t)$ for the current timestamp t that is processed using *forget*, *input*, and *output* gates. The hidden state and cell state are known as short-term memory and long-term memory, respectively. An LSTM cell first determines what information to forget from the previous timestamp using a forget gate, f_t. It uses the last hidden state h_{t-1} and input x_t to compute the value determining the amount of forgotten information using equation 1. The computed value is multiplied by the previous cell state c_{t-1} to update the amount of the last state information:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

Further, LSTM computes the amount of new information to be included in the current cell state using an input gate. In this process, it first finds new information (i.e., current input) using equation 2. Next, the input gate using a tanh function computes the new information $\tilde{C}_t$ using equation 3 to the cell state. Finally, the cell state at the current time stamp t, C_t, is

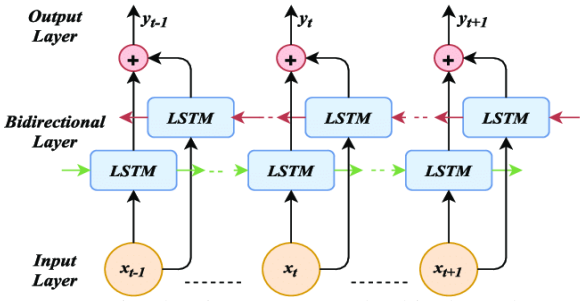


Figure 8. A simple BiLSTM network (Ihianle et al., 2020).

updated using equation 4, where the first part erases some information from the earlier cell state C_{t-1} and the second part adds the new information:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\widehat{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_c) \tag{3}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \widehat{C}_t \tag{4}$$

Further, the LSTM cell using the output gate determines what part of the current cell state C_t will be the cell output. First, a value o_t is computed using a sigmoid function as in equation 5 to determine the amount of output information. Next, the current cell state C_t is passed through a hyper-tangent function and multiplied by the o_t to compute the final LSTM hidden state/output h_t as in equation 6. The W and b in the above equations are the weight and bias vectors, respectively, whereas $\otimes$ performs element-wise multiplication:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{6}$$

This study uses BiLSTM to capture the contextual information in both directions using a pair of forward and backward LSTMs. In the context of LA and this study, we use BiLSTM to learn the representation incorporating both the forward and backward context. For example, to learn the representation for a particular weekday, backward LSTM learns the representation for the day including the VLE information of previous days of that week, whereas forward LSTM learns the representation including the VLE-extracted information of the remaining days of the week. In the process, the forward and backward LSTM generates two hidden states, $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$, using equations 7 and 8, respectively. The BiLSTM network computes the final hidden representation h_t using equation 9. Figure 8 shows an example of a BiLSTM network. In empirical evaluation, researchers observe that deep RNN is better at learning low-level feature representation with better model complexity (Pascanu et al., 2014). Therefore, this paper uses a two-layer stacked BiLSTM to process the sequential behavioural vectors:

$$\overleftarrow{h}_t = \overrightarrow{LSTM}(f_t) \tag{7}$$

$$\overrightarrow{h}_t = \overleftarrow{LSTM}(f_t) \tag{8}$$

$$h_t = \sigma(\overleftarrow{h}_t, \overrightarrow{h}_t) \tag{9}$$

2.3.2 CNN

The four sequential behaviour representations are processed using separate stacked-BiLSTM networks. BiLSTM efficiently extracts temporal patterns from sequential information but cannot extract spatial features. We organized the student diurnal interaction information weekly and modelled it using the CNN to extract the student inter-day and inter-week interaction patterns. We modelled the student diurnal interaction over the semester as two-dimensional tensor $M \in R^{N_w \times 7}$, where N_w represents the number of weeks in the semester, which is 14 for the given dataset. CNN efficiently extracts features from grid data like image (LeCun et al., 2015). In the existing literature, researchers exploit its competency in retrieving good semantic and spatial features, efficiently employing these features in several natural language processing (NLP) applications. CNN also extracts local and position-invariant features (Yin et al., 2017). Therefore, we apply CNN over the diurnal interaction information to extract efficient low-level spatial and position-invariant features. A CNN typically performs *convolution* and *pooling* operations, where a convolution operation extracts local features using different kernels and pooling extracts vital features from the extracted features maps. A CNN uses kernels of various sizes. In a deep CNN, higher layers capture rich and complex features from lower-layer feature maps (Roy et al., 2020). Therefore, this study uses a two-layer CNN with a max pooling layer after the second layer to extract the rich local features.

2.4 Attention Layer

The encoded representations from the stacked BiLSTM and CNN layers are passed through attention layers to assign feature weights, which are high for important features and low for insignificant ones. In NLP and other computing problems, attention is a technique to capture the important features from the input feature representation. In the process, it assigns higher weights to important features for the underlying tasks and lower weights to trivial features. Because NLP represents the text in a sequential pattern, this study also presents student interaction behaviour in a sequential form. Additionally, each value of every interaction behaviour vector is not of equal importance; rather some are more important than others. Therefore, the attention mechanism is suitable for the presented scenario where the attention will assign each interaction value a weight representing its relative importance within the vector. Attention assigns high weights to important VLE interaction factors. This study uses word-level attention by Bahdanau and colleagues (2015). If the encoded hidden representation of a behaviour vector f from one of the BiLSTM/CNN layers is h_f, then it is passed to the attention layer for further contextual weight scoring, where h_f is passed to a dense layer using the hyper-tangent activation function to learn the hidden representation h'_f as given in equation 10. The activation function tanh takes the product of a trainable weight matrix W_w and h_f added to a bias vector b_w. Thereafter, equation 11 computes the normalized similarity, α_f, of the hidden representation h'_f. Finally, equation 12 computes the product of the normalized similarity α_f and the feature vector h_f to find its attention-based representation:

$$h'_f = \tanh(W_w h_f + b_w) \tag{10}$$

$$\alpha_f = \frac{\exp(h'_f)}{\sum_f \exp(h'_f)} \tag{11}$$

$$s_i = \sum_{i=1}^{t} \alpha_f h_f \tag{12}$$

2.5 Concatenation Layer

This layer concatenates the four BiLSTM-encoded representations with the CNN-encoded diurnal weekly behaviour. Further, it concatenates the continuous assessment information that is passed through a dense layer followed by a softmax layer to classify student performance into *high-performer*, *moderate-performer*, and *at-risk* categories.

3. Results and Discussion

In this section, we empirically evaluate ASIST over three real datasets. In the following subsections, we discuss the experimental results, perform comparative analysis, and conduct other empirical evaluations.

3.1 Evaluation Metrics

This study is a classification problem to predict student performance. In the existing literature, researchers generally evaluate classification models using accuracy, precision, recall, and fscore. The presented model predicts student categorizations into

three categories. The datasets for the three modules are greatly imbalanced because they have a relatively small number of *high-performing* and *at-risk* students compared to the *moderate-performers*. In an imbalanced dataset, performing experimentation using only accuracy is not considered good evaluation (Mubarak et al., 2021; Wang et al., 2017). Researchers have used accuracy, including other evaluation measures like area under the curve (AUC) and fscore, to evaluate the efficacy of their models (Raga & Raga, 2019; Waheed et al., 2020; Hidalgo et al., 2022). Therefore, we evaluate the proposed model using a comprehensive set of five metrics: accuracy (Acc), precision (Pre), recall (Rec), fscore (Fsc), and area under the receiver operating characteristic curve (AUC).

3.2 Experimental Setting

During experimentation, we trained ASIST using a learning rate of 0.001 and *Adam* as the optimization algorithm. The model was trained on 80% of the dataset and tested on the remaining 20%. The model was trained for 50 epochs using categorical cross entropy as the loss function because we classify student performance in more than two categories. The proposed model uses a dropout value of 0.5 to avoid over-fitting. ASIST processes the four sequential behaviour vectors using an attention-aware stacked BiLSTM network. In the stacked BiLSTM network, the first and second BiLSTM layers use 128 and 64 memory units, respectively. ASIST employs two CNN layers with a dropout layer after the first layer to process the diurnal weekly interactions. The first CNN layer uses 128 filters with size 3×3, whereas the second layer uses 64 filters each of size 2×2. After the two CNN layers, the model applies a max pooling layer with a pooling operation of two. The proposed model adjusts the batch size to 16 in the training procedure. The output layer uses the softmax function with three neurons to classify student performance into three categories.

3.3 Comparison Methods

We evaluated ASIST's performance with the following baseline methods:

BiLSTM_CNN: This baseline analyzes the impact of attention on ASIST's performance. To this end, we remove the attention layers to retrain ASIST and predict student performance to analyze the impact of the attention layer on the model's performance.

LSTM_CNN_Attn: We created this baseline to analyze the impact of using LSTM instead of BiLSTM. Therefore, we replace the BiLSTM layers with LSTM to analyze ASIST's performance.

Simple Model: ASIST uses stacked BiLSTM and CNN. In this baseline, we use a single BiLSTM and a single CNN layer to investigate the impact of stacked BiLSTM and CNN layers on ASIST's performance.

ANN: This baseline concatenates four sequential behaviour vectors, $\mathscr{D}$, $\mathscr{I}_W$, $\mathscr{E}_W$, and $\mathscr{E}_C$; diurnal weekly behaviour $\mathscr{I}_W$; and midterm performance values $\mathscr{A}$ into a single vector. Further, the concatenated vector is passed to a deep ANN with two hidden layers with 100 and 50 neurons, respectively. Finally, the softmax layer predicts student performance.

Decision Tree: In this baseline, we concatenate all of the behaviour representations into a single vector and pass it to a decision tree using entropy as a splitting criterion for student performance prediction.

SVM: This baseline also concatenates all of the behaviour representations into a single vector and passes it to a support vector machine with a radial basis kernel to predict student performance.

Random Forest: This baseline also concatenates all of the behaviour representations into a single vector and passes it to a random forest with 50 decision trees to predict student performance. Random forest is an ensemble classifier that fits multiple decision trees to various random bootstrap samples. In classification, the final class is the mode value of all of the decision tree predictions.

XGBoost: This stands for extreme gradient boosting. It is the open-source library of gradient-boosted decision trees under the gradient boosting framework with improved speed and performance. Like the random forest, it is also an ensemble learning method for classification and regression problems. This baseline also concatenates all of the behaviour representations into a single vector.

Table 5. Performance results of ASIST over University of Limerick dataset.

Approach	Module-1					Module-2					Module-3				
	Acc	Pre	Rec	Fsc	AUC	Acc	Pre	Rec	Fsc	AUC	Acc	Pre	Rec	Fsc	AUC
ASIST	0.7519	**0.7776**	0.7247	**0.7491**	**0.8858**	**0.8245**	**0.8442**	0.8191	**0.8131**	**0.9040**	**0.7207**	0.6990	**0.7083**	0.7029	**0.8614**
BiLSTM_CNN	0.7434	0.7391	0.7012	0.7408	0.8682	0.8190	0.8056	**0.8245**	0.8003	0.9025	0.7125	0.7102	0.6982	**0.7051**	0.8185
LSTM_CNN_Attn	0.7492	0.7726	0.7245	0.7478	0.8791	0.8161	0.8401	0.8169	0.8103	0.8994	0.7157	0.7022	0.6879	0.6949	0.8555
Simple Model	0.6987	0.7056	0.6760	0.6901	0.8531	0.8004	0.8030	0.7956	0.7992	0.8988	0.6819	0.6671	0.6589	0.6627	0.8289
ANN	0.6667	0.6664	0.6635	0.6649	0.8529	0.7721	0.7662	0.7644	0.7653	0.8785	0.6529	0.6514	0.6489	0.6502	0.8215
Decision Tree	0.6879	0.6812	0.6859	0.6825	0.7704	0.7587	0.7476	0.7524	0.7496	0.6571	0.5917	0.5949	0.5955	0.5939	0.6987
Random Forest	0.7288	0.7301	0.7212	0.6994	0.8411	0.7929	0.7108	0.8149	0.7446	0.7903	0.7184	**0.7150**	0.6899	0.6530	0.8186
XGBoost	**0.7610**	0.7695	**0.7564**	0.7432	0.8790	0.7954	0.7864	0.8173	0.7597	0.7665	0.6918	0.6943	0.6982	0.6847	0.8434
SVM	0.7519	0.7499	0.7340	0.7131	0.8500	0.7891	0.7411	0.8149	0.7341	0.6983	0.6725	0.6937	0.6530	0.5896	0.7685

3.4 Results

In this section we evaluate ASIST over the three datasets in comparison to the eight baseline models, including four designed DL models, *simple model*, *BiLSTM_CNN*, *LSTM_CNN_Attn*, and *ANN*, and four classical ML models, *decision tree*, *random forest*, *XGBoost*, and *support vector machine*. Table 5 presents the results of ASIST and baseline models. The best performance considering each evaluation metric on each dataset is highlighted in bold typeface. The table shows that ASIST outperforms the comparison methods in all but five cases. ASIST also performs best over the module-2 dataset and worst over the module-3 dataset. The table also shows that considering AUC, the proposed model outperforms the comparison methods over all three datasets. This is significant because AUC is a good evaluation metric for imbalanced datasets. The second row of the table shows that the exclusion of the attention mechanism has a minute impact on ASIST's performance. The third row indicates that excluding the second BiLSTM and CNN layers significantly impacts ASIST's performance. We also evaluated how the model would perform if instead of BiLSTM, LSTM were used; the underlying results are presented in the fourth row of the table. The result shows that the model performance is slightly decreased, though not significantly; therefore, the presented model uses BiLSTM. We can also observe from the fifth row that the ANN baseline performs poorly. Among the four classical ML models, XGBoost performs best, whereas the decision tree shows the worst performance. XGBoost also performs best considering accuracy and recall over the module-1 dataset. Additionally, the ensemble classifiers XGBoost and random forest show better performance than decision tree and SVM. This establishes the learning and classification efficacy of ensemble classifiers.

3.4.1 Behaviour Ablation Analysis

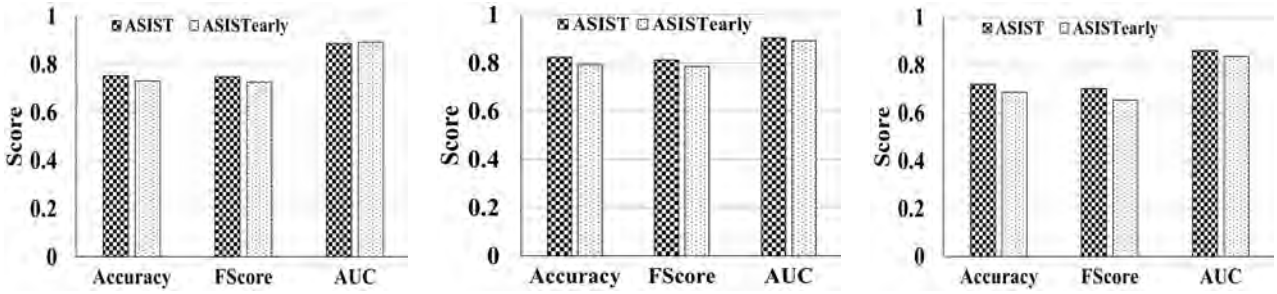
The presented DL model employs different categories of information to model different student behaviours. We use these information sequences separately so that the model can capture patterns from each behaviour. Therefore, this section performs behaviour ablation analysis to investigate the impact of each behaviour component on ASIST's performance. We investigate the effect of a particular behaviour by excluding its underlying information sequence from the proposed model. For example, we exclude the demographic and academic information sequence, $\mathscr{D}$, to analyze its influence on ASIST. The second row of Table 6 presents the results of the underlying updated model. The demographic information shows a moderate adverse impact on ASIST's performance, in line with the findings of existing literature (Waheed et al., 2022; Selwyn, 2020). However, the use of demographic information also includes privacy issues and concerns. This process of ablation analysis is repeated for each behaviour component to analyze their effect on ASIST. We can observe from Table 6 that weekly event count E_W has the highest impact on ASIST's performance. This behaviour component shows the highest impact considering Acc, Fsc, and AUC over module-2 and considering Acc and Fsc over module-1. It exhibits a moderate effect on the module-3 dataset. It reflects that the type of activity also includes the signal to predict its performance. The weekly interaction behaviour, $\mathscr{I}_W$, also adversely impacts ASIST. Table 6 presents the results for behaviour components with the highest impact on ASIST's performance in bold typeface. The table also shows that $\mathscr{I}_D$ has the minimum effect, representing the fact that the daily interaction count arranged weekly trivially affects the performance. Also, E_C and $\mathscr{A}$ show moderate impact, reflecting the fact that assessments and the kind of activity performed over a VLE are good indicators of student performance. An interesting observation is that excluding some behaviour components improves ASIST's performance over the module-1 and module-2 datasets. The impact of component exclusion on the model performance is more significant over module-1 than over the other two modules' datasets. However, it is also significant over the module-2 dataset, considering precision. The ablation analysis result is not given for the $\mathscr{A}$ component for the module-2 dataset because continuous assessment information is unavailable for the module.

3.4.2 Early Prediction of Student Performance

The early prediction of performance in LA is relevant because it gives institutions and concerned teachers sufficient time for early intervention to guide at-risk students. To this end, we extend ASIST to predict student performance early to provide

Table 6. Experimental results for behaviour ablation analysis with exclusion of demographic and academic behaviour ($\mathscr{D}$), weekly interaction behaviour ($\mathscr{I}_W$), weekly event behaviour (E_W), cumulative event performance (E_C), diurnal weekly interaction ($\mathscr{I}_D$), and continuous assessment ($\mathscr{A}$).

Approach	Module-1			Module-2			Module-3		
	Acc	Fsc	AUC	Acc	Fsc	AUC	Acc	Fsc	AUC
ASIST	0.7519	0.7491	0.8858	0.8245	0.8131	0.9040	0.7207	0.7029	0.8614
ASIST-$\mathscr{D}$	0.7299 (0.022 ↓)	0.7273(0.0218 ↓)	**0.8443**(**0.0415** ↓)	0.8100(0.0145 ↓)	0.8097(0.0034 ↓)	0.8947(0.0093 ↓)	**0.6879**(**0.0328** ↓)	0.6869(0.016 ↓)	**0.8344**(**0.027** ↓)
ASIST-$\mathscr{I}_W$	0.7444(0.0075 ↓)	0.7189(0.0302 ↓)	0.8614(0.0244 ↓)	0.8179(0.0066 ↓)	0.8081(0.005 ↓)	0.9010(0.003 ↓)	0.7187(0.002 ↓)	0.6795(0.0234 ↓)	0.8376(0.0238 ↓)
ASIST-E_W	**0.7231**(**0.0288** ↓)	**0.7098**(**0.0393** ↓)	0.8529 (0.0329 ↓)	**0.8052**(**0.0193** ↓)	**0.8015**(**0.0116** ↓)	**0.8925**(**0.0115** ↓)	0.7115(0.0092 ↓)	0.6949(0.008 ↓)	0.8566(0.0048 ↓)
ASIST-E_C	0.7615(−0.0096 ↑)	0.7593(−0.0102 ↑)	0.8899(−0.0041 ↑)	0.8139(0.0106 ↓)	0.8087(0.0044 ↓)	0.9005(0.0035 ↓)	0.7290(0.0083 ↓)	0.7129(−0.01 ↑)	0.8682 (−0.00682 ↑)
ASIST-$\mathscr{I}_D$	0.7568(−0.0049 ↑)	0.7619(−0.0128 ↑)	0.9011(−0.0153 ↑)	0.8084(0.0161 ↓)	0.8154(−0.0023 ↑)	0.9108(−0.0068 ↑)	0.7103(0.0104 ↓)	0.6861(0.0168 ↓)	0.8481(0.0133 ↓)
ASIST-$\mathscr{A}$	0.7423(0.0096 ↓)	0.7331(0.016 ↓)	0.8786(0.0072 ↓)	-	-	-	0.6940(0.0267 ↓)	**0.6741**(**0.0288** ↓)	0.8433 (0.0181 ↓)



(a) Module-1 **(b)** Module-2 **(c)** Module-3

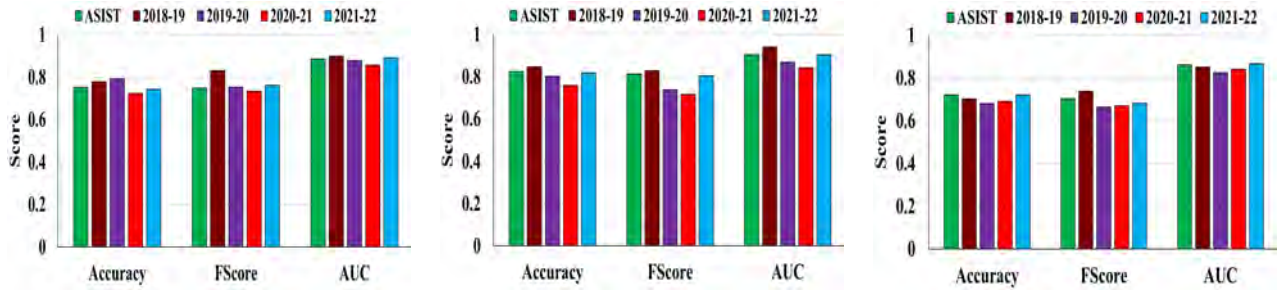
Figure 9. Comparative results between ASIST and ASIST$_{early}$.

timely intervention to at-risk students. To this end, we executed experiments using only the first seven weeks of information for $\mathscr{I}_W$, E_W, E_C, and $\mathscr{I}_D$. We call the underlying updated model ASIST$_{early}$. The demographic and academic behaviour vector is the same because it has static information. We also drop the continuous assessment because midterm assessments are generally held late in the semester. Figure 9 presents the comparative results of ASIST with ASIST$_{early}$. It shows that considering only the first seven weeks of interaction information shows a trivial impact on ASIST. Therefore, ASIST$_{early}$ exhibits comparative performance with ASIST. The figure also indicates that ASIST$_{early}$ shows the highest impact over module-3, where Acc, Fsc, and AUC are reduced by 4, 5, and 3 points, respectively. The figure also demonstrates that impact is greatest considering Fsc and least considering AUC. Finally, this evaluation establishes ASIST's efficacy as an early predictor of student performance.

The ASIST module trained on a full semester of data contributes to the scholarly literature, thanks to its methodological novelty, and from a practical perspective, it enabled us to understand the variables contributing to student success in our context. However, prediction of end-of-course outcomes using a full semester's worth of data is not particularly useful if only the accuracy of the predictions is considered. The potential danger is that by selecting a narrow set of data spanning the first seven weeks of the semester (when we have an opportunity to intervene in students' behaviour), we lose the accuracy of the prediction. The comparison of ASIST (full semester) and ASIST$_{early}$ (first seven weeks) in Figure 9 provided reassurance that we were establishing student intervention on solid methodological ground. The fact that a "real-world" implementation of ASIST with far less data (ASIST$_{early}$) holds in terms of comparative performance with the full model reassures us about the suitability of the model as a tool for student early intervention during the most vulnerable weeks of the transition period to university. Therefore, presenting results on full-semester data as primary findings and early prediction as secondary results balances the contribution of this paper to the literature while providing crucial data for decision making.

3.4.3 Yearly Prediction of Student Performance

The datasets of this study were extracted from data accumulated during four academic years, from 2018/19 to 2021/22. The model was trained and evaluated considering Acc, Fsc, and AUC over the dataset of each academic year to investigate the performance of ASIST in each academic year. Figure 10 shows the underlying results for each academic year over the three datasets. It demonstrates that ASIST trained over the 2018/19 dataset performs best and even outperforms the model trained over the combined dataset, except considering Acc and AUC over module-3. The figure also indicates that ASIST's performance degrades considering all metrics except in one instance when trained over module-1 and module-2 datasets from 2018/19 to 2020/21, but it improves over the 2021/22 dataset. Also, ASIST's performance for each year over the module-3 dataset is random. We cannot ascertain the reason behind this drift in model performance over the different academic years. The model's performance over the dataset of each year, considering AUC, is relatively more stable than Acc and Fsc over all three module datasets.



(a) Module-1 **(b)** Module-2 **(c)** Module-3

Figure 10. Experimental results of ASIST for each academic year over the three datasets.

Table 7. Experimental results of ASIST over the three datasets with training and validation on first-year data and model evaluation over the next three years of datasets.

Approach	Module-1			Module-2			Module-3		
	Acc	Fsc	AUC	Acc	Fsc	AUC	Acc	Fsc	AUC
$ASIST_{2018-19}$	0.7783	0.8312	0.8989	0.8459	0.8285	0.9405	0.7018	0.7375	0.8517
$ASIST_{2019-20}$	0.7012	0.6871	0.7562	0.7345	0.7017	0.8056	0.6381	0.6142	0.7277
$ASIST_{2020-21}$	0.6590	0.6887	0.7756	0.6629	0.6541	0.7518	0.6211	0.5971	0.6951
$ASIST_{2021-22}$	0.6443	0.6734	0.7823	0.5961	0.6221	0.7132	0.6081	0.6354	0.7187

More interestingly, however, we trained the ASIST model over the first-year dataset 2018/19 and investigated its performance over the next three years of datasets to investigate the transfer of model learning capabilities and drift in accuracy. It would be closer to an important use case when the system is deployed in the future. We represent the model for each academic year as $ASIST_{year}$, where "year" is replaced with the corresponding year. Table 7 presents the underlying results considering the accuracy, fscore, and AUC. The investigation shows, however, that the model trained over the year 2018/19 dataset performs poorly when used for prediction in forthcoming academic years. This finding is hardly surprising due to the period in which the investigation happened, capturing the sudden pivot to remote delivery imposed by the COVID-19 pandemic. While in 2018/19, the reliance on VLEs was much lower across all modules from a pedagogical design perspective, by 2021, it had become way more integrated into teaching, learning, and assessment practices across the board in the institution, and the educational experience and expectations around blended learning of student cohorts had radically changed. Also, delivery and assessment design of each module evolve from year to year in multiple ways. This showcases the complexities of EDM in real practice when dealing with contextualized and ever-changing teaching and learning scenarios.

4. Conclusion and Future Works

This study introduces a novel DL model for student performance prediction. The presented model contributes to the existing knowledge by utilizing and integrating ANN components with minimal manual feature engineering to predict students' performance at the end of a module in their first year of study. The proposed model, ASIST, uses a basic set of demographic data, academic records, and VLE interaction information. The results show that (a) integration of stacked BiLSTM and CNN improves the effectiveness of the model, and (b) the attention mechanism shows little impact but still has an impact on the model performance. Empirical investigation through ablation analysis also revealed that VLE information showing students' weekly event performing behaviour has the highest impact on student performance. We also contributed to the existing literature by demonstrating a real-life example of implementation of the model, presenting an early detection version of ASIST, called $ASIST_{early}$, to predict student performance using only the first seven weeks of the dataset. From an institutional practice perspective, the results demonstrated the efficacy of the ASIST model and its practical application as an early predictor of student performance. It also facilitated decision-making around early and contextually relevant interventions for the students of first-year student cohorts.

The demonstrated efficacy of ANN components can enlighten researchers elsewhere to improve the DL model presented. The insights from this data-driven study, which demonstrates the possibility of practical application using a reduced dataset, can also be applied elsewhere as a helpful springboard from which policy stakeholders can formulate pedagogical policies and support guidelines. However, limitations to the study exist because the institution where this study took place collects a limited amount of demographic data, which could have a strong impact on student success. While acknowledging this limitation, it was

comforting to find that the results of ASIST over three datasets demonstrate that student interaction with VLE information and academic data (entry points to college) are good indicators of student performance. It also provides a workaround solution to the ethical problems highlighted by critical authors with the use of personal and demographic data (Selwyn, 2020), by selecting instead a narrow amount of behavioural data, whose use for student support interventions is less likely to provoke unintended negative consequences related to surveillance and labelling.

We also acknowledge the fact that the investigation of accuracy drift shows that the model performs moderately on forthcoming academic years' data. The finding was hardly surprising due to the period in which the investigation happened, capturing the sudden pivot to remote delivery imposed by the COVID-19 pandemic. While in 2018/19, reliance on the VLE was much lower across all modules from a pedagogical design perspective, by 2021, it had become widely integrated into teaching, learning, and assessment practices across the institution, and the educational experience and expectations around blended learning of student cohorts had radically changed. Also, delivery and assessment design of each of the modules evolved from year to year in multiple ways, independent of their reliance on the VLE for delivery. It showcases the complexities of EDM in real practice when dealing with contextualized and ever-changing teaching and learning scenarios and the need for further testing of the model in a post-pandemic setup.

In future research, prediction results using data spanning entire academic years and even programs of study could be used to design guidelines for intelligent decision-making. Using complete datasets of students' journey in higher education could greatly contribute to mining students' behaviour and its relationship with their performance. Also, we can strengthen the analysis by adding other demographic and contextual variables, such as native language and teaching approaches, enabling targeted supports for specific student populations, as shown by previous contextual evidence (Walsh & Risquez, 2020). Also, the ASIST model could be augmented to incorporate student feedback data by applying the latest NLP techniques. However, rather than the time-span of the available dataset, the main contribution of this study is focused on the methodological novelty of the model presented and its demonstrated application as a useful prediction mechanism to enhance students' chances of success.

Declaration of Conflicting Interest

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

Funding

This study was financially supported by the National Forum for the Enhancement of Teaching and Learning, Ireland.

References

Abulaish, M., Kumari, N., Fazil, M., & Singh, B. (2019). A graph-theoretic embedding-based approach for rumor detection in Twitter. In P. Barnaghi, G. Gottlob, Y. Manolopoulos, T. Tzouramanis, & A. Vakali (Eds.), *Proceedings of the 2019 IEEE/WIC/ACM International Conference on Web Intelligence* (WI 2019), 13–17 October 2019, Thessaloniki, Greece (pp. 466–470). ACM. https://doi.org/10.1145/3350546.3352569

Alam, M. M., Mohiuddin, K., Das, A. K., Islam, M. K., Kaonain, M. S., & Ali, M. H. (2018). A reduced feature based neural network approach to classify the category of students. In *Proceedings of the Second International Conference on Innovation in Artificial Intelligence* (ICIAI 2018), 9–12 April 2018, Shanghai, China (pp. 28–32). ACM. https://doi.org/10.1145/3194206.3194218

Ang, K. L.-M., Ge, F. L., & Seng, K. P. (2020). Big educational data and analytics: Survey, architecture and challenges. *IEEE Access*, *8*(7), 116392–116414. https://doi.org/10.1109/ACCESS.2020.2994561

Arbel, N. (2018). *How LSTM networks solve the problem of vanishing gradients* (tech. rep.). Medium. https://medium.datadriveninvestor.com/how-do-lstm-networks-solve-the-problem-of-vanishing-gradients-a6784971a577

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *Proceedings of the Third International Conference on Learning Representations* (ICLR 2015), 7–9 May 2015, San Diego, California, USA (pp. 1–15). ICLR. https://arxiv.org/abs/1409.0473

Christian, T. M., & Ayub, M. (2014). Exploration of classification using NBTree for predicting students' performance. In *Proceedings of the 2014 International Conference on Data and Software Engineering* (ICODSE 2014), 26–27 November 2014, Bandung, Indonesia (pp. 1–6). IEEE Computer Society. https://doi.org/10.1109/ICODSE.2014.7062654

Chui, K. T., Fung, D. C. L., Lytras, M. D., & Lam, T. M. (2020). Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. *Computers in Human Behavior*, *107*(2), 1–7. https://doi.org/10.1016/j.chb.2018.06.032

Dowah, H. A., Al-Samarraie, H., & Fauzy, W. M. (2019). Educational data mining and learning analytics for 21st century higher education: A review and synthesis. *Telematics and Informatics*, *37*(1), 13–49. https://doi.org/10.1016/j.tele.2019.01.007

Essa, A. (2019). Is data dark? Lessons from Borges's "Funes the Memorius." *Journal of Learning Analytics*, *6*(3), 35–42. https://doi.org/10.18608/jla.2019.63.7

Fazil, M., & Abulaish, M. (2018). A hybrid approach for detecting automated spammers in twitter. *IEEE Transactions on Information Forensics and Security*, *13*(11), 2707–2719. https://doi.org/10.1109/TIFS.2018.2825958

Fazil, M., Sah, A. K., & Abulaish, M. (2021). DeepSBD: A deep neural network model with attention mechanism for socialbot detection. *IEEE Transactions on Information Forensics and Security*, *16*(8), 4211–4223. https://doi.org/10.1109/TIFS.2021.3102498

Gasevic, D., Dawson, S., & Siemens, G. (2015). Let's not forget: Learning analytics are about learning. *TechTrend*, *59*, 64–71. https://doi.org/10.1007/s11528-014-0822-x

Hai-tao, P., Ming-qu, F., Hong-bin, Z., Bi-zhen, Y., Jin-jiao, L., Chun-fang, L., Yan-ze, Z., & Rui, S. (2021). Predicting academic performance of students in Chinese-foreign cooperation in running schools with graph convolutional network. *Neural Computing and Applications*, *33*(1), 637–645. https://doi.org/10.1007/s00521-020-05045-9

Hernandez-Blanco, A., Herrera-Flores, B., Tomas, D., & Navarro-Colorado, B. (2019). A systematic review of deep learning approaches to educational data mining. *Complexity*, *2019*(1), 1–22. https://doi.org/10.1155/2019/1306039

Hidalgo, A. C., Ger, P. M., & Valentin, L. D. L. F. (2022). Using meta-learning to predict student performance in virtual learning environments. *Applied Intelligence*, *52*(7), 3352–3365. https://doi.org/10.1007/s10489-021-02613-x

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

Hu, Q., & Rangwala, H. (2019). Reliable deep grade prediction with uncertainty estimation. In *Proceedings of the Ninth International Conference on Learning Analytics and Knowledge* (LAK 2019), 4–8 March 2019, Tempe, Arizona, USA (pp. 76–85). ACM. https://doi.org/10.1145/3303772.3303802

Huang, S., & Fang, N. (2010). Prediction of student academic performance in an engineering dynamics course: Development and validation of multivariate regression models. *International Journal of Engineering Education*, *26*(4), 1008–1017. https://www.ijee.ie/contents/c260410.html

Ihianle, I. K., Nwajana, A. O., Ebenuwa, S., Otuka, R. I., Owa, K., & Orisatoki, M. O. (2020). A deep learning approach for human activities recognition from multimodal sensing devices. *IEEE Access*, *8*(1), 179028–179038. https://doi.org/10.1109/ACCESS.2020.3027979

Jeslet, S., Komarasamy, D., & Hermina, J. J. (2021). Student result prediction in Covid-19 lockdown using machine learning techniques. In *Proceedings of the International Conference on Innovative Technology for Sustainable Development* (ICITSD-2021), 27–29 January 2021, Chennai, India (pp. 1–9). IOP Publisher. https://doi.org/10.1088/1742-6596/1911/1/012008

Karimi, H., Derr, T., Huang, J., & Tang, J. (2020). Online academic course performance prediction using relational graph convolutional neural network. In A. N. Rafferty, J. Whitehill, V. Cavalli-Sforza, & C. Romero (Eds.), *Proceedings of the 13th International Conference on Educational Data Mining* (EDM 2020), 10–13 July 2020, online (pp. 444–450). ACM.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(1), 436–444. https://doi.org/10.1038/nature14539

Li, X., Zhang, Y., Cheng, H., Li, M., & Yin, B. (2022). Student achievement prediction using deep neural network from multi-source campus data. *Complex & Intelligent Systems*, *8*, 5143–5156. https://doi.org/10.1007/s40747-022-00731-8

Long, P., & Siemens, G. (2011). Penetrating the fog: Analytics in learning and education. *Educause Review*, *46*(5), 31–40. https://er.educause.edu/articles/2011/9/penetrating-the-fog-analytics-in-learning-and-education

Marbouti, F., Diefes-Dux, H. A., & Madhavan, K. (2016). Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education*, *103*(9), 1–15. https://doi.org/10.1016/j.compedu.2016.09.005

Migueis, V., Freitas, A., Garcia, P. J., & Silv, A. (2018). Early segmentation of students according to their academic performance: A predictive modelling approach. *Decision Support Systems*, *115*(9), 36–51. https://doi.org/10.1016/j.dss.2018.09.001

Mishra, T., Kumar, D., & Gupta, S. (2014). Mining students' data for performance prediction. In *Proceedings of the Fourth International Conference on Advanced Computing & Communication Technologies* (ACCT 2014), 8–9 February 2014, Rohtak, Haryana, India (pp. 255–262). IEEE Computer Society. https://doi.org/10.1109/ACCT.2014.105

Mubarak, A. A., Cao, H., & Hezam, I. M. (2021). Deep analytic model for student dropout prediction in massive open online courses. *Computers and Electrical Engineering*, *93*(6), 1–14. https://doi.org/10.1016/j.compeleceng.2021.107271

Palacios, C. A., Reyes-Suarez, J. A., Bearzotti, L. A., Leiva, V., & Marchant, C. (2021). Knowledge discovery for higher education student retention based on data mining: Machine learning algorithms and case study in Chile. *Entropy*, *23*(4), 1–23. https://doi.org/10.3390/e23040485

Pascanu, R., Gulcehre, C., Cho, K., & Bengio, Y. (2014). How to construct deep recurrent neural networks. In *Proceedings of the International Conference on Learning Representations* (ICLR 2014), 14–16 April 2014, Banff, Alberta, Canada (pp. 1–13). Arxiv. https://arxiv.org/abs/1312.6026

Priya, S., Ankit, T., & Divyansh, D. (2021). Student performance prediction using machine learning. In D. Hemanth, M. Elhosney, T. Nguyen, & S. Lakshmanan (Eds.), *Proceedings of the Virtual International Conference on Advances in Parallel Computing Technologies and Applications* (ICAPTA 2021), 15–16 April 2021, online (pp. 167–174). IOS Press. https://doi.org/10.3233/APC210137

Qiu, L., Liu, Y., Hu, Q., & Liu, Y. (2018). Student dropout prediction in massive open online courses by convolutional neural networks. *Soft Computing*, *23*(10), 10287–10301. https://doi.org/10.1007/s00500-018-3581-3

Raga, R. C., & Raga, J. D. (2019). Early prediction of student performance in blended learning courses using deep neural networks. In *Proceedings of the Fifth International Symposium on Educational Technology* (ISET 2019), 2–4 July 2019, Hradec Králové, Czechia (pp. 39–43). IEEE. https://doi.org/10.1109/ISET.2019.00018

Ramanathan, K., & Thangavel, B. (2021). Minkowski Sommon Feature Map-based Densely Connected Deep Convolution Network with LSTM for academic performance prediction. *Concurrency and Computation: Practice and Experience*, *33*(13), 1–17. https://doi.org/10.1002/cpe.6244

Rizvi, S., Rientiesa, B., & Khoja, S. A. (2019). The role of demographics in online learning; a decision tree based approach. *Computers & Education*, *137*(4), 32–47. https://doi.org/10.1016/j.compedu.2019.04.001

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

Roy, P. K., Tripathy, A. K., Das, T. K., & Gao, X. (2020). A framework for hate speech detection using deep convolutional neural network. *IEEE Access*, *8*(11), 204951–204962. https://doi.org/10.1109/ACCESS.2020.3037073

Santos, J. L., Klerkx, J., Duval, E., Gago, D., & Rodriguez, L. (2014). Success, activity and drop-outs in MOOCs an exploratory study on the UNED COMA courses. *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge* (LAK 2014), 24–28 March 2014, Indianapolis, Indiana, USA, 98–102. https://doi.org/10.1145/2567574.2567627

Selwyn, N. (2020). Re-imagining "Learning Analytics" . . . a case for starting again? *The Internet and Higher Education*, *46*(7), 1–9. https://doi.org/10.1016/j.iheduc.2020.100745

Shafiq, D. A. K., Marjani, M., Habeeb, R. A. A., & Asirvatham, D. (2022). Student retention using educational data mining and predictive analytics: A systematic literature review. *IEEE Access*, *10*(7), 72480–72503. https://doi.org/10.1109/ACCESS.2022.3188767

Sharada, N., Shashi, M., & Xiong, X. (2018). Modeling student knowledge retention using deep learning and random forests. *Journal of Engineering and Applied Sciences*, *13*(6), 1347–1353. https://doi.org/10.3923/jeasci.2018.1347.1353

Strecht, P., Cruz, L., Soares, C., Mendes-Moreira, J., & Abreu, R. (2015). A comparative study of classification and regression algorithms for modelling students' academic performance. In O. Santos, J. Boticario, C. Romero, M. Pechenizkiy, A. Merceron, P. Mitros, J. Luna, C. Mihaescu, P. Moreno, A. Hershkovitz, S. Ventura, & M. Desmarais (Eds.), *Proceedings of the International Conference on Educational Data Mining* (EDM 2015), 26–29 June 2015, Madrid, Spain (pp. 1–4). International Educational Data Mining Society. https://www.educationaldatamining.org/EDM2015/proceedings/short392-395.pdf

Waheed, H., Hassan, S.-U., Aljohani, N. R., Hardman, J., Alelyani, S., & Nawaz, R. (2020). Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human Behavior*, *104*(1), 1–13. https://doi.org/10.1016/j.chb.2019.106189

Waheed, H., Hassan, S.-U., Nawaz, R., Aljohani, N. R., Chend, G., & Gasevic, D. (2022). Early prediction of learners at risk in self-paced education: A neural network approach. *Expert Systems with Applications*, *213*(9), 1–13. https://doi.org/10.1016/j.eswa.2022.118868

Walsh, J. N., & Risquez, A. (2020). Using cluster analysis to explore the engagement with a flipped classroom of native and non-native English-speaking management students. *The International Journal of Management Education*, *18*(2), 1–9. https://doi.org/10.1016/j.ijme.2020.100381

Wang, W., Yu, H., & Miao, C. (2017). Deep model for dropout prediction in MOOCs. In *Proceedings of the Second International Conference on Crowd Science and Engineering* (ICCSE 2017), 6–9 July 2017, Beijing, China (pp. 26–32). ACM. https://doi.org/10.1145/3126973.3126990

Wasif, M., Waheed, H., Aljohani, N. R., & Hassan, S.-U. (2019). Understanding student learning behavior and predicting their performance. *Cognitive Computing in Technology-Enhanced Learning*, *1*(1), 1–28. https://doi.org/10.4018/978-1-5225-9031-6.ch001

Xing, W., & Du, D. (2018). Dropout prediction in MOOCs: Using deep learning for personalized intervention. *Journal of Educational Computing*, *57*(3), 547–570. https://doi.org/10.1177/0735633118757015

Yagci, M. (2022). Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, *9*(11), 1–19. https://doi.org/10.1186/s40561-022-00192-z

Yin, W., Kann, K., Yu, M., & Schutze, H. (2017). Comparative study of CNN and RNN for natural language processing. *arXiv:1702.01923v1*. https://doi.org/10.48550/arXiv.1702.01923

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, *13*(3), 55–75. https://doi.org/10.1109/MCI.2018.2840738

Zhang, L., Qi, G.-J., Wang, L., & Luo, J. (2019). AET vs. AED: Unsupervised representation learning by auto-encoding transformations rather than data. In *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR 2019), 15–20 June 2019, Long Beach, California, USA (pp. 2542–2550). IEEE/CVF. https://doi.org/10.1109/CVPR.2019.00265