The Journal of Supercomputing (2023) 79:4146–4163
https://doi.org/10.1007/s11227-022-04804-w

Evaluation of e-learners' concentration using recurrent neural networks

Young-Sang Jeong[1] · Nam-Wook Cho[2]

Accepted: 28 August 2022 / Published online: 22 September 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract

Recently, interest in e-learning has increased rapidly owing to the lockdowns imposed by COVID-19. A major disadvantage of e-learning is the difficulty in maintaining concentration because of the limited interaction between teachers and students. The objective of this paper is to develop a methodology to predict e-learners' concentration by applying recurrent neural network models to eye gaze and facial landmark data extracted from e-learners' video data. One hundred eighty-four video data of ninety-two e-learners were obtained, and their frame data were extracted using the OpenFace 2.0 toolkit. Recurrent neural networks, long short-term memory, and gated recurrent units were utilized to predict the concentration of e-learners. A set of comparative experiments was conducted. As a result, gated recurrent units exhibited the best performance. The main contribution of this paper is to present a methodology to predict e-learners' concentration in a natural e-learning environment.

Keywords E-learning · Concentration · E-learner · Recurrent neural networks (RNN) · Gated recurrent units(GRU) · Long short-term memory (LSTM)

✉ Nam-Wook Cho
nwcho@seoultech.ac.kr

Young-Sang Jeong
videorighter@seoultech.ac.kr

[1] Department of Data Science, Seoul National University of Science and Technology, 232 Gongreung-ro, Nowon, Seoul 01811, South Korea

[2] Department of Industrial Engineering, Seoul National University of Science and Technology, 232 Gongreung-ro, Nowon, Seoul 01811, South Korea

1 Introduction

The COVID-19 pandemic has forced many schools and universities to switch to e-learning, which is also known as online distance learning [15]. E-learning involves using the Internet and other related technologies for learning, teaching, and regulating courses in an organization [1]. E-learning has been widely accepted as a significant educational platform not only by organizations but also by teachers and students [27]. Besides the COVID-19 pandemic, the expansion of e-learning is due to the benefits of e-learning itself. The main advantages of e-learning include a variety of learning materials, cost-effectiveness, and self-pacing [1, 28, 32]. Among the many advantages of e-learning, time and place flexibility are among the most crucial advantages, largely contributing to the spread of e-learning.

Despite the advantages of e-learning, one of the main disadvantages is the lack of interaction between teachers and students. It is evident that some aspects of education, including learning with peers and interactions with professors, cannot be replaced by online [15]. These disadvantages often lead to the ineffectiveness of education, leading to the learning loss in many students. In particular, e-learning requires immense self-motivation and self-discipline from students, which poses significant challenges. Several attempts have been made to overcome these limitations. Enhancing the interaction between teachers and students and installing monitoring systems for learning progress are often considered appropriate approaches to the limitations.

Concentration plays an essential role in learning. This has become more critical for online education. Effective and efficient assessment of the concentration of e-learners is crucial for providing necessary feedback to learners and tutors. The development of effective and customizable intelligent tutoring systems (ITS) has been proposed to understand the cognitive state of a learner's knowledge, emotions, and concentration [17]. De Carolis et al. [8] argued that it is important to develop personalized e-learning environments that can customize the learning experience of students.

This paper aims to develop a methodology to predict e-learners' concentration by applying recurrent neural network models to eye gaze and facial landmark data extracted from e-learners' video data. One hundred eighty-four video data of ninety-two e-learners were obtained, and their features were extracted using the OpenFace 2.0 toolkit. The data were then divided into 5-s units, and their concentration levels were labeled by education experts. The recurrent neural network(RNN), long short-term memory(LSTM), and gated recurrent unit(GRU) models were utilized in the comparative experiments. It is expected that the proposed methodology can predict the concentration level of students in a natural e-learning environment, thereby increasing the effectiveness of education by facilitating feedback between students and e-learning systems.

The structure of the paper is as follows. The relevant theories and literature are reviewed in Sect. 2. Section 3 explains the proposed RNN-based concentration classification model for e-learners. The experimental results are presented in Sect. 4. Finally, Sect. 5 discusses the benefits and limitations of our methodology.

2 Literature review

2.1 Recurrent neural networks

Recurrent neural networks (RNNs) are a variant of artificial neural networks (ANNs). They are capable of selectively passing information across sequence steps while processing sequential data one element at a time [22]. By overcoming a major limitation of ANN, the assumption of independence among data, RNNs have been proposed to deal with sequential data. RNNs can model input and/or output consisting of sequences of elements that are not independent. Furthermore, recurrent neural networks can simultaneously model sequential and time dependencies on multiple scales.

RNNs have been successfully applied to numerous applications, including time-series prediction [34, 37], speech recognition [9, 16], image classification [26], and video analysis [40], where a model effectively captures the dynamics of sequences via cycles in the network nodes.

Training time-series data often requires dealing with input information in the past and future of a specific time frame [24], for which bidirectional RNNs have been proposed. It splits the state neurons of a regular RNN into a forward state (positive time direction) and a backward state (negative time direction). Outputs from forward states are not connected to inputs of backward states and vice versa. As a bidirectional RNN has shown good performance in modeling time-series data, it has been adopted in our model.

Another limitation of RNNs is the vanishing gradient of traditional RNNs. To overcome this limitation, Hochreiter and Schmidhuber [12] introduced a long short-term memory(LSTM) model primarily to overcome the problem of vanishing gradients of RNNs. Unlike traditional RNNs, LSTM has feedback connections, thereby better dealing with the entire sequence of data.

Gated recurrent units (GRUs) are another notable approach to vanishing gradients. GRUs create shortcut paths that bypass multiple temporal steps [7]. These shortcuts allow the error to be back-propagated easily, minimizing vanishing as a result of passing through multiple bounded nonlinearities, thus reducing the difficulty due to vanishing gradients.

A GRU adaptively makes each recurrent unit capture the dependencies of different time scales. Similar to the LSTM unit, the GRU has gating units that modulate the flow of information inside the unit without having separate memory cells. While the LSTM consists of input, forget, and output gates, the GRU has a small number of parameters because this function is performed at the reset and updated gates without memory cells, increasing the computational efficiency. In this study, three models using bidirectional RNNs, LSTM, and GRU were proposed. Comparison experiments were conducted using video data collected from real e-learners.

2.2 Related works on e-learning

Recently, there has been an increasing interest in research within areas related to e-learning. To overcome the lack of interaction between teachers and students, Troussas et al. [36] proposed an alternative educational tool over a Social Network Service for students. Lopez et al. [24] presented a comparative study of the effectiveness of face-to-face and remote educational escape rooms. Stevens [33] presented a comparative study on the learning outcomes within and between online and face-to-face education.

Face retrieval or recognition is essential in computer vision and e-learning environments as well [25]. Lin et al. [21] proposed a cloud-based face video retrieval system with deep learning. Cognitive theory has also been utilized in the research related to e-learning. Wen et al. [38] presented chaos optimization cognitive learning model, where the learning process of distance learning has been formulated into a multi-objective optimization problem. Liu and Peng [23] proposed an online user focus evaluation system, where eye tracking and face recognition technologies were combined with the cognitive theory to evaluate the concentration of students.

Several attempts have been made to determine the concentration level of e-learners based on their behavior and biological information. Asteriadis et al. [2] presented a neuro-fuzzy inference system that utilizes the position and movement of the eyes and irises of an e-learner to determine the concentration level in the context of reading an electronic document. To monitor the concentration level of e-learners, Lee et al. [19] utilized the pupillary responses and eye-blinking patterns of students. A one-class support vector machine (SVM) was used to determine the concentration levels. Li et al. [20] utilized data collected by a webcam and a mouse to determine the concentration levels of e-learners. SVM techniques were applied to identify useful features for recognizing human attention levels.

Convolutional neural networks (CNNs) have been widely used for image classification [35]. They have also been used to determine the concentration levels of students. Hasnine et al. [10] utilized CNNs to detect the concentration level of students. Six types of basic emotions were extracted using a pre-trained CNN, and those were used to detect the concentration level of students in a virtual classroom. Sharma et al. [31] proposed a CNN-based machine learning system for student engagement detection using emotion analysis, eye tracking, and head movement by using a web camera. Although the CNN-based methods are noteworthy, they still have their weaknesses; as they are based on still images, they are unable to capture the sequential and temporal nature of e-learners' responses. As a result, the actual e-learning environment can hardly be represented. Therefore, the application of RNNs has attracted the interest of researchers to effectively capture the dynamics of sequential data obtained from videos.

Sharma et al. [30] presented LIVELINET to estimate the liveliness of educational videos. While LIVELINET combines audio and visual information to predict the liveliness of educational videos using convolutional neural networks and LSTM, it does not utilize the behavior and biological information of e-learners.

De Carolis et al. [8] presented a method to determine the concentration, also referred to as engagement, of e-learners using LSTM. The OpenFace Toolkit was used to extract the necessary features from the video data. LSTM was applied to the features consisting of eye gaze, facial landmark, head pose, and facial expressions, and the degree of concentration was predicted. The subjective evaluation of the engagement from a questionnaire based on the psychological notion of "flow" was used in this study. Although the proposed method is noteworthy, the limited data set and subjective nature of a questionnaire can pose limitations in terms of practical applications; students had to answer questionnaires to assess their own engagement. In practical applications, the need for questionnaires or special instruments requires additional costs, causing difficulties in real e-learning environments. Therefore, research is needed to determine the degree of learning concentration by extracting various features using only the videos obtained in an actual e-learning environment.

3 Methods

3.1 Overview

Figure 1 shows the overall procedures of our study. First, video data of e-learners were collected and preprocessed as sequential temporal data so that they could be used as input data for RNN. Each dataset was labeled with its concentration levels prior to the application of supervised learning tasks. Three different RNN models, vanilla RNN, LSTM, and GRU, were used in the experiment, along with an SVM baseline model.

3.2 Participants

Ninety-two undergraduate students between the ages of 20 and 31 participated in the experiment. Prior to the video recording, a consent form for providing personal information and utilizing information was provided to the participants. The shooting resolution was 480×640 pixels, with a frame rate of 30 frames per second. During the filming process, interference with participants was minimized; participants were guided only in the direction of the experiments.

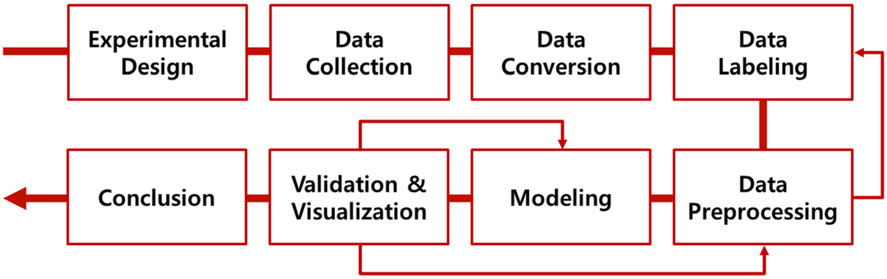


Fig. 1 Overview of procedures

3.3 Procedures

Two distinct online lectures were used for the experiments. An interesting lecture and an interest-inhibiting lecture were shown to collect learners' different behaviors. During the experiments, participants watching the lectures were unaware of the differences between the lectures.

The first lecture to evoke interest was a famous history lecture. The second lecture on mathematics intended to evoke boredom in students was selected from MIT Open Courseware. All participants watched the first lecture for about 9 min and the second for approximately 15 min. To effectively control the environmental variables, video recording was performed only in a laboratory with a camera located in the upper center of the monitor. One hundred eighty-four video data were obtained from the participants.

3.4 Data preprocessing

The video data were converted to structured data using the OpenFace Toolkit, a tool for facial behavior analysis [3]. The output data provided by the OpenFace Toolkit consist of a point distribution model (PDM) of facial landmark location, head pose, eye gaze, facial expressions, and facial action units (AU). Among the data obtained from the Toolkit, facial landmark location, head pose, and eye gaze information were mainly utilized in our model. Figures 2 and 3 represent 2D eye landmarks and 2D facial landmarks, respectively, as detected by OpenFace Toolkit.

Each PDM data point encompasses three-dimensional coordinates (X, Y, Z). Among the PDM data, sixteen iris data points from the eye landmark data and seventeen face contour data points from the facial landmark data were utilized. The data points #20 ~ #27 and #48 ~ #55 were used from the eye landmarks in Fig. 2. The data points #0 ~ #16 were used from the facial landmarks in Fig. 3. In addition, two eye gaze data with (X, Y, Z) and one eye gaze direction data with (X, Y) were used in our model. A total of 109 features were used, and their details are presented in Table 1.

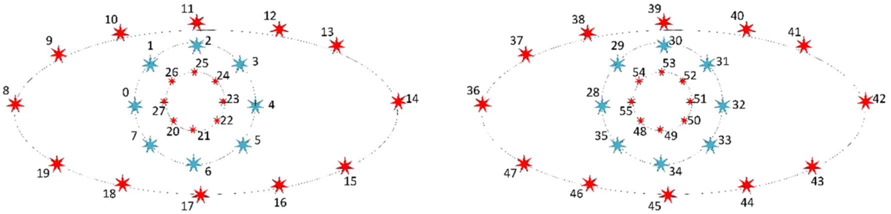


Fig. 2 2D eye landmarks as detected by OpenFace

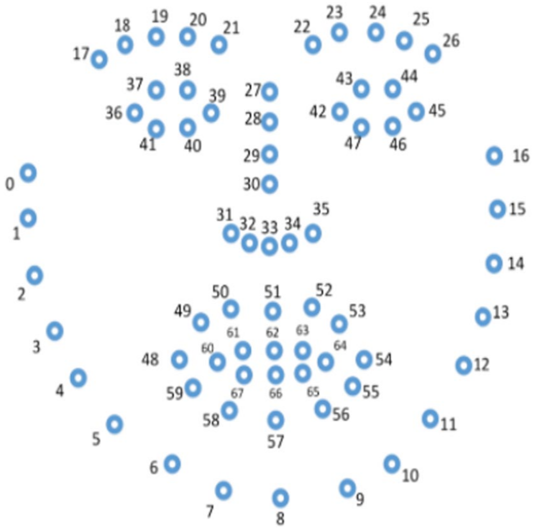


Fig. 3 2D facial landmarks as detected by OpenFace

Table 1 Description of data features

Data source	Description	Type	# of features
Eye landmark data	16 iris data points	3D	48
Face landmark data	17 face contour data points	3D	51
Eye gaze data	2 eye gaze data	3D	6
	Eye gaze direction data	2D	2
Basic	Face detection confidence score, face detection success rate	Numerical	2

3.5 Data sets

The video data were divided into 5-s units. Each video was labeled as binary, depending on the concentration of the learner of the video. Three education experts reviewed the videos, and a voting method was used in the labeling process.

Even though the recording proceeded with the learner located in the center, each participant had a different location on the screen and often changed their position during the experiment. Thus, data scaling was conducted so that the head positions of the participants as located equidistant as much as possible.

The video data were preprocessed to obtain 150 frames for each. As each video data contain motion noise for shooting preparation, we took the video from t = 150. A total of 27,026 data were used in the experiment. Each 5-s clip was modeled as a temporal sequence$\{x_1, x_2, \ldots., x_t, \ldots., x_T\}$, where x_t (t = 1, 2, …,150) is a vector representing the input data at time instant t. The data were divided into training, validation, and test datasets at a ratio of 8:1:1.

3.6 Modeling

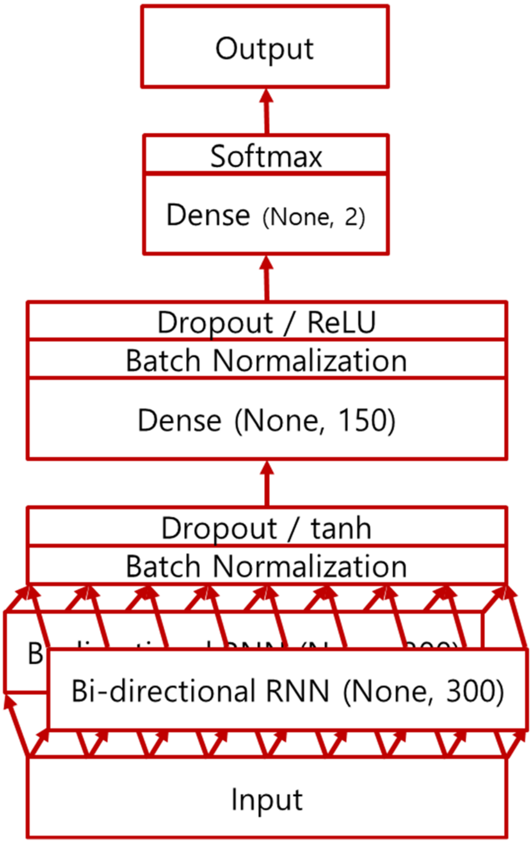
Figure 4 illustrates the overall architecture of the proposed RNN model. The sequential temporal data are fed into the bidirectional RNN layers, go through the normalization process, and pass through deep neural network layers to generate a binary classification of concentration levels.

(1) RNN

For given a sequence $x=(x_1; x_2;\ldots\ldots; x_T)$, the recurrent state h_t is determined from the recurrent state h_{t-1} at the previous time and the current input x_t through a transition function [7, 15] and, consequently, the output of the RNN's cell state (o_t) is determined:

$$h_t = f\left(x_t, h_{t-1}; \theta\right) = \tanh\left(W_x x_t + W_h h_{t-1} + b_h\right) \tag{1}$$

Fig. 4 Architecture of RNN



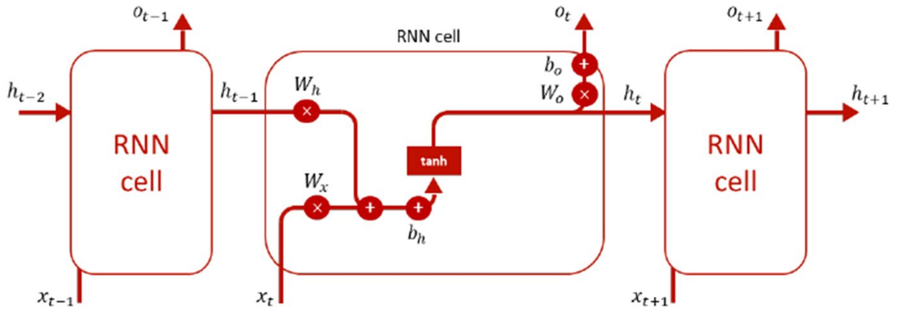


Fig. 5 Illustration of RNNs

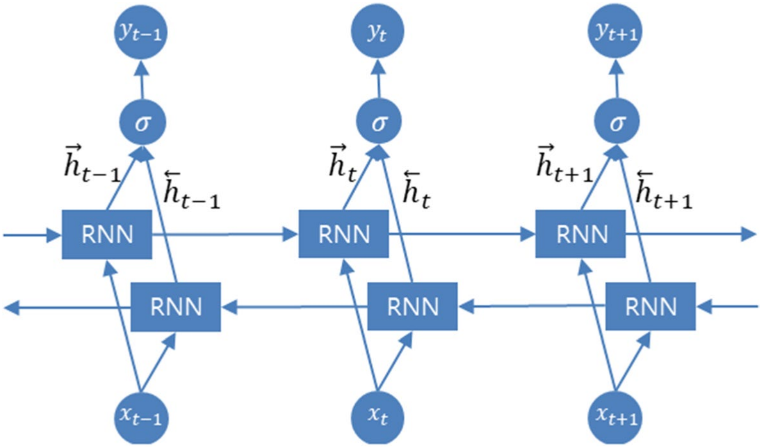


Fig. 6 Illustration of bidirectional RNNs

$$o_t = W_o h_t + b_o \tag{2}$$

where $h_0 = 0$ and θ are the parameters of the function f. W and b are the weight matrix and the bias vector between the input and the output layers. The hyperbolic tangent activation function, *tanh()*, guarantees that the output(o_t) of the RNN unit should be within the range of (− 1, 1). Figure 5 shows the structure of RNNs used in our experiment.

As illustrated in Fig. 6, a bidirectional RNN computes both the forward hidden sequence $\vec{h}$ and the backward hidden sequence $\overleftarrow{h}$ [9, 29]. The output sequence is given by iterating the backward layer from $t = T$ to 1 and the forward layer from $t = 1$ to T.

(2) LSTM

As shown in Fig. 7, each LSTM unit maintains a memory C_t at time t [7]. The activation of the LSTM unit h_t is

$$h_t = o_t * \tanh\left(C_t\right), \tag{3}$$

where o_t is an output gate. The output gate is determined by

$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right), \tag{4}$$

where σ is a logistic sigmoid function and *bo* is a diagonal matrix. Then, the memory cell C_t and the new memory cell $\widetilde{C}_t$ are

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t, \text{and} \tag{5}$$

$$\widetilde{C}_t = tanh\left(W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right). \tag{6}$$

A forget gate f_t and an input gate i_t are given by

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \text{and} \tag{7}$$

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right). \tag{8}$$

(3) GRU

As shown in Fig. 8, the GRU [5, 7] is designed to adaptively capture dependencies of different time scales using a more sophisticated transition function. The transition function h_t is given as

$$h_t = \left(1 - z_t\right) \odot \widetilde{h}_t + z_t \odot h_{t-1}, \tag{9}$$

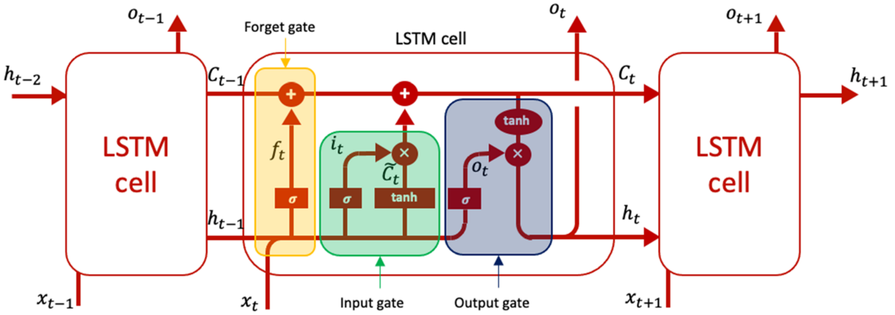


Fig. 7 Illustration of LSTM. *i*, *f*, and *o* represent the input, forget, and output gates, respectively. *C* is the memory cell and $\widetilde{C}$ is the new memory cell

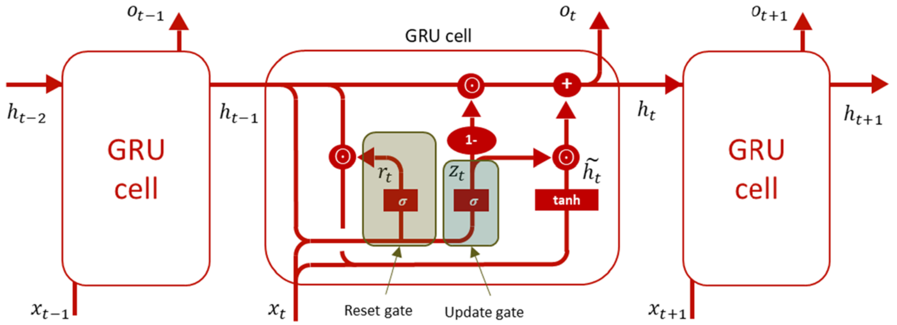


Fig. 8 Illustration of gated recurrent units. *r* and *z* are the reset and update gates, respectively

where

$$z_t = \sigma\left(W_{xz}x_t + W_{hz}h_{t-1} + b_z\right), \tag{10}$$

$$r_t = \sigma\left(W_{xr}x_t + W_{hr}h_{t-1} + b_r\right), \text{ and} \tag{11}$$

$$\widetilde{h}_t = tanh\left(W_{xh}x_t + W_{hh}\left(r_t \odot h_{t-1}\right)\right). \tag{12}$$

Note that, ⊙ denotes an element-wise multiplication operator.

(4) Configurations

Six configurations were used in training, including one to two layers for bidirectional RNNs and one to three layers for deep neural networks. RNNs, LSTM, and GRU were applied to each configuration, along with batch normalization and dropout. Note that, both LSTM and GRU were constructed on the basis of the bidirectional RNNs.

Prior to comparative experiments, the following requirements were considered to select a proper baseline classifier. First, the classifier needed to perform well with a limited number of data samples while minimizing overfitting. Secondly, the classifier is required to classify elements nonlinearly [18]. In addition, the classifier needs to be utilized in related works [19, 20]. Upon a review of machine learning approaches based on relevant literature, SVMs were identified as the baseline classifier most suited to meet the requirements.

By standardizing the inputs to a layer for each mini-batch, batch normalization stabilizes the learning process and accelerates the training of deep neural nets. It eliminates internal covariate shifts and changes in the distributions of the internal nodes of a deep network [14].

In the course of optimizing the binary cross-entropy loss function, as shown in Eq. (12), Nesterov-accelerated adaptive moment estimation, or Nadam, was used.

Nadam is an extension of the adaptive moment estimation [17] algorithm that incorporates Nesterov's accelerated gradient (NAG) and can result in better performance of the optimization algorithm [6].

$$\text{Loss Function} = -\sum_{i=1}^{C=2} t_i \log\left(f\left(s_i\right)\right) = -t_1 \log\left(f\left(s_1\right)\right) - \left(1 - t_1\right)\log(1 - f\left(s_1\right)) \tag{13}$$

Learning rate is "the single most important hyper-parameter" [4] in training neural networks. Learning rate decay (lrDecay) is a de facto technique for training modern neural networks, where we adopt an initially large learning rate and then decay it by a certain factor after pre-defined epochs. Popular deep networks such as ResNet [11] and DenseNet [13] are all trained by Stochastic Gradient Descent (SGD) with lrDecay.

As it has been empirically observed that learning rate decay helps to learn complex patterns [39], the learning rate decay is set to 1×10^{-5} with a learning rate of 1×10^{-4}. Even though the initial epoch was set to 300, the training terminated if the validation loss during 50 epochs did not decrease. The batch size was set to 256.

The specifications of the computational machine include an AMD Ryzen 7 3.20 GHz processor with 32 GB of RAM, and an NVIDIA GeForce RTX 3070 GPU running the 64-bit Windows 10 operating system. The Keras Python library was used on top of a source build of TensorFlow.

4 Experimental results

The experimental results are summarized in Table 2. Overall, the RNNs performed better than the baseline SVM method. Among the RNNs, a GRU method with one RNN layer and two FF layers provided the best performance, with an accuracy of 0.8431. The recall and precision of the GRU method were 0.8512 and 0.9077, respectively.

Figures 9–11 present the comparison results of RNN models, which illustrate an accuracy and loss plot and ROC curves. Figure 9 shows the accuracy/loss and AUC plot of Vanilla bidirectional RNN with two RNN layers and three FF layers. It shows that the validation loss is minimum with 90 epochs, and the AUC is 0.8664.

Figure 10 shows the accuracy/loss and AUC plot of the LSTM with one RNN layer and two FF layers. It shows that the validation loss is minimum with 15 epochs, and the AUC is 0.9076. Note that, the LSTM model tends to converge to the minimum loss relatively faster than the other two models, but it shows overfitting after certain epochs.

Figure 11 shows the accuracy/loss and AUC plot of the GRU with one RNN layer and one FF layers. It shows that the validation loss is minimum with 42 epochs, and the AUC is 0.9210. While the GRU model reaches the minimum loss gently, it shows instability after certain epochs.

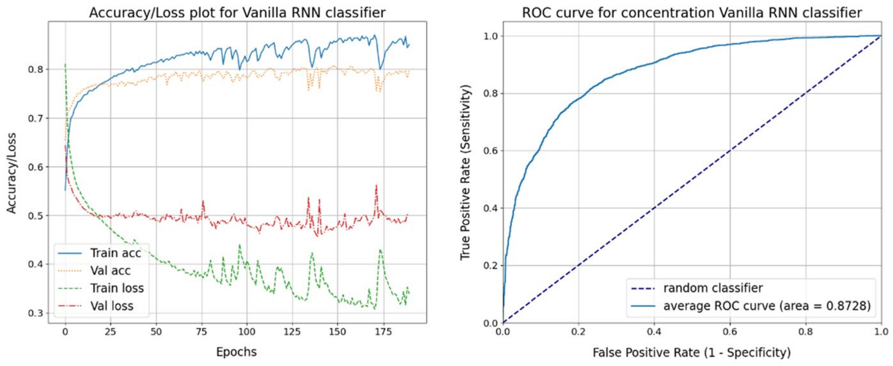
Table 2 Summary of Experiments. “True” means that a participant of the video is concentrating on learning, and “False” means otherwise. Each experiment has been repeated five times. The mean and standard deviation are reported in the table

Model	RNN layer	FF layer	Concentration	Accuracy	AUC	Recall	Precision	F1-score
SVM	NA	NA	True	0.7141	0.6889	0.6899	0.824	0.751
			False			0.7544	0.5935	0.6644
Vanilla RNN	1	1	True	0.7575 ± 0.0249	0.8152 ± 0.0709	0.7713 ± 0.0473	0.8488 ± 0.0193	0.8077 ± 0.0318
			False			0.728 ± 0.0172	0.6179 ± 0.0372	0.6676 ± 0.0171
	1	2	True	0.7805 ± 0.0109	0.8623 ± 0.0058	0.8048 ± 0.0119	0.8568 ± 0.0075	0.83 ± 0.0077
			False			0.7324 ± 0.0131	0.6535 ± 0.0258	0.6905 ± 0.0185
	1	3	True	0.7762 ± 0.0042	0.8585 ± 0.0063	0.8036 ± 0.0077	0.8497 ± 0.0052	0.826 ± 0.0021
			False			0.723 ± 0.0028	0.6536 ± 0.0186	0.6864 ± 0.01
	2	1	True	0.7803 ± 0.0105	0.8604 ± 0.0088	0.8163 ± 0.0101	0.837 ± 0.0204	0.8264 ± 0.0096
			False			0.7171 ± 0.0219	0.6858 ± 0.025	0.7007 ± 0.0139
	2	2	True	0.7786 ± 0.0063	0.863 ± 0.008	0.8097 ± 0.006	0.8441 ± 0.0087	0.8265 ± 0.0052
			False			0.7205 ± 0.0107	0.6692 ± 0.0136	0.6938 ± 0.0092
	2	**3**	True	**0.783 ± 0.0068**	**0.8664 ± 0.0066**	**0.8145 ± 0.0093**	**0.8454 ± 0.0069**	**0.8296 ± 0.0046**
			False			**0.725 ± 0.0081**	**0.6789 ± 0.021**	**0.701 ± 0.0125**
LSTM	**1**	**1**	True	**0.8318 ± 0.006**	**0.9076 ± 0.004**	**0.8452 ± 0.0117**	**0.8951 ± 0.0083**	**0.8693 ± 0.0036**
			False			**0.8062 ± 0.0081**	**0.7262 ± 0.0262**	**0.7639 ± 0.0123**
	1	2	True	0.8241 ± 0.0037	0.9038 ± 0.001	0.8437 ± 0.0069	0.8821 ± 0.0072	0.8624 ± 0.0027
			False			0.7875 ± 0.0078	0.7274 ± 0.0158	0.7561 ± 0.0072
	1	3	True	0.8226 ± 0.0067	0.9033 ± 0.003	0.8334 ± 0.0056	0.8953 ± 0.0135	0.8632 ± 0.0059
			False			0.8013 ± 0.0187	0.7016 ± 0.0143	0.7479 ± 0.0081

Table 2 (continued)

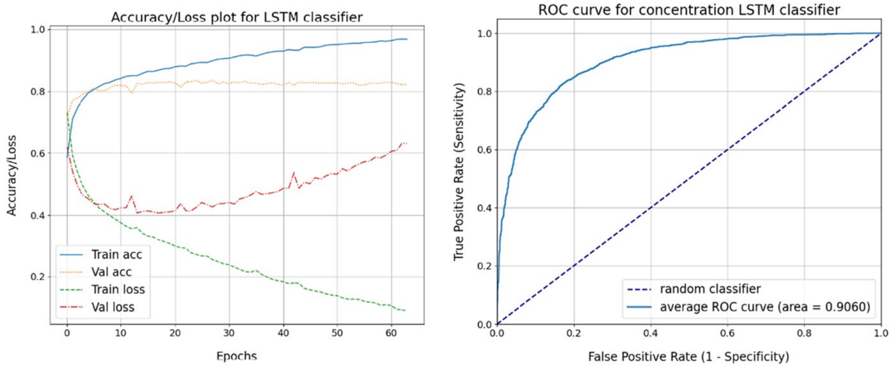
Model	RNN layer	FF layer	Concentration	Accuracy	AUC	Recall	Precision	F1-score
	2	1	True	0.8084 ± 0.003	0.887 ± 0.0035	0.8313 ± 0.0068	0.8701 ± 0.0128	0.8502 ± 0.0034
			False			0.7656 ± 0.0137	0.7055 ± 0.0183	0.7341 ± 0.0054
	2	2	True	0.8005 ± 0.0063	0.8771 ± 0.0026	0.8232 ± 0.0057	0.8672 ± 0.0164	0.8445 ± 0.0063
			False			0.7577 ± 0.0196	0.6893 ± 0.0164	0.7216 ± 0.0064
	2	3	True	0.8077 ± 0.0052	0.8822 ± 0.005	0.8227 ± 0.004	0.8825 ± 0.0085	0.8515 ± 0.0044
			False			0.7773 ± 0.012	0.683 ± 0.0091	0.7271 ± 0.0067
GRU	**1**	**1**	True	**0.8431 ± 0.0056**	**0.921 ± 0.0035**	**0.8512 ± 0.0063**	**0.9077 ± 0.0106**	**0.8785 ± 0.0047**
			False			**0.8273 ± 0.0144**	**0.7355 ± 0.0148**	**0.7786 ± 0.0079**
	1	2	True	0.8412 ± 0.0031	0.9185 ± 0.0045	0.8521 ± 0.0059	0.9027 ± 0.0146	0.8766 ± 0.0038
			False			0.8207 ± 0.0193	0.7387 ± 0.0164	0.7772 ± 0.0014
	1	3	True	0.8368 ± 0.0065	0.9166 ± 0.0043	0.8462 ± 0.0064	0.9032 ± 0.0117	0.8737 ± 0.0055
			False			0.8186 ± 0.0164	0.7262 ± 0.0149	0.7695 ± 0.0089
	2	1	True	0.8292 ± 0.0065	0.9112 ± 0.0029	0.8401 ± 0.0092	0.8979 ± 0.0141	0.8679 ± 0.0054
			False			0.8082 ± 0.0178	0.7148 ± 0.0225	0.7583 ± 0.0106
	2	2	True	0.8266 ± 0.0061	0.9083 ± 0.0038	0.8349 ± 0.0054	0.9007 ± 0.009	0.8666 ± 0.0049
			False			0.8097 ± 0.0134	0.7032 ± 0.0121	0.7526 ± 0.0085
	2	3	True	0.8349 ± 0.0064	0.9125 ± 0.0031	0.8418 ± 0.0073	0.9063 ± 0.0056	0.8728 ± 0.0047
			False			0.8209 ± 0.0088	0.716 ± 0.0155	0.7648 ± 0.0104

Bolded numbers are the best performance



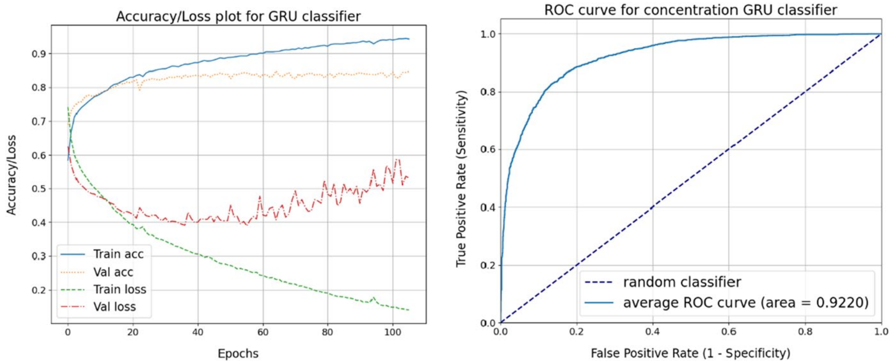
(a) Accuracy/loss **(b)** AUC

Fig. 9 Accuracy/loss and AUC plot of RNNs



(a) Accuracy/loss **(b)** AUC

Fig. 10 Accuracy/loss and AUC plot of LSTM



(a) Accuracy/loss **(b)** AUC.

Fig. 11 Accuracy/loss and AUC plot of GRU

5 Conclusion

This study explored the use of RNNs to determine the concentration of students in an e-learning environment. Three RNN models, namely bidirectional RNNs, LSTM, and GRUs, were utilized in our model, along with an SVM baseline model. A total of 27,026 datasets obtained in a natural e-learning environment were used in the experiment. Overall, the RNN models demonstrated that they are suitable for predicting the concentration of students, showing better performance than the baseline model. Among the RNN models, GRUs exhibited the best performance, with an overall accuracy of 84.3%.

The contributions of this work are summarized as follows. Our main contribution lies in designing a prediction model for e-learners' concentration in an actual e-learning environment. Our model is one of the studies that are implemented in the most actual e-learning environment. The proposed model does not require any additional questionnaires or special instruments, which easily enables its implementation in an online education system. The detailed procedures of a model, including data collection, preprocessing, data modeling, and testing, were presented, which can stimulate research in this area. To effectively evaluate e-learner's concentration, the architectures and configurations of the bidirectional RNN, LSTM, and GRU models have been proposed. In addition, comparative experiments were conducted to demonstrate the usefulness of the proposed model. Finally, the applicability of the proposed models has been examined.

Despite our contributions, we cannot help admitting the limitations of our approach. Significantly, our model was applied only to well-structured process models. Video data should be transformed and preprocessed for application to our model, which requires additional time and effort in an actual application. Automation of such processes would facilitate the usability of the proposed system.

As our approach has focused on the RNNs, LSTM, and GRU, expanding our model to other architectures such as CNNs and temporal convolutional networks (TCNs) would be our future work. Another limitation of our approach is the robustness of the model. The experiments were conducted in a controlled environment. However, real e-learning environments involve unexpected situations, which were not considered in our model. For example, students can excessively change their postures or even leave their seats during an online lecture, which may cause problems in our model. Thus, the development of a model that can effectively handle such situations would be a suitable topic for future research.

Acknowledgements This study was supported by the Research Program funded by the SeoulTech (Seoul National University of Science and Technology).

Funding The data that support the findings of this study are available from the corresponding author upon reasonable request (U2021-0234).

Declarations

Conflict of interest All authors declare that they have no conflicts of interest.

References

1. Arkorful V, Abaidoo N (2015) The role of e-learning, advantages and disadvantages of its adoption in higher education. Int J Instr Technol Distance Learn 12:29–42
2. Asteriadis S, Tzouveli P, Karpouzis K, Kollias S (2009) Estimation of behavioral user state based on eye gaze and head pose — application in an e-learning environment. Multim Tools Appls 41:469–493. https://doi.org/10.1007/s11042-008-0240-1
3. Baltrusaitis T, Zadeh A, Lim YC, Morency LP (2018) Openface 2.0: Facial behavior analysis toolkit. In: 2018 13th IEEE International Conference on Automatic Face and Gesture Recognition, pp 59–66
4. Bengio, Y (2012) Practical recommendations for gradient-based training of deep architectures. In: Neural networks: Tricks of the trade, pp. 437–478
5. Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078
6. Choi D, Shallue CJ, Nado Z, Lee J, Maddison CJ, Dahl GE (2019) On empirical comparisons of optimizers for deep learning. arXiv preprint arXiv:1910.05446
7. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555
8. De Carolis B, Errico FD, Macchiarulo N, Palestra G (2019) Engaged Faces: Measuring and Monitoring Student Engagement from Face and Gaze Behavior. In: IEEE/WIC/ACM International Conference on Web Intelligence-Companion, pp 80–85
9. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing pp. 6645–6649
10. Hasnine MN, Bui HT, Tran TTT, Nguyen HT, Akçapınar G, Ueda H (2021) Students' emotion extraction and visualization for engagement detection in online learning. Procedia Comput Sci 192:3423–3431
11. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778
12. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
13. Huang G, Liu Z, Van Der Maaten L, Weinberger, KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708
14. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456
15. Kalenzi C, Back D, Yim M (2020) The future of online education: lessons from South Korea. World Economic Forum. https://www.weforum.org/agenda/2020/11/lessons-from-south-korea-on-the-future-of-online-education/. Accessed 12 June 2021.
16. Kim HG, Jang GJ, Oh YH, Choi HJ (2020) Speech and music pitch trajectory classification using recurrent neural networks for monaural speech segregation. J Supercomput 76:8193–8213. https://doi.org/10.1007/s11227-019-02785-x
17. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980
18. Koo B, La S, Cho NW, Yu Y (2019) Using support vector machines to classify building elements for checking the semantic integrity of building information models. Autom Constr 98:183–194
19. Lee G, Ojha, A, Lee M (2015) Concentration monitoring for intelligent tutoring system based on pupil and Eye – blink, In: Proceedings of the 3rd International Conference on Human-Agent Interaction, pp 291–294
20. Li J, Ngai G, Leong HV, Chan SCF (2016) Multimodal human attention detection for reading from facial expression, Eye Gaze, and mouse dynamics. ACM SIGAPP Appl Comput Rev 16:37–49
21. Lin FC, Ngo HH, Dow CR (2020) A cloud-based face video retrieval system with deep learning. J Supercomput 76(11):8473–8493
22. Lipton Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019
23. Liu L, Peng N (2021) Evaluation of user concentration in ubiquitous and cognitive artificial intelligence-assisted English online guiding system integrating face and eye movement detection. Int J Commun Syst 34(6):e4580

24. López-Pernas S, Gordillo A, Barra E, Quemada J (2021) Comparing face-to-face and remote educational escape rooms for learning programming. IEEE Access 9:59270–59285
25. Martínez-Díaz Y, Méndez-Vázquez H, Luevano LS, Nicolás-Díaz M, Chang L, Gonzalez-Mendoza M (2022) Towards accurate and lightweight masked face recognition: an experimental evaluation. IEEE Access 10:7342–7353. https://doi.org/10.1109/ACCESS.2021.3135255
26. Paoletti ME, Haut JM, Plaza J, Plaza A (2020) Scalable recurrent neural network for hyperspectral image classification. J Supercomput 76:8866–8882. https://doi.org/10.1007/s11227-020-03187-0
27. Radha R, Mahalakshmi K, Kumar VS, Saravanakumar AR (2020) E-Learning during lockdown of covid-19 pandemic: a global perspective. Int J Control Autom 13:1088–1099
28. Sankar JP, Kalaichelvi R, John J, Menon N, Elumalai KV, Alqahtani M, Abumelha M (2019) Factors affecting the quality of E-learning during the covid-19 pandemic from the perspective of higher education students. J Inf Technol Educ Res 19:731–753. https://doi.org/10.28945/4628
29. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681. https://doi.org/10.1109/78.650093
30. Sharma A, Biswas A, Gandhi A, Patil S, Deshmukh O (2016) LIVELINET: A multimodal deep recurrent neural network to predict liveliness. in: educational videos. In: International Educational Data Mining Society
31. Sharma P, Joshi S, Gautam S, Maharjan S, Filipe V, Reis MJ (2019) Student engagement detection using emotion analysis, eye tracking and head movement with machine learning. arXiv preprint arXiv:1909.12913
32. Soni VD (2020) Global Impact of E-learning during COVID 19. SSRN Electron J. https://doi.org/10.2139/ssrn.3630073
33. Stevens GJ, Bienz T, Wali N, Condie J, Schismenos S (2021) Online university education is the new normal: but is face-to-face better? Interact Technol Smart Edu 18(3):278–297
34. Tealab A (2018) Time series forecasting using artificial neural networks methodologies: a systematic review. Futur Comput Informatics J 3:334–340. https://doi.org/10.1016/j.fcij.2018.10.003
35. Toshpulatov M, Lee W, Lee S (2021) Generative adversarial networks and their application to 3D face generation: a survey. Image Vis Comput 108:104119
36. Troussas C, Krouska A, Giannakas F, Sgouropoulou C, Voyiatzis I (2021) An alternative educational tool through interactive software over Facebook in the Era of COVID-19. In: Novelties in Intelligent Digital Systems: 3–11
37. Wang ZH, Horng GJ, Hsu TH et al (2020) Heart sound signal recovery based on time series signal prediction using a recurrent neural network in the long short-term memory model. J Supercomput 76:8373–8390. https://doi.org/10.1007/s11227-019-03096-x
38. Wen J, Zhang W, Shu W (2019) A cognitive learning model in distance education of higher education institutions based on chaos optimization in big data environment. J Supercomput 75(2):719–731
39. You K, Long M, Wang J, Jordan MI (2019) How does learning rate decay help modern neural networks? arXiv preprint arXiv:1908.01878
40. Zaghari N, Fathy M, Jameii SM, Sabokrou M, Shahverdy M (2021) Improving the learning of self-driving vehicles based on real driving behavior using deep neural network techniques. J Supercomput 77:3752–3794. https://doi.org/10.1007/s11227-020-03399-4

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.