Temporal-Based Transformed Recurrent Neural Network for Dropout Prediction in Online Learning

Abdullah Alshehri

Department of Information Technology
Faculty of Computer Science and Information Technology
Al Baha University, Alaqiq 65779-7738, Saudi Arabia

Summary
Online learning has provided flexible learning opportunities worldwide to a wide range of individuals. The rapid advances in the online learning industry have allowed numerous universities and institutes to offer online courses that provide students with adequate learning experiences. Although online learning provides great advantages to individuals, the increase in dropout and course incompletion rates brings a crucial challenge that adversely influences the effectiveness of online learning. To address the dropout issue, many studies have been directed at student performance prediction to determine the intention to dropout. The prediction of student performance relies on the analysis of log data learning. However, the sparsity and high dimensionality of learning data have brought a complex feature extraction to build a reliable prediction model. Moreover, the learning behavior may change over time due to the change in learning settings for various reasons, such as using different devices to complete the course. Nevertheless, the current prediction models fail to capture the subtle change in learning behavior where the prediction can significantly be improved. This paper presents a novel model to develop a prediction model for student dropout in online learning. The model proposes a novel Transformed Remember Gated-based Long Short-Term Memory TRG-LSTM to structure temporal feature space from multivariate time series learning activities. TRG-LSTM is employed to map the log data of learning activities in non-linear temporal domain as multi-sequences representation. Thus, the behavior learning profile is built upon non-linear feature space thus to learn the inter-relations between temporal dependencies. Moreover, the proposed TRG-LSTM can handle the subtle changes in learning behavior by modifying the activation function value of remember gate to capture subtle information over temporal granularity. The evaluation, using online learning dataset, has shown that the proposed model has outperformed the benchmarked models to predict student dropout more accurately.

Keywords:
Deep Learning, LSTM, Online Learning, Student Performance Prediction.

1. Introduction

Nowadays, online learning is a popular form of E-Learning where the learning is delivered over the Internet, either synchronously or asynchronously. For example, Massive Open Online Courses (MOOCs) have become prevalent as they provide learners with flexible and affordable learning. Moreover, during the COVID-19 pandemic, tremendous learning providers worldwide have shifted to offer online learning to the affected students due to the traveling ban and lockdown restrictions [1]. Thus, in such a circumstance, online learning is no longer a choice to supplement in-person classes; it is instead an indispensable necessity.

Despite the advantages of online learning, there are some emerging challenges where solutions are critically important [2]. A concern behind the utilization of online learning is the increase in dropout rates in which many students did not complete the course successfully. One reason is that the number of enrolled students in such online courses would be significantly high, which makes it hard to monitor the learning performance of each individual student. For instance, as of June 2022, the enrolled number of learners in the Machine Learning Specialization course at Coursera was 9565. This, in turn, would make monitoring students' performance, where identifying the limitations that lead to dropout, such as the lack of engagement, a complicated task. Another reason it causes the increase in dropout rates in online courses is that the course instructors and students do not communicate with one another directly, i.e., face-to-face communication; this typically occurs when the online course is introduced asynchronously. Direct contact, as in the case of on-campus learning or traditional classes, would allow the instructor to assess the student's performance individually to predict the learning performance and anticipated outcomes easily. Whatever the cause of dropout in online learning, the need to find a sophisticated solution to this issue is desirable to ensure the effectiveness of the online learning paradigm.

Manuscript received September 5, 2022
Manuscript revised September 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.9.101

Student performance prediction is a well-studied topic in online learning [3]. In this context, investigating the student's performance is essential to anticipate the intention of dropping out. The student dropout (or completion) prediction problem has been studied in some depth by the communities of learning analytics and educational data mining to reduce the incompletion rate and increase the effectiveness of online learning. Nevertheless, the study of student performance heavily depends on the log data of learning activities, such as watching videos, answering quizzes, and so on, to analyze learning patterns and behavior [4]. Many studies have used Machine Learning (ML) approaches to accomplish performance prediction models [5]. Most ML algorithms have used shallow learning approaches that mainly focus on linear models for regression and classification tasks to dropout prediction. However, the log data of learning activities is stochastic and largely unstructured, which requires extensive feature engineering to build a robust prediction model. Therefore, Deep Learning (DL), an ML subfield, has been brought to address this drawback. The intuition of using DL is that it serves to develop a nonlinear model where features are extracted automatically based on the nonlinear mapping between variables data samples.

Although DL has merit for student performance prediction, the prediction model is affected by several flaws. First, log data learning consists of a high-dimensional structure encompassing many values at each instance in the data records. For example, when the student starts watching a lecture video, multiple values would be recorded simultaneously, including start and end time, topic name, duration, platform information, student ID, student name, and so on. This, in turn, indicates that multi variables can be involved in the data structure where these variables are paramount to capturing more explicit patterns of learning. Therefore, the predictive model needs to consider the complex structure of log data learning in that the hidden temporal dependence relations should be learned for better prediction. It is worth mentioning that missing some data while building the feature space in DL can result in a weak prediction model. Secondly, log data stream fluctuations can adversely affect the prediction model. This can be noticed when the student may use various platforms and devices to complete the online course, such as mobile, PC, and tablets, where related-device characteristics can be recorded in the log data. However, whatever devices are used, they should reflect the student's pattern and behavior of learning. These changes can underline a subtle shift in the streaming flow, although it remarks the performance of a unique student. Thus, the prediction model should be aware of the subtle changes in the log data stream and maintain a stable analysis of student performance for accurate dropout prediction.

This study proposes a novel model to address the abovementioned drawbacks for student performance prediction. The model uses Long Short-Term Memory (LSTM), a well-known Recurrent Neural Network (RNN) flavor, with the temporal representation of log data learning. More specifically, the model represents log data as a multivariate time series structure to map abstracted temporal feature space. In this context, the LSTM maintains the complicated temporal dependent relationships of features to capture the nonlinear relations between hidden dependencies in the data samples. To this end, we develop two attentions in the encoder and decoder as follows: i) decoder-influenced attention and ii) decoder-temporal dependencies attention. The former serves to map the attention layer in the encoder for the new arrival temporal features to select the most valuable non-predictive columns for the target series. Thus, the decoder-temporal dependencies attention can be learned on the complicated temporal dependencies and the target series variation rule with time. To capture the subtle changes, we develop new modified attention based on extending the save gate range that can obtain the hidden information (hidden stats) for long dependencies changes. Therefore, the TRG-LSTM attention is developed to avoid the miss discovering of subtle changes over the temporal intervals and to relatively capture learning patterns under different settings. In this context, the prediction model considers implicit and explicit hidden relations in nonlinear separable feature space considering the temporal dependencies for learning activities.

This study endeavors to provide the following contributions:

i) The proposed prediction model allows to map nonlinear relationships between various variables in the log data samples to extract abstracted feature space for better prediction of student dropout.

ii) The study provides a novel model where modified LSTM attention is developed to learn complex and nonlinear temporal dependencies and the target series variation rule with time to predict at-risk dropout possibility.

iii) The proposed prediction model maintains the temporal order of dependencies to capture subtle changes to learn hidden patterns under uncertain feature space.

iv) The proposed model attempts to resolve the sparsity of log data in online learning using multivariate time series representation where it underlines dependent temporal streams for entire learning activities. Thus, the model feature mapping considers n-streams to learn the hidden inter-relations in the data instances for better prediction modeling.

The remainder of this paper is structured as follows. Section 2 introduces the related work. In Section 3, the proposed model has been introduced in further detail. Section 4 presents the evaluation and experiment results of the proposed model. Finally, Section 5 provides the conclusions and future directions where the proposed model can be further extended and improved.

2. Related Work

Students' performance prediction is a well-studied online learning application that has been employed to determine the possibility of dropping out of the course before successful completion [2],[3],[5]. The traditional approaches have utilized machine learning-based mechanisms to map the log data of learning activities, such as watching videos, answering exams, and solving assignments, into a classification problem. In this context, the prediction model employs shallow learning algorithms that conduct the prediction using linear functions of feature space. Moreover, the features are handcrafted to map explicit space in that the correlations can be identified correctly. For instance, in [6], a linear Support Vector Machine (SVM) was trained on activity learning data extracted from clickstream information. The model was trained to compute linear prediction function using maximized hyperplane between positive and negative data points. The study presented in [7] has used multiple linear SVM kernels, including the linear kernel, poly kernel, and RBF kernel, to predict dropout students in MOOCs. The classifier was trained on log data learning obtained from KDD Cup 2015 dataset [8]. The study introduced in [9] utilized the sequential features of watching the learning lectures to predict the student's performance using the Random Forest (RF) classification approach. The RF classifier was trained on aggregated features based on the temporal characteristics of selected learning activities. In [10], educational log data (log files) was used to develop a predictive model for recognizing students' performance. The log data, in this context, was represented as a time series of n-grams of the lecture video data, and the prediction model used Decision Tree with J48 classifier to predict the students at risk of course completion.

Further study is presented in [11], where the prediction model was built upon Natural Language Processing (NLP) approach for monitoring the completion rate in online learning platforms. The proposed model used lasso-regularized logistic regression [12], a well-known ML statistical approach, to learn from the unstructured text to predict at-risk dropouts. However, the prediction model was limited to textual features, which can relatively reduce the prediction accuracy; if the feature space consists of more optimal features, it can represent accurate learning behavior. In most situations, the use of shallow ML models to predict dropout in online learning fails to extract accurate learning features due to the gradual increase in online data. The learning activities have provided high dimensional data characteristics, nonlinear relations, and dynamic sparsity.

Deep learning has been introduced to bring solutions as features are automatically engineered. Moreover, DL models can yield nonlinear mapping to features to learn latent correlations in high-dimensional feature space. In the literature, DL is quietly used to address various limitations to student performance prediction (readers can refer to [13] and [14] for recent surveys on students' performance and dropout prediction using DL methods). In [15], the authors employed a recurrent neural network with sort-long memory to predict the potential dropout in online MOOCs offered on Coursera and edX. The authors in [16] suggested a dropout prediction model that combines a variational information bottleneck and a multiscale convolutional network. By building a multiscale complete convolutional network, the model extracts multiscale features from learning activity streams and then employs a variational information bottleneck to suppress the impact of noise on the prediction outcomes. The study presented in [17] has introduced a convolutional neural network (CNN) feature extractor, which builds feature relations to predict dropout potentials in MOOCs. The model employed CNN to obtain abstracted features, calculating attention on extracted dimensions. In [18], a CNN with LSTM prediction model was proposed using a scalable Xgboost classifier [19] for dropout prediction. The authors considered a cost-sensitive method in the loos function to address the drawback of misclassification costs in class imbalance. Thus, the study attempted to reduce the false-negative rate caused by imbalanced data, due to the sparsity of online learning data, for a better prediction model. Moreover, the prediction model was improved using the context-aware feature interaction network (CFIN) with one-dimensional CNN by incorporating user and course information [18]. The study introduced in [20] used CNN to automatically extract streaming features using the word2 vector encoding method. The extracted vectors were combined using LSTM and then passed to the RF classifier to predict the dropout intention in online learning platforms. Also, the authors in [21] have introduced self-attention and multi-head attention for auto labeling and extracting features to train linear transformed CNN classifier for dropout prediction problem in online learning.

The inter-relationships between dependent temporal features in learning data are neglected in the literature. Moreover, the temporal dependencies can underline significant changes over time granularity which can reflect representative learning patterns for student performance prediction. Thus, the proposed study introduces a novel solution using temporal representation and modified LSTM

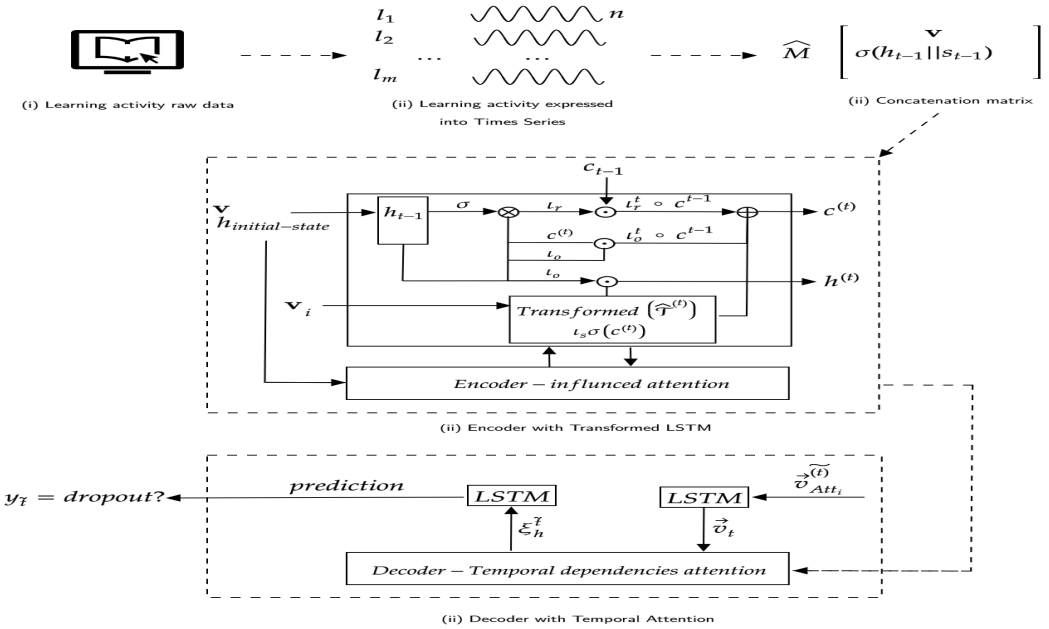

Fig.1 The overall structure of the proposed TRG-LSTM prediction model.

attention to effectively map inter-relationships and subtle changes to predict dropout possibility in online and MOOC platforms.

3. Proposed Work

As given in the introduction to this paper, the proposed model is agreed to handle the dropout issue as a multivariate time series prediction problem where TRG-LSTM coupled with two attentions in the encoder and decoder stages are used to map the complex temporal inter-relations and subtle changes. The problem is formulated in that, given a sequence of n-sequences of activity learning, the model should estimate whether the student will complete the course entirely. This section introduces the proposed model in detail. Figure 1 illustrates the overall prediction model based on TRG-LSTM that considers working with encoder-influenced attention and decoder-temporal dependencies attention. The model begins by molding learning activities in a multivariate-time series structure. Thus, we first describe the notation of learning activity streams to develop the input feature space. Then, the input streams are expressed into an embedded vector to map abstracted features in which the significant correlations are captured in the multi-temporal sequences of learning activities. Therefore, the mapping procedure of hidden dependencies is performed using attention-influenced in the encoder mechanism. During the encoder stage, the input vectors are parametrized into mini batches to underline hidden information in the temporal granularity. The next step is concerned with learning the internal relations and subtle changes using TRG-LSTM with the encoder-influenced attention. Finally, the decoder is used with LSTM and temporal dependencies attention to calculating a probability vector for the final prediction state.

3.1 Learning Activity Representation

The learning process at such an online platform is realized as a set of learning actions relevant to a number of learning materials, such as watching videos, browsing slides, writing texts, and answering questions. These actions produce a massive raw of multiple variables such that n variables are recorded simultaneously. In this study, learning activities are compressed into a multivariate time series structure. Formally, given a single learning activity l it represents a series of learning events such that $l = (l_{e1}, l_{e2}, \dots, l_{ei})$. Therefore, a set of learning activities L represents the entire learning process such that $L = (l_1, l_2, \dots, l_m)^\intercal = (l^1, l^2, \dots, l^n) \in \mathbb{R}^{n \times m}$, where $\mathbb{R}^{n \times m}$ is the size of the time window. Recall that $l^{(t)} = (l_1^{(t)}, l_2^{(t)}, \dots, l_n^{(t)})^\intercal \in \mathbb{R}^n$ defines n time series vector at time (t). In this context, the structure of learning activity represents multivariate time series columns consisting of non-predictive and target sequences. Thus, given the past $l^{(t)}$, the LSTM model is trained to predict the dropout potential over time (t).

3.2 Decoder-Influenced Attention

In this step, the individual streams are embedded into vectors to map encoded-influenced attention. Given a learning activity $l = (l_{e1}, l_{e2}, \dots, l_{ei})$, it is transformed into embedded vector $\mathbf{v}$ such that $\mathbf{v} \in \mathbb{R}^{n}$, for $i \in \{1, \dots, n\}$. The embedded vectors are randomly created to map the abstracted feature. Given embedded vectors, the encoder-influenced attention creates matrix concatenation to the previously hidden cells, thus it serves to combine the hidden relations over the attention method. To prevent the abundant hidden information in the temporal dependences, we use rectified linear unit (ReLU) to map the latent information in matrix concatenation. Thus, the generated concatenation matrix $\widehat{\mathbf{M}}$ can be formally written as follows (Eq.1):

$$\widehat{\mathbf{M}} = \boldsymbol{\sigma}\,(\mathrm{h}_{\mathrm{t}-1} || \mathrm{s}_{\mathrm{t}-1}) \tag{1}$$

where $\mathrm{h}_{\mathrm{t}-1}$ and $\mathrm{s}_{\mathrm{t}-1}$ are the hidden and cell states at time $t-1$ respectively, and σ is the activation ReLU function.

The temporal variation rule will experience unneeded interference from this mapping method; therefore, this mapping is only used to record subtle changes during the encoder stage in order to minimize the possible influences accordingly.

3.2.1 Transformed Remember Gated-LSTM

The temporal dependencies of long-term dynamic changes are hard to be properly reflected in the classic LSTM network without a particular universal mechanism for learning subtle changes. Thus, traditional approaches have employed Dropout [22] and Zoneout [23] regularizations to enhance the generalization performance of the LSTM model. Recalling that dropout regularizer selects the activation value based on approximated Bayesian inference, where zoneout regularizer randomly preserves hidden state activation over tuned zonout probabilities. However, these regularization techniques will continue to lose data flow content at random temporal granularity, making it difficult to detect subtle information in continuous time series data.

In this study, we propose the Transformed Remember Gated TRG-LSTM solution to avoid the miss discovering of subtle changes in the temporal learning data for student dropout prediction. More formally, a resulted hidden layer h_t is controlled by three non-linear gated vectors: save vector $\iota_s^{(t)}$, remember vector $\iota_r^{(t)}$ and focus vector $\iota_f^{(t)}$. The equations (Eq.2, Eq.3, and Eq.4) determines the mathematical description of each gated vector. Thus, the gated vectors serve to control a memory cell $c^{(t)}$(Eq.6) over time t to capture long-term dependencies. To obtain the whole learning subtle information, a nonlinear function transformation $\hat{\mathcal{T}}^{(t)}$ (Eq.5) is used to change the numerical range of the remember gate output value. The TRG-LSTM considers the subsequent focus gate processed after the remember gate calculation. Thus, the activation function computes the focus gate (output gate) value to interval of [0.2,1.0] where the hidden layer is computed as in Eq.7.

$$\iota_s^{(t)} = \sigma\,(W_s[l^{(t)}; h_{(t-1)}] + b_s) \tag{2}$$

$$\iota_r^{(t)} = \sigma\,(W_r[l^{(t)}; h_{(t-1)}] + b_r) \tag{3}$$

$$\iota_o^{(t)} = \sigma\,(W_o[l^{(t)}; h_{(t-1)}] + b_o) \tag{4}$$

$$\hat{\mathcal{T}}^{(t)} = 1 - ReLU(\iota_r^{(t)}) \tag{5}$$

$$c^{(t)} = \hat{\mathcal{T}}^{(t)} \odot c_{t-1} \oplus \iota_r^{(t)} \odot ReLU(W_g[l^{(t)}; h_{(t-1)}] + \mathrm{b_g}) \tag{6}$$

$$h_t = \iota_o^{(t)} \odot ReLu(c^{(t)}) \tag{7}$$

In this context, the data flow after the remember gated value is governed by the following instructions. The obtained values close to 0 are regarded as fully rejected when the transformed ReLU function (Eq.5) is triggered, whereas values near 1 are all considered to be passed. In order to maintain the information in the data before and after the time, it is clear that the focus gated value is essential. As the process of learning subtle changes in temporal granularity takes place over a number of time steps, creating suitable output rules for the remember gate is the crucial step in resolving the issue of recording subtle data. The remember gated output value range is transferred by the proposed transformation to the interval [0.2,1.0] instead of [0,1]. The initial value, which was close to 1, decreased to 0.2 through the transformation method, whereas the original output value, which was close to 0, increased to be close to 1. The original values near the center are compressed to 0.5 in the center, which is something to keep in the memory cell. The data's numerical range is condensed to the portion of the interval with the most pronounced changes, which is better for capturing the relationship between the data, especially when there are sudden changes in the data. In other words, the modification in remember gated output value range is proposed to strengthen the relationship extraction between temporal dependencies on subtle information. Moreover, the TRG-LSTM works to avoid supersaturation intervals by compressing temporal flows to the region with the most significant change. Thus, the subtle change of supersaturation interval is decreased, and the issue that the classic LSTM can rarely learn subtle information in the supersaturation period is resolved.

3.2.2 TRG-LSTM Recurrent Propagation

In the proposed model, we also examined how the recurrent calculation arrives at the partial derivative of the $1 - ReLU\left(\iota_s^{(t)}\right)$ function. Thus, given an input (tensor of learning activity) l to replace the remember gate output $\boldsymbol{\iota_r^{(t)}}$, the original expansion of the function is derived as follows (Eq.8):

$$f(\iota) = 1 - \ ReLU\left(\iota_r^{(t)}\right) = 1 - \frac{1 - e^{-2\iota}}{1 + e^{-2\iota}} \tag{8}$$

Thus, to explicit calculation of the recurrent propagation in the proposed model, a new recurrent transformation $\acute{f}(\iota)$ has been introduced to the original function $f(\iota)$ as follows (Eq.9):

$$\acute{f}(\iota) = f^2(\iota) - 2f(\iota) \tag{9}$$

In this context the partial derivatives of $\hat{\mathcal{T}}^{(t)}$ and $\iota_r^{(t)}$ of the transformation mechanism are computed using a loss function $\mathcal{L}$ based on the gradient descent method. This loss function $\mathcal{L}$ is then extended by the chain rule to a sequence covering all time steps (t^i) for $i \in \{1, \dots, n\}$ so the cumulative gradients can be obtained as follows (Eq.10):

$$\frac{d\mathcal{L}}{dW_h} = \sum\nolimits_{i=1}^{n} \frac{d\mathcal{L}_i}{dW_{hi}} \tag{10}$$

where $\frac{d\mathcal{L}}{dW_h}$ indicates the partial derivative of $\mathcal{L}$ to the weight W_h in the TRG-LSTM, which is obtained by the sum of partial derivatives of each time tick (i.e., vector input as mini-batches over interval time). Consequently, to determine the ideal TRG-LSTM parameters, multiple repetitions of recurrent computations can experimentally minimize $\mathcal{L}$ function of tasks until the TRG-LSTM is completed. The value range of the gradient information flow always remains in the most significant range due to the partial derivatives of recurrent weights and the continual manufacturing of transformation gated vectors. This indicates that the suggested transformation technique is helpful in capturing potential subtle changes before and after a particular instant in forward propagation and for the recurrent (backward) propagation stage. Therefore, the proposed transformation mechanism has the ability to enhance the learning capacity of subtle changes for the remember gated output value.

3.2.3 Influenced-based Attention Flow

It can be observed that the subtle information in various non-predictive series affects the target series differently. Thus, the influence attention mechanism is proposed to receive the influence information of each non-predictive temporal sample on the target series at the same time step to learn these various degrees of influence information. To this end, the encoder is given the influence attention mechanism of the exact temporal step. It is worth mentioning that non-predictive time series frequently influence the target series in various ways, producing extraordinarily complicated influence data. Thus, the proposed model employs this attention mechanism to adaptively capture the various influences between the target series and each temporal stream. Given the input feature vector (i.e., temporal stream) $\mathbf{v}_i \longmapsto l_i^{(t)} l = (l_{e1}, l_{e2}, \dots, l_{ei})$, the influenced attention score $att_i^{(t)}$ is computed in Eq.12 as follows:

$$\acute{h}^{(t-1)} = \boldsymbol{\sigma}\left([\boldsymbol{h_{t-1}}||\boldsymbol{s_{t-1}}]\right) \tag{11}$$

$$att_i^{(t)} = \frac{\mathbf{v}_s^{\mathrm{T}} \boldsymbol{\sigma}\left(\boldsymbol{W_s}\acute{h}^{(t-1)} + \boldsymbol{U_s}\boldsymbol{l_i}| + \boldsymbol{b_s}\right.}{\sum_{j=1}^{n-1}[\mathbf{v}_s^{\mathrm{T}} \boldsymbol{\sigma}\left(\boldsymbol{W_s}\acute{h}^{(t-1)} + \boldsymbol{U_s}\boldsymbol{l_j}| + \boldsymbol{b_s}\right.} \tag{12}$$

where $([\boldsymbol{h_{t-1}}||\boldsymbol{s_{t-1}}])$ represents the element-wise concatenation of previous hidden layer $\boldsymbol{h_{t-1}}$and cell state $\boldsymbol{s_{t-1}}$. Thus equation Eq.11 is used to underline the state dependency in the encoder and generate a new hidden layer $\acute{h}^{(t-1)}$after the concatenation process. Recall that, in Eq.12, the values of $\mathbf{v_s}$ and $\mathbf{b_s} \in \mathbb{R}^{\boldsymbol{n}}$, where $\boldsymbol{W_s} \in \mathbb{R}^{\boldsymbol{n \times 2m}}$ and $\boldsymbol{U_s} \in \mathbb{R}^{\boldsymbol{n \times n}}$. These values represent the parameters that the model should be learned. The final attention score implies the influenced attention to each non-predictive column in temporal learning data. By doing this, we can obtain the output vector of the current temporal interval at time (t) as follows (Eq.13):

$$x^{s(t)} = \left(att_1^{(t)}\mathbf{v}_1^{(t)}, att_2^{(t)}\mathbf{v}_2^{(t)}, \dots, att_n^{(t)}\mathbf{v}_n^{(t)}\right)\intercal \tag{13}$$

where $x^{s(t)} \in \mathbb{R}^{n+n}$ represents the obtained vector. Thus, given a vector $x^{s(t)}$ is passed to the encoder to get the new current hidden dependency layer where the subtle change is captured accordingly.

3.2 Decoder-Temporal Dependencies Attention

The periodicity patterns should be captured via the temporal attention technique to collect the subtle changes indirectly. To counteract this performance deterioration, in this case, a temporal attention mechanism is employed to adaptively choose the pertinent hidden state from the encoder and gather more hidden state data. As a result, the model is able to represent the target dynamic temporal correlation. To be more precise, we compute an attention vector $\vec{v}_{Att_i}^{(\tilde{t})}$ across each hidden layer of the encoder at each output time $(\tilde{t})$, as follows (Eq.14):

$$\vec{v}_{Att_i}^{(\tilde{t})} = \vec{\mathbf{v}}_d\, \sigma\left(\widetilde{W}_d[d_{\tilde{t}-1}||\tilde{s}_{\tilde{t}-1}] + U_d h_{Att_i} + b_d\right) \tag{14}$$

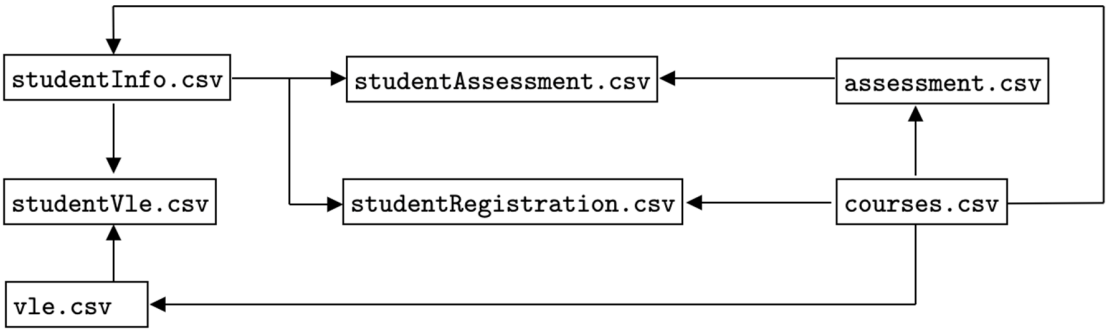

Fig. 2 The structure of OULAD dataset.

$$score_{Att_i}^{(\tilde{t})} = \frac{\exp(\vec{v}_{Att_i}^{(\tilde{t})})}{\sum_{j=1}^{n} \vec{v}_{Att_i}^{(\tilde{t})}} \quad (15)$$

where the variables $\vec{\mathbf{v}}_d, b_d \in \mathbb{R}^{\boldsymbol{m}}$, $\widetilde{W}_d \in \mathbb{R}^{\boldsymbol{2p \times m}}$, and $U_d \in \mathbb{R}^{\boldsymbol{m \times m}}$ are expressed and learned by the model. In this context, the temporal attention score $score_{Att_i}^{(\tilde{t})}$ is computed in Eq.15 to determine the sum of all encoder's hidden dependencies weights which results in temporal-generated vector $\vec{v}_t$ such that (Eq.16):

$$\vec{v}_t = \sum_{j=1}^{n} score_{Att_k}^{(\tilde{t})} h_k \quad (16)$$

3.3 Prediction Procedure

During the prediction process, the temporal-generated vector is summed with the previous output dependency $y_{\tilde{t}-1}$ in the decoder to update the encoder hidden layer $\xi_h^{\tilde{t}}$ as follows (Eq.17):

$$\xi_h^{\tilde{t}} = d_{LSTM}(d_{\tilde{t}-1}, [\vec{v}_t || y_{\tilde{t}-1}] \quad (17)$$

where d_{LSTM} indicates the LSTM decoder.

This results in new hidden information using the concatenation of temporal-generated victor $\vec{v}_t$ with the updated encoder hidden layer $\xi_h^{\tilde{t}}$, thus the final prediction can be computed as follows (Eq.18):

$$y_{\tilde{t}} = \mathbf{v}_y^{\mathrm{T}}(W_y[\vec{v}_t || \xi_h^{\tilde{t}}] + \mathbf{v}_p) + \boldsymbol{b}_y \quad (18)$$

where $W_y \in \mathbb{R}^{p(p+m)}$ and $\mathbf{v}_p \in \mathbb{R}^p$. Recall that $[\vec{v}_t || \xi_h^{\tilde{t}}]$ represents the concatenation between temporal-generated victor and updated encoder hidden layer.

4. Evaluation

The effectiveness of the suggested model is assessed in this section. The primary goal of the evaluation is to demonstrate how well the inter-relations can be captured with subtle changes in the temporal dependences using TRG-LSTM for predicting student dropout possibility in online learning. Moreover, the evaluation introduces the impact of tweaking LSTM parameters on detection accuracy. Finally, the evaluation demonstrates how well the proposed model is performed compared to traditional linear dropout prediction techniques.

4.1 Dataset

The proposed model has been evaluated using publicly available online learning dataset. The dataset is obtained from [24]; it is an online learning dataset known as anonymized Open University Learning Analytics Dataset (OULAD). The dataset includes information from online courses offered at the Open University (OU). The dataset is distinctive because it combines demographic information with clickstream data from students' interactions in the virtual learning environment (VLE). This makes it possible to analyze the behavior of students as shown by their deeds. The collection includes details on 22 courses, 32,593 students, their assessment outcomes, and records of their VLE interactions, which are displayed as daily summaries of student clicks (10,655,280 entries). The dataset consists of seven .csv files (expressed in tabular format). For clarity, Figure 2 shows the schema of the dataset, which consists of seven csv files in which further description is given in the following subsection of each file.

4.1.1 Dataset description

studentInfo.csv: This file of tabular data of demographic information of each student. It also includes the obtained results of the modules the students have studied.

courses.csv: This file is dedicated to storing a list of all modules and their presentations.

studentRegistration.csv: The file contained the temporal information when the student enrolled for the module presentation.

assessments.csv: This csv file stores the assessments information of each module presentations. Recall that every presentation has several assessments followed by the final exam.

studentAssessment.csv: The csv file stores the assessment results of students concerning each module in the course.

studentVle.csv: This file includes the learning activities in the MOOC platform; it includes temporal information of activity interactions and stream clicks (the number of times the student interacted with learning materials).

vle.csv: This file consists of the materials information that are supported by the MOOC platform, including HTML pages and pdf files.

4.2 Evaluation Metrics

To evaluate the performance of the proposed model, we employ confusion matrix to compute Precision (Pre.), Recall (Rec.) and F1-score (F1) values. The fraction of samples predicted as dropouts that really do so is known as (Pre.), and it is an indicator of how accurately the model can identify dropouts. The (Rec.) is a coverage metric that measures how well a model can identify all dropouts by indicating the percentage of all dropout samples that are adequately predicted. However, only considering (Pre.) or (Rec.) computations is not sufficient for the dropout prediction challenge. That is the value of (Rec.) will be 100% and (Pre.) will be low if the model correctly predicts that all samples will drop out. In contrast, if the model only detects a small number of dropouts, (Pre.) may be significantly high and (Rec.) will be poor. Therefore, the (F1) score can be used to combine (Pre.) and (Rec.) to provide a more accurate assessment of the model because it is biased toward the smaller of the two harmonic means of (Pre.) and (Rec.). Nevertheless, these metrics are frequently used in the literature for dropout prediction. The value of each metric is computed as in the following equations (Eq.19, Eq.20, and Eq.21) respectively:

$$\text{Pre.} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{19}$$

$$\text{Rec.} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{20}$$

$$\text{F1} = \frac{\text{TP}}{\text{TP} + \frac{1}{2}(\text{FP} + \text{FN})} \tag{21}$$

Also, the area under the curve (AUC) is used as an evaluation metric along with the receiver operating characteristic curve (ROC) to determine the performance of prediction at different threshold values. AUC is computed using the true positive rate (TPR) and the false positive rate (FPR) where they represent the vertical and horizontal axes respectively in the ROC curve. Note that the ROC curve is produced by altering the classification threshold to the plot the probability distribution of TPR and FPR. Thus, whenever the AUC

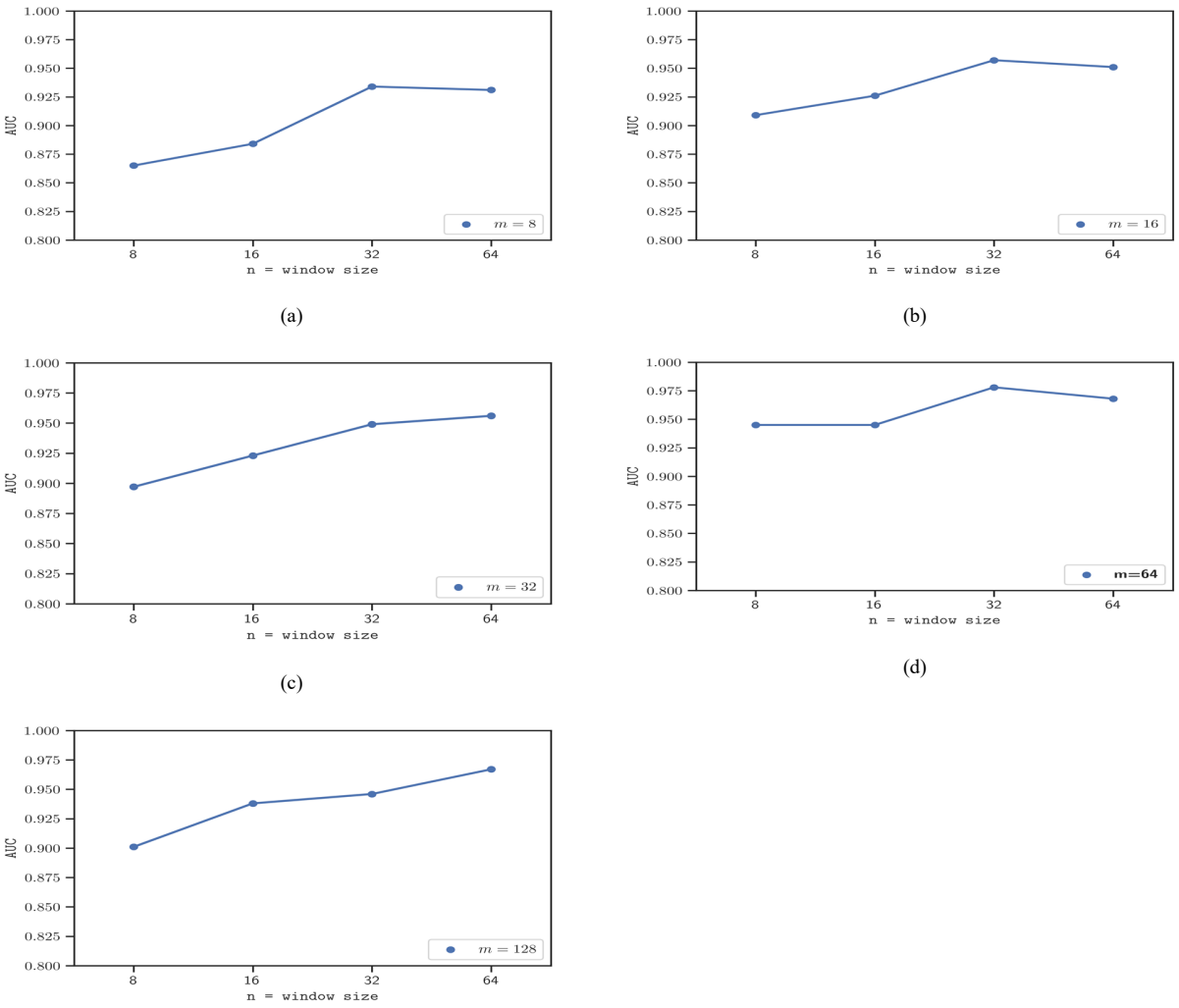

Fig. 3 The AUC results corresponding to hyper-parameters of window size (n) and hidden layers (m). (a) m = 8, (b) m = 16, (c) m = 32, **(d) m = 64**, and (e) m = 128.

value tends to be 1 it indicates better measure of separability where the value tends to 0 indicates poor measure of separability. Nevertheless, the values of TPR and FPR can be calculated as in Eq.22 and Eq.23 respectively.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{22}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{23}$$

It is worth mentioning that in the dataset used for this study, the ratio of dropouts to nondropouts is roughly 5:1. This indicates that the use of classification accuracy as measurement is not significant due to the imbalance in data samples. Thus, AUC is the appropriate measure for the proposed model as it is not influenced by the ratio of imbalanced samples in the dataset.

4.3 Experiments Setting and Results

The experiments have been implemented using TensorFlow framework [25]. All experiments are implemented with Adam optimizer [26] for the training process. The Adam algorithm parameters α, β_1 and β_2 are set to 0.001, 0.9 and 0.99 respectively. Recall that αrepresents the learning rate in which the weights are updated, where β_1 and β_2 refer to the first- and second-time estimations of the exponential decay rate. Moreover, the model is tuned under different hyper-parameters to pick up the best performance. Thus, the

values of window size n and hidden layer m are obtained using grid search within a range of values as follows: n ={8,16,32,64}, m ={8,16,32,64,128}. With respect to the values of mini-batch size and epochs are also selected using grid search, such that mini-batch-size ={32,64,128}, epochs ={50,100} to obtain the optimal parameters for model prediction.

The optimal values of n and m are recorded at $n = 32$ and $m = 64$, as presented in Figure 3 (b). The figure shows that best n and m values have yielded AUC of 0.97. It can be observed from the figure that the value of n has a notable positive impact whenever it increases. Nonetheless, in the evaluation, the window size selection was carefully considered to optimally capture subtle changes based on the size of temporal dependencies. In other words, whenever the temporal learning data consists of extensive short-term dependencies, the window size can be set to a small value, i.e., n = 8 was an experimentally ideal option. In contrast, when the temporal learning data has substantial long-term dependencies, the value of window size is best to be larger. Thus, the intuition of grid search is justified by the fact that the scale of temporal dependencies has a significant impact on the prediction performance. As mentioned earlier, when n was set to 32, it yielded the best AUC; thus, it was selected as the best hyperparameter value.

However, it is essential to clarify that the bigger value of n would increase computational complexity, which can affect the prediction efficiency for real-time dropout prediction applications. The prediction efficiency is important because some online courses are presented synchronously in one session, where the need to anticipate at-risk dropouts in short courses within a limited period is important. Note that the efficiency performance is beyond the scope of this paper, but it is worth to be mentioned for future consideration.

Concerning the batch-size and epoch values, the best results were recorded with batch-size=128 and epoch=100 where best AUC was obtained as depicted in Figure 4. The figure shows that both parameters have a moderate impact on the training dataset despite the bigger batch size having produced adequate performance in terms of the AUC result.

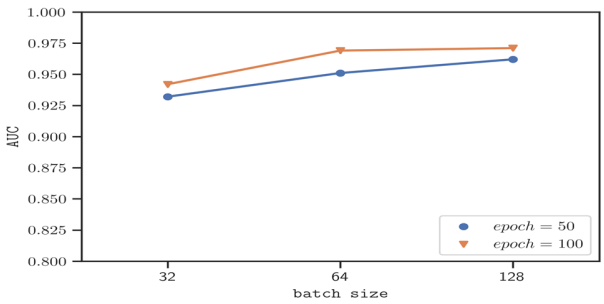

Fig. 4 The AUC result corresponds to the hyper-parameters of batch size and epochs. The best AUC result is recorded with batch-size=128 and epoch=100.

4.4 Baseline Comparison

The evaluation was conducted to determine the prediction of the proposed model compared to the linear models. To this end, several linear models were selected for the evaluation, including Linear Regression (LR) [27], Support Vector Machine (SVM) [28], K-Nearest Neighbor (KNN) [29], Decision Tree (DT) [30], and Random Forest (RF) [31]. Furthermore, the performance of TRG-LSTM was compared to classic LSTM to discover the effectiveness of the proposed prediction model. Nonetheless, in the context of the proposed predictive model, the plan was to adopt each model for dropout prediction on the selected dataset and demonstrates the performance compared to the proposed TRG-LSTM. Note that the linear models have been selected as they were commonly used for student performance prediction. Nevertheless, Figure 5 shows the ROC curves of linear models and classic LSTM in comparison with TRG-LSTM. The figure shows that TRG-LSTM recorded the best value of AUC. For clarity, Table 1 introduces the results in terms of (Pre.), (Rec.), and (F1) for all comparable methods in which the TRG-LSTM has outperformed the other benchmarked methods with a significant margin.

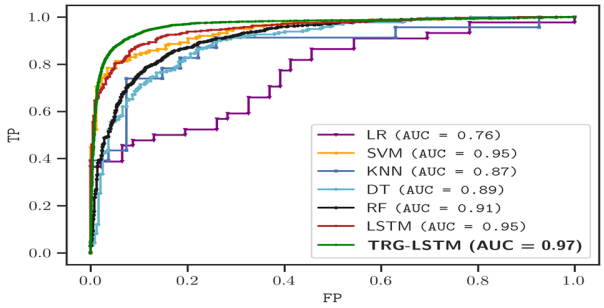

Fig. 5 The obtained ROC curves of the baseline methods and TRG-LSTM.

Table 1: Performance evaluation, in terms of Pre., Rec. and F1., of different baseline methods in comparison with TRG-LSTM.

Methods	Pre.	Rec.	F1.
LR	67.71	45.56	0.32
SVM	91.98	68.98	0.44
KNN	87.61	59.44	0.41
DT	90.87	67.39	0.62
RF	89.46	65.81	0.57
LSTM	92.34	68.00	0.69
TRG-LSTM	**97.21**	**73.32**	**0.85**

5. Conclusion

This study introduces a novel transformed remember gated LSTM (TRG-LSTM) to predict student performance for dropout potentials in online learning platforms. The TRG-LSTM maps learning activities in multivariate time series structure to learn the hidden inter-relations between nonlinear dependent variables for accurate learning pattern extraction. The relationships are nonlinearly captured using transformed remember gated based on developed encoder-influenced attention and decoder-temporal dependencies attention. In this context, an attention layer is mapped to the new arrival temporal information to identify the most valuable non-predictive dependencies for the target series. Moreover, TRG-LSTM extends the remember gate range to collect subtle changes in long temporal dependencies where latent learning patterns can relatively be discovered for accurate dropout prediction. Thus, the decoder-temporal dependencies can be trained to focus on the complex temporal dependencies and the target series variation rule with time to capture relative correlations and subtle changes. It is worth mentioning that the TRG-LSTM attention prevents missing the discovery of subtle changes across temporal intervals and captures learning patterns related to their respective contexts. The evaluation using OULAD dataset has shown that the proposed model outperformed the baseline models with the best AUC value of 0.97.

The proposed work can be improved in the future by incorporating additional features, such as spatiotemporal features, that can extend the learning of latent inter-relations for different online learning settings. For example, in a short-synchronous environment learning where the dependencies size is short, incorporating spatio-information features can help in mapping adequate feature abstraction for dropout prediction. Furthermore, the proposed model can examine different transformation criteria to maintain subtle changes in short-term dependencies. For instance, the value of remember gate can be dynamically captured according to the size of dependent information in stream data.

Acknowledgments

The author would like to thank Al Baha University in Saudi Arabia for funding this work under the grant 8/1440.

References

[1] P. Qiao, X. Zhu, Y. Guo, Y. Sun, and C. Qin, "The development and adoption of online learning in pre-and post-COVID-19: Combination of technological system evolution theory and unified theory of acceptance and use of technology," *J. Risk Financ. Manag.*, vol. 14, no. 4, p. 162, 2021.

[2] B. Albreiki, N. Zaki, and H. Alashwal, "A systematic literature review of student'performance prediction using machine learning techniques," *Educ. Sci.*, vol. 11, no. 9, p. 552, 2021.

[3] A. Khan and S. K. Ghosh, "Student performance analysis and prediction in classroom learning: A review of educational data mining studies," *Educ. Inf. Technol.*, vol. 26, no. 1, pp. 205–240, 2021.

[4] A. Hellas *et al.*, "Predicting academic performance: a systematic literature review," in *Proceedings companion of the 23rd annual ACM conference on innovation and technology in computer science education*, 2018, pp. 175–199.

[5] R. Alamri and B. Alharbi, "Explainable student performance prediction models: a systematic review," *IEEE Access*, vol. 9, pp. 33132–33143, 2021.

[6] M. Kloft, F. Stiehler, Z. Zheng, and N. Pinkwart, "Predicting MOOC dropout over weeks using machine learning methods," in *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs*, 2014, pp. 60–65.

[7] S. Ardchir, M. A. Talhaoui, H. Jihal, and M. Azzouazi, "Predicting MOOC dropout based on learner's activity," *Int. J. Eng. \& Technol.*, vol. 7, no. 4.32, pp. 124–126, 2018.

[8] L. Qiu, Y. Liu, and Y. Liu, "An integrated framework with feature selection for dropout prediction in massive open online courses," *IEEE Access*, vol. 6, pp. 71474–71484, 2018.

[9] C. Ye *et al.*, "Behavior prediction in MOOCs using higher granularity temporal information," in *Proceedings of the second (2015) ACM conference on Learning@ Scale*, 2015, pp. 335–338.

[10] C. Brooks, C. Thompson, and S. Teasley, "A time series interaction analysis method for building predictive models of learners using log data," in *Proceedings of the fifth international conference on learning analytics and knowledge*, 2015, pp. 126–135.

[11] C. Robinson, M. Yeomans, J. Reich, C. Hulleman, and H. Gehlbach, "Forecasting student achievement in MOOCs with natural language processing," in *Proceedings of the sixth international conference on learning analytics & knowledge*, 2016, pp. 383–387.

[12] M. Taddy, "Multinomial inverse regression for text analysis," *J. Am. Stat. Assoc.*, vol. 108, no. 503, pp. 755–770, 2013.

[13] C.-A. Lee, J.-W. Tzeng, N.-F. Huang, and Y.-S. Su, "Prediction of student performance in Massive Open Online Courses using deep learning system based on learning behaviors," *Educ. Technol. \& Soc.*, vol. 24, no. 3, pp. 130–146, 2021.

[14] Z. Sun, A. Harit, J. Yu, A. I. Cristea, and L. Shi, "A brief survey of deep learning approaches for learning analytics on MOOCs," in *International Conference on Intelligent Tutoring Systems*, 2021, pp. 28–37.

[15] M. Fei and D.-Y. Yeung, "Temporal models for predicting student dropout in massive open online courses," in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, pp. 256–263.

[16] Z. Shou, P. Chen, H. Wen, J. Liu, and H. Zhang, "MOOC Dropout

Prediction Based on Multidimensional Time-Series Data," *Math. Probl. Eng.*, vol. 2022, 2022.
[17] Q. Fu, Z. Gao, J. Zhou, and Y. Zheng, "CLSA: A novel deep learning model for MOOC dropout prediction," *Comput. \& Electr. Eng.*, vol. 94, p. 107315, 2021.
[18] W. Feng, J. Tang, and T. X. Liu, "Understanding dropouts in MOOCs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 01, pp. 517–524.
[19] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.
[20] L. Cai and G. Zhang, "Prediction of moocs dropout based on wclsrt model," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 2021, vol. 5, pp. 780–784.
[21] S. Yin, L. Lei, H. Wang, and W. Chen, "Power of attention in MOOC dropout prediction," *IEEE Access*, vol. 8, pp. 202993–203002, 2020.
[22] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.
[23] D. Krueger *et al.*, "Zoneout: Regularizing rnns by randomly preserving hidden activations," *arXiv Prepr. arXiv1606.01305*, 2016.
[24] Z. Z. Kuzilek J. Hlosta M., "Open University Learning Analytics dataset Sci. Data 4:170171." 2017.
[25] Martín~Abadi *et al.*, "{TensorFlow}: Large-Scale Machine Learning on Heterogeneous Systems." 2015.
[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv Prepr. arXiv1412.6980*, 2014.
[27] G. A. F. Seber and A. J. Lee, *Linear regression analysis*, vol. 329. John Wiley & Sons, 2012.
[28] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
[29] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers-a tutorial," *ACM Comput. Surv.*, vol. 54, no. 6, pp. 1–25, 2021.
[30] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst. Man. Cybern.*, vol. 21, no. 3, pp. 660–674, 1991.
[31] M. Pal, "Random forest classifier for remote sensing classification," *Int. J. Remote Sens.*, vol. 26, no. 1, pp. 217–222, 2005.

Abdullah Alshehri received the M.S degree in Software Engineering from Wollongong University, Australia. He obtained a Ph.D. degree in Computer Science, majored in Data Mining and Machine Learning, from The University of Liverpool, United Kingdom, 2018. He is currently an Assistant Professor at the Faulty of Computer Science and Information Technology at Al Baha University in Saudi Arabia. His research interests include Artificial Intelligence (AI), Machine Learning, Deep Learning, Time Series Data Mining (TSDM), and Applied AI in Cybersecurity, Biometrics, and Authentication. Dr. Alshehri has published many refereed publications on AI-related research, and he has chaired sessions at several international conferences in AI disciplinary.