Towards Finding the Impact of Deep Learning in Educational Time Series Datasets – A Systematic Literature Review

Vanitha.S[1], Jayashree.R[2*]
Department of Computer Applications-College of Science and Humanities,
SRM Institute of Science and Technology, Kattankulathur, Tamilnadu, India

***Abstract*—Besides teaching in the education system, instructors do a bunch of background processes such as preparing study material, question paper setting, managing attendance, log book entry, student assessment, and the result analysis of the class. Moreover, Learning Management System(LMS) is mandatory if the course is online. The Massive Open Online Course (MOOC) is an example of the worldwide online education system. Nowadays, educators are using Google to efficiently formulate study material, question papers, and especially for self-preparation. Also, student assessment and result analysis tools are available to get instant results by feeding student data. Artificial Intelligence (AI) is driving behind these applications to deliver the most precise outcome. To accomplish that, AI requires historical data to train the model, and this sequential (year-wise, month-wise, etc) information is called time series data. This Systematic Literature Review (SLR) is conducted to find the contribution of time series algorithms in Education. There are enormous changes in algorithm architecture analogized to the traditional neural network to endure all kinds of data. Though it significantly raises the performance, it expands the complexity, resources, and execution time as well. Due to this, comprehending the algorithm architecture and the method of the execution process is a challenging phase before creating the model. But it is essential to have enough knowledge to select the suitable technique for the right solution. The first part reviews the time series problems in educational datasets using Deep Learning(DL). The second part describes the architecture of the time series model, such as the Recurrent Neural Network (RNN) and its variants called Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), the differences between each other, and the classification of performance metrics. Finally, the factors affecting the time series model accuracy and the significance of this work are summarized to incite the people who desire to initiate the research in educational time series problems.**

Keywords*—Deep learning; education; gated recurrent unit; long-short term memory; recurrent neural network; time series

ABBREVIATIONS

SLR	Systematic Literature Review
PRISMA	Preferred Reporting Items for Systematic Review and Meta-Analyses
LMS	Learning Management System
MOOC	Massive Open Online Courses
CNN	Convolutional Neural Network
AE	Auto Encoder
DBN	Deep Belief Network
GAN	Generative Adversarial Network
DRL	Deep Reinforcement Learning
FFNN	Feed Forward Neural Network
MLP	Multi Layer Perceptron
ERNN	Elman Recurrent Neural Network
ESN	Echo State Network
TCN	Temporal Convolutional Network
LR	Linear Regression
NB	Naive Bayes
SVM	Support Vector Machine
DT	Decision Tree
RF	Random Forest
GBM	Gradient Boosting Machine
AUC	Area Under Curve
ADAM	Adaptive Moment Optimization algorithms
SGD	Stochastic Gradient Descent
SMOTE	Synthetic Minority Oversampling Technique
HMM	Hidden Markov Model

I. INTRODUCTION

Human education assists in automating massive work with less intervention of human resources. That education system itself automated with the help of AI. Machine Learning is a subset of AI. Likewise, Deep Learning (DL) is a division of broader machine learning based on the Neural Network (NN) designed to mimic the human brain. DL is becoming an imperative buzzword in data handling technology due to the potential of prediction using extensive data. However, the prediction system was enlightened only after the innovation of RNN. The RNN is chosen for this study due to its architecture to operate on sequential nature data. For example, Predicting learner dropout rate in MOOC using LMS interaction data (user click events, weekly assignments, etc). All educational institutions switched online to continue the classes during the corona lockdown period. Many online courses started and then boomed. The MOOC is one of the popular platforms for online education. But, the course completion rate is significantly lower than the number of registration due to being free of cost. RNN helps to predict the success and dropout rate of MOOC learners. RNN applications are unrestricted in all the fields, such as finance [1], [2], medicine [3], [4], [13], and nature-related forecasts, such as weather, rainfall, temperature, and wind speed [5]-[8]. Also, enough surveys are available to enrich the existing work on those domains. But in education, reviews still need to be conducted

*Corresponding Author, jayashrr@srmist.edu.in

to find the related work in sequential data collectively. Hernández et al.,[21] did the same job, but that did not focus on time series. This work fills this research gap with the time series model architecture, the difference from the conventional neural network, and the parameters influencing the model performance. The following are the research questions identified for this work:

RQ1: Finding the impact of Deep Learning in educational time series problem.

RQ2: Identify the architecture of time series model and how it differs from the traditional approach.

RQ3:Discover the significant factors affecting the time series model accuracy.

The remaining paper encloses five sections. Section II defines the methodology of this work, and Section III describes the review results including previous work using the deep learning model,working methodology of RNN, LSTM, and GRU, and metrics used for the model. Section IV outlines the contribution of this paper through discussion. Finally, Section V explains the conclusion and future work of this article.

II. Research Methodology

This section elucidates the research methodology followed in carrying out this review process and the filtration of the downloaded papers. The following research repositories are accessed: Google Scholar and IEEE Xplore. The keyword used for this work is the following: "Deep Learning", "RNN", “Time Series", "Student”, and “Education”. The google search result showed many research papers, and all are evaluated manually to select the suited one for this work. The selection process considers the journal articles using time series data in Education and valid conference papers. The inclusion and exclusion criteria of this study is mentioned in Table I.

The article selection process followed the PRISMA method to carry out this study. PRISMA is an abbreviation of the "Preferred Reporting Items for Systematic Review and Meta-Analyses". Fig. 1 explains the step-by-step article inclusion and elimination details through PRISMA 2020 flowchart.

1) Identification: The initial search retrieved two hundred and ninety-one (n= 291) documents from Google scholar and the IEEE database. The filter is applied for the last five years (2018-2022) to restrict the search before 2018 and after 2022. Then removed, four duplicate records from various databases.

2) Screening: There are two screening steps to check the paper's eligibility. i) Preliminary check ii) Full-text analysis. Step 1 investigates the title and abstract to verify the document's relevance. It removed one hundred-six (n=106) reports and included eighty-one (n=81) articles for full-text retrieval. Then sixteen (n=16) documents are eliminated due to paid version. Step 2 inquiry prevents invalid articles, conferences, and other documents irrelevant to this context.

3) Included: The previous stage gives twenty-two reports (n=22), and the selected articles are used as a source for this Systematic Literature Review(SLR) or meta-analysis.

TABLE I. Selection and Rejection Criteria of the Study

Criteria	Inclusion	Exclusion
Year	Documents published between 2018 to 2022	Documents published before 2018 and after 2022
Language	Articles in English	Other language articles
Domain & Data	Educational time-series documents	Other domains and the data which are not using time-series
Article type	Journal and conference	- Articles less than 6 pages - Articles having less than 20 citation
Algorithm	Deep Learning	Machine Learning

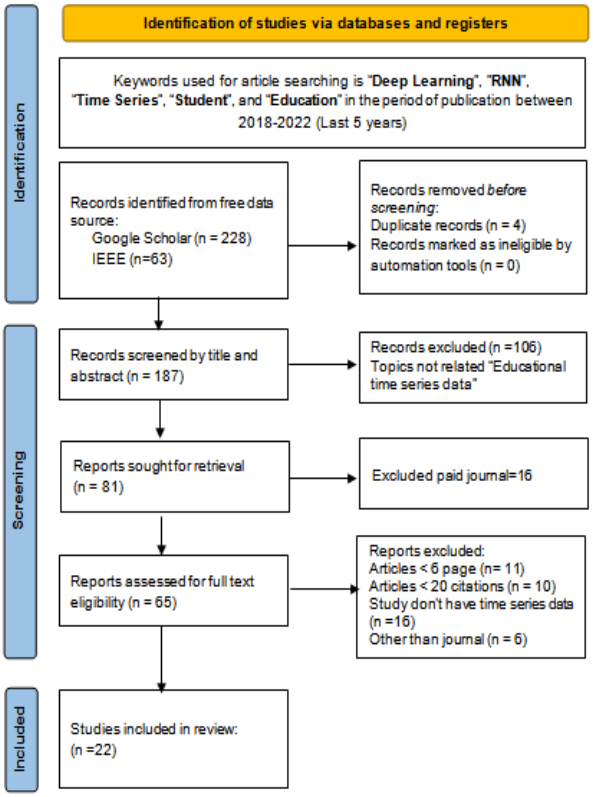

Fig. 1. PRISMA 2020 flow diagram of the study.

III. Results

This section describe the review results obtained through previous section. Fig. 2. depicts the publisher's contribution to this topic. It shows that most well-known publishers are involved, but springer published more articles than others.

Fig. 3. explains the number of publications year-wise. It depicted the growing trend from 2018 to 2021 and decreased in 2022. Due to corona, online education peaked in 2021, and most of the research was conducted on various dimensions using vast online data such as MOOC and other LMS platform.

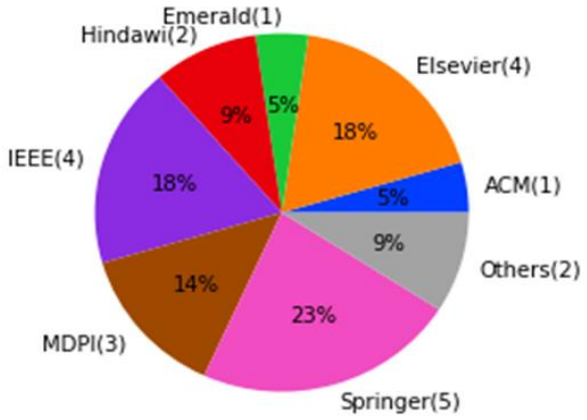

Fig. 2. Publisher contribution.

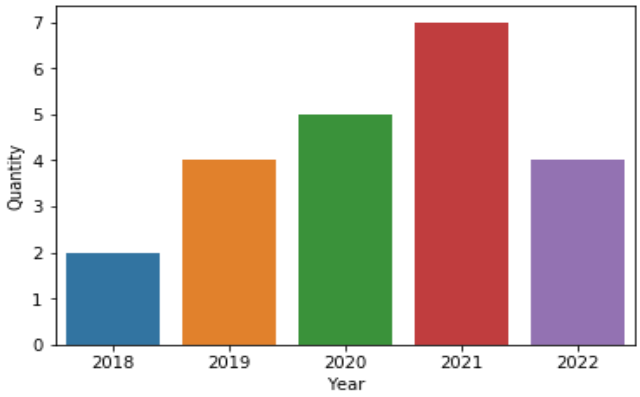

Fig. 3. Year wise publication.

A. *Contribution of Deep Learning in Educational Time Series Data*

Wang et al.,[24] proposed two novel methods to predict student learning status. The first one is to retrieve compelling features and performance using Conv-GRU. The second one, xNN (Explainable Neural Network) explains the relevance of student positive/negative results to improve the weak area. This approach helps to identify the hidden pattern of student behavior and early notification to improve the particular section.

Waheed et al., used the same dataset (OULAD) in both of their papers [25, 29], but followed different methodologies to predict the student category. The first work gives the highest accuracy (93%) than the second using DNN(84%), with a notable difference. The deep neural network(DNN) proves its power by providing the highest accuracy. Mubarak et al., [26] predict learner's weekly performance using video click stream for timely intervention. This model is created in such a way that it can adjust a variable window length routinely, which helps it to fit the RNN layer dimensions with different sizes of input data.

In studies [27, 28, 39, 42], all the authors used the same dataset (KDD cup 2015) to predict student dropout, and the result shows above 85% performance in all. It contains 39 courses and seven kinds of student behavioral information such as Access, video, wiki, discussion, navigate, page_close and problem. These multiple parameters allow applying a multi-variate time series approach. Though the dataset is the same, imbalanced data is handled only in [28, 39].

Zhang et al., [30] introduced a predictive model to pick the micro-level pattern from student learning behavior. To avoid data sparsity, the author divides the data into five clusters based on the nature of the student's learning behavior. Because, the author believes that every student's online learning behavior will change depending on their free time. An auto-encoder is used to encode time-series data. The significant difference between recall and accuracy values shows that classification errors need to fix in this model.

He and Gao [31] proposed a student performance predictive model by collecting student learning behavior information through terminal data acquisition tools to find the student concentration level in the classroom and explore the influencing factors of learning concentration. Aljaloud [32] suggested a model to predict student learning outcomes by selecting the number of essential features and evaluating the result by reducing the number of features. There are seven features(f1,f2,f3,f4,f5,f6,f7) and seven courses used in this LMS, and the final result shows the best accuracy in the more number of attribute combination.

Chen et al., [33] created an intelligent framework to handle imbalanced datasets and spatiotemporal information. This LMS contains eight learning features (F1-F8): assignment, file, forum, homepage, label, page, quiz, and URL. Course length is 16 weeks, but this model helps to warn the at-risk students much earlier than other models with higher accuracy.

Karim et al., [34] conducted ablation tests on time series data using LSTM. In this experiment the LSTM block is substituted by other techniques such as GRU, RNN, and Dense block. But LSTM-FCN performance was higher than others.

Chen et al., [35] provided a comparative study between deep learning and conventional machine learning using the data retrieved from the Learning Management System (LMS). This data tells the temporal behavior of the student activity in the form of time series. The author used classification and clustering techniques to predict early identification of at-risk students, and then compared the results using AUC.

Li et al., [36] also did a comparative study using higher education data such as student grades and levels to predict the performance. Prabowo et al., [37] tried dual input, the combination of categorical and numerical time series data. The proposed dual-input hybrid model combines MLP and LSTM networks and then compares the perrmance with the individual model.

Asish et al., [38] offered a comparative study using CNN, LSTM, and CNN-LSTM to classify the student distraction level using eye gaze data. The author collected the data through a Virtual Reality Environment. Wuet al., [39] proposed a hybrid model called CNN-Net to predict student dropout in MOOCs. Moreover, the author handled the class imbalance due to the massive dropout ratio of students.

Shin et al., [40] created a model to predict student performance using time series data by clustering the students using the k-shape technique. Each cluster helps to identify the student category to give a warning from the instructors. Bousnguar et al., [41] proposed a model for enrolment prediction using LSTM and statistical machine learning. The statistical model gives the highest accuracy than deep learning due to the insufficient data for training.

Qiu et al., [42] developed a model for dropout prediction using CNN. The author compares the results with baseline models, including LR, Naïve Bayes, Decision Tree, Random Forest, Gradient Tree Boosting, and SVM. Among those, CNN with windows size 10 showed better results than the others. Chen et al., [43] created a model for predicting course performance using an imbalanced dataset. The SMOTE sampling technique is applied to balance the minority data.The author used the KNN algorithm to fill in the arbitrary missing values.

Aljohani et al., [44] proposed a model to find the at-risk student in the early stage based on weekly performance sequence data. But it achieved the highest accuracy only after 38 weeks. He et al., [45] suggested a model for student performance prediction. The author used two fully connected neural networks for demographic information; RNN for handling student assessment and click stream time series data. The proposed method provided better performance than the baseline models.

Tables II provides the vital points of this review and Fig. 4 represent the classification of time series use case and workflow found in this review.

TABLE II. REVIEWS OF PREVIOUS WORK IN EDUCATIONAL TIME SERIES DATA

Source	Model	Dataset	Samples/Train-Test	Purpose/Findings	Individuality
[24]	ML, BPNN, RNN, GRU,LSTM, Conv-GRU-MaxP, Conv-GRU-AvgP	WorldUC, Liru Online Course Dataset	Datasets 1 - 7543 students, Dataset 2 - 347 students	Predictive model for student performance ML-75%, BPNN-76%, RNN-78%,LSTM-80%, GRU-81.3%, Conv-GRU-MaxP-81.8%, Conv-GRU-AvgP-82.2%	Weighted average pooling is used instead of max pooling in Conv-GRU to achive better performance.
[25]	LR,SVM,Deep ANN	Open University Learning Analytics Dataset (OULAD)	2014-2015 32,593 student log records. 70% - train 30% - test	The inclusion of legacy data and assessment-related data impact the model significantly. LR-85%, SVM - 89%, Deep ANN - 93%.	Instead of week- wise, at-risk students are identified in each quarter Q1,Q2,Q3, and Q4 along with distinction, pass, fail and withdrawal.
[26]	LR, SVM, Deep ANN, LSTM	MOOC - Stanford University Dataset	Student records for each dataset. course 1 - 5346 course 2 - 2135 course 3 - 3022 course 4- 2497 60%- training 30%- testing 10% - validation	Predicting learner performance and early dropout using video file click stream events and quiz score. LR=84%, SVM=85%, Deep ANN=85%, LSTM=93%	More than 90% accuracy in real time dataset.
[27]	CSLA(Hybrid model using CNN+Bi-LSTM+Attention Mechanism)	MOOC - KDD Cup 2015 dataset	2013-2014 79,186 students records 80% - training, 20%- testing	Predicting the student dropout rate based on learners' behavior data with accuracy - 87.6% and f1 score-86.9%.	Combined three different strategies to increase the performance over 2.8% . CNN - Feature selection LSTM - Time series Attention Mechanism- Assigning weight
[28]	LR, SVM, DNN, CONV-LSTM	MOOC- Dataset 1- Stanford University Dataset 2- KDD cup 2015.	Dataset 1- 78,623 records Dataset 2- 120,542 records 65% - training 20% - testing 15%-validation	Predicting Student Dropout using MOOC data with f1 score 89% and 90% for two datasets respectively.	Custom loss function applied to rectify classification error instead of SMOTE and other techniques used for imbalanced dataset.
[29]	LR, SVM, DT, GBT, KNN, ANN, LSTM	OLUAD	2014-2015 32,593 student log records. 75% - training 25% - testing	Predicting learners behavior using student online log data. LR-73%, SVM – 73%, DT-79%, GBT-78%, KNN-78%, ANN-83%, LSTM-84%.	Time series data(clickstream logs) converted into aggregated format for the purpose of applying classical machine learning.
[30]	LSTM - Encoder	Blackboard LMS	2014-2016 4706 students 3,625,619 log records	Predicting at-risk students using micro level behavioral pattern and time series clustering with accuracy-92%	Auto encoder used to extract best featuresStudent clustering based on the learning behavior
[31]	HMM, Machine Learning (ML), LSTM, CNN-LSTM	UCI - HAR dataset	10,299 data samples 70% - training 30 % - testing	Classroom attention behavior recognition using sensor data. HMM-72%, ML-78%, LSTM-89%, CNN-LSTM - 92%.	Identifying student concentration level through wearable device and mobile interaction data.
[32]	CNN, LSTM, CNN-LSTM	Blackboard LMS	35,000 students 1, 715, 000 records	Predicting student learning outcomes in LMS with fl	This study aims to find thedominate features called KPI

			70% - training 30%- testing	scoreLSTM - 90%, CNN - 92% CNN-LSTM - 93%	(Key Performance Indicator) to improve the model performance.
[33]	LR,SVM, LSTM, CNN-LSTM, Conv-LSTM	MOODLE-Gadjah Mada University	977 students 202,000 log records	Early prediction of at-risk students. LR-64%,SVM-80%,LSTM-85%, CNN-LSTM-88%, Conv-LSTM-91%	Hybrid SMOTE technique used for imbalance dataset More than 90% accuracy in predicting the first few weeks instead of final week.
[34]	LSTM-FCN, Attention LSTM-FCN, GRU, RNN, Dense block	University of California-Riverside (UCR) time series repository	Not specified due to the large number of experiment.	Z-normalization is recommended, if the training data having good representation of global population, Appling dimension shuffles before the LSTM block increases the performance.	Series of experiments(3627) using educational data, the State-of-Art performance for classifying the time series signal
[35]	NN,LR,NB, SVM,DT,RF, GBM,LSTM	Canadian university. (LMS log data-https://moodle.org/)	290 (semester 1)-train, validation, test data 311(semester 2)-Test data	Predicting the students' performance using LMS activity. LSTM (AUC: >60%) performs better than classical machine learning (AUC <60%).	SMOTE sampling technique is used to overcome the imbalanced dataset.
[36]	Linear Regression, LSTM	Multi-disciplinary university	2007-2016-training 2017-2019-testing	In Predicting student performance, deep learning offers the highest accuracy than the Linear model. MAE: 0.593 and RMSE: 0.785	Extracting informative data as a feature with corresponding weights. Multiple updated hidden layers were used for designing neural networks automatically
[37]	LSTM, MLP-LSTM	Bina Nusantara University	2011-2017 46,670 -univariate time-series and tabular data	Predicting student GPA using tabular and historical data. Hybrid model with dual input gives highest accuracy MSE:0.41, MAE:0.34, R-square:0.48.	Dual input to the hybrid model using tabular data and time series data
[38]	CNN, LSTM, CNN-LSTM	Student classroom data using Virtual Reality (VR) Environment	3.4 M data points 70%-Training 30%-Testing	Finding the student distraction level using Virtual Reality data by creating a deep learning model. The hybrid model achieves the highest accuracy at 89.8%	The large amount of data points were used for this experiment.
[39]	CNN-Net, CNN-LSTM,CNN-RNN, Classical Machine Learning	MOOC - KDD Cup 2015 dataset	79,186 students records 1,20,542 data points 80% - training, 20%- testing	Early prediction of student dropout in MOOC and Hybrid Model gives highest accuracy AUC: 91.5% than classical model.	The effort was given for pre-processing the data to handle categorical and imbalanced dataset.
[40]	RNN, LSTM, Deep LSTM	Dataset collected from Star Math-Formative Assessment tool	2017-2018, 10,107- records, 80% -training,10% test, 10% validation	Predicting student performance using previous test assessment. The short history (3 data points) prediction gives highest accuracy.	Student performance is predicted then categorized using clustering technique based on their performance.
[41]	ARIMA, LSTM, Exponential Smoothing, and Fuzzy Time Series algorithms	IBN ZOHR University	18 years data	Developed four different forecasting models using Time Series algorithms to predict the new student enrollment. Highest RMSE score Fuzzy Time Series: 211, ARIMA: 452,ES: 461, LSTM: 1152.	Comparison between Statistical and Deep Learning model.
[42]	CNN	MOOC - KDD Cup 2015 dataset	79,186 students records	Predicting student dropout in online courses.Precision: 86%, Recall: 87%, F1 score: 86%, AUC: 86%	CNN model is newly applied for student dropout prediction using click stream data.
[43]	NN, LR, NB, GBM SVM, DT, KNN, RF, and LSTM with SMOTE	Moodle LMS data –Canadian university	527 students 72% - training 28% - testing	Analyze student online temporal behavior using their LMS data for the early prediction of course performance. LSTM - AUC Score 80.1.	Separate models are created for 28, 48, 56, and 70 days data to evaluate the course performance for each semester.
[44]	LR, ANN, SVM, LSTM	Open University Learning Analytics Dataset (OULAD)	2014-2015 32,593 students with 20 different activity data.	Finding the at-risk students in the early stage using virtual learning environment video stream click event and demographic data. Recall score is LR=80, SVM=78, ANN=85, LSTM=95.	Student week-wise activities are stacked and given to the model for early prediction. The last week's data provides better performance.
[45]	RNN,GRU and LSTM	Open University Learning Analytics Dataset (OULAD)	2014-2015 32,593 student records.	Static and sequential informationswere combined for performance prediction. GRU gives better performance than LSTM due to minimal length data. The accuracy of the proposed model is above 80% in the last week.	The joint neural network is proposed to fit both static and sequential data, where the data completion mechanism is also adapted to fill the missing stream data.

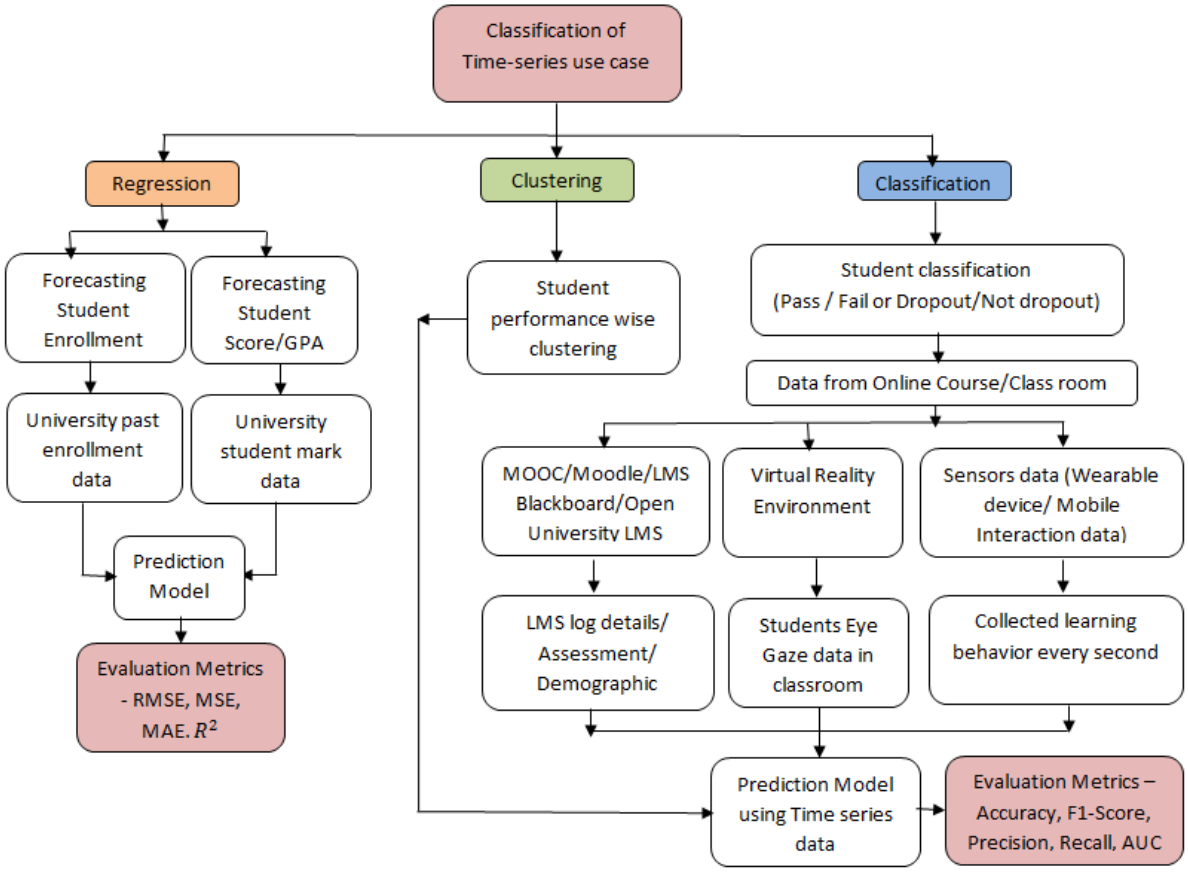

Fig. 4. Classification of time series problem and workflow found in this review.

B. *The Architecture of Time Series Model and the Difference between Traditional Approach*

This section provides the history and technical background of Recurrent Neural Networks. Even though a few studies used other models (CNN and hybrid models) for time series problems, those are excluded and not specific to handle temporal data.

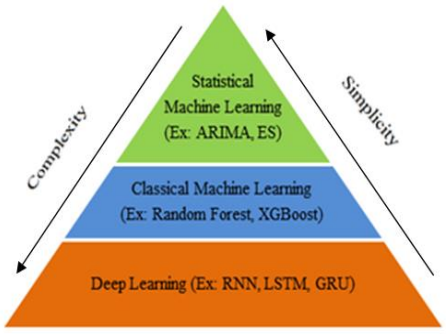

Fig. 5. Classification of time-series algorithms.

Initially, statistical methods are beneficial in predicting time series problems, but they are ineffective in handling nonlinear data. Therefore deep learning came into existence to overcome the liabilities of conventional time series algorithms such as ARIMA and Exponential smoothing techniques [9], [10], [20]. Similarly, few classical machine learning algorithms (XGBoost) apply to time series problems. Fig. 5 illustrates the complexity and the simplicity level of different time series algorithms.

The properties of time series data are Trend, Seasonal, Cyclic, and Irregular. Fig. 6 describes the pictorial representation of each property.

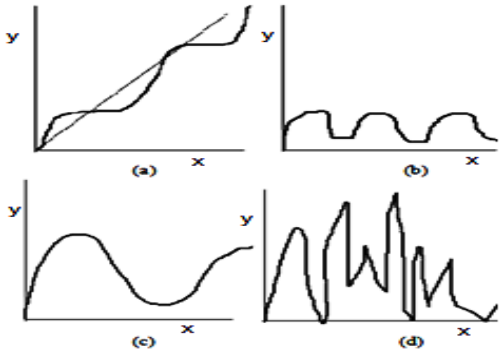

Fig. 6. Time series properties (a). Trend (b). Seasonal (c). Cyclic (d). Irregular.

Here ‘x’ denotes any time unit such as minutes, hours, months, years, etc. And the ‘y’ represents the numerical value such as weight, height, price, quantity, etc. Fig. 6 explains how the ‘y’ value is changed based on time. The appropriate algorithm has to prefer built on the type of the dataset. RNN

has introduced around the 1980s. However, it got renowned after the invention of LSTM in 1990 to overcome the weaknesses of RNN. The most common use case for RNN is time series problems [11] and natural language processing [12]. Fig. 7 depicts the workflow difference between traditional Feed Forward Neural Networks (FFNN) and RNN.

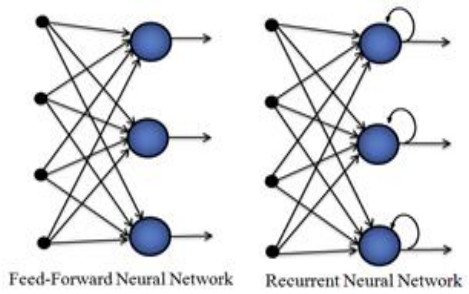

Fig. 7. FFNN vs RNN.

1) Recurrent Neural Network (RNN): RNN is a type of Artificial Neural Network (ANN) specially designed to capture sequential information with the aid of memory cells. This memory cell retains the previous report for further processing, and the decision is based on the prior and current state. RNN shares the same weight parameters within each layer, whereas the traditional neural network shares different weights. There are three crucial components in RNN Input, hidden neuron, and activation function, as described in Fig. 8 and 9.

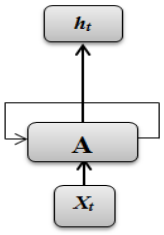

Fig. 8. Simple RNN.

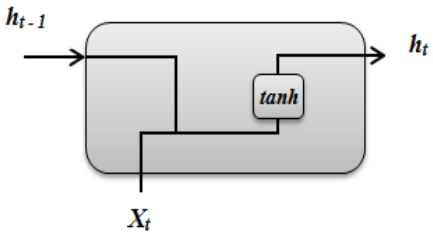

Fig. 9. Internal architecture of simple RNN.

$$h_t = tanh(\mathrm{U}.x_t + W.h_{t-1}) \quad (1)$$

Eq. (1) calculates the hidden state where h_t is a hidden neuron at time t, x_t is the input at time *t*, *U* is the weight of the hidden layer and *W* is the transition weight of the hidden layer. The input and previous state informations are combined to go through the *tanh*activation function to produce a new hidden state. RNN suffers from the vanishing gradient problem while handling long sequence data. But it is rectified by Long Short-Term Memory[19], another variant of RNN.

2) Long Short Term Memory (LSTM): LSTM is capable of processing long-term dependency data. It manages the previous context more effectively than RNN using three gates. They are the input gate, forget gate, and output gate, as depicted in Fig. 10. The input gate updates the memory cell, forget gate decides whether the information has to be kept or not. The output gate is responsible for determining the next hidden state.The loop structure of RNN and LSTM helps to choose the better weight parameter. The formula for each variable in LSTM is defined below:

$$f_t = \sigma(W_f.X_t + U_f.h_{t-1} + b_f) \quad (2)$$

$$i_t = \sigma(W_i.X_t + U_i.h_{t-1} + b_i) \quad (3)$$

$$S_t = tanh(W_c.X_t + U_c.h_{t-1} + b_c) \quad (4)$$

$$C_t = i_t * S_t + f_t * S_{t-1} \quad (5)$$

$$o_t = \sigma(W_o.X_t + U_o.h_{t-1} + V_o.C_t + b_o) \quad (6)$$

$$h_t = o_t * tanh(C_t) \quad (7)$$

where i_t, f_t, o_t refers to the input gate, forget gate, and out gate respectively. *W, U,* and *V* are the weight matrices, *b* is the bias vectors, X_t is the input vector to the memory cell at time t, h_t is the value of the memory cell at time t, and C_t, S_t are the candidate state and state of the memory cell at time *t*, respectively. Here sigmoid (σ) and *tanh* are the activation functions.

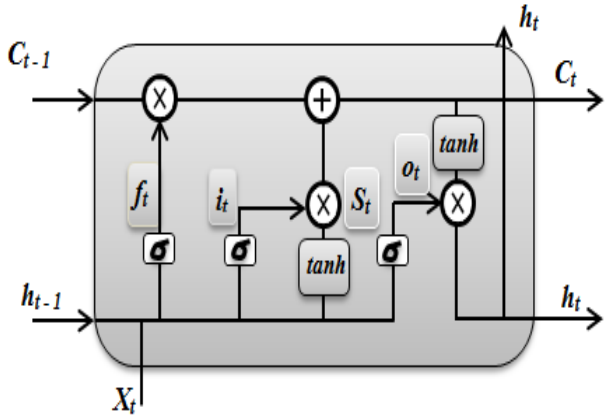

Fig. 10. LSTM architecture.

3) Gated Recurrent Unit (GRU): GRU is a simple version of RNN in terms of architecture. It is uncomplicated to implement and has a quick performance than LSTM, but the functionalities of both architectures are identical. GRU uses fewer parameters, so it requires less hardware and training time. Therefore, GRU attracts the user to involve in many applications. The three gates are reduced into two gates update and reset gate, defined in Fig. 11.

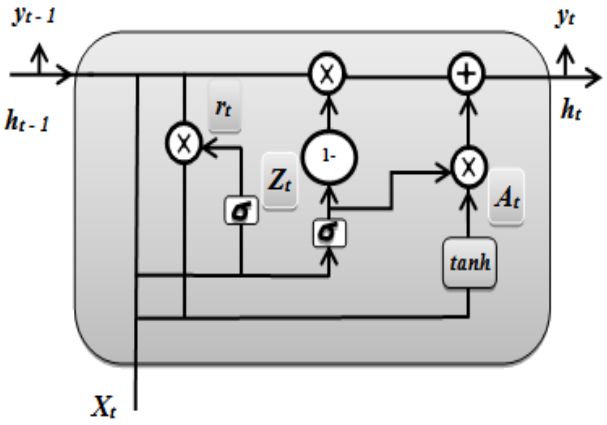

Fig. 11. GRU architecture.

The update gate is a combination of the input, and a forget gate in LSTM. It decides whether the particular information has to be kept or discarded. The reset gate will determine the amount of data that should forget. The following formula defines each variable in GRU:

$$r_t = \sigma(W_r.X_t + U_r.h_{t-1} + b_r) \quad (8)$$

$$Z_t = \sigma(W_z.X_t + U_z.h_{t-1} + b_z) \quad (9)$$

$$A_t = tanh(W_h.X_t + U_h.(r_t * h_{t-1}) + b_h) \quad (10)$$

$$h_t = (1 - Z_t) * h_{t-1} + Z_t * A_t \quad (11)$$

where r_t and Z_t are the two gates for reset and update respectively. A_t is memory content, h_t is the final memory of the current time step and the σ and *tanh* are the activation functions. The two gates have values between 0 and 1 through the sigmoid function (σ). While doing, the memory content (A_t), using the reset gate store the significant information from the previous value between the range −1 to 1 over *tanh*.

4) Metrics used for time-series data: Choosing the right metric is essential to evaluating the model's performance. All the decision, such as tuning the hyper-parameter and selecting the suitable model, is made on the result only. Here the notable thing is before deciding the metrics, need to check the following entities: the nature of the dataset, the values going to handle, and whether there is any need to compare other datasets. If so, are they all on the same scale or different ones? Table III and Table IV illustrate the various metrics available for the time series problem [14]-[18]. Fig. 12 shows the percentage of performance metrics reported in this study.

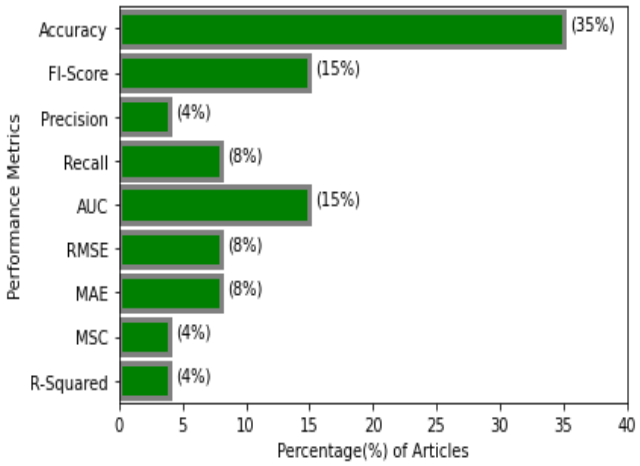

Fig. 12. Percentage of performance metrics applied in this review.

TABLE III. TYPES OF PERFORMANCE METRICS IN REGRESSION PROBLEM

Metrics for Regression			
Scale-Dependent	**Percentage-Error**	**Relative-Error**	**Scale-Free Error**
• Mean Absolute Error(MAE) • Mean Squared Error (MSE) • Root Mean SquaredError (RMSE)	• Mean Absolute Percentage Error(MAPE) • Symmetric Mean Absolute Percentage Error (SMAPE)	• Median Relative Absolute Error (MdRAE) • Geometric Mean Relative Absolute Error (GMRAE)	• Mean Absolute Scaled Error (MASE)
Description			
Error metrics are articulated in the units of the underlying data (Example: Dollars, Inches, etc.)	Scale independent and used to compare forecast performance between different time series	Compare your model's performance with the baseline or benchmark model.	Scale the error based on the in-sample MAE from a random walkforecast method
Advantage			
Easy to calculate and interpret	Scale- independency and easy interpretability	Scale-independence	Scale free and suitable metric for time series data with zeros
Disadvantage			
Scale dependency	- Infinite or undefined values for zero or close-to-zero actual values - Heavier penalty on negative than on positive errors - Cannot be used when using percentages make no sense.	When the calculated errors are small it leads to division by zero error.	Not to use when all historical observations are equal or all of the actual values during the in-sample period were zeros.

TABLE IV. TYPES OF PERFORMANCE METRICS IN CLASSIFICATION PROBLEM

Metrics for Classification			
Name	**Description**	**Advantage**	**Disadvantage**
AUC/ROC	The **AUC** measures the entire two-dimensional area under the curve at all possible classification thresholds.ROC is a plot to explain the true and false positive rates.	Using graph representation to show the trade-off between the TPR and FPR.	Not suitable for the highly imbalanced dataset and concentrates only on TPR and FPR.
Confusion Matrix	Identify the model correctness all the way. The Four elements of this table are TP, TN, FP, and FN, which helps to derive the following metrics.	Find the issue where the model failed to understand.	Interpreting the result is complex.
Accuracy	The degree of model correctness. ***Accuracy=(TP+TN)/(TP+FN+TN+FP)***	Easy to interpret	Misleading the result where the sample of minority class is very less.
Precision	Ability of the model to identify only the relevant data points. ***P=TP/(TP+FP)***	Identify the proportion of correct positive identifications	It doesn't consider the type II classification error.
Recall (Sensitivity)	Ability of the model to find all the relevant data points. ***R=TP/(TP+FN)***	Identify the proportion of correct actual positives.	It doesn't consider the type I classification error.
F1-Score	A single score that balances both the concerns of precision and recall in one number. ***F1-Score = 2 *(P*R)/(P+R)***	The harmonic mean of precision and recall value	It is a combined result of precision and recall, so a bit harder to interpret.

TABLE V. LIST OF HYPER-PARAMETERS

Hyper-Parameters	Description
Train/Test ratio	Splitting the dataset into train and test. (Example: 80:20)
Hidden Layer	The layer between input and output and it determines the depth of the neural network (Usually 1 or 2 layers).
Optimizer	It is an algorithm used to update the weight of each layer after each iteration (Example: Gradient descent, Adam [26,27,39,43,45,29,31,32,33])
Learning Rate	It defines how quickly the network updates its parameters (0.0-0.1)
Activation function	Allowing deep learning models to learn non-linear prediction boundaries [22] (Example: Sigmoid [28, 39], ReLU[24,28,32], Tanh[26,32], Leaky Relu[45])
Number of Epochs	Number of iterations to pass the whole dataset in training.
Batch-Size	Number of sample that the network used to update the weights
Momentum	It speeds up the learning process by preventing the oscillation in the convergence of the method.
Weight initialization	It defines the starting point of the optimization.
Dropout	It helps to avoid over-fitting by eliminating the randomly selected neurons in training [26, 27, 28, 41,43, 45].
Regularization	It prevents over-fitting by stopping the weights that are too high(L1,L2) [26]
Units	It determines the level of knowledge that is extracted by each layer.

C. Factors Affecting the Time Series Model Accuracy

There are several factors affecting the model performance which are the techniques used for pre-processing,train-test ratio, and the selection of model hyper-parameters. Table V represents the hyper-parameters that are affecting the model accuracy [23].

1) Pre-processing: Removing unwanted data and filling in the missing values are the initial step inpre-processing. Several methods are available for imputation, such as mean, median,mode, interpolation, weighted average [24], and k-nearest neighbor[43]. Mean and Weighted Average is the widelyused techniques. The first one returns the average value of the feature column, and the second substitute the average of the most frequent information.

The next step is to encode all the categorical information into numerical value for model understanding using any technique such as label encoding or one hot encoding [39]. Each method has merits and demerits of its own. After encoding, re-scaling the data (feature) is very important since it makes the model less sensitive to the scale of features and allows converging with better weights. There are two significant types of scaling: Standardization(z-score) and Normalization (min-max scalar). Standardization assumes that the values are in Gaussian distribution and centered on the zero mean with unit standard deviation. It is less sensitive to outliers, so Karim et al., [34] used z-score normalization to handle the outlier values, and Wang et al., used batch normalization for scaling in [24]. It is specific to each layer and batch of input in the neural network.

2) Train-test split: Normally, 80:20 is the suggested ratio for a train-test split if the samples are distributed evenly across the dataset. Wu et al., [27,39] split the dataset into 80:20 for training and testing. Shin et al., carry the exact ratio in [40], but 10 % for validation from test data. In researches [25,38, 31, 32], the percentage used for training/testing is 70:30; the remaining studies [43, 26, 28, 29] use slightly different ratios.

3) Imbalanced dataset: Most of the real-time dataset is imbalanced and should be handled appropriately to avoid classification errors. In studies [35, 39, 43, 33], the authors used synthetic samples (SMOTE) to balance the target class count. But Mubarak et al.,[28] introduced a cost-sensitive technique in the loss function to avoid type 2 classification error[28]. Dimension reduction is also another issue where the feature count is vast. Waheed et al.,[25] using Singular Value Decomposition (SVD) method to find the top 30 efficient features.

4) Hyper-parameters: The number of hidden layers is significant in deep learning because it shows the complexity of the problem. Bousnguar et al., [41] used three LSTM layers and 50 cells for each layer. Qiu et al.,[42] involved two convolutional and two fully connected layers for binary classification with the sigmoid activation function. Aljohani et al.,[44] applied three LSTM layers, and each layer is assigned 100 to 300 units of neurons. Deep ANN is appliedin [25,45] and uses a minimum of three layers and up to seven hidden layers. Next to hidden layers, select the suitable optimizer to update the weight for every iteration. The Adam optimizer is majorly used [39, 43, 45, 26, 27, 29, 31, 32, 33] among others, such as gradient descent, stochastic gradient descent, and RMSProp.

Then the learning rate (0.0-0.1) assigns the speed of the network parameter update. Frequently used values are 0.0025 [27], 0.001 [29, 32, 33], and 0.1 [31]. The activation function is another hyper-parameter that helps to predict complex non-linear data. This parameter differentiates neural networks compared with machine learning models. Relu, Leaky Relu, tanh, and sigmoid are the activation functions equally used in all the papers. After fitting these parameters training the model by mentioning the number of epochs is mandatory. Sometimes less training, such as 15, 20, and 25, gives better performance than massive iteration [27, 31]. The Dropout is the last layer of the neural network to avoid overfitting, so it is majorly used in all the experiments. The frequent values are 0.1, 0.2, 0.3, and a maximum of 0.5 by Mubarak et al., in [28].

Concerning batch size, the authors adjusted the value to improve the accuracy by doing several experiments. Waheed et al., found the batch size from the value of 64 increased the model performance for all the weeks, but when the batch size was increased additionally from 1364, the model performance degraded with AUC decreasing by a value of 0.04. Regularization is rarely used [26] in the experiment. The model setup does not explain the other parameters, such as Weight initialization and Momentum.

IV. DISCUSSION

This section discuss the contribution of this paper referred to in the introduction.

A. Finding the Impact of Deep Learning in Educational Time Series Problem

To answer RQ1, this SLR proved the success of deep learning models in educational time series data retrieved from various sources. Hernández et al.,[21] also confirmed in their review the number of publications recently enriched after raising the application of the DL model. But the first publication commenced in 2015. Section III describes the previous work, and all the information is summarized in Table II to explain the types of models used, the paper's findings, individuality, and the dataset details. The CNN-LSTM is the majorly used hybrid technique, and the LSTM is the widely used single model in many research works. In Education, student performance prediction is the typical use case executed in multiple investigations. Moreover, clustering the time sequence data was also applied to categorize the students based on their performance. Due to its sequential nature, most of the work was done on MOOC online data than the offline mode to predict student dropout.

B. Identify the Architecture of Time Series Model and How it Differs from the Traditional Approach

To answer RQ2, Section IV describes the internal structure of RNN, LSTM, and GRU using the required diagrams and formulas. It represents the improvement and differences between each other. Numerous investigations involve LSTM rather than RNN and GRU though the architecture is intricate. Also, LSTM merged with CNN to retrieve spatiotemporal features effectively. Self-connected neurons helps to maintain the previous information, and this is different from feed-forward neural network.

C. Discover the Significant Factors Affecting the Time Series Model Accuracy

To answer RQ3, Table V provides information on the factors influencing the model accuracy. Tuning the Neural Network is necessary because it improves the model's performance. The number of hidden layers, epochs, batch size, dropout layer, and optimizers are the commonly used hyper-parameters due to their high impact on the outcome. Most authors use manual selection to pick the best hyperparameters instead of any optimization technique, such as grid search and Bayesian method.

V. CONCLUSION

Understanding the DL methodology and the previous work done in a particular domain is fundamental before implementing the research idea. This study is the first work that gives a background for young researchers who want to involve Deep Learning in the Education time series problem. Accessed Google Scholar and IEEE Xplore scientific websites to collect relevant research papers. Then the collected documents(n=291) are analyzed manually and selected twenty-two (n=22) papers for this SLR by following PRISMA methodology. The essence of this survey is deep learning applies widely, but the hybrid model gave the highest accuracy than the individual model. Student classification, clustering, forcasting the student enrolment/grade, and dropout prediction using online course log data are the normally used problem statement. Large sequential data are rarely used compared with other domains which helps to avoid complex models. Finally discussed the RNN architecture, types of metrics, and the factors influencing the model accuracy.

VI. FUTURE PERSPECTIVE

This deliberation clearly explains the previous work done in the educational domain using time series data and will involve all this learning in the implementation work to fill the research gap identified.

REFERENCES

[1] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18569–18584, Dec. 2016, doi: 10.1007/s11042-016-4159-7.

[2] S. Ji, J. Kim, and H. Im, "A Comparative Study of Bitcoin Price Prediction Using Deep Learning," *Mathematics*, vol. 7, no. 10, p. 898, Sep. 2019, doi: 10.3390/math7100898.

[3] M. O. Alassafi, M. Jarrah, and R. Alotaibi, "Time series predicting of COVID-19 based on deep learning," Neurocomputing, vol. 468, pp. 335–344, Jan. 2022, doi: 10.1016/j.neucom.2021.10.035.

[4] H. T. Rauf et al., "Time series forecasting of COVID-19 transmission in Asia Pacific countries using deep neural networks," Personal and Ubiquitous Computing, Jan. 2021, doi: 10.1007/s00779-020-01494-0.

[5] P. Hewage, M. Trovati, E. Pereira, and A. Behera, "Deep learning-based effective fine-grained weather forecasting model," Pattern Analysis and Applications, Jun. 2020, doi: 10.1007/s10044-020-00898-1.

[6] D. Sun, J. Wu, H. Huang, R. Wang, F. Liang, and H. Xinhua, "Prediction of Short-Time Rainfall Based on Deep Learning," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–8, Mar. 2021, doi: 10.1155/2021/6664413.

[7] P. P. Sarkar, P. Janardhan, and P. Roy, "Prediction of sea surface temperatures using deep learning neural networks," *SN Applied Sciences*, vol. 2, no. 8, Jul. 2020, doi: 10.1007/s42452-020-03239-3.

[8] S. Biswas and M. Sinha, "Performances of deep learning models for Indian Ocean wind speed prediction," *Modeling Earth Systems and Environment*, vol. 7, no. 2, pp. 809–831, Sep. 2020, doi: 10.1007/s40808-020-00974-9.

[9] A. P, "Higher Education Institution (HEI) Enrollment Forecasting Using Data Mining Technique," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 2, pp. 2060–2064, Apr. 2020, doi: 10.30534/ijatcse/2020/179922020.

[10] B. Siregar, I. A. Butar-Butar, R. Rahmat, U. Andayani, and F. Fahmi, "Comparison of Exponential Smoothing Methods in Forecasting Palm Oil Real Production," *Journal of Physics: Conference Series*, vol. 801, p. 012004, Jan. 2017, doi: 10.1088/1742-6596/801/1/012004.

[11] A. Graves,"Long short-term memory," *Supervised sequence labelling with recurrent neural networks*, pp. 37-45, 2012.

[12] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Oct. 2019, doi: 10.2478/jaiscr-2019-0006.

[13] R. Jayashree, "Enhanced Classification Using Restricted Boltzmann Machine Method in Deep Learning for COVID-19," *Understanding COVID-19: The Role of Computational Intelligence*, pp. 425–446, Jul. 2021, doi: 10.1007/978-3-030-74761-9_19.

[14] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and practice. Melbourne: OTexts, 2021.

[15] C. Chen, J. Twycross, and J. M. Garibaldi, "A new accuracy measure based on bounded relative error for time series forecasting," *PLOS ONE*, vol. 12, no. 3, p. e0174202, Mar. 2017, doi: 10.1371/journal.pone.0174202.

[16] V. R. Jose, "Percentage and relative error measures in forecast evaluation," Operations Research, vol. 65, no. 1, pp. 200–211, 2017, doi:10.1287/opre.2016.1550

[17] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, Jul. 2016, doi: 10.1016/j.ijforecast.2015.12.003.

[18] V. Cerqueira, L. Torgo, J. Smailović and I. Mozetič, "A Comparative Study of Performance Estimation Methods for Time Series Forecasting," 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Tokyo, Japan, 2017, pp. 529-538, doi: 10.1109/DSAA.2017.7.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] S. Vanitha and R. Jayashree, "A Prediction On Educational Time Series Data Using Statistical Machine Learning Model -An Experimental Analysis," Journal of Theoretical and Applied Information Technology, vol. 100, no. 14, pp. 5189–5200, Jul. 2022.

[21] A. Hernández-Blanco, B. Herrera-Flores, D. Tomás, and B. Navarro-Colorado, "A Systematic Review of Deep Learning Approaches to Educational Data Mining," *Complexity*, vol. 2019, pp. 1–22, May 2019, doi: 10.1155/2019/1306039.

[22] A. Farzad, H. Mashayekhi, and H. Hassanpour, "A comparative performance analysis of different activation functions in LSTM networks for classification," *Neural Computing and Applications*, vol. 31, no.7, pp. 2507–2521, Oct. 2017, doi: 10.1007/s00521-017-32106.

[23] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework," *IEEE Access*, vol. 6, pp. 49325–49338, 2018, doi: 10.1109/access.2018.2868361.

[24] X. Wang, P. Wu, G. Liu, Q. Huang, X. Hu, and H. Xu, "Learning performance prediction via convolutional GRU and explainable neural networks in e-learning environments," *Computing*, vol. 101, no. 6, pp. 587–604, Jan. 2019, doi: 10.1007/s00607-018-00699-9.

[25] H. Waheed, S.-U. Hassan, N. R. Aljohani, J. Hardman, S. Alelyani, and R. Nawaz, "Predicting academic performance of students from VLE big data using deep learning models," *Computers in Human Behavior*, vol. 104, p. 106189, Mar. 2020, doi: 10.1016/j.chb.2019.106189.

[26] A. A. Mubarak, H. Cao, and S. A. M. Ahmed, "Predictive learning analytics using deep learning model in MOOCs' courses videos," *Education and Information Technologies*, vol. 26, no. 1, pp. 371–392, Jul. 2020, doi: 10.1007/s10639-020-10273-6.

[27] Q. Fu, Z. Gao, J. Zhou, and Y. Zheng, "CLSA: A novel deep learning model for MOOC dropout prediction," *Computers & Electrical Engineering*, vol. 94, p. 107315, Sep. 2021, doi: 10.1016/j.compeleceng.2021.107315.

[28] A. A. Mubarak, H. Cao, and I. M. Hezam, "Deep analytic model for student dropout prediction in massive open online courses," *Computers & Electrical Engineering*, vol. 93, p. 107271, Jul. 2021, doi: 10.1016/j.compeleceng.2021.107271.

[29] H. Waheed, S. U. Hassan, R. Nawaz, N. R. Aljohani, G. Chen, and D. Gasevic, "Early prediction of learners at risk in self-paced education: a neural network approach," *Expert Systems with Applications*, vol. 213, no. Part A, p. 118868, Mar. 2023, doi: 10.1016/j.eswa.2022.118868.

[30] M. Zhang, X. Du, K. Rice, J.-L. Hung, and H. Li, "Revealing at-risk learning patterns and corresponding self-regulated strategies via LSTM encoder and time-series clustering," *Information Discovery and Delivery*, vol. 50, no. 2, pp. 206–216, Jun. 2021, doi: 10.1108/idd-12-2020-0160.

[31] K. He and K. Gao, "Analysis of Concentration in English Education Learning Based on CNN Model," *Scientific Programming*, vol. 2022, pp. 1–10, Jul. 2022, doi: 10.1155/2022/1489832.

[32] A. S. Aljaloud *et al.*, "A Deep Learning Model to Predict Student Learning Outcomes in LMS Using CNN and LSTM," *IEEE Access*, vol. 10, pp. 85255–85265, 2022, doi: 10.1109/access.2022.3196784.

[33] H.-C. Chen *et al.*, "Week-Wise Student Performance Early Prediction in Virtual Learning Environment Using a Deep Explainable Artificial Intelligence," *Applied Sciences*, vol. 12, no. 4, p. 1885, Feb. 2022, doi: 10.3390/app12041885.

[34] F. Karim, S. Majumdar, and H. Darabi, "Insights Into LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 7, pp. 67718–67725, 2019, doi: 10.1109/access.2019.2916828.

[35] F. Chen and Y. Cui, "Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course

Performance," *Journal of Learning Analytics*, vol. 7, no. 2, pp. 1–17, Sep. 2020, doi: 10.18608/jla.2020.72.1.

[36] S. Li and T. Liu, "Performance Prediction for Higher Education Students Using Deep Learning," *Complexity*, vol. 2021, pp. 1–10, Jul. 2021, doi: 10.1155/2021/9958203.

[37] H. Prabowo, A. A. Hidayat, T. W. Cenggoro, R. Rahutomo, K. Purwandari, and B. Pardamean, "Aggregating Time Series and Tabular Data in Deep Learning Model for University Students' GPA Prediction," *IEEE Access*, vol. 9, pp. 87370–87377, 2021, doi: 10.1109/access.2021.3088152.

[38] Asish, S.M., Hossain, E., Kulshreshth, A.K. and Borst, C.W., "Deep Learning on Eye Gaze Data to Classify Student Distraction Level in an Educational VR Environment", In *ICAT-EGVE 2021-International Conference on Artificial Reality and Telexistence and Eurographics Symposium on Virtual Environments*, Vol. 20211326, *https://doi.org/10.2312/egve*.

[39] N. Wu, L. Zhang, Y. Gao, M. Zhang, X. Sun, and J. Feng, "CLMS-Net: dropout prediction in MOOCs with deep learning," *Proceedings of the ACM Turing Celebration Conference - China*, pp.1-6, 2019.

[40] J. Shin, F. Chen, C. Lu, and O. Bulut, "Analyzing students' performance in computerized formative assessments to optimize teachers' test administration decisions using deep learning frameworks," *Journal of Computers in Education*, Aug. 2021, doi: 10.1007/s40692-021-00196-7.

[41] H. Bousnguar, L. Najdi, and A. Battou, "Forecasting approaches in a higher education setting," *Education and Information Technologies*, vol. 27, no. 2, pp. 1993–2011, Aug. 2021, doi: 10.1007/s10639-021-10684-z.

[42] L. Qiu, Y. Liu, Q. Hu, and Y. Liu, "Student dropout prediction in massive open online courses by convolutional neural networks," *Soft Computing*, vol. 23, no. 20, pp. 10287–10301, Oct. 2018, doi: 10.1007/s00500-018-3581-3.

[43] F. Chen and Y. Cui, "Utilizing Student Time Series Behaviour in Learning Management Systems for Early Prediction of Course Performance," *Journal of Learning Analytics*, vol. 7, no. 2, pp. 1–17, Sep. 2020, doi: 10.18608/jla.2020.72.1.

[44] N. R. Aljohani, A. Fayoumi, and S.-U. Hassan, "Predicting At-Risk Students Using Clickstream Data in the Virtual Learning Environment," *Sustainability*, vol. 11, no. 24, p. 7238, Dec. 2019, doi: 10.3390/su11247238.

[45] Y. He *et al.*, "Online At-Risk Student Identification using RNN-GRU Joint Neural Networks," *Information*, vol. 11, no. 10, p. 474, Oct. 2020, doi: 10.3390/info11100474.