Received: November 16, 2023. Revised: February 4, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Ed-Net: Multivariate Time Series Approach for Uncovering Student Learning Outcome in Higher Education Using Blended Deep Learning Technique

Vanitha S[1] **Jayashree R[1]***

[1]*Department of Computer Applications, Faculty of Science and Humanities,*
SRM Institute of Science and Technology, Kattankulathur, Chennai, India
* Corresponding author's Email: jayashrr@srmist.edu.in

Abstract: The higher education system (HES) in every country depicts the progress and prominence of that nation. So, the government endures the most care at all levels to enrich the quality of education on both the educator and learner side. The new academic policies also urge for worthwhile education. The HES abides by multiple steps for students' welfare in the academic curriculum, including a revamped syllabus, teaching methodology, and evaluation system. Nevertheless, students' performance is falling yearly, particularly in undergraduate programs. After COVID-19, students' study behaviour transformed abruptly due to online classes where students use mobile phones for education. They spend valuable time on the internet and gaming applications, which makes the students addicted. The evolution of Artificial Intelligence and massive student data permits us to get back the next generation by doing periodic assessments during their study period. This paper proposed a blended deep learning binary classification model (Ed-Net) using convolutional neural network and bidirectional long short term memory to predict the students' performance. Multivariate time series (MTS) student academic data is employed to train the model. To accomplish this systematic research, we followed two stages in the experiment. Stage one identifies the superlative student input data, approach (tabular/time-series), and algorithm (machine/deep learning) for better classification. Stage two executes the proposed system (Ed-Net) to attain the highest accuracy with less classification error. The synthetic minority oversampling technique (SMOTE) is applied to balance the students' pass-fail ratio. Finally, the experimental result exhibits the proposed method surpassed the baseline models with 98% accuracy, 97% precision, 94% recall, and 95% fl-score. The proposed model also uses a benchmark dataset to simulate the data and evaluate its efficacy. Moreover, any educational institution can quickly fit the academic data into this model to identify the students lacking in studies early for giving proper intervention to the parents, teachers, and students.

Keywords: Multivariate time series, Student performance classification, Machine learning, Deep learning, Bidirectional long short term memory, Convolutional neural network.

1. Introduction

Higher education system (HES) majorly comprises three modes of courses: Full-time, part-time, and distance education. The full-time classes are taught physically (Offline) in the classroom every day. On the other hand, evening or weekend classes are conducted for part-time and distance education. All these courses follow semester-wise evaluation to produce the result. Apart from this, many online certification courses are available for learners who want to learn irrespective of age, location, and background. Online courses aid a lot of learners in the past pandemic situation. Even though there is some benefit and liabilities, that implication is still in higher education [1, 2].

The prevalent online course providers are Coursera, SWAYAM, EDx, Udemy, etc. Working people, students, and educators predominantly use these online certification courses. Higher education recently oriented a credit-based course completion method [3]. Accordingly, a regular student can fulfill 20% to 40% of credits through MOOCs available in the SWAYAM platform [4]. Here, the week-wise assignment permits student evaluation to produce the results. As per the review [5], the significant problem statement in offline and online education is student performance and dropout prediction, respectively. Both issues help the students' successful completion and promote the pass ratio. However, the former is more crucial since a lot of money, time, infrastructure, and other resources are required to complete a regular course. Conversely, online courses are less expensive, short duration, and the infrastructure is optional.

Also, the student's failure or dropping out will not significantly impact the learner's career. But, the student's performance in regular classes decides the placement and higher studies. These are all the motivations to prioritize the problem statement: "student performance prediction using higher education data" instead of taking any online course dataset. Also, considerable research studies have been accomplished on student dropout prediction using popular online LMS datasets [22-32], such as MOOC, OULAD, and KDD Cup 2015. These studies widely follow a time series approach but are limitedly applied in offline courses that follow the traditional tabular data method. However, Li and Prabowo [33, 34] used sequential data to predict the final GPA of regular students. The main objective of this research is to prove that the multivariate time series approach and algorithms give better predictions than traditional tabular data using classical machine learning algorithms. Moreover, this study finds the most straightforward input and method for projecting the weak students from the educational institution's standpoint.

Time series models are becoming popular due to their tremendous implementation over sequential data. Time series data implies assembling the data based on equal time intervals, for example, seconds, minutes, hours, weeks, months, quarters, and years. There are two types: Univariate and Multivariate. The prior one refers to single sequential information, and multivariate refers to the collection of multiple sequences. Time series data have more than one entry in the dataset and are related to the previous one by time. Oppositely, each record in tabular data is independent, and all the attributes are correlated only with the particular entry, which is known as static data. Deep learning (DL) has demonstrated remarkable efficacy in a variety of disciplines, including computer vision [6], speech recognition [7], natural language processing [8], time series data, and gaming.

One of the most significant evolutions in deep learning is the emergence of convolutional neural networks (CNNs) for multi-dimensional data and recurrent neural networks (RNNs) for sequential information. These two efficient strategies are integrated in this work to construct a dedicated model for student performance classification. Benchmark studies use different student details, approaches, and algorithms to classify the student category. However, this study applies a multivariate time series approach using a blended model to fuse the spatiotemporal characteristics.

The research questions and contribution of this paper are as follows:

- Which is the most suitable student data and approach to predict the learning outcome?
  Three distinct models are developed to evaluate three input approaches: static, static + time series, and time series to find the best student data for prediction.
- Which algorithm performs better while following the various data approaches?
  Three different algorithms, ML, hybrid (MLP-RNN), and RNN, are applied to various student information to identify the best one for performance classification.
- What is the impact of integrating spatio-temporal features in a multivariate time-series dataset?
  Ed-Net (proposed) is created by combining CNN and Bi-LSTM to improve the precision level of the students' imbalanced dataset by reducing the false positive and false negative ratio. RQ1 and RQ2 select the best input data and algorithm for this proposed model.

The remaining paper encloses the following sections: section 2 explains the related work, and section 3 illustrates all the elements related to this research work, such as the dataset, working principles of deep learning, and the metrics employed. The experimental results are discussed in section 4, and finally concluded the findings and future perspectives in the last section.

2. Related work

2.1 Classical machine learning

Francis et al., [11] suggested an ensemble model by combining two techniques: classification and clustering. In the first step author identifies the best student feature combination using classification and then the K-Means clustering to categorize the low, medium, and high-performance students. The result shows a notable association between the behaviour of students and the learner's academic performance. Abu et al., [12] explore the feasibility of creating a prediction model using a small dataset with 50 records. The author suggested the hierarchical clustering technique to find the student's critical features (age, grade, course detail, instructor detail).

But Xu et al., [13] created an ML model with internet usage data, such as online usage time, volume of data, and connection frequency using higher education students. The result exhibits that online connection frequency entirely correlates with educational performance, whereas Internet traffic volume is negatively associated with academic performance. Ghorbani et al., [14] tried to solve the data imbalance issue through this work by comparing different sampling techniques such as borderline SMOTE, random over sampler, SMOTE, SMOTE-ENN, SVM-SMOTE, and SMOTE-Tomek using two other datasets.

Chui et al., [15] proposed a hybrid model reduced training vector-based support vector machine (RTV-SVM) to indicate risky and borderline students. It applies binary and multiclass classification for each course. This proposed method removes repeated training vectors to decrease the training time for large datasets. Zeineddine et al., [16] suggested an automated machine learning model by choosing the best hyper-parameters using the AutoML tool from Weka software. It enhances the precision of indicating student performance using the data obtainable before commencing the course. This specified model is an ensemble one based on the voting system. The auto-generated hybrid model predicts falling students with a correctness of 83% accuracy and kappa 0.5 after resampling the data using the SMOTE technique.

Hussain et al., [17] suggested a dual model to classify the student grades and forecast the student marks using the decision tree and genetic algorithm. Both regression and classification are executed in this model to work simultaneously. The author collected the required data from intermediate and secondary education. The genetic algorithm determines the best feature, and the grade classification accuracy is above 90%. Yakubu et al., [18] developed a model to predict student performance using logistic regression. The result revealed that student age does not predict academic success; female students are one or two times more likely to achieve higher CGPA than male students. Students with high CGPA scores and those from wealthy families and cities have more possibility to succeed academically. The precision obtained by the model is 83.5%.

Yağcı [19] designed a new model to predict the students' final grades using the past midterm exam scores. This study compares the results among classical machine learning algorithms and the proposed model gives 70-75% accuracy. Rose et al., [20] suggested a model to predict at-risk students in the early stage of the course in a cloud virtual learning environment. The dataset contains 530 records with 46 features, including student demographic details, academic progress, learning style, and other online usage information. The maximum accuracy achieved is 89%. Christou et al., [21] developed a model that is a grammatical evolution-based feature selection and construction method for radial basis function (FSC4RBF) networks. It predicts the student's future grade and study time using past data. Here, the prediction of grade value and study duration addresses the multiclass classification. The proposed method FSC4RBF achieves 78.18% in grade prediction and 79.56% for study duration.

2.2 Deep learning

Qiu et al., [22] suggested a student dropout prediction model using the CNN algorithm. The author tries the CNN model on click stream data and compares the results with traditional machine learning models. CNN outperforms the other model with an 86% f1-score, and this study proves that CNN can also work effectively in time series data. Based on week-wise sequential data, Aljohani et al. [23] created a model utilizing LSTM to identify the risky students in advance. However, it took 38 weeks to reach its peak level of accuracy. Two new approaches for predicting student learning status were proposed by Wang et al. [24]. The first one is to use Conv-GRU to retrieve significant attributes. The author tried weighted average pooling rather than maximum pooling layer, showing acceptable accuracy (f1-score: 81%). The next one, called xNN (explainable neural network), emphasizes the relationship between students' positive and negative outcomes for strengthening the weak areas of students. This technique aids in uncovering hidden patterns in student behaviour and provides early warning to boost the lacking area.

Wu et al., [25] created a hybrid CLMS-Net model to classify the dropout students in MOOCs online courses. Furthermore, the author addressed the problem of class disparity with an AUC score of 91.5%. He et al., [26] developed a model for predicting student category. The author employed two fully connected neural networks for demographic details, and RNN was applied to student assessment data using LMS interaction data. Comparing the suggested method to the baseline models, the proposed one provides the highest accuracy (above 80%). Another work Chen et al., [27] offer is comparing deep learning and machine learning algorithms using LMS data. The author predicted the at-risk pupils' early using classification

Table 1. Contribution of proposed model (Ed-Net)

Reference No / Author / Year	Online Course (LMS)	Offline Course (University)	Univariate Time series	Multivariate Time series	Handled data Imbalance	Comparing Baseline model (ML)	Comparing Baseline model (DL)	Comparing another dataset	Hybrid	Train / Test comparison	Class wise (0/1) comparison
[22] L. Qiu (2018)	✓	×	×	✓	×	✓	✓	×	×	×	×
[23] N. R. Aljohani (2019)	✓	×	✓	×	×	✓	✓	×	×	×	×
[26] Y. He (2020)	✓	×	✓	×	×	✓	✓	×	×	×	×
[27] F. Chen (2020)	✓	×	✓	×	✓	✓	✓	×	×	×	×
[28] H. Waheed (2020)	✓	×	✓	×	×	✓	✓	×	×	×	×
[29] A. A. Mubarak (2020)	✓	×	✓	×	×	✓	✓	×	×	✓	×
[30] Q. Fu (2021)	✓	×	×	✓	×	✓	✓	×	✓	×	×
[31] A. A. Mubarak (2021)	✓	×	×	✓	×	✓	✓	✓	✓	✓	×
[34] H. Prabowo (2021)	×	✓	✓	×	✓	×	✓	×	✓	✓	×
[36] A. S. Aljaloud (2022)	✓	×	×	✓	×	×	✓	×	✓	✓	×
[37] H.-C. Chen (2022)	✓	×	×	✓	✓	✓	✓	×	✓	×	×
[38] H. Waheed (2023)	✓	×	×	✓	×	✓	✓	×	×	×	×
Ed-Net (Proposed)	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	✓

and clustering approaches and compared the outcomes using the AUC metric with the above 60% score.

Waheed et al., used a similar dataset (OULAD) in their two different work [28, 38], though the authors involved various procedures. Preferably, the first study exhibits the power of deep neural network (DNN) with 93% accuracy than the later one (84%), with a significant disparity. Rather than predicting week-wise, the weak learners are recognized in every quarter Q1, Q2, Q3, and Q4 with the classification of pass, fail, distinction, and withdrawal. Mubarak et al., [29] forecast student's week-wise assessment using video click streams for timely intervention.

All authors in [22, 25, 30, 31] applied the same dataset (KDD Cup 2015) to predict student dropout. Its thirty-nine courses included seven types of student online interaction data: portal access, video, wiki, discussion, navigation, page close, and problem. Fu et al., [30] created a hybrid model known as CSLA-Net by fusing three approaches: CNN for selecting local features, LSTM for sequential data, and Attention Mechanism for assigning weight. It improves performance by over 2.8%, and the f1-score is 86.9%. However, the dataset is identical, only [25, 31] deals with the data imbalance issue. A predictive model was developed by Zhang et al. [32] to identify the micro-level pattern in student learning behaviour. The author separates the attributes into five categories depending on the student's learning manners to prevent data sparsity. Because the author assumes every student's online learning habits vary according to their leisure time. The substantial discrepancy between recall and accuracy indicates the classification error that this model needs to correct. LSTM - Encoder achieves up to 92% accuracy.

Li et al., [33] conducted a comparative analysis using the student grade and levels to classify the student category. An automatic neural network with many hidden layers pulls informative elements with associated weights. Prabowo et al., [34] attempted the dual-input method, integrating categorical and numerical time series data. The suggested dual-input hybrid model combines MLP and LSTM networks and analogizes the accuracy with the individual model. Both [33, 34] works are regression problems to forecast the students' final grades using regular course academic performance. Shin et al., [35] offered a deep LSTM model to predict student performance by clustering the students using the K-shape method. Each bunch allows the identification of the pupil class to deliver a warning from the instructors. The accuracy reaches up to 90%.

Aljaloud [36] proposed a CNN-LSTM model to foretell pupil understanding levels by preferring the number of critical components and assessing the impact by decreasing the number of attributes. This

Received: November 16, 2023. Revised: February 4, 2024.

LMS dataset employs seven features and seven courses, and the final result shows the best accuracy (93%) in the more attribute combinations. Chen et al., [37] designed a hybrid model (Conv - LSTM) to manipulate imbalanced datasets. This LMS data encloses eight online interactive features (F1-F8): assignment, homepage, label, page, quiz, file, forum, and URL. The course duration is sixteen weeks, but this model assists in warning the dull students earlier than other models with improved accuracy (91%).

The contribution of existing and proposed method is discussed in Table 1. The results revealed from the existing studies are as follows: Most ML model shows exactness between 70 - 80%, and the DL earns 80 - 90%. The f1-score metric is widely applied, but some studies [17, 28, 29, 37] have not used fl-score or other metrics to evaluate Type I and Type II errors. Likewise, in few of the existing studies [22, 23, 26, 28, 37, 38], model efficiency is not shown in terms of train test convergence and the class wise performance with each other separately. The model in [23, 32] shows less recall but highest accuracy. Though related research carries on this topic, it is necessary to look into the above issues to create a generalized model. This study aims to gain improved accuracy (above 95%) and f1-score to develop a reliable model from all perceptions, such as over-fitting, fewer classification errors, model stability, and randomness. In addition, machine learning models widely use numerous student features to predict the performance [11, 14, 17]. But, this research aims to find the simple learners' data and model to attain the most increased performance using the multivariate time series approach not widely involved in the offline course investigation. This multivariate approach also makes it easy for an educational institution to apply the student academic data through this method without any tedious conversion.

3. Methods and materials

3.1 Dataset

The dataset is collected from undergraduate students (UG) from various higher education institutions, including multidisciplinary university and affiliated colleges. The students belong to different specializations and faculties. The total number of records processed in this study is 4408, and the number of students involved is 1102, of which 895 pass (label=1) and 207 fail (label=0). There are 50 features categorized into five types: student demographic details, family information,

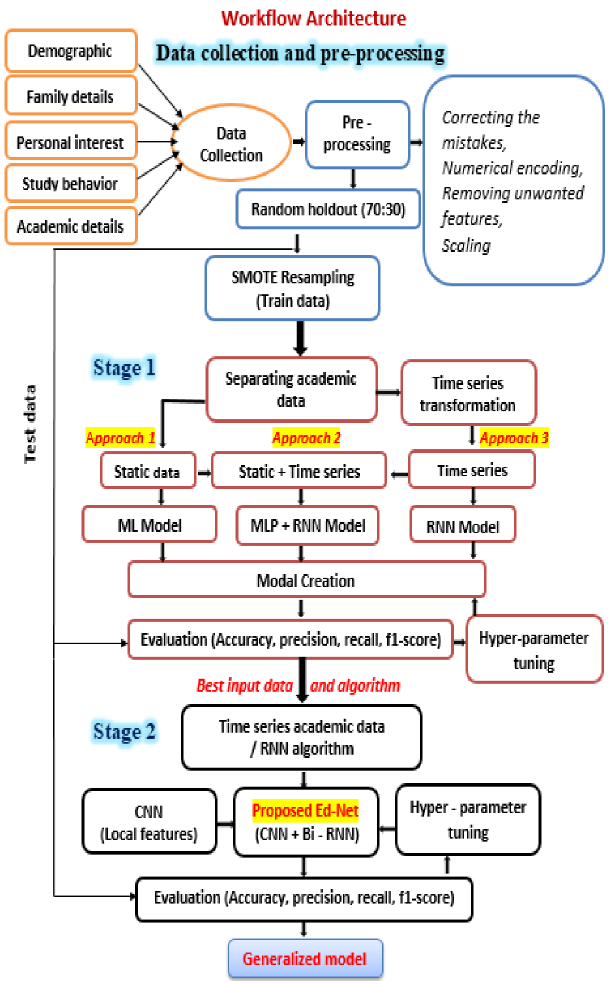


Figure. 1 Workflow architecture

study behaviour, personal interest, and academic details. Overall, 26 columns are utilized for the student's educational data, and the remaining 24 are for other details. The educational data includes secondary, higher secondary and semester-wise grades up to four semesters. Appendix A describes the collected student features. Most data is categorical because the previous semester's scores are collected as a grade letter. For Example, an O (outstanding) grade refers to the mark between 91-100; likewise, 'A+' refers to 81-90, A is for 71-80, B+ is for 61-70, B refers to 51-60, C is for 41-50, P is for 40(just pass), F is for 0-39(fail), and Ab is for Absent; Absent also considered as fail. The semester-wise information increases the number of features, but it is required for this study to identify the adequate student information to predict. This study only focuses on undergraduate students since they are adolescents rather than post-graduates.

Fig. 1 explains the overall workflow of this research work. This experiment is divided into two stages after pre-processing. Stage one is to find the best input data, approach, and model. Then, stage 2 combines the student local and temporal features to improve the precision level of the proposed model (Ed-Net).

3.2 Pre-processing

Converting the collected data into the required format is essential, and it is accomplished through data pre-processing using the following steps:

Step1: Correcting the mistakes: Students' typing mistakes are fixed manually in the field of secondary and higher secondary total marks.
Step2: Numerical Encoding: The categorical information, including demographics, family details, study behaviour, personal interest, and academic details converted into ordinal values. Then, the ordinal values of student grades are converted into marks based on the rank using random number generation.
Step 3: Removing unwanted features: Email address, number of siblings, name of college/University, and department details are not required for this study and are removed.
Step 4: Scaling: Scaling helps to converge the model quickly, and this study uses MinMaxScalar to scale the encoded values.
Step5: Train-Test split: The random hold-out method divides the pre-processed data into train and test (70:30) and uses stratification to split the student data evenly for both classes (pass/fail).
Step6: Resampling: The SMOTE resampling technique is involved in training data to increase the number of samples in the minority class to balance the data distribution of both categories.
Step7: Time-series transformation: After resampling, semester-wise scores are separated and transformed into time-series data. The remaining features are used as tabular (static) data for model creation.

3.3 Synthetic minority oversampling technique (SMOTE)

SMOTE is a well-known up-sampling technique introduced by Chawla et al., (2002) to fix the class imbalance issue. It generates synthetic data points to augment the number of samples rather than replicating the same data. SMOTE generates new data points between the existing data using interpolation. The steps followed in SMOTE are as follows:

- Select the random sample from the class that needs to increase the data points.
- Identify the k nearest neighbours (k=n) of the selected data using the distance measure algorithm (euclidian).
- Choose any of those neighbours and calculate the vector between the current and selected neighbour data.

The nearest neighbour value n=5 is used in this experiment, and the number of synthetic samples created for the minority class is 483. SMOTE performs better than other combinations, such as SMOTE-Tomek link, SMOTE-ENN, and SVM-SMOTE.

3.4 Recurrent neural network (RNN)

The RNN is an improved artificial neural network (ANN) algorithm that uses memory cells to remember sequential information. This memory cell saves prior information for subsequent processing, and the selection is made by considering the past and present states. RNN shares the same weight parameters for each layer, but typical neural networks share distinct weights. The three RNN building blocks are input, hidden neuron, and activation function.

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

where h_t is a hidden neuron, x_t is the input data, b_h refers the bias value, U is the weight of the hidden layer and W is the transition weight. Thus the Eq. (1) determines the hidden state at time t. To create a new hidden state, the input and prior state information are merged and passed through the *tanh* activation function. RNN is affected by the vanishing gradient issue when dealing with long sequence data. However, the other variants of RNN, such as LSTM [9] and GRU rectify this issue.

3.5 Long short-term memory (LSTM)

LSTM is appropriate for processing long-term dependency data. It retains the prior context better than RNN with three types of gates. The input gate restores the memory cell; the forget gate determines whether the preceding information is maintained; the output gate is liable for identifying the subsequent hidden state. The RNN loop configurations make it easier to decide the ideal weight parameter. Each component in LSTM is calculated in the following equations:

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

Received: November 16, 2023. Revised: February 4, 2024.

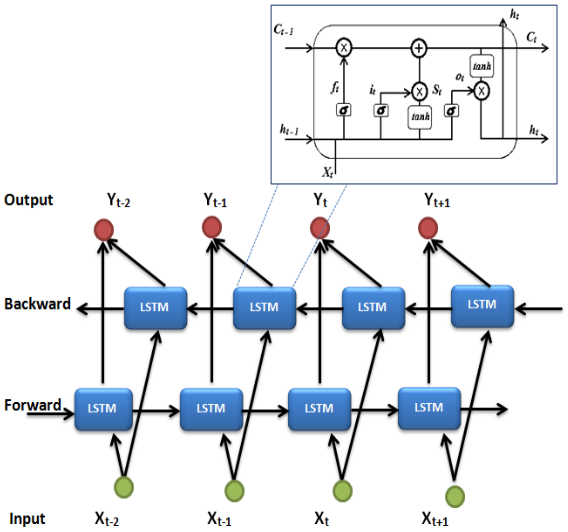


Figure. 2 Bidirectional LSTM architecture

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

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

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

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

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

i_t - input gate	b - bias value
f_t- forget gate	C_t - candidate state
o_t - output gate	S_t - state of the memory cell
X_t - input vector	W, U, V - weight matrices
h_t - memory cell	σ , $tanh$ - activation functions

3.6 Bidirectional LSTM (Bi-LSTM)

The LSTM model considers one direction of information on a sequence, which reduces its significance. Furthermore, the multi-directional LSTM offers the best performance of the model by involving future data for prediction. Hence, the forward and backward directions in the row came together to form bidirectional long short-term memory, known as Bi-LSTM [10]. Fig. 2 explains the Bi-LSTM Architecture.

The basic principle of the Bi-LSTM technique is that it studies a particular series in both directions. In this scenario, an LSTM layer is used for forward processing, whereas the final layer is used for backward processing. The network can record the failed student from its history and its future. To comprehend this concept, imagine an input sequence p with q elements. The frontward LSTM's direction is $\{p_1, p_2, \ldots, p_q\}$, while the backward LSTM's direction is $\{p_q, p_{q-1}, \ldots, p_1\}$. Once trained, both LSTM's are calculated independently and then combined as follows:

Table 2. Raw data of a student assessment details

Student Id	Semester	Paper 1	Paper 2	Paper 3	Paper 4	Paper 5	Paper 6	Label
1001	1	74	59	49	66	58	56	1
1001	2	69	65	54	54	67	61	1
1001	3	65	57	69	68	45	58	1
1001	4	70	40	71	70	77	62	1
1002	1	66	37	60	62	68	66	0
1002	2	55	4	49	58	70	45	0

$$h_t^f = tanh\left(U^f.x_t^f + W^f.h_{t-1}^f + b_h^f\right) \quad (8)$$

$$h_t^b = tanh\left(U^b.x_t^b + W^b.h_{t-1}^b + b_h^b\right) \quad (9)$$

$$y_t = W^f.h_t^f + W^b.h_t^b + b_y) \quad (10)$$

where h_t^f and h_t^b represent the outputs of the forward and backward LSTMs, respectively. Then the Bi-LSTM combines the forward and backward directions using the Eq. (10). Finally, a fully connected dense layer receives the output from the Bi-LSTM to produce the outcome of student performance.

3.7 Multivariate sequential input labelling and sub sequencing

The pre-processing result of a student academic semester wise data shown in Table 2.

Multivariate input refers to the sequence of student time series data with multiple features. It includes the student's last four-semester score in six subjects (paper) utilized to create the proposed model. The number of records used in this experiment is 4408 (*r*), a long sequence. This sequence is divided into sub-sequences using the time step 4 *(T)* referring to students' past four-semester scores; again, each semester is a collection of six papers *(p1, p2, p3, p4, p5, p6)* representing the multivariate features *(f)*. The total number of sub sequences (*SS*) is determined through the Eq. (11):

$$SS = \frac{r}{T} \quad (11)$$

The number of input features *(IF)* involved in each subsequence is identified using the below Eq. (12):

Table 3. Pseudo code for converting sequences into samples

Pseudocode for dividing sequence into samples

1. **START the procedure**
2. **The set of long sequence, time step: = 4**
3. **X:= list(), y = list() where X for multivariate features, y for label**
4. **for (start_index:=0, start_index <length of sequences)**
5. **end_index = 4* start_index + time step**
6. **if (end_index > length of sequences)**
7. **end loop**
8. **X_sequence, y_sequence = sequences[4* start_index: end_index, :-1], sequences[end_index -1, -1] where X_seq for input, y_seq for output**
9. **X: = append X_sequence**
10. **y:= append y_sequence**
11. **return X, y as numpy array**
12. **END procedure**

$$IF = T*f \tag{12}$$

Each subsequence owns 24 multivariate features and a separate label column for each semester to classify the student category. Students' semester-wise performance is manually verified while assigning the label value. The occurrence of one or more fail or absent in any semester assumes the student class is fail (0). The following Eqs. (13) and (14) explain the structure of multivariate input elements:

$$\begin{bmatrix} \{sm_1, sm_2, sm_3, sm_4\} \in SS_1 \\ \{sm_1, sm_2, sm_3, sm_4\} \in SS_2 \\ \vdots \\ \{sm_1, sm_2, sm_3, sm_4\} \in SS_n \end{bmatrix} = S \tag{13}$$

$$\begin{bmatrix} sm_1p_1, sm_1p_2, sm_1p_3 \dots sm_1p_6 \\ sm_2p_1, sm_2p_2, sm_2p_3 \dots sm_2p_6 \\ sm_3p_1, sm_3p_2, sm_3p_3 \dots sm_3p_6 \\ sm_4p_1, sm_4p_2, sm_4p_3 \dots sm_4p_6 \end{bmatrix} = SS \tag{14}$$

Where *sm* refers to the semester, and *p* refers to the papers accomplished by the student every semester. Thus, the long sequence (S) includes subsequence $\{ SS_1, SS_2 \dots SS_n \} \in S$ and subsequence (SS) includes $\{ sm_1, sm_2 \dots sm_4 \} \in SS$ semester-wise student scores. The pseudo-code in Table 3 transforms the multivariate data into two-dimensional samples after assigning the label.

The two-dimensional input matrix (M) of each student for the model is described in Eq. (15). The *n* refers to the number of papers in each semester *(sm)*.

$$M = \begin{bmatrix} x_t^1 & x_t^2 & \dots & x_t^n \\ x_{t+1}^1 & x_{t+1}^2 & \dots & x_{t+1}^n \\ \dots & \dots & \dots & \dots \\ x_{t+sm-1}^1 & x_{t+sm-1}^2 & \dots & x_{t+sm-1}^n \end{bmatrix} \tag{15}$$

3.8 Ed-net (proposed)

CNN is a type of deep learning model introduced by Lecun et al., in 1998. CNN is fit for working in a grid-like topology applicable in image processing [6]. Nowadays, it is also used in time series data to extract the local features effectively. The gain of applying CNN is reducing the number of parameters through sharing the weight. To process input data, CNN uses a sequence of layers involving a particular operation: The convolution layer is the fundamental building block that applies several filters and kernels to the input information. Then, the non-linear activation function is applied to the feature maps created by the convolutional layer, and the convolution output is downsized using the pooling layer. This process is repeated a few times to establish the hierarchy of features. The outcome is flattened and passed through one or more fully connected layers or any other network.

The proposed (Ed-Net) architecture is created with the combination of CNN and Bi-LSTM to classify the student category (pass/fail). In the first step, all the inputs are fed into two CNN 1D convolutional layer to capture the local features. CNN cannot encode the temporal intrinsic dependencies in the sequence; it can learn only the regional characteristics of the series. So, in the second step, the CNN output is transmitted into a Bi-LSTM layer to solve this problem. Finally, a fully connected (dense) layer is added to predict the student class (0/1).

Fig. 3 shows the proposed architecture and configuration details. The s1, s2…s6 refers the subjects in the above Fig. 3. In step one, after connecting the CNN ID layer, the max pooling layer is utilized to reduce the extracted feature's complexity and flatten the values. Also, the dropout (0.1) method is used as a part of the regularization technique in the CNN and Bi-LSTM layer to inactivate a portion of neurons connected in the network.

3.9 Experimental setup

In this research work, Python programming language is used in all experiments with the following libraries: pandas, numpy, scikit, imblearn matplotlib, and keras. Keras functional and sequential APIs create multi-input, sequential models, respectively.

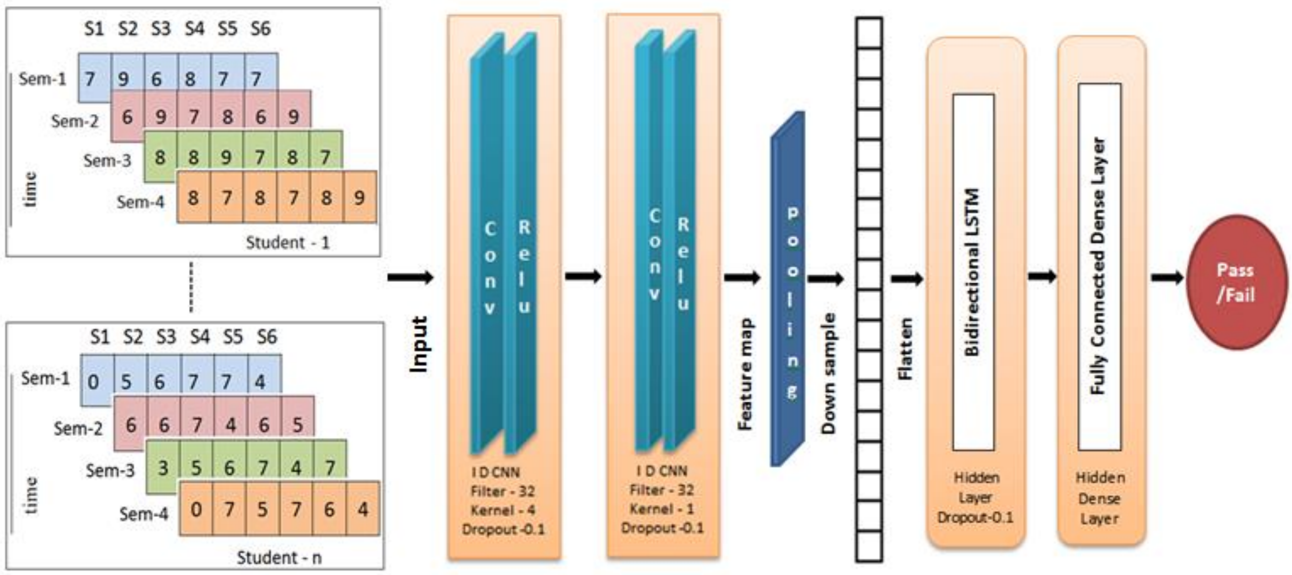


Figure. 3 Proposed Ed-Net architecture

Table 4. Approach 2 model configuration details

Layer (type)	Output Shape	Param #	Connected to
Timeseries_Input (InputLayer)	[(None, 4, 6)]	0	[]
Hidden_Layer_1 (GRU)	(None, 4, 128)	52224	['Timeseries_Input[0][0]']
Dropout_Layer_1 (Dropout)	(None, 4, 128)	0	['Hidden_Layer_1[0][0]']
Static_Input (InputLayer)	[(None, 22)]	0	[]
Hidden_Layer_2 (GRU)	(None, 64)	37248	['Dropout_Layer_1[0][0]']
Dense_Layer_1 (Dense)	(None, 64)	1472	['Static_Input[0][0]']
Dropout_Layer_2 (Dropout)	(None, 64)	0	['Hidden_Layer_2[0][0]']
Dense_Layer_2 (Dense)	(None, 32)	2080	['Dense_Layer_1[0][0]']
Concatenated_Layer (Concatenate)	(None, 96)	0	['Dropout_Layer_2[0][0]', 'Dense_Layer_2[0][0]']
Dense_Layer_3 (Dense)	(None, 64)	6208	['Concatenated_Layer[0][0]']
Output_Layer (Dense)	(None, 1)	65	['Dense_Layer_3[0][0]']

Total params: 99297 (387.88 KB)
Trainable params: 99297 (387.88 KB)
Non-trainable params: 0 (0.00 Byte)

Stage 1 includes three input approaches: static, hybrid, and time series to find the best input data and model using different set of student information.

Approach 1: This approach uses machine learning with static data for student classification. The model and the hyper-parameter setting used in this experiment are as follows: Logistic regression, SVM classifier ('rbf'), decision tree, random forest (n_estimators=150), KNN (n_neighbours=9), and Naive Bayes. Majorly, default parameters perform better than the specific setting.

Approach 2: This multi-input single output model uses Keras functional API to combine static and time series input data. The configuration details are explained in Table 4.

The L2 regularization is used in both inputs of

Table 5. Approach3 model configuration details

Model: "sequential_11"

Layer (type)	Output Shape	Param #
gru_6 (GRU)	(None, 128)	52224
dropout_11 (Dropout)	(None, 128)	0
dense_22 (Dense)	(None, 64)	8256
dense_23 (Dense)	(None, 1)	65

Total params: 60,545
Trainable params: 60,545
Non-trainable params: 0

this model to avoid over-fitting with the value of 0.01 and the dropout ratio of 0.1. Then, these two static and time series inputs are concatenated and pass through the dense layer with the *'relu'* activation function. Finally, an output layer is added with a sigmoid activation function for prediction. The batch size is 32 and 40 epochs commonly used in stage one models. The RMSprop optimization algorithm uses the following parameters: learning_rate=0.001, rho=0.9, momentum=0.9, epsilon=1e-07, and centered=False. The loss function used in this model is binary cross-entropy.

Approach 3: The same regularization technique and values are used in this sequential model and used Adam optimization with a learning rate 0.01. The model configuration details are displayed in Table 5.

Stage 2 includes the proposed system to find the impact of the blending spatial and temporal information. The model settings are visible in Table 6.

In the proposed Ed-Net, two time distributed convolutional 1D layers are added, with the filter

Table 6. Ed-Net configuration details

```
Layer (type)                 Output Shape              Param #
================================================================
time_distributed (TimeDistr  (None, None, 1, 32)       800
ibuted)

time_distributed_1 (TimeDis  (None, None, 1, 32)       0
tributed)

time_distributed_2 (TimeDis  (None, None, 1, 32)       1056
tributed)

time_distributed_3 (TimeDis  (None, None, 1, 32)       0
tributed)

time_distributed_4 (TimeDis  (None, None, 1, 32)       0
tributed)

time_distributed_5 (TimeDis  (None, None, 32)          0
tributed)

bidirectional (Bidirectiona  (None, 256)               164864
l)

dropout_2 (Dropout)          (None, 256)               0

dense (Dense)                (None, 64)                16448

dense_1 (Dense)              (None, 1)                 65

================================================================
Total params: 183,233
Trainable params: 183,233
Non-trainable params: 0
```

size 32 and kernel size 4, 1 for each layer, respectively. Subsequently, the Max-pooling layer is added with the pool size 1. The hyper-parameter settings are as follows: Adam optimizer with the learning rate = 0.001, epochs 60, and batch size 32.

3.10 Evaluation metrics

The metrics employed in this experiment are as follows: *Accuracy* tells the model correctness by evaluating the ratio between correctly identified data and the total number of points. However, it provides misleading outcomes in imbalanced datasets with a relatively small sample of the minority class. Therefore, the confusion matrix used to calculate the precision, recall, and f1-score. These metrics allow us to locate the problem the model failed to understand. *Precision* is a measure to determine the ratio of correctly identified classes among the total number of positive classes the model predicted. *Recall* is a measure to determine the ratio of correctly predicted classes among the total number of positive classes. The harmonic mean of precision and recall is the *f1-score*. The following Eq. (16) to (19) describes the metrics.

$$Accuracy = (TP + TN)/(TP + FN + TN + FP) \quad (16)$$

$$Precision(P) = TP/(TP + FN) \quad (17)$$

$$Recall(R) = TP/(TP + FN) \quad (18)$$

Table 7. Performance of ML models

Technique	Minority (0) / Majority (1) Class	Precision	Recall	f1-score	Train Accuracy	Test Accuracy
LR	0	0.35	0.66	0.46	0.74	0.70
	1	0.90	0.71	0.80		
DT	0	0.42	0.55	0.48	1.00	0.78
	1	0.89	0.83	0.86		
RF	**0**	**0.61**	**0.50**	**0.55**	**1.00**	**0.85**
	1	**0.89**	**0.93**	**0.91**		
KNN	0	0.28	0.76	0.41	0.79	0.59
	1	0.91	0.55	0.68		
Naive Bayes	0	0.36	0.66	0.47	0.71	0.72
	1	0.90	0.73	0.81		
SVM	0	0.40	0.61	0.48	0.86	0.75
	1	0.90	0.78	0.84		

$$f1 - score = 2 * (P * R)/(P + R) \quad (19)$$

In binary classification, there will be four situations, which are as follows:

True positive (TP): The number of instances predicted by the model is positive and correct. i.e., the student class predicted by the model is pass, and the actual label of the student is also pass.

False positive (FP): The number of instances predicted by the model is positive but incorrect. i.e., the student class indicated by the model is pass, but the actual label of the student class is fail.

False negative (FN): The number of instances predicted by the model is negative, but it is wrong. i.e., the student class denoted by the model is fail, but the actual label of the student class is pass.

True negative (TN): The number of instances predicted by the model is negative and correct. i.e., the student class indicated by the model is fail, and the actual label of the student class is also fail.

4. Results and discussion

Approach 1 follows the traditional way of student performance prediction using static data with machine learning algorithms. Static data includes all the student details except the current academic information. The correlation table (heat map) identifies the student's essential features. The result shows the school type and mode of study, transport and travel time, father and mother's education, listening classes, taking notes, and all study behavioural values are highly correlated. However, the gender and address fields have less correlation, and these fields are removed.

Overall, the range of accuracy of all the models is 70 – 80%, except for the random forest. It shows 85% accuracy, but a significant difference exists between train and test data. Before resampling, all the models failed to detect the minority class, but it improved moderately after applying SMOTE resampling. Table 7 shows the approach 1 results.

More specifically, logistic regression and Naive Bayes show less over-fitting even though the accuracy level is less than 75%. Random forest, decision tree, and support vector classifier (kernel='rbf') reveal large over-fitting with high accuracy. However, the KNN shows the worst performance in both cases. All the models did not perform well for the minority class, even after resampling. All the models did not perform well for the minority class, even after resampling. This issue is rectified by approach 2.

Approach 2 combines two types of input data (Static + Time Series) to study the hybrid data approach performance. Multi-layer perceptron (MLP) is used to handle the static data, and RNN is used to govern the sequential information. Students' past semester scores are transformed into time series and other student details are used as static data. The MLP-LSTM, MLP-GRU, and MLP–Bi GRU hybrid models were developed and evaluated. The MLP-GRU works better than other combinations with good recall and precision value. Table 8 and Fig. 4 describe the approach 2 model performance

In this technique, the overall performance is enhanced by 5% compared to the previous approach, predicting the students up to 90% accuracy. The minority class prediction is also increased up to 60% in precision and recall, but there is a large amount of over-fitting. However, approach 3 reduces the over-fitting by using the time series data.

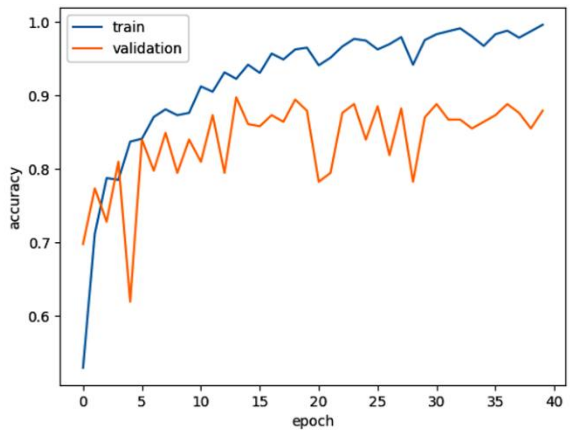
In Approach 3, student academic information is used to check the time series significance using RNN algorithms and it increases the accuracy level up to 95%. It also decreases over-fitting, and 89% precisely identifies the minority class which is more than 20% of previous approach. Compared to prior methods, each model in this approach provides stable results in multiple runs, and converges gradually along with the test data. However, there is a fluctuation in validation data but it is rectified by the proposed model Ed-Net in the next section. Table 9 and Fig. 5 represent the performance of approach 3 models given below:

ED-Net (proposed): The previous section concluded that the Time series technique provides better results than the other two. Also, the GRU algorithm proved its proficiency in both time series and hybrid methods. Therefore, students' current academic sequential information and RNN is utilized in the proposed system. However, there is a necessity to find local features to improve the model's ability.

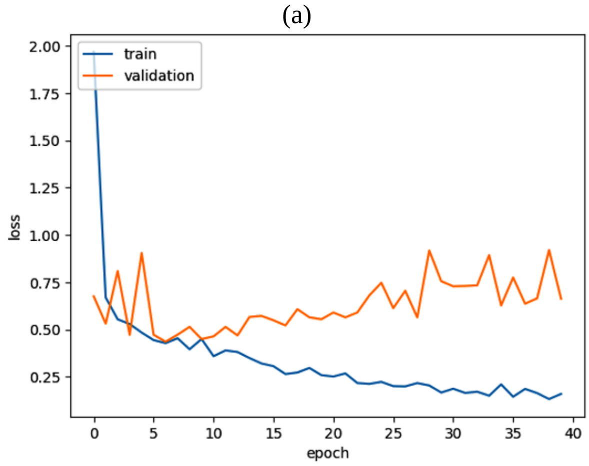
CNN is a dedicated algorithm for selecting the local attributes. So, the proposed system follows a

Table 8. Performance of hybrid approach

Technique	Minority (0) / Majority (1) Class	Precision	Recall	f1-score	Train Accuracy	Test Accuracy
MLP-LSTM	0	0.69	0.65	0.67	0.97	0.88
	1	0.92	0.93	0.93		
MLP-GRU	0	**0.75**	**0.71**	**0.73**	**0.98**	**0.90**
	1	**0.93**	**0.94**	**0.94**		
MLP-Bi GRU	0	0.67	0.68	0.67	0.98	0.88
	1	0.93	0.92	0.92		



(a)



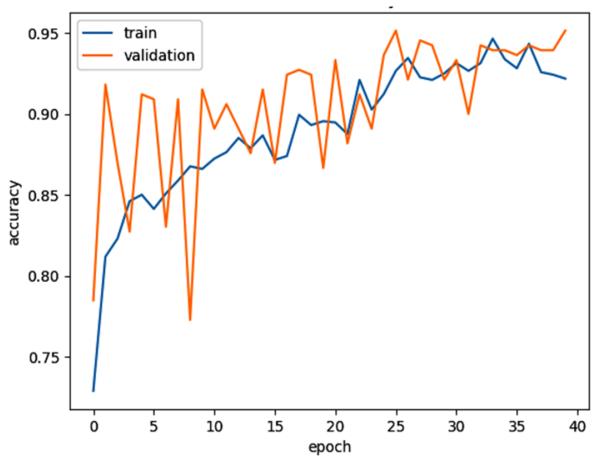
(b)

Figure. 4 Hybrid data model using MLP – GRU: (a) accuracy and (b) loss

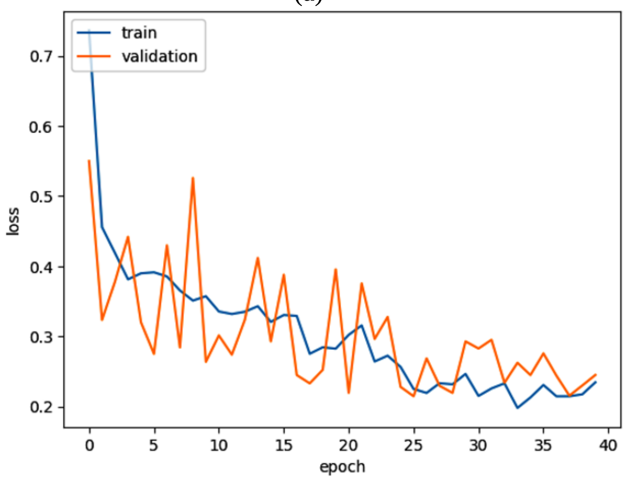
Received: November 16, 2023. Revised: February 4, 2024.

Table 9. Performance of time series models

Technique	Minority (0) / Majority (1) Class	Precision	Recall	f1-score	Train Accuracy	Test Accuracy
LSTM	0	0.70	0.87	0.78	0.89	0.91
	1	0.97	0.91	0.94		
GRU	0	**0.86**	**0.89**	**0.87**	**0.94**	**0.95**
	1	**0.97**	**0.97**	**0.97**		
Bi-GRU	0	0.77	0.89	0.83	0.94	0.93
	1	0.97	0.94	0.96		



(a)



(b)

Figure. 5 GRU model performances: (a) accuracy and (b) loss

hybrid strategy to combine CNN and RNN. Here, CNN-LSTM, CNN-GRU, CNN-Bi GRU, and CNN-Bi LSTM combinations are developed for student classification and compared the results. The CNN-LSTM and CNN-Bi GRU show over-fitting (maximum 2%) after 30 to 40 epochs. Still, the

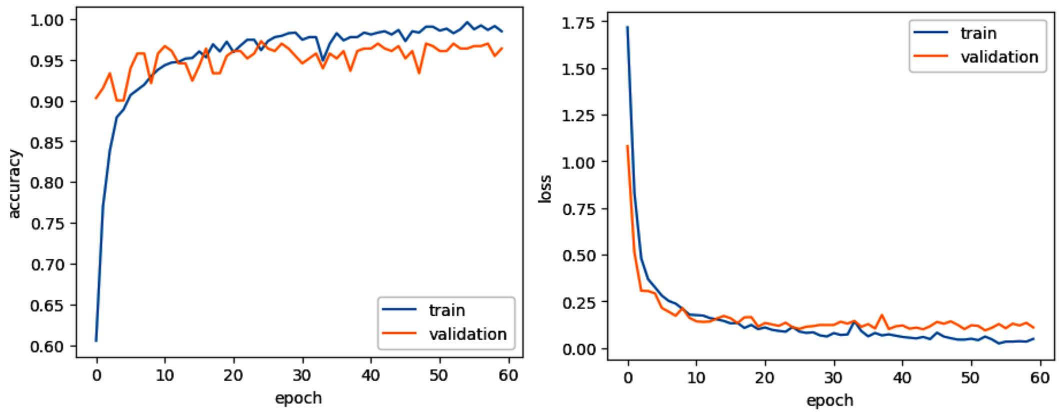
Table 10. Performance of proposed approach models

Technique	Runs	Minority (0) / Majority (1) Class	Precision	Recall	f1-score	Train Accuracy	Test Accuracy	Average Accuracy (%)
CNN + LSTM	1	0	0.93	0.87	0.90	0.984	0.963	94.9
		1	0.97	0.99	0.98			
	2	0	0.75	0.95	0.84	0.978	0.933	
		1	0.99	0.93	0.96			
	3	0	0.85	0.90	0.88	0.984	0.951	
		1	0.98	0.96	0.97			
CNN + GRU	1	0	0.91	0.94	0.92	0.978	0.969	97
		1	0.98	0.98	0.98			
	2	0	1.00	0.90	0.95	0.982	0.981	
		1	0.98	1.00	0.99			
	3	0	0.89	0.90	0.90	0.984	0.96	
		1	0.98	0.97	0.98			
CNN + Bi-GRU	1	0	0.89	0.77	0.82	0.988	0.939	95
		1	0.95	0.98	0.96			
	2	0	0.85	0.87	0.86	0.987	0.948	
		1	0.97	0.97	0.97			
	3	0	1.00	0.80	0.89	0.988	0.963	
		1	0.96	1.00	0.98			
Ed-Net	1	0	0.89	0.94	0.91	0.979	0.966	**97.4**
		1	0.98	0.97	0.98			
	2	**0**	**0.97**	**0.94**	**0.95**	**0.986**	**0.981**	
		1	**0.99**	**0.99**	**0.99**			
	3	0	0.92	0.95	0.94	0.981	0.975	
		1	0.99	0.98	0.99			

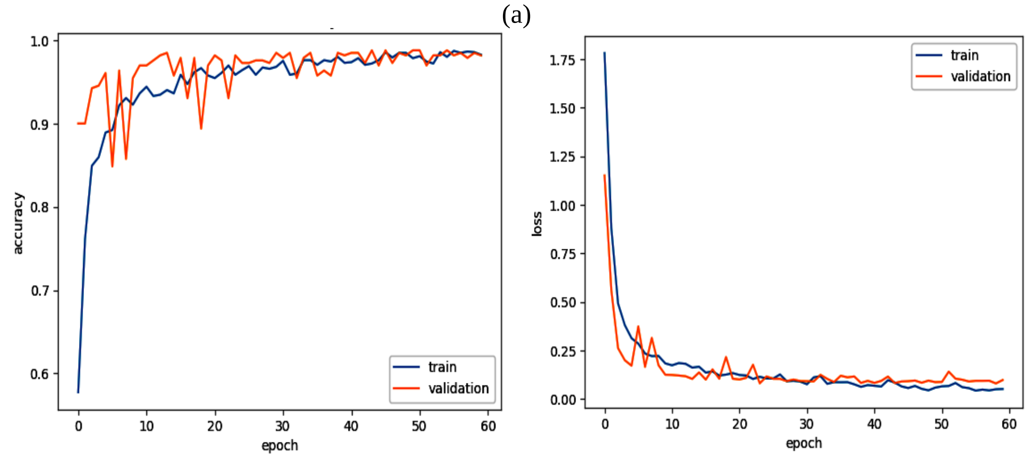
CNN-GRU and CNN-Bi LSTM show gradual convergence until 60 epochs and provide higher accuracy than the other combinations. The accuracy increased to 98% for a single run, and the average of 3 runs reached 97.4 % by CNN - Bi-LSTM with 1 % over-fitting. Table 10 shows the result in various metrics and Fig. 6 shows the graphical representation of train and test performance.

The proposed system significantly reduces the test data fluctuation and classification errors. The further increment of the epoch leads to over-fitting, so it quit at 60. The overview of the hyper-parameter setting in the deep learning model is as follows: Stage one uses 40 epochs for convergence, but the model in stage two needs 60 epochs to attain the global minima. The batch size of 64 gives significant fluctuation in convergence, so the batch size 32 is applied commonly in stages 1 and 2. The Adam optimizer is widely used and performs better than the RMSProp except in approach 2. In CNN,

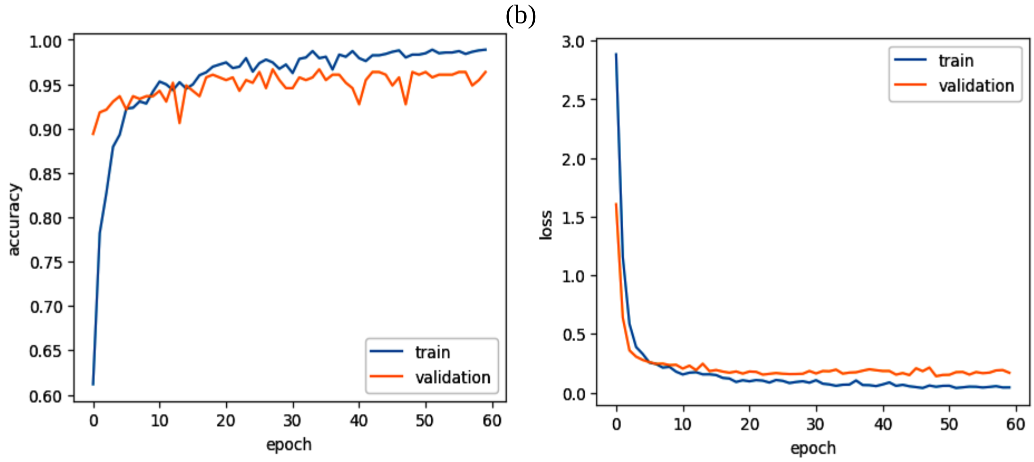
Received: November 16, 2023. Revised: February 4, 2024.



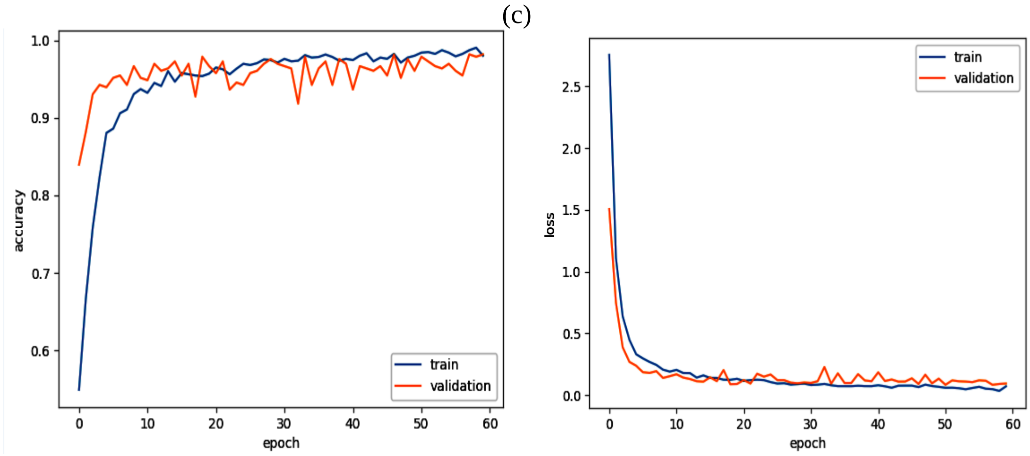
(a)



(b)



(c)



(d)

Figure. 6 Performance of proposed approach: (a) CNN-LSTM, (b) CNN-GRU, (c) CNN-Bi GRU, and (d) Ed-Net

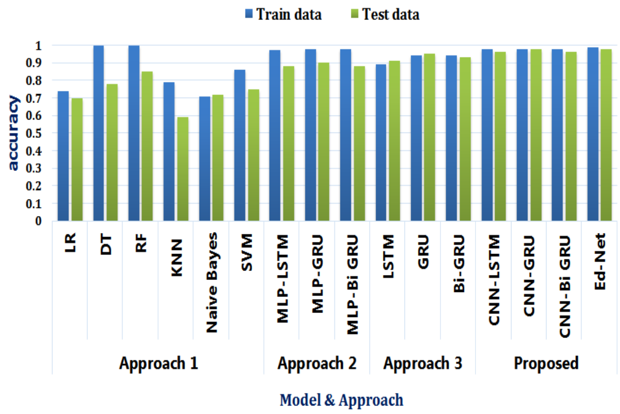


Figure. 7 Performance comparison of all approach using accuracy

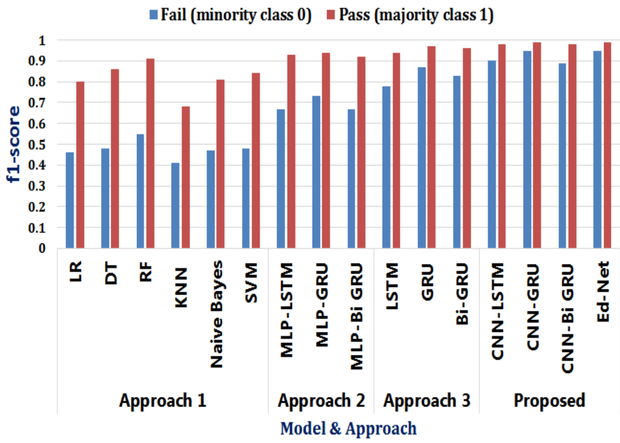


Figure. 8 Performance comparison of all approach using f1-score

filter size 32 works better than filter 16 and improves accuracy by 1 %.

Figs. 7 and 8 show the comparative analysis of all the models created in stages one and two. Due to the imbalanced dataset, the f1-score is preferred to measure the model's correctness. SMOTE well-trained the minority class, and the Ed-Net misclassified only six students, of which four failed and two passed. Fig. 9 describes the classification of both classes. Next to the Ed-Net, CNN-GRU performs well in all scenarios. The proposed approach exceeds the baseline model LSTM, GRU, and Bi-LSTM used in approach 3. Also, ML and Hybrid data methods used in approaches 1 and 2, delivers notable disparity in results. Thus, the proposed system reduces the over-fitting and improves the smooth convergence along with model stability and randomness.

Fig. 10 represent the performance comparison of the proposed method with existing work that exclusively uses CNN with LSTM and Bi-LSTM combinations. The result shows the f1 - score of Ed-Net, which is 95% higher than others.

	precision	recall	f1-score	support
0	0.97	0.94	0.95	62
1	0.99	0.99	0.99	268
accuracy			0.98	330
macro avg	0.98	0.96	0.97	330
weighted avg	0.98	0.98	0.98	330

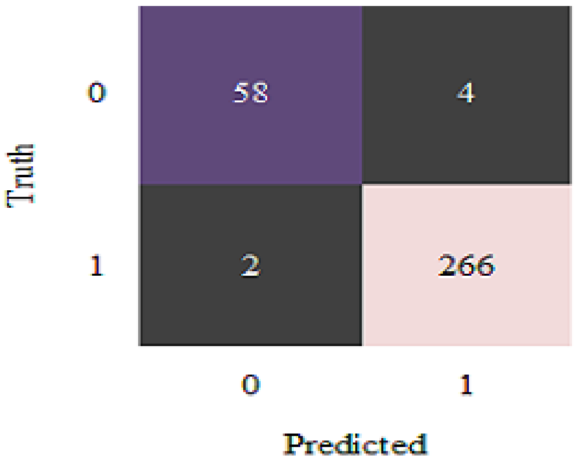


Figure. 9 Confusion matrix of Ed-Net

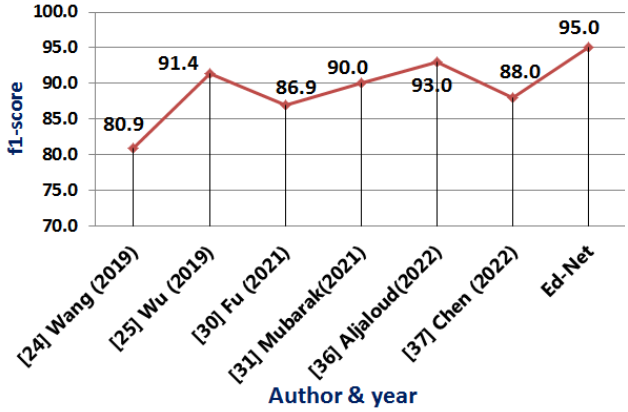


Figure. 10 Existing vs. proposed (Ed-Net) performance

Furthermore, this proposed model is validated with the popular open university learning analytics (OULA) dataset [40]. It contains the students' demographic, assessment, and LMS interaction details of 32,593 students. The course duration is 9 months, with various courses conducted during 2014–2015. However, not all the students registered for all the courses. To evaluate the proposed system, students' assessment scores are utilized as test data. We selected a total of 258 students (175 pass, 83 fail) who applied for more than one course and attended a minimum of 20 test series to match the number of proposed input features (24). The test score is 0 to 100, the pass mark is 40, and the student will fail the course if any test score is less than 40. Each student's test series is converted into a 2D matrix suitable for Ed-Net and predicts the students' class (0/1)

Table 12. Performance of Ed-Net using OULA dataset

Technique	Minority (0) / Majority (1) Class	Precision	Recall	f1-score	Train Accuracy	Test Accuracy
Validation without Training	0	0.83	0.47	0.60	0.97	0.80
	1	0.79	0.95	0.87		
Validation with Training	0	0.87	0.71	0.78	0.91	0.86
	1	0.85	0.94	0.89		

based on the results of all registered courses. The missing values are filled with zero, and we imputed a maximum of four data points using the mean value.

In Table 12, the first setup, validation without training shows the capability of Ed-Net, which predicts the other dataset up to 80% correctly. In the second setup, validation with training, the accuracy increases to 86% and the over-fitting decreases from 17% to 7%. It concludes that further training with more samples will give a better result than state-of-the-art methods that use the similar OULA dataset [26, 38] where the accuracy starts at 80%. And the simulation test reached adequate accuracy of 86% with better recall and precision values. Due to the variation in course length and number of test series, the samples are limited to avoid data imputation, which is used to increase the sequence of data points.

Finally, the overall research outcome proves the following facts: Apart from academic information, the other student details are not sufficient to classify the student category correctly. Because the students differ in knowledge level, aim, interests, time management, and family situation. A student may utilize the long travel time for his studies, but others may not. Likewise, hostel students may spend all their leisure time with their roommates and doing other activities. Again, a student from an educated family may not be interested in studies, but a poor one does. The study time also varies based on the student's intelligence level. Though the students perform well in school, their behaviour may change while entering higher education. According to Yakubu (2021), age is not a predictor of academic success [18]. Bilal (2022) also proved that student academic information is the key feature of performance prediction compared to other details [39]. The researchers widely utilize pre-academic data and previous grade points as they provide more accurate predictions [16, 17, 19, 21]. Thus, this study concludes that current academic details only tell the student's progress exactly. Besides, apart from current academic data, other information such as personal details, study behaviour, and demographic data complicates the prediction system, and it takes time to collect from students.

Moreover, the multivariate time series approach gives more training to the model than the univariate data (semester wise GPA) by tracing students' semester-wise information in parallel [34]. Usually, educational time series data is short in sequence [35], so the GRU performs better than LSTM [26], which is suitable for long sequence data. However, in this study, LSTM yields good results while combining CNN and the bidirectional method. Because the individual LSTM and the bidirectional model performance are unsatisfactory in stage one models, it only showed excellence while merging CNN. Similarly, CNN proved its fitness in [30, 36, 37].

5. Conclusion

This study emphasizes the impact of the multivariate time series approach in student learning outcomes conducted in two stages. Stage one recognizes the most appropriate student data to identify the weak students in advance. Stage two receives the best input from stage one and enhances accuracy by implementing the proposed system. Stage one comprises three approaches: tabular data with classical machine learning (ML), tabular data and academic time series information using MLP+RNN, and finally, time-series data using RNN. Stage two executes the proposed blended model combination of CNN and the bidirectional LSTM. This blended approach enhanced the accuracy to 98%, the average accuracy on multiple runs reached 97.4 %. The dataset is balanced through the SMOTE up sampling method and boosted the f1-score in both classes (0=95%, 1=99%). Furthermore, the proposed system achieved an accuracy of 86% when evaluating the benchmark dataset (OULAD) using student assessment data. This end-to-end model proves the quality and simplicity in input using student semester-wise multivariate information. The advantage is the availability of student academic detail in all educational institutions makes it feasible to predict the at-risk students in advance to form a warning system. Therefore, this research follows two stages and multiple approaches using numerous student features to find the best data and model to attain the most satisfactory result. The future perspective is to apply a massive dataset collected from educational bodies and to use optimization techniques to choose the best hyper-parameter combination. Also, future work will be able to adjust the variable length of student academic data instead of the fixed one.

Conflicts of interest

The authors declare that there is no conflict of interest concerning the publication of this paper.

Author contributions

The author's contributions to this paper are as follows: Conceptualization, methodology, validation, formal analysis, investigation, resources, data curation, writing review, original draft preparation, and visualization have been done by Vanitha. S. The supervision and editing have been done by Jayashree. R.

References

[1] C. Su and Y. Guo, "Factors impacting university students' online learning experiences during the COVID -19 epidemic", *Journal of Computer Assisted Learning*, Vol. 37, No. 6, Nov. 2021, doi: https://doi.org/10.1111/jcal.12555.

[2] V. J. G. Morales, A. G. Moreno, and R. M. Rojas, "The transformation of higher education after the COVID disruption: Emerging challenges in an online learning scenario", *Frontiers in Psychology*, Vol. 12, pp. 1–6, Feb. 2021, doi: https://doi.org/10.3389/fpsyg.2021.616059.

[3] H. M. Naveen, "UGC Regulations on Credit Framework for Online Courses through SWAYAM, 2021", *International Journal of Scientific Research in Science and Technology*, pp. 232–238, Jul. 2022, doi: https://doi.org/10.32628/ijsrst229438.

[4] A. Singh and K. B. Kakkar, "Program inclusive, credit-based SWAYAM MOOCs in higher educational institutions of India", *International Journal of Educational Development*, Vol. 97, p. 102727, Mar. 2023, doi: https://doi.org/10.1016/j.ijedudev.2023.102727.

[5] S. Vanitha and R. Jayashree, "Towards finding the impact of deep learning in educational time series datasets – A systematic literature review", *International Journal of Advanced Computer Science and Applications*, Vol. 14, No. 3, pp. 804–815, Mar. 2023. doi:10.14569/ijacsa.2023.0140392

[6] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection", *Frontiers in Plant Science*, Vol. 7, Sep. 2016, doi: https://doi.org/10.3389/fpls.2016.01419.

[7] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review", *IEEE Access*, Vol. 7, pp. 19143–19165, 2019, doi: https://doi.org/10.1109/access.2019.2896880.

[8] M. A. Maleh and S. Desouki, "Arabic text summarization using deep learning approach", *Journal of Big Data*, Vol. 7, No. 1, Dec. 2020, doi: https://doi.org/10.1186/s40537-020-00386-7.

[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 15 Nov. 1997, doi: https://doi.org/10.1162/neco.1997.9.8.1735.

[10] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks*, Vol. 18, No. 5–6, pp. 602–610, Jul. 2005, doi: https://doi.org/10.1016/j.neunet.2005.06.042.

[11] B. K. Francis and S. S. Babu, "Predicting Academic Performance of Students Using a Hybrid Data Mining Approach", *Journal of Medical Systems*, Vol. 43, No. 6, Apr. 2019, doi: 10.1007/s10916-019-1295-4.

[12] L. M. A. Zohair, "Prediction of student's performance by Modelling Small Dataset Size", *International Journal of Educational Technology in Higher Education*, Vol. 16, No. 1, 2019. doi: 10.1186/s41239-019-0160-3

[13] X. Xu, J. Wang, H. Peng, and R. Wu, "Prediction of academic performance associated with internet usage behaviors using machine learning algorithms", *Computers in Human Behavior*, Vol. 98, pp. 166–173, 2019. doi: https://doi.org/10.1016/j.chb.2019.04.015

[14] R. Ghorbani and R. Ghousi, "Comparing Different Resampling Methods in Predicting Students' Performance Using Machine Learning Techniques", *IEEE Access*, Vol. 8, pp. 67899–67911, 2020, doi: https://doi.org/10.1109/access.2020.2986809.

[15] K. T. Chui, D. C. L. Fung, M. D. Lytras, and T. M. Lam, "Predicting at-risk university students in a virtual learning environment via a machine learning algorithm", *Computers in Human Behavior*, Vol. 107, p. 105584, Jun. 2020, doi: 10.1016/j.chb.2018.06.032.

[16] H. Zeineddine, U. Braendle, and A. Farah, "Enhancing prediction of student success: Automated machine learning approach", *Computers & Electrical Engineering*, Vol. 89, p. 106903, Jan. 2021, doi: 10.1016/j.compeleceng.2020.106903.

[17] S. Hussain and M. Q. Khan, "Student-

Performulator: Predicting Students' Academic Performance at Secondary and Intermediate Level Using Machine Learning", *Annals of Data Science*, Jun. 2021, doi: 10.1007/s40745-021-00341-0.

[18] M. N. Yakubu and A. M. Abubakar, "Applying machine learning approach to predict students' performance in higher educational institutions", *Kybernetes*, Vol. 51, No. 2, pp. 916–934, Jun. 2021, doi: 10.1108/k-12-2020-0865.

[19] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms", *Smart Learning Environments*, Vol. 9, No. 1, Mar. 2022, doi: 10.1186/s40561-022-00192-z.

[20] A. L. P. J. Rose and A. C. T. Mary, "An Early Intervention Technique for At-Risk Prediction of Higher Education Students in Cloud-based Virtual Learning Environment using Classification Algorithms during COVID-19", *International Journal of Advanced Computer Science and Applications*, Vol. 13, No. 1, 2022, doi: https://doi.org/10.14569/ijacsa.2022.0130174.

[21] V. Christou, I. Tsoulos, V. Loupas, A. T. Tzallas, C. Gogos, P. S. Karvelis, N. Antoniadis, E. Glavas, and N. Giannakeas, "Performance and early drop prediction for higher education students using machine learning", *Expert Systems with Applications*, Vol. 225, p. 120079, 2023. doi: https://doi.org/10.1016/j.eswa.2023.120079.

[22] 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, 2018, doi: 10.1007/s00500-018-3581-3.

[23] 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.

[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] N. Wu, L. Zhang, Y. Gao, M. Zhang, X. Sun, and J. Feng, "CLMS-Net: dropout prediction in MOOCs with deep learning", In: *Proc of the ACM Turing Celebration Conference*, China, pp. 1-6, 2019, doi: 10.1145/3321408.3322848.

[26] Y. He, R. Chen, X. Li, C. Hao, S. Liu, G. Zhang, and B. Jiang, "Online At-Risk Student Identification using RNN-GRU Joint Neural Networks", *Information*, Vol. 11, No. 10, p. 474, 2020, doi: 10.3390/info11100474.

[27] 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, 2020, doi: 10.18608/jla.2020.72.1.

[28] 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, 2020, doi: 10.1016/j.chb.2019.106189.

[29] 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, 2020, doi: 10.1007/s10639-020-10273-6.

[30] 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, 2021, doi: 10.1016/j.compeleceng.2021.107315.

[31] 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.

[32] 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, 2021, doi: 10.1108/idd-12-2020-0160.

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

[34] 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.

[35] 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*, 2021, doi:

10.1007/s40692-021-00196-7.
[36] A. S. Aljaloud, D. M. Uliyan, A. Alkhalil, M. A. Elrhman, A. F. M. Alogali, Y. M. Altameemi, M. Altamimi, and P. Kwan, "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.
[37] H. C. Chen, E. Prasetyo, S. S. Tseng, K. T. Putra, Prayitno, S. S. Kusumawardani, and C. E. Weng, "Week-Wise Student Performance Early Prediction in Virtual Learning Environment Using a Deep Explainable Artificial Intelligence", *Applied Sciences*, Vol. 12, No. 4, p. 1885, 2022, doi: 10.3390/app12041885.
[38] 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, 2023, doi: 10.1016/j.eswa.2022.118868.
[39] M. Bilal, M. Omar, W. Anwar, R. H. Bokhari, and G. S. Choi, "The role of demographic and academic features in a student performance prediction", *Scientific Reports*, Vol. 12, No. 1, 2022, doi:10.1038/s41598-022-15880-6.
[40] J. Kuzilek, M. Hlosta, and Z. Zdrahal, "Open University Learning Analytics Dataset", *Scientific Data*, Vol. 4, No. 1, 2017, doi:10.1038/sdata.2017.171.

Appendix: A

Student Features	**Category**	**Data type**
College name Department Gender Address School type Mode of study Working part time Transportation to the college Travel time to college Accommodation type	Demographic Information	Tabular /Static data
Mothers education Fathers education Number of sisters/brothers Mothers occupation Fathers occupation	Family Details	
Willing to do higher studies Attending seminars/conferences? Other activities	Personal Interest	
Attendance to classes Listening in classes Taking notes in classes Exam preparation Exam preparation time Weekly study time	Study Behaviour	
10 th Mark 12 th Mark	Previous Academic Performance	
Sem 1 - Paper 1 Sem 1 - Paper 2 Sem 1 - Paper 3 Sem 1 - Paper 4 Sem 1 - Paper 5 Sem 1 - Paper 6 Sem 2 - Paper 1 Sem 2 - Paper 2 Sem 2 - Paper 3 Sem 2 - Paper 4 Sem 2 - Paper 5 Sem 2 - Paper 6 Sem 3 - Paper 1 Sem 3 - Paper 2 Sem 3 - Paper 3 Sem 3 - Paper 4 Sem 3 - Paper 5 Sem 3 - Paper 6 Sem 4 - Paper 1 Sem 4 - Paper 2 Sem 4 - Paper 3 Sem 4 - Paper 4 Sem 4 - Paper 5 Sem 4 - Paper 6	Current Academic Information	Time series data