



A Multi-Source Deep Learning Model Utilizing Campus Data to Enhance Early Academic Performance Prediction

Arun Prasath N¹

Department of Computer Science, PSG College of Arts & Science, Coimbatore 641014, Tamil Nadu, India

Corresponding Author Email: arunphd25@gmail.com

Copyright: ©2025 The author. This article is published by IIETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/mmep.120422>

ABSTRACT

Received: 14 December 2024

Revised: 17 February 2025

Accepted: 25 February 2025

Available online: 30 April 2025

Keywords:

student performance prediction, ML, deep learning, temporal traits, correlation features, BiGRU, ELM

The quality of education depends on the early identification of students who are at risk of learning inefficiency. Most current research has utilized Machine Learning (ML) models to forecast pupils' academic performance according to their behavioral information. This process involves manually extracting behavioral features with the help of expert knowledge and experience. However, the growing diversity and volume of behavioral data have made it difficult to recognize higher-level handcrafted attributes. Therefore, this manuscript introduces a new Multi-Source Deep Learning Model (MSDLM) for predicting student performance utilizing various data sources. First, academic, demographic, and campus activity data are gathered to create a student database, which is pre-processed and fed into the MSDLM. In this model, an embedding layer is adopted to learn dense vectors of log-format behavior data, such as web page viewing behavior followed by the one-dimensional Convolutional Neural Network (1DCNN) to shorten the length of behavior sequences. The Bidirectional Gated Recurrent Unit (BiGRU) model is then used to extract temporal characteristics of all behavioral attributes, which are transformed into a feature tensor. This tensor is given to a two-dimensional CNN (2DCNN) to extract correlation characteristics between different behaviors. These temporal and correlation characteristics are further fused with academic and demographic attributes to form a single feature vector. This vector is used to train the Extreme Learning Machine (ELM) classifier for predicting students' academic performance. Finally, experiments demonstrate that the MSDLM achieves 91.1% accuracy compared to existing models for predicting students' academic performance.

1. INTRODUCTION

Evaluating student's academic achievement is crucial, and their learning achievement plays a significant factor in the assessment process. Research has shown that struggling students are more likely to experience stress, depression, and a higher risk of dropping out of school. Students may miss classes due to mental health issues, family or social problems, or lack of support from teachers, putting their academic progress at risk [1]. It is essential for schools to quickly identify at-risk students and provide the necessary support and intervention. Instructors can identify pupils who need more help, additional sessions, or inspiration to avert negative activities such as poor grades and dropping out. Effective methods to predict students' academic performance are needed [2].

Research on improving the academic performance of underachieving students is best conducted by focusing on high school or college students. Because their grades will determine their college options, higher education pupils are currently the ideal population to study [3]. Data collected from pupils, including demographic and academic records, can be used to find students with low academic performance [4, 5].

Nevertheless, owing to a huge population of pupils and limited resources, it is challenging for educators and schools to assess each student's academic progress effectively.

Various ML algorithms have been utilized to forecast students' academic progress, including early failure detection, placement rate prediction, student forecasting, at-risk student identification, and final exam forecasting [6, 7]. Identifying and managing at-risk students has garnered significant attention in the scientific community. However, the success rate of early student risk prediction is largely dependent on the characteristics of the dataset used, which are diverse and complex. Most research has focused on common student traits, such as academic, personal, and demographic characteristics [8]. Data on daily living behaviors, such as eating, shopping, using libraries, browsing the internet, and more, are a crucial source of information about student behavior on campus. However, existing studies do not utilize this behavior data to accurately predict student achievement. Zhai et al. [9] created prediction models by extracting variables like breakfast incidence, web usage, neatness, attentiveness, and sleep patterns from unprocessed behavioral information using ML methods. However, these models often require manual feature extraction, which is time-consuming and dependent on expert

knowledge. Also, existing ML models fail to fully capture multifaceted behavioral characteristics that influence student performance, such as campus activities, internet usage, library entries, and daily habits.

To address these limitations, this article introduces a novel MSDLM that leverages various campus data sources to enhance the accuracy of student performance prediction. Unlike traditional models that rely solely on academic and demographic data, this MSDLM incorporates behavioral data, such as campus activities and web usage, to offer an extensive analysis of student involvement in learning. The key contributions of this study are:

- 1) First, this study collects information from the student database, encompassing academic, demographic, and campus activity attributes, which are then pre-processed using various pre-processing methods, like data cleaning, deduplication, merging, etc.
- 2) Second, the log behavior data is fed into the 1DCNN via an embedding layer to obtain dense vectors representing key features of log-based behavior data, with the sequence length.
- 3) Third, the features extracted by the 1DCNN are given to the BiGRU model along with transaction behavior data to capture temporal characteristics across all behavior types. These temporal features are converted into a feature tensor and fed to the 2DCNN to capture correlation features among different behaviors.
- 4) Fourth, the temporal and correlation features from the BiGRU and 2DCNN are combined with demographic and academic data to create a unified feature vector. This vector is then trained using the ELM classifier to predict students' learning performance.

1.1 Ethical considerations in student data usage

Using student data to predict academic performance raises ethical concerns regarding privacy, consent, and responsible data handling. This study rigorously prioritizes privacy protection, ensuring that all student information is anonymized. The violation of students' privacy is prevented during both the data collection and processing phases. The student IDs in the raw data are pseudonymous. The realism of the students' spatiotemporal trend is diminished. All data about the specific date and location of a behavior's occurrence have been omitted. Consequently, reidentifying individuals within the gathered dataset would be relatively challenging.

The paper is structured as follows: Section 2 reviews prior studies on predicting student performance using ML and DL models. Section 3 presents the MSDLM and Section 4 evaluates its efficiency. Section 5 summarizes the work.

2. LITERATURE SURVEY

This section explores previous studies based on ML and DL models for student performance prediction.

2.1 Review on student performance prediction models

A multiclass forecasting method [10] was presented that utilizes J48, Naive Bayes (NB), Support Vector Machine (SVM), Linear Regression (LR), and Random Forest (RF). However, its accuracy was low while increasing the number of pupils' records. A hybrid Deep Neural Network (DNN) [11] was presented to forecast student performance based on past

data. However, it needs multiple attributes to increase the prediction accuracy. During the COVID-19, the K-Nearest Neighbor (KNN) and SVM classifiers [12] were applied to measure students' fulfillment in online education. However, the SVM's high complexity and the KNN's slower training led to a decline in performance. Multivariate distribution models [13] were developed using quiz and assignment assessments to predict a weighted score for an engineering mathematics course and assess its impact on the final grade. However, due to the limited data, the predictions were not accurate. To predict students' performance, Light Gradient Boosted Machine (LightGBM), Category Boosting (CatBoost), and Extreme Gradient Boosting (XGBoost) [14] were utilized. However, more factors such as sociodemographic details and ranks obtained in the enrolled syllabus were necessary to boost the precision of the predictions. In the study conducted by Poudyal et al. [15], a hybrid 2DCNN was presented to predict academic performance. However, it has a low sensitivity due to a limited dataset.

2.2 Review on early detection of at-risk students

Using data sources and algorithms, numerous studies have identified at-risk students for early notification and feedback, enhancing student performance prediction and preventing low-performing students from completing final exams. An augmented education model [16] using the Long Short-Term Memory (LSTM) network followed by ML models such as XGBoost, KNN, SVM, RF, and Gradient Boost Regression Tree (GBRT) was created to forecast learning success based on students' behavior data. However, to make accurate predictions, more details on the students' activities are needed. Ensemble techniques including additional trees and XGBoost with Shapley additive explanations [17] were used to forecast student achievement and find at-risk pupils. However, using datasets with more properties could improve performance. A forecasting model [18] was developed using RF, SVM, KNN, additional tree, AdaBoost, gradient boosting, and Artificial Neural Network (ANN) to forecast performance scores and find at-risk pupils. However, to improve prediction accuracy, more textual characteristics connected to the students' input were required. Data from a 4-year open institution [19] was used to develop a DNN-based predictive model for forecasting students' academic performance in new subjects. To enhance the model's efficacy, integrating additional semester data was required.

An ensemble model [20] was created utilizing various ML algorithms to forecast at-risk students during the pandemic. However, its accuracy was limited due to lack of student-specific characteristics.

2.3 Research gap

The studies mentioned above use different ML and DL models to predict students' academic performance. Classical ML models rely on manually extracted features, which can introduce biases and degrade prediction performance. Many studies consider academic and demographic data, neglecting information about student activities or behavioral insights. In contrast, DL models such as DNNs and CNNs struggle to capture changes in student behavior over time. Also, these models were trained using limited datasets, which can lead to overfitting and poor generalizability. Hence, this study aims to develop the MSDLM using a large dataset containing student records from various sources to enhance prediction accuracy and model generalizability.

3. PROPOSED METHODOLOGY

This section explains the MSDLM for predicting student performance. It encompasses data collection, pre-processing,

temporal feature extraction using BiGRU, correlation feature extraction using 2DCNN, and prediction using ELM classifier, as shown in Figure 1.

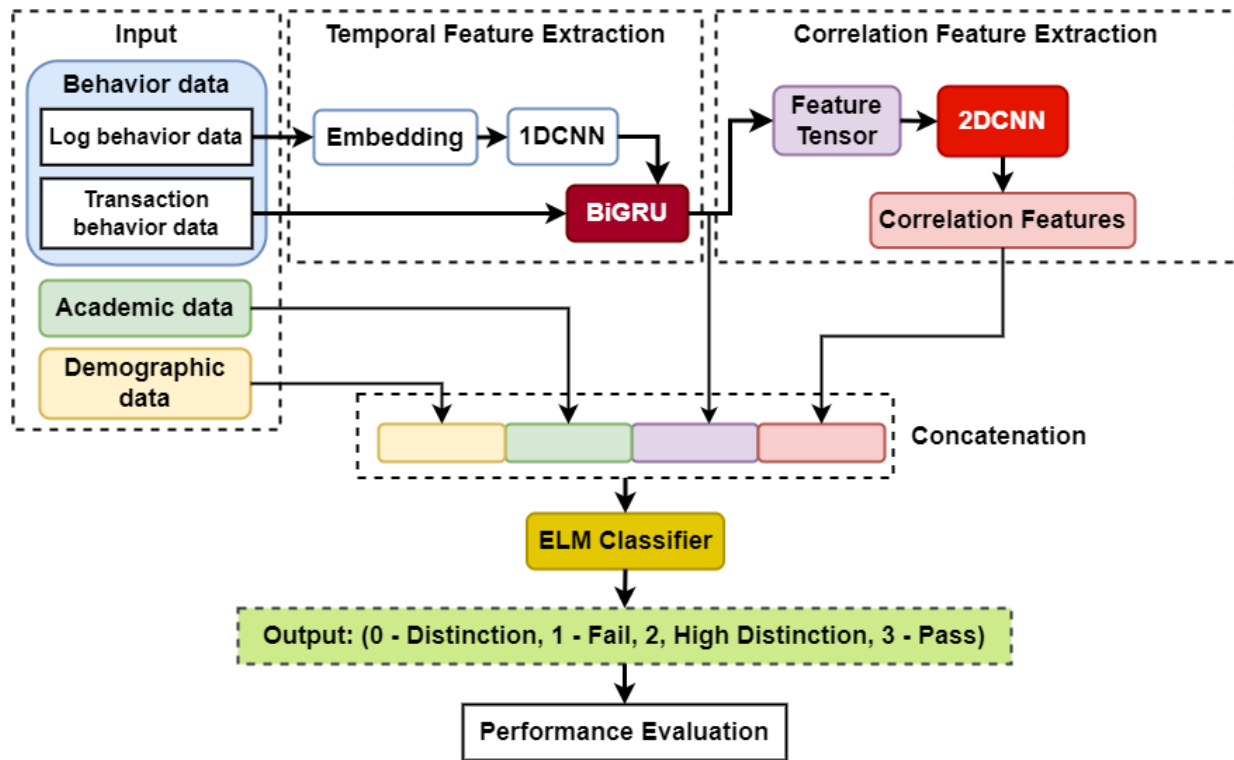


Figure 1. Overall structure of the presented model

3.1 Data collection

In this study, the dataset was created by gathering academic and demographic records of 80,000 students from government and private engineering colleges in Coimbatore, Tamil Nadu over 120 days. It comprises 133 attributes, 80,000 instances, and 1 class attribute. The academic attributes are the number of students, course name, type of college (public or private), subject grades, study materials, teaching style, class size, smartphone allowance, etc. The demographic attributes are name, age, sex, home place (rural, urban, or semi-urban), family type (nuclear or joint), occupation, academic skills of family members, parental homework help, social circle, TV viewing habits, home internet connection, and other details. Also, information was collected using ETL tools on four distinct pupil actions on campus: consumption activity in the canteen, perusing the web, entering a library, and logging into a gateway.

The purpose of this paper is to use student behavior data to predict academic success on campus. To achieve this, certain conditions were established to exclude student samples with minimal behavior records. Precisely, students were required to have at least 1,000 web page browsing behavior records and at least 20 records for breakfast, lunch, dinner, and gateway login behaviors per semester.

3.2 Pre-processing

Effective pre-processing is crucial to ensure that the input data is clean and structured for model training. A few pre-processing methods applied in this study are discussed below.

3.2.1 Handling date and time

The raw behavior data includes timestamps stored in the “yyyy-mm-dd hh:mm:ss” format, which is unsuitable for direct input into the model. Hence, preprocessing of the date and time was necessary. The date was transformed into a numerical format, beginning with 1, symbolizing the first academic day in the university calendar. This allows for sequential representation of time when keeping consistency with the semester schedule. Besides, time was divided into K intervals of size τ to represent distinct periods during which behaviors occurred.

Each interval is assigned a numerical value (1 to K), making it easier for the model to process behavioral sequences. Different activities require different τ to prevent redundant log entries:

Web browsing behavior: τ was set to 4 hours to prevent repeated logging of the same website visits within a short time. For example, a browsing log at 10:45 AM would be assigned to the 8 AM-12 PM interval.

Other behaviors (library entry, cafeteria transactions, and gateway logins): τ was set to 15 minutes to capture short-term behaviors while reducing redundancy. For example, a cafeteria purchase at 1:30 PM is assigned to the 1:15 PM-1:30 PM interval.

3.2.2 Data deduplication and merging

Behavioral logs can contain duplicate records if the same activity is recorded multiple times within a short period. Therefore, duplicate records are merged to reduce storage overhead and prevent bias in model training. Different types of behaviors have different merging logics as outlined below.

- For cafeteria transaction behavior, if two purchases

occur at the same time, date, and place, they are merged into one record with the sum of their consumption quantities.

- For gateway login behavior, if multiple logins occur within the same interval, the total session duration and network traffic usage are summed into a single record.
- For library entry behavior, repeated library entries within the same interval are merged into a single log.
- For web browsing behavior, if the same website is visited multiple times within the same interval, only the first entry is retained to reduce redundancy.

3.2.3 Handling missing data

For academic and demographic data, missing numerical attributes (e.g., grade, attendance percentages, etc.) are imputed using mean values. Additionally, missing categorical variables (e.g., gender, family type, etc.) are imputed using mode values. In the case of behavioral data, if a student has missing behavior records (e.g., no cafeteria transactions logged), a zero-value placeholder is assigned to maintain a uniform data structure.

3.2.4 Feature scaling and encoding

To enhance model training, numerical attributes like grades, attendance percentages, etc., are scaled using min-max normalization. This prevents features with varying numerical ranges from skewing the model. Alternatively, categorical attributes such as gender and college type are transformed using one-hot encoding.

3.3 MSDLM model for student performance prediction

This MSDLM comprises the following key components:

- 1DCNN: It processes log-based student behavior data to extract relevant features, reduce dimensionality and identify behavior trends.
- BiGRU: It is a variant of the GRU network that analyzes past and future information to capture temporal traits in sequential students' behavior data.
- 2DCNN: It captures relationships between different behavioral attributes.
- ELM: It is a fast classification method that uses a single-layer neural network to efficiently learn from input features and make predictions.

Figure 2 shows the flowchart of the MSDLM, which aids in comprehending the functioning of this model for predicting students' performance.

3.3.1 Input

The MSDLM contains various categories of student details such as academic, demographic, and behavioral attributes. Each category is a time series, meaning all records have a timestamp, yet different students have different attributes. Here, $X_i = (X_{i1}, \dots, X_{ij}, \dots, X_{iN})$ represents the N categories of multi-source attributes of student i , where $X_{ij} = [x_{ij}^1, \dots, x_{ij}^t, \dots, x_{ij}^{T_{ij}}]$ is the j^{th} attribute of i , x_{ij}^t ($1 \leq t \leq T_{ij}$). Each X_{ij} has a vector of single event record information at period t , such as a single consumption record or gateway login record. T_{ij} represents the length of the j^{th} attribute of i .

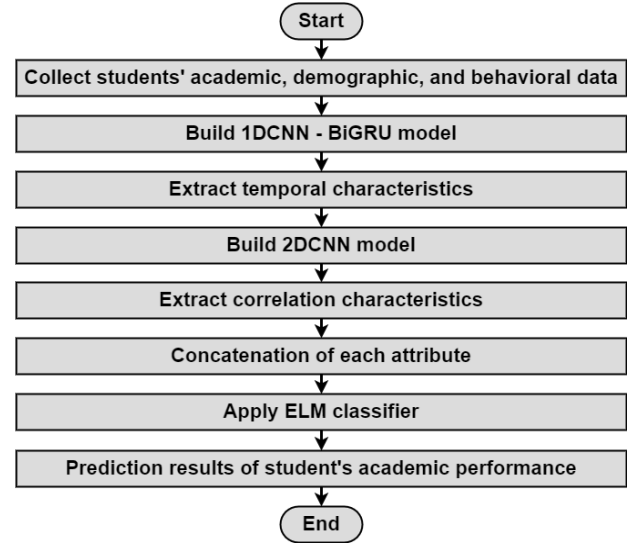


Figure 2. Flowchart of MSDLM for students' performance prediction

After applying the pre-processing methods in Section 3.2, the data can be directly used as inputs to the MSDLM.

3.3.2 Temporal feature extraction using BiGRU model

This study uses the BiGRU network, a variant of LSTM, which can capture the sequential patterns in the data and learn the dependencies between different time steps. This makes it suitable for analyzing and predicting student behavior over time. Also, it can effectively handle the temporal nature of the data and improve the accuracy of behavior prediction.

Campus behavior data can be categorized into transaction and log behavior data based on how they are generated. Transaction behavior data consists of single records for each activity event, like consumption, library entry, and gateway login behavior. These data are typically input into BiGRU after one-hot encoding or normalization. On the other hand, log behavior data, like web page browsing behavior, can generate hundreds or thousands of records for a single event. Modeling log behavior data with BiGRU poses challenges due to the many URL domains and long sequences.

To address these challenges, an embedding layer is used to create dense vectors for URL domains, and a one-dimensional convolutional network is employed to reduce sequence length before applying BiGRU for modeling.

Embedding Layer for URL domain representation: This study adopts the embedding layer in DL to learn URL domain vectors for the academic performance prediction task. This procedure involves: (1) determining the frequency of URL domain accesses in the dataset; (2) creating a domain index table sorted by access frequency and assigning indexes in descending order; (3) selecting high-frequency domain names from the index table; (4) converting web browsing behavior sequences into index values for domain names; (5) incorporating the embedding layer into the deep neural model configuration.

Shortening the length of a behavior sequence: The BiGRU model is effective at capturing long information dependencies but struggles with extremely long sequences of web page browsing behaviors. To address this issue, this study applies 1DCNN to the behavior sequence to extract local time features. Pooling layers are then used to filter out redundant features, effectively reducing the sequence length while retaining important behavioral details.

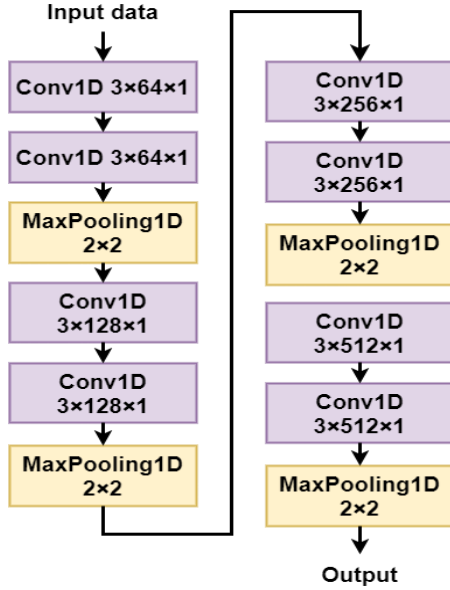


Figure 3. Structure of 1DCNN

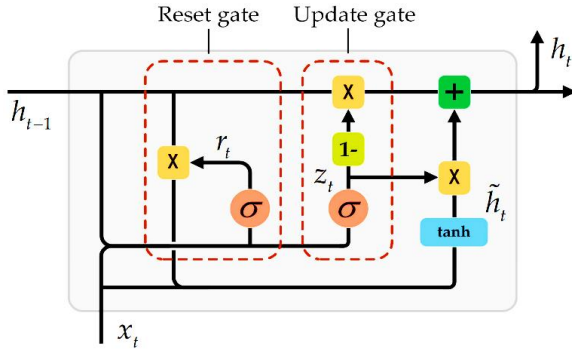


Figure 4. Structure of BiGRU network

The 1DCNN model portrayed in Figure 3 is designed to shorten the behavior sequence length. It consists of two consecutive convolution layers followed by a pooling layer. Conv1D $3 \times k \times 1$ represents a convolution layer with k 1D convolutions using a kernel size of 3 and a step size of 1. The kernel size of 3 is chosen to increase the network's nonlinear expression ability by adding depth while maintaining the same receptive field as a larger convolution kernel. The values of k can be 64, 128, 256, or 512. MaxPooling1D 2×2 is a 1D maximum pooling layer with a kernel size of 2 and a step size of 2. Thus, this model significantly reduces the sequence length from L to $L - 60/16$.

BiGRU network: Figure 4 illustrates the structure of BiGRU network. The hypothesis is that the output at time t may be influenced by both past and future input. Assuming that the neural network computes the j^{th} hidden unit, it first combines the hidden state and cell state. After that, it produces the reset gate q_j , which is calculated by Eq. (1).

$$q_j = \sigma([W_r x]_j + [U_r h(t-1)]_j) \quad (1)$$

In Eq. (1), σ is the sigmoid function, $[\cdot]_j$ is the j^{th} element of a vector, x and $h(t-1)$ denote the input and former hidden state vectors, respectively, W_r and U_r are weight matrices. Then, it merges the forget and input gates into a unified update gate z_j as Eq. (2):

$$z_j = \sigma([W_z x]_j + [U_z h(t-1)]_j) \quad (2)$$

After that, the actual activation of the h_j is calculated by Eqs. (3) and (4).

$$h_j(t) = z_j h_j(t-1) + (1 - z_j)(\tilde{h}_j) \quad (3)$$

$$\tilde{h}_j(t) = \tanh([W x]_j + [U(q \odot h(t-1))]_j) \quad (4)$$

At last, an element-wise sum is adopted to add forward and backward states generated by BiGRU as the output of the j^{th} element. This is represented in Eq. (5).

$$h_j(t) = [\overrightarrow{h_j(t)} \oplus \overleftarrow{h_j(t)}] \quad (5)$$

Thus, the BiGRU network learns the temporal relationship between behavioral data to obtain temporal feature vectors.

3.3.3 Correlation feature extraction using 2DCNN model

Academic performance data from multiple sources for a comparable student should be linked based on different characteristics. This is achieved by converting the temporal feature vectors of each behavior data into a 3D tensor using a tensor method. The 2DCNN is then employed to capture the relationship between various characteristics, enabling the extraction of correlation features across the data.

In this context, a picture is represented as a tensor (ω, h, c) , where ω, h , and c denote the width, height, and number of channels, respectively. The 2DCNN is used to extract picture features. Similarly, the temporal attribute vectors of N different types of attributes are transformed into a 3D tensor, with the M dimension of the temporal attribute vector represented as $\omega \times h = N, c = M$.

By applying the 2DCNN on the tensor, effective correlation characteristics can be extracted. This procedure aids in analyzing the relationship between various characteristics in the student academic performance data.

3.3.4 Student performance prediction using ELM classifier

This study focuses on predicting students' learning achievements by classifying them into Distinction, Fail, High Distinction, and Pass. The outcomes are represented as $y \in \{0: \text{Distinction}, 1: \text{Fail}, 2: \text{High Distinction}, 3: \text{Pass}\}$. The evaluation procedure is as follows:

Grade Point Averages (GPAs) are a common way to measure students' academic performance. GPAs are calculated using numerical values derived from academic scores. To determine high distinction and fail, all student scores are sorted from highest to lowest GPA. High distinction usually includes the top $k\%$ of students with scores ranging from 85% to 100%, while fail includes the bottom $k\%$ with scores from 0% to 49%. Scores between 75% and 84% are considered a pass, and scores between 50% and 64% are classified as a distinction.

The ELM classifier determines the grade for students' academic performance using the fused temporal, correlation, academic, and demographic attributes. It utilizes a single-layer feed-forward network, as shown in Figure 5, to predict students' performance in class. This classifier employs randomly initialized hidden layer weights to optimize the output layer's weights using the Moore-Penrose generalized inverse. This approach reduces the computational complexity of parameter optimization.

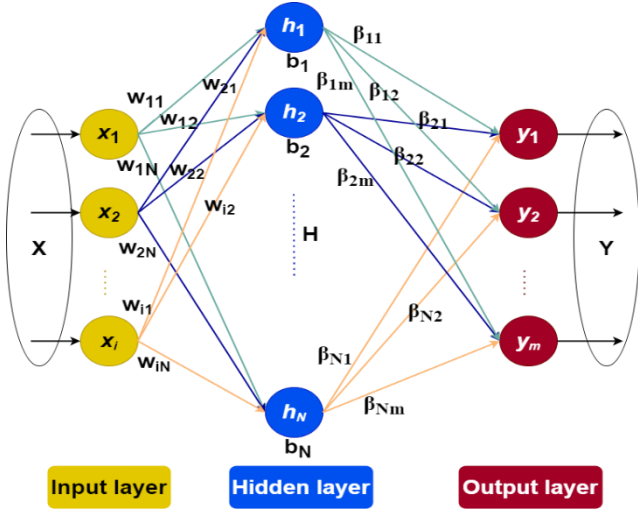


Figure 5. Structure of ELM classifier

The objective is to learn the relationship between m attributes $(x_i, y_i), i = 1, \dots, m$, where $x_i \in R^m$ and $y_i \in R^m$, to predict students' learning outcomes. The result of ELM with N hidden neurons is represented by Eq. (6).

$$y = \sum_{i=1}^N \beta_i f(x, w_i, b_i) \quad (6)$$

In Eq. (6), N is the total hidden nodes, β_i is the weight value associating i^{th} hidden and output nodes, f is the activation function, w_i is the weight value associating i^{th} hidden and input nodes and b_i is the bias of i^{th} hidden node. Eq. (6) can be represented by Eqs. (7) and (8).

$$Y = H\beta \quad (7)$$

where,

$$H = \begin{pmatrix} f(x_1, w_1, b_1) & \cdots & f(x_1, w_N, b_N) \\ \vdots & \ddots & \vdots \\ f(x_M, w_1, b_1) & \cdots & f(x_M, w_N, b_N) \end{pmatrix} \quad (8)$$

After deciding on the total hidden nodes and the ELM's activation function, every parameter, except for β_i , is chosen at random. Then, the least-square form is used to determine the ELM norm, as given in Eqs. (9) and (10):

$$L(X, Y; \beta) = \|Y - H\beta\|^2 \quad (9)$$

where,

$$\beta = H^+ Y \quad (10)$$

In Eq. (10), H^+ is the Moore-Penrose generalized inverse of H . Additionally, the dropout is applied before ELM to alleviate overfitting, a weighted cross-entropy error is considered as the loss factor, and Adam is utilized as the optimizer. The weighted cross-entropy error is defined by Eqs. (11) and (12).

$$\text{loss} = \frac{1}{N} \sum_{k=1}^N \sum_{c=1}^M w_c y_c^k \log(p_c^k) \quad (11)$$

where,

$$w_c = \frac{N}{M * N_c} \quad (12)$$

In Eqs. (11) and (12), w_c is the weight of the tag c , N is the sum quantity of pupil information, N_c is the total records in specific c , M is the total tags, y_c^k is the real score of k^{th} instance in c , and p_c^k is the predicted score possibility. Thus, the MSDLM can be used to predict students' performance by analyzing multi-source campus data in conjunction with their academic and demographic information.

4. RESULTS AND DISCUSSION

In this section, the efficiency of the MSDLM is evaluated against conventional ML/DL models. MATLAB 2019b is used as the software tool.

4.1 Dataset

To ensure fairness in performance evaluation, proposed and existing models were trained and evaluated on the same dataset. It consists of student records including academic, demographic, and behavioral data from government and private engineering colleges in Coimbatore, Tamil Nadu. This study involves 80000 instances of student data. Of these, 64000 instances (16000 for each grade level) are used for training and 16000 instances (4000 for each grade level) for testing. More information about the attributes in this dataset are presented in Section 3.1.

4.2 Model configuration

To maintain an unbiased evaluation, the hyperparameters of each model were optimized efficiently. Table 1 presents the parameter settings for the proposed MSDLM and existing models such as SVM [10], KNN [12], XGBoost [14], ANN [18], and DNN [19]. All models were trained under similar computational conditions to ensure a fair comparison.

4.3 Performance metrics

This study focuses on analyzing accuracy, precision, recall, and F-measure, as these metrics provide valuable insights into prediction performance compared to other metrics. These metrics are defined as follows:

- **Accuracy:** It is the percentage of correct predictions of students' grades out of the total number of predictions made. It is calculated by Eq. (13).

$$\text{Accuracy} = \frac{\text{True Positive (TP)} + \text{True Negative (TN)}}{\text{TP} + \text{TN} + \text{False Positive (FP)} + \text{False Negative (FN)}} \quad (13)$$

For instance, let's assume there are two classes: pass and fail. TP represents the percentage of positive data (pass) that are predicted to pass, TN represents the percentage of negative data (fail) that are predicted to fail, FP represents the percentage of negative data that are predicted to pass, and FN represents the percentage of positive data that are predicted to fail.

- **Precision:** It is calculated by Eq. (14).

$$Precision = \frac{TP}{TP + FP} \quad (14)$$

- **Recall:** It is determined by Eq. (15).

$$Recall = \frac{TP}{TP + FN} \quad (15)$$

- **F-Measure:** It is calculated by Eq. (16).

$$F - measure = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (16)$$

Table 1. Parameter settings for existing and proposed models

Models	Parameters	Range
SVM [10]	Kernel type	Linear
	Regularization parameter	1.0
	Penalty	0.1
	Gamma	0.01
KNN [12]	No. of neighbors	5
	Distance metric	Euclidean
	Weights	Distance-based
XGBoost [14]	Number of trees	200
	Learning rate	0.05
	Maximum tree depth	6
	Subsample	0.8
	Column sampling	0.7
	Gamma	0.1
	Lambda (L2 regularization)	1.0
ANN [18]	No. of hidden layers	3
	No. of neurons per layer	[64, 128, 64]
	Activation function for hidden layers	Rectified Linear Unit (ReLU)
	Activation function for output layer	Sigmoid
	Optimizer	Adam
	Learning rate	0.001
	Loss function	Categorical cross-entropy
	Batch size	32
	No. of epochs	100
DNN [19]	No. of hidden layers	4
	No. of neurons per layer	[128, 256, 128, 64]
	Activation function for hidden layers	ReLU
	Activation function for output layer	Softmax
	Batch size	32
	Learning rate	0.0005
	Optimizer	Adam
	No. of epochs	100
	Loss function	Categorical cross-entropy
BiGRU	No. of layers	2
	GRU units per layer	128
	Dropout rate	0.3
	Recurrent dropout	0.2
	Activation function for hidden layers	ReLU
	Optimizer	Adam
	Learning rate	0.0001
	Batch size	32
	Loss function	Weighted cross-entropy
	No. of epochs	100
Proposed MSDLM	No. of convolutional layers	4
	Filters per layer	[64, 128, 256, 512]
	Kernel size	(3,3)
	Activation function	ReLU
	Dropout rate	0.4
	Optimizer	Adam
	Learning rate	0.0005
	Batch size	32
	No. of epochs	100
	Loss function	Weighted cross-entropy
	Number of hidden nodes	500
	Activation function	Softmax
ELM	Regularization parameter	10 ³
	Solver	Moore-Penrose inverse

4.4 Experimental results

Figure 6 illustrates the confusion matrix of the MSDLM for predicting students’ academic performance. It is a well-known representation of the model’s performance across different classes (grades) of prediction. In this illustration, the rows indicate the predicted grades, while the columns signify the actual grades. The diagonal green boxes signify exactly predicted instances, while the red cells indicate inaccurately predicted instances.

Using this matrix, TP, FP, FN, and TN values for each class are measured, which are given in Table 2. These values are utilized to determine the accuracy, precision, recall, and F-measure values of MSDLM. It is observed that the proposed MSDLM accurately predicted 3640 distinctions, 3650 fails, 3642 high distinctions, and 3644 pass instances (i.e., 14576 out of 16000 instances accurately predicted), achieving an overall accuracy of 91.1%.

Testing Set					
Actual \ Predicted	Distinction	Fail	High Distinction	Pass	Total
Distinction	3640 22.8%	105 0.7%	150 0.9%	105 0.7%	4000 91.0% 9.0%
Fail	110 0.7%	3650 22.8%	125 0.8%	115 0.7%	4000 91.3% 8.8%
High Distinction	120 0.8%	102 0.6%	3642 22.8%	136 0.8%	4000 91.0% 9.0%
Pass	105 0.7%	121 0.8%	130 0.8%	3644 22.8%	4000 91.1% 8.9%
Total	3975 91.6% 8.4%	3978 91.8% 8.2%	4047 90.0% 10.0%	4000 91.1% 8.9%	14576 / 16000 91.1% 8.9%

Figure 6. Confusion matrix for MSDLM

Table 2. Detailed statistics for each class prediction using MSDLM

Class	TP	FP	FN	TN
Distinction	3640	360	335	11665
Fail	3650	350	328	11672
High-distinction	3642	358	405	11595
Pass	3644	356	356	11644

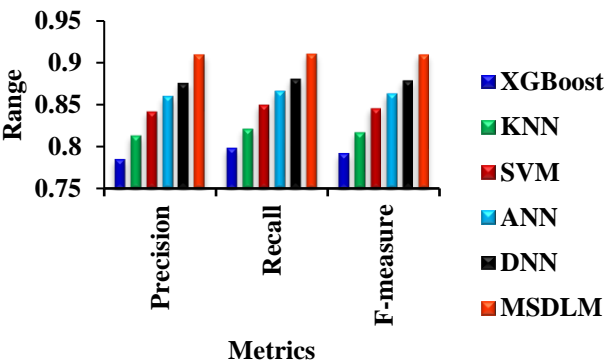


Figure 7. Comparison of precision, recall, and f-measure for different student performance prediction models

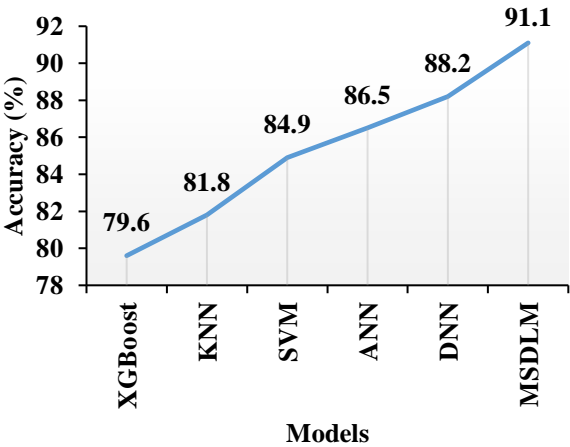


Figure 8. Comparison of accuracy for different student performance prediction models

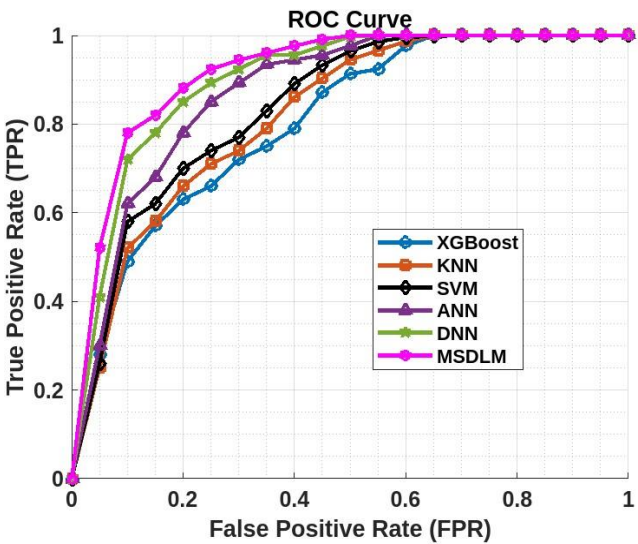


Figure 9. Comparison of ROC curve for different student performance prediction models

Figure 7 displays a comparison of precision, recall, and F-measure for various student performance prediction systems. It can be noticed that the precision of MSDLM is significantly higher, with increases of 15.92%, 11.93%, 8.08%, 5.81%, and 3.88% compared to XGBoost, KNN, SVM, ANN, and DNN, respectively. Similarly, the recall is also notably improved, with increases of 14.16%, 10.96%, 7.18%, 5.2%, and 3.41% compared to the same models. The F-measure follows a similar trend, showing improvements of 14.9%, 11.38%, 7.57%, 5.44%, and 3.53% compared to XGBoost, KNN, SVM, ANN, and DNN, respectively.

Figure 8 compares the accuracy of different student performance prediction models. MSDLM has higher accuracy than XGBoost, KNN, SVM, ANN, and DNN by 14.45%, 11.37%, 7.3%, 5.32%, and 3.29% respectively. The superior performance of MSDLM is attributed to its ability to learn temporal and correlation features from multi-source campus data. This data includes a diverse set of information, encompassing not only student academic and demographic records but also behavior attributes. By leveraging this multi-source data, MSDLM is able to capture complex patterns and relationships that contribute to a more accurate prediction of student performance.

Figure 9 shows Receiver Operating Characteristic (ROC) curves for the proposed MSDLM and existing models for predicting students' academic performance. It represents the relationship between the True Positive Rate (TPR) and False Positive Rate (FPR) at different prediction thresholds. Each point on the curve indicates the balance between accurately predicting academic grades for each student. The closer the curve is to the top-left corner, the better the MSDLM is at distinguishing between different grades.

In summary, these findings highlight MSDLM as a robust and effective model for predicting student performance. Its superior accuracy, when compared to other established models, underscores the significance of incorporating temporal and correlation features from multi-source campus data in the predictive modeling process. This approach can offer valuable insights into understanding and forecasting student outcomes in an educational setting.

4.5 Potential limitations

Although MSDLM demonstrates remarkable improvement compared to existing models, there are possible limitations that need to be considered. Its ability to generalize to other educational institutions with varying curriculum frameworks, student populations, and institutional policies needs to be further confirmed. The use of past data also implies that sudden shifts in student behaviors or campus activities may affect prediction accuracy. Therefore, future studies should investigate the scalability of MSDLM in diverse academic settings and incorporate additional contextual factors like students' physiological attributes for a more comprehensive predictive model.

4.6 Real-world implementation and challenges

Implementing the proposed MSDLM in real educational settings requires attention to infrastructure, data availability, and ethical considerations. Institutions must integrate this model with historical student information to ensure seamless data collection and processing. Faculty and administrators need training to interpret predictions and apply them in academic interventions effectively.

Challenges may arise in data privacy and compliance with regulations like the General Data Protection Regulation (GDPR) and the Family Educational Rights and Privacy Act (FERPA), obliging robust data anonymization and encryption measures. Some stakeholders and students may be resistant to the model since it uses continuous behavioral data, which raises concerns about surveillance and potential misuse of data. To address these concerns, institutions should prioritize transparency, obtain informed consent, and establish clear data usage guidelines.

Scalability is another significant challenge, especially for institutions with limited computational resources. Cloud-based solutions and federated learning approaches can help overcome these limitations and facilitate broader adoption in diverse learning environments.

5. CONCLUSION AND FUTURE WORK

This paper introduces the MSDLM that predicts student academic performance using daily campus behavior data with academic and demographic attributes. It addresses the

challenge of manually extracting features from multi-source heterogeneous students' behavior data. It uses 1DCNN to shorten behavior sequences, an embedding layer to learn the dense vector of nominal attributes, and BiGRU to capture temporal features. Besides, 2DCNN extracts correlation features between different behaviors. These temporal and correlation characteristics are combined with academic and demographic attributes to create a unified feature vector. This vector is then fed into the ELM classifier to predict students' academic performance. Furthermore, results from extensive experiments proved that the MSDLM on the large-scale students' dataset has 91.1% accuracy, 0.91 precision, 0.911 recall, and 0.91 F-measure compared to the XGBoost, KNN, SVM, ANN, and DNN models.

This study highlights the importance of integrating student information systems into the MSDLM for educators. Thus, they can better understand students' performance in the classroom. This information can be used to create modified intervention strategies for students who may be at risk of low academic achievement. Additionally, administrators can use this model to allocate resources more effectively for student support programs based on predictive insights.

Future work could explore integrating additional data types, such as emotional or psychological data, to enhance prediction accuracy. This could involve sentiment analysis from student feedback, stress levels, or attendance metrics for deeper insights into academic performance. Furthermore, using explainable AI could improve the interpretability of MSDLM, aiding educators in understanding the reasoning behind predictions and making more informed interventions.

REFERENCES

- [1] Ogresta, J., Rezo, I., Kožljan, P., Paré, M.H., Ajduković, M. (2021). Why do we drop out? Typology of dropping out of high school. *Youth & Society*, 53(6): 934-954. <https://doi.org/10.1177/0044118X20918435>
- [2] Pedditzi, M.L., Fadda, R., Lucarelli, L. (2022). Risk and protective factors associated with student distress and school dropout: A comparison between the perspectives of preadolescents, parents, and teachers. *International Journal of Environmental Research and Public Health*, 19(19): 12589. <https://doi.org/10.3390/ijerph191912589>
- [3] Hassan, E.M.G. (2023). Addressing academic challenges: A quasi-Experimental study on the effect of remedial exam strategy for nursing students with low academic performance. *Belitung Nursing Journal*, 9(4): 369. <https://doi.org/10.33546/bnj.2699>
- [4] Liu, J., Peng, P., Zhao, B., Luo, L. (2022). Socioeconomic status and academic achievement in primary and secondary education: A meta-analytic review. *Educational Psychology Review*, 34(4): 2867-2896. <https://doi.org/10.1007/s10648-022-09689-y>
- [5] Heppt, B., Olczyk, M., Volodina, A. (2022). Number of books at home as an indicator of socioeconomic status: Examining its extensions and their incremental validity for academic achievement. *Social Psychology of Education*, 25(4): 903-928. <https://doi.org/10.1007/s11218-022-09704-8>
- [6] Yağcı, M. (2022). Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*, 9(1): 11. <https://doi.org/10.1186/s40561-022-00192-z>

- [7] Baashar, Y., Alkawsi, G., Mustafa, A., Alkahtani, A.A., Alsariera, Y.A., Ali, A.Q., Hashim, W., Tiong, S.K. (2022). Toward predicting student's academic performance using artificial neural networks (ANNs). *Applied Sciences*, 12(3): 1289. <https://doi.org/10.3390/app12031289>
- [8] Batool, S., Rashid, J., Nisar, M.W., Kim, J., Kwon, H.Y., Hussain, A. (2023). Educational data mining to predict students' academic performance: A survey study. *Education and Information Technologies*, 28(1): 905-971. <https://doi.org/10.1007/s10639-022-11152-y>
- [9] Zhai, M.Y., Wang, S.T., Wang, Y.Z., Wang, D.J. (2022). An interpretable prediction method for university student academic crisis warning. *Complex & Intelligent Systems*, 8(1): 323-336. <https://doi.org/10.1007/s40747-021-00383-0>
- [10] Bujang, S.D.A., Selamat, A., Ibrahim, R., Krejcar, O., Herrera-Viedma, E., Fujita, H., Ghani, N.A.M. (2021). Multiclass prediction model for student grade prediction using machine learning. *IEEE Access*, 9: 95608-95621. <https://doi.org/10.1109/ACCESS.2021.3093563>
- [11] Yousafzai, B.K., Khan, S.A., Rahman, T., Khan, I., Ullah, I., Ur Rehman, A., Baz, M., Hamam, H., Cheikhrouhou, O. (2021). Student-performulator: Student academic performance using hybrid deep neural network. *Sustainability*, 13(17): 9775. <https://doi.org/10.3390/su13179775>
- [12] Abdelkader, H.E., Gad, A.G., Abohany, A.A., Sorour, S.E. (2022). An efficient data mining technique for assessing satisfaction level with online learning for higher education students during the COVID-19. *IEEE Access*, 10: 6286-6303. <https://doi.org/10.1109/ACCESS.2022.3143035>
- [13] Nguyen-Huy, T., Deo, R.C., Khan, S., Devi, A., Adeyinka, A.A., Apan, A.A., Yaseen, Z.M. (2022). Student performance predictions for advanced engineering mathematics course with new multivariate copula models. *IEEE Access*, 10: 45112-45136. <https://doi.org/10.1109/ACCESS.2022.3168322>
- [14] Saidani, O., Menzli, L.J., Ksibi, A., Alturki, N., Alluhaidan, A.S. (2022). Predicting student employability through the internship context using gradient boosting models. *IEEE Access*, 10: 46472-46489. <https://doi.org/10.1109/ACCESS.2022.3170421>
- [15] Poudyal, S., Mohammadi-Aragh, M.J., Ball, J.E. (2022). Prediction of student academic performance using a hybrid 2D CNN model. *Electronics*, 11(7): 1005. <https://doi.org/10.3390/electronics11071005>
- [16] Zhao, L., Chen, K., Song, J., Zhu, X., Sun, J., Caulfield, B., Mac Namee, B. (2020). Academic performance prediction based on multisource, multifeature behavioral data. *IEEE Access*, 9: 5453-5465. <https://doi.org/10.1109/ACCESS.2020.3002791>
- [17] Sahlaoui, H., Nayyar, A., Agoujil, S., Jaber, M.M. (2021). Predicting and interpreting student performance using ensemble models and shapley additive explanations. *IEEE Access*, 9: 152688-152703. <https://doi.org/10.1109/ACCESS.2021.3124270>
- [18] Adnan, M., Habib, A., Ashraf, J., Mussadiq, S., Raza, A. A., Abid, M., Bashir, M., Khan, S.U. (2021). Predicting at-Risk students at different percentages of course length for early intervention using machine learning models. *IEEE Access*, 9: 7519-7539. <https://doi.org/10.1109/ACCESS.2021.3049446>
- [19] Nabil, A., Seyam, M., Abou-Elfetouh, A. (2021). Prediction of students' academic performance based on courses' grades using deep neural networks. *IEEE Access*, 9: 140731-140746. <https://doi.org/10.1109/ACCESS.2021.3119596>
- [20] Karalar, H., Kapucu, C., Gürüler, H. (2021). Predicting students at risk of academic failure using ensemble model during pandemic in a distance learning system. *International Journal of Educational Technology in Higher Education*, 18(1): 63. <https://doi.org/10.1186/s41239-021-00300-y>