

A unified deep learning framework for accurate student performance-prediction towards sustainable education

Taneja Sanjay Devkishan¹, Sanjay Kumar Singh¹, Ajay Kumar Bharti²

¹Amity Institute of Information Technology, Amity University Uttar Pradesh Lucknow Campus, Lucknow, India

²Department of Computer Science and Engineering, Ambalika Institute of Management and Technology, Lucknow, India

Article Info

Article history:

Received Feb 1, 2026

Revised Jul 30, 2026

Accepted Aug 20, 2026

Keywords:

Attention mechanism

CNN-GRU architecture

Deep neural networks

Heterogeneous data

Hybrid deep learning

Predictive analytics

Sequential attribute predictor

ABSTRACT

Forecasting how students will perform early in their studies is difficult in higher education domain, reason being varying types of academic and behavioral data. Traditional deep learning (DL) techniques face challenges with the selection of influential attributes and feature relationships, resulting in poor performance prediction. To overcome this, our research presents a sequential attribute predictor (SAP) and an attention-enhanced hybrid deep learning (HDL) framework which combines feature-selection using SAP with an attention mechanism and examines five DL models: long short-term memory (LSTM), gated recurrent unit (GRU), convolutional neural network (CNN), hybrid LSTM-GRU, and hybrid CNN-GRU. These models were trained on an extensive dataset collected from students of various universities in India and other countries. The assessment results showed that performance of hybrid models is superior to that of individual DL models. The best performing model was SAP based attention-enhanced hybrid CNN-GRU, which had lowest training MSE of 0.0434 and a validation MSE of 0.0284. A comparatively higher coefficient of determination ($R^2=0.9746$) of proposed model clearly reflects a better fit between predicted and actual outcomes due to CNN's capability to extract discriminative features, GRU's effectiveness in modeling sequential patterns, and the attention mechanism's ability to emphasize influential attributes.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Taneja Sanjay Devkishan

Amity Institute of Information Technology, Amity University Uttar Pradesh Lucknow Campus

Lucknow, India

Email: sdtaneja@lko.amity.edu

1. INTRODUCTION

The academic success of a student is strongly linked to their mental and emotional well-being. The performance in the classroom is a key measure of learning efficiency and quality of education. It has been demonstrated in previous studies that low-performing students are much more vulnerable to anxiety, depression, and even suicidal moods than high-performing students. The very purpose of forecasting academic achievement is to identify high-risk students in advance to detect this problem and handle it earlier. This vision will assist administrators, educators, and students in making corrective measures within a certain time and a certain direction in order to mitigate adverse consequences of course failure, dropout, and absenteeism. Consequently, prediction of success of students has received much attention and research. New areas that have high potential in the education field are in deep learning (DL) assessment and student performance evaluation [1], [2]. The methods use the most advanced neural network architecture to forecast how well children would do in the institution based on things like their past grades, attendance history, behavior patterns, and social

contacts. Predictive models effectively enable the early detection and early intervention of students with the potential of performing poorly in academics, hence promoting the achievement of better learning outcomes.

Furthermore, automated DL algorithms have been applied in marking specific assignments and tests and thus have assisted teachers in offering their students with effective and prompt feedback [3]–[5]. DL has also been incorporated into learning analytics and learning data mining, which has helped educators to have a better insight into the learning behaviors and cognition patterns of the students. Having such insight, evidence-based instruction is attained thereby making the whole learning process more effective. Moreover, DL is a vital feature of the intelligent tutoring systems (ITS), which allow personalized learning to occur in the case of dynamic modifications of the educational material and the way of teaching it according to the requirements and learning preferences of specific learners. The constant analysis of interactions and successes of a student helps the ITS to change its teaching strategy and become more efficient in meeting the needs of a certain student. Also, natural language processing (NLP) applications based on DL have been deployed to automatically rate student performance by explicitly examining their content, structural consistency, and linguistic effectiveness [6].

Figure 1 demonstrates instructions for gathering and managing evaluation labels or student assessment data for machine learning (ML) model training. Using the most up-to-date evaluation data, the trained model can then predict performance labels for incoming students or track their development from one semester to the next. In short, the figure represents the full workflow starting from the process of students' data collection to performance evaluation and predictions. Potential of DL has also been implemented to carry out performance analysis using massive open online courses (MOOCs). This also has allowed the platforms to experiment with the efficiency of the course material, areas in need of enhancement and capitalize on learning experience in the scale of masses [7]–[9]. Moreover, DL has been used in sentiment and emotion analysis, which can provide useful information about the emotional experience of students during the learning process. This has facilitated the formation of emotionally insightful learning areas that support the well-being of students and inspire their eagerness to learn. Even with these new ideas, though, the ethical and data privacy issues need to be talked about when DL models are put into academic institutions. Evaluating educational data helps with understanding how students learn and spotting trends in their performance. It also helps with data-driven choices in education.

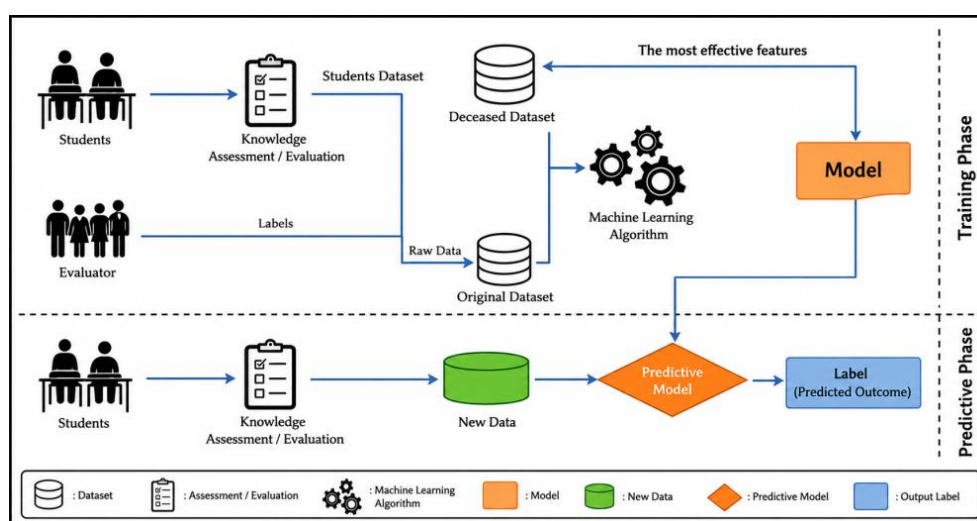


Figure 1. Student performance evaluation

If we talk about the importance of educational data analysis and the use of DL in predicting academic performance, then it's worth to discuss that DL takes data analysis to another level. It makes the analysis of large amounts of complex data easier by detecting relationships of varying degrees of complexity that were not easily discernible before. Some of the models that fulfill this requirement include convolutional neural networks (CNNs), recurrent neural networks (RNNs), long short-term memory (LSTM), and gated recurrent units (GRUs). They are particularly useful for student performance data as they capture learning challenges and help provide timely interventions on learning. Educational data analysis together with DL helps understand the challenges of the learners and helps them to learn in the way that helps them learn the best. It also helps support learner retention and the improvement of the educational institution overall.

Since information about students should not be compromised and data security must not be jeopardized, it is essential that data collection and analysis should be conducted in a responsible

manner [5], [10], [11]. It is imperative to proceed with the analysis of new trends and advances that have been introduced recently in IEEE publications and other scholarly sources to narrow down on the analysis of student performance through DL methods, as studies in this field have advanced at a very fast pace [12].

Academic performance can be more accurately assessed using a hybrid neural network (HNN) model that combines precision-driven LSTM systems; CNNs, as well as high-efficiency GRUs. With this update done, the model can then make the effective use of student data to extract spatial and temporal information [13], [14]. The hybrid network's use of LSTM and GRU allows it to identify patterns and correlations in data that persist over time, including students' study habits and exam scores [15]–[17]. The LSTM memory-storage gating mechanism extended to GRUs allows for the consideration of both short-term and long-term temporal dependencies. With this knowledge, we can attempt to make performance predictions for students and analyze the variables that impact their learning outcomes [18], [19]. This technique is a noteworthy advancement to the assessment system. It allows educators to have improved insight to help them design personalized interventions and support pathways for students [20], [21]. The progressive HNN structure embeds LSTM functioning alongside GRUs and provides a more powerful mechanism to assess and understand educational outcomes [22].

Considering here the research problem, traditional academic performance prediction systems utilize ML techniques, and statistics. These systems struggle to capture the complex and nonlinear relationships that exist in educational data. Moreover, they struggle to describe the dynamics of learning over time, as well as the relationships among features in a high-dimensional space. The combination of these limitations may lead to insufficient prediction accuracy to facilitate the timely detection of at-risk students. Therefore, a more sophisticated DL approach is necessary to extrapolate complex patterns from various educational datasets and provide improved predictions of students' academic performance within the appropriate timeframe.

Currently, most student performance prediction systems only utilize academic data to the exclusion of behavioral information, socio-economic data, and health data. Most importantly, current DL systems fail to capture the spatial and temporal dimensions, thereby resulting in higher prediction error. Therefore, to remedy the gaps in performance prediction derived from behavioral and health data, a multi-dimensional approach leveraging a hybrid deep learning (HDL) model is essential.

Shortcomings of previous research: earlier research relied on standard ML models that struggled to pick up on complex, nonlinear relationships in educational data. Many of these methods didn't do a good job of showing how learning patterns change over time or how different factors depend on each other, which made their predictions less accurate. On top of that, the lack of focus on prioritizing features or using attention mechanisms made it harder to explain the results and less effective at identifying at-risk students early.

As far as the significance of the proposed model is concerned, this research has significance as it aims to employ DL for the identification of students who are at risk of failing early on. The model consists of multiple factors, including academic achievement, attendance, classroom demeanor, and learning skills, which can help accurately, foresee students' performance. The predictions are more accurate because it is able to follow complex patterns over time; this means that higher education systems can intervene and provide support earlier. It also aids in the development of improved educational systems to cater for the individual student, maintaining engagement and reducing drop-out rates. In summary, this model assists Colleges and Universities with making decisions based on actual data, so that they can effectively promote student success in an early manner rather than merely taking a reactive approach after a student is already behind.

Research objectives are important to be mentioned as it clearly defines what the research intends to achieve thereby providing various methods and direction to perform the entire study effectively. Some of the most applicable research objectives are mentioned below:

- The study aims to collect multidimensional student data encompassing academic, behavioural, socio economic, and lifestyle factors.
- Feature selection is performed using the sequential attribute predictor (SAP) method. The research develops DL models, including standalone LSTM, GRU, and CNN architectures, alongside hybridized LSTM-GRU and CNN-GRU frameworks.
- Attention mechanisms are integrated to improve the learning of feature importance.
- Standalone and hybrid models are compared to identify the most effective architecture for student academic performance prediction.

A research hypothesis presents us a tentative explanation or prediction, that through a systematic investigation can be tested. Following are the few listed that applies best to our ongoing research work:

- H1: Factors like academic performance, behavior, socio-economic status, and health play a crucial role in how well students succeed.
- H2: Spotting students, who might be struggling early on, by looking at their study habits and engagement data, can really boost their academic success through timely support.
- H3: Using SAP-based feature selection greatly enhances our ability to pinpoint the key factors that impact student performance.

- H4: HDL models offer a significant edge in predicting student performance when compared to traditional models.
- H5: Incorporating attention mechanisms greatly improves our understanding of student-related features and helps lower prediction errors.
- H6: Among all the models we evaluated, the hybrid CNN–GRU model stands out as the most accurate in predicting student performance.

The research contributions cover all key contributions of this study mainly including the following:

- Created a comprehensive dataset that captures a variety of factors: academic, behavioural, socio-economic, and health, that influence how students perform.
- Pinpointed and examined the main factors affecting student performance using SAP-based feature selection.
- Introduced hybrid models that utilize DL to improve the accuracy of predictions regarding student performance.
- Incorporated an attention mechanism to make the models more interpretable and to spotlight the key factors related to students.
- Showed that the hybrid CNN–GRU model results in the lowest prediction error but highest efficiency.
- Provided valuable insights into how students learn, allowing early identification of those who are truly at risk.
- Recommended a data-driven approach to support decision-making in education and boost student performance.

2. LITERATURE REVIEW

2.1. Student performance studies

There have been numerous studies on student achievement using data-driven prediction models. Another method was based on a ML system that was able to accurately predict academic performance, demonstrating how predictive analytics can be used in educational settings [23]. The study of technology-based learning contexts has also been substantial, with a view of improving learning outcomes by analysing student behaviour in these contexts and emphasizing the vital role of institutional and household tutoring for improving student performance [24]. In addition, the analysis of the assessment methods in political theory and ideology courses revealed the lack of connection between the assessment results and the knowledge of the students, which is a reason for the need for the development of better assessment systems [25].

2.2. Deep learning and machine learning methods

There are many ML and DL approaches proposed for predicting student performance. Notably, when used in conjunction with back propagation neural network (BPNN), random forest (RF), and Naïve Bayes (NB), the results are promising, especially with BPNN having the best accuracy rate. To overcome the data sparsity problem, researchers proposed an enhanced conditional generative adversarial network (CGAN) for generating synthetic data and used multiple kernel functions and deep support vector machine (DSVM) models. This integration had resulted in an improvement in sensitivity, specificity, and area under the curve (AUC) values. Moreover, DL algorithms have been used to improve educational assessment systems, highlighting the growing impact of sophisticated artificial intelligence (AI) on educational analytics [26].

Table 1 provides a summary of some of the research studies that have used various approaches to predict or analyze students' academic performance. This table summarizes the techniques used and the performance results obtained; this helps to compare and understand which techniques are really successful in terms of performance measures such as accuracy and root mean square error (RMSE). It helps us steer our research in a more innovative direction.

Moreover, the table provides an indication that the performance of prediction is different for various algorithms. Although there have been various studies on applying ML and DL to predict students' performance, most of them have focused on studying academic history, or studying behavioral, socio-economic and lifestyle factors. Moreover, many studies are based on single DL models that are limited in their ability to capture the spatial and temporal characteristics of student data. This suggests that there is a clear need for a HDL model that can enhance the performance of the predictive model by using multidimensional student data and feature selection strategies [27].

2.3. Critical analysis of existing studies

Many investigations have been conducted related to predictive tools of student performance prediction based on ML and DL approaches. Some of these traditional approaches such as RF, SVM, DB, decision trees (DT), and evolutionary algorithms (EA) have been successful but they are not able to capture the complex,

nonlinear relationships and temporal dependencies inherent in the educational data. This restriction can affect their ability to achieve accurate predictions when it comes to a diverse array of students, including those based on academics, behavior, demographics, and socio-economic status.

Table 1. Literature synopsis

Ref.	Method	Performance matrix
Waldeyer <i>et al.</i> (2022) [28]	Diagonally weighted least square	RMSE=0.3, TLI=0.96, CFI=0.97
Mai <i>et al.</i> (2022) [29]	Louvain method	Accuracy=74.00% and F1-Score=73.5%
Alshabandar <i>et al.</i> (2020) [30]	RF	RMSE=8.1
Pereira <i>et al.</i> (2019) [31]	Evolutionary algorithm	Accuracy=75.55%
Rimadana <i>et al.</i> (2019) [32]	Time structure questionnaire (TSQ)	Accuracy=84.00%
Zohair (2019) [33]	Multiple ML algorithms	Accuracy=76.30%

2.3.1. Research gap

Through a thorough examination of current studies, several research gaps have been recognized:

- There's a real gap in having effective ways to prioritize the attributes that would help us choose the most impactful indicators of student performance.
- The use of feature prioritization along with attention mechanisms in models that predict performance is quite limited.
- We're not fully utilizing both spatial feature interactions and temporal learning dependencies within a single architectural framework.
- A thorough comparison of different DL architectures using the same datasets and performance metrics hasn't been adequately explored.
- There's not enough focus on minimizing prediction errors while still ensuring strong classification performance for the early detection of students who may be at risk.

2.3.2. Novelty of the proposed work

In order to tackle these challenges, this study proposes a novel approach, termed the SAP, and an attention-enabled hybrid CNN-GRU model for student performance prediction. The most salient feature of this work is the inclusion of a SAP mechanism that is able to rank and select the most influential academic, behavioral, demographic, and socio-economic attributes prior to the start of model training.

In addition, an attention mechanism is introduced to focus on key features for predictions dynamically, which makes the model more interpretable and can also enhance the predictive accuracy. The hybrid CNN-GRU architecture is specifically designed to capture the spatial feature relationships by CNN and to handle the temporal learning dependency by GRU to give a more comprehensive representation of student data. The proposed framework integrates SAP, attention, CNN, and GRU into a single prediction model and has not been fully investigated in the current educational data mining literature. In addition, a detailed comparative analysis of different models, such as CNN, LSTM, GRU, LSTM-GRU, CNN-LSTM, CNN-GRU, and their corresponding SAP-enhanced models, is provided to demonstrate the benefits of the proposed model. The findings showed considerable improvements in the MAE, MSE, RMSE, and MAPE, which helped to increase the reliability of the prediction and provide adequate support for early intervention for students who were at academic risk.

3. RESEARCH APPROACH AND METHOD ADOPTED

We carefully selected a wide and comprehensive range of data from a range of Indian and international universities, across a wide variety of disciplines and settings. This data set contained twenty-one relevant predictive attributes that were examined in detail and included academic achievement measures, behavioral and psychological measures, socio-economic and family background measures, co-curricular and extra-curricular engagement measures, social media engagement measures, health and wellness measures, and independent study measures. These various parameters have provided a robust analytical frame for the creation of a data-driven predictive model that can provide accurate and reliable predictions in terms of students' performance and academic progress.

The one we proposed aims to estimate student performance and predict it using a DL framework that combines five architectures namely LSTM, GRU, CNN, LSTM-GRU, and CNN-GRU. We used techniques such as normalization, label encoding, and class balancing using SMOTE-ENN to counteract bias and promote diversity in the learning environment.

This was necessary because our data includes both academic and behavioral data for students. Using exploratory data analysis (EDA), we were able to visually and statistically analyze the data trends and

relationships. The models were constructed using the adaptive based moment estimation also known as Adam Optimizer and mean squared error (MSE) as the loss function, and after running for 100 iterations (epochs), through the Keras Sequential API. The GRU and LSTM networks were able to learn the temporal change or the relationships over time, while the CNN learned the spatial patterns. The new hybrid model is innovative since it uses recurrent and convolutional layers simultaneously, which enables the model to learn short-term and long-term dependencies in the Scholastic data.

The proposed diagram showing the workflow in Figure 2 clarifies the chronological steps followed in the creation and testing of a useful predictive model based on the results of students. The workflow is a sequence used to build and test a predictive model especially in predicting student performance by applying ML procedures. The process is organized in the following significant steps:

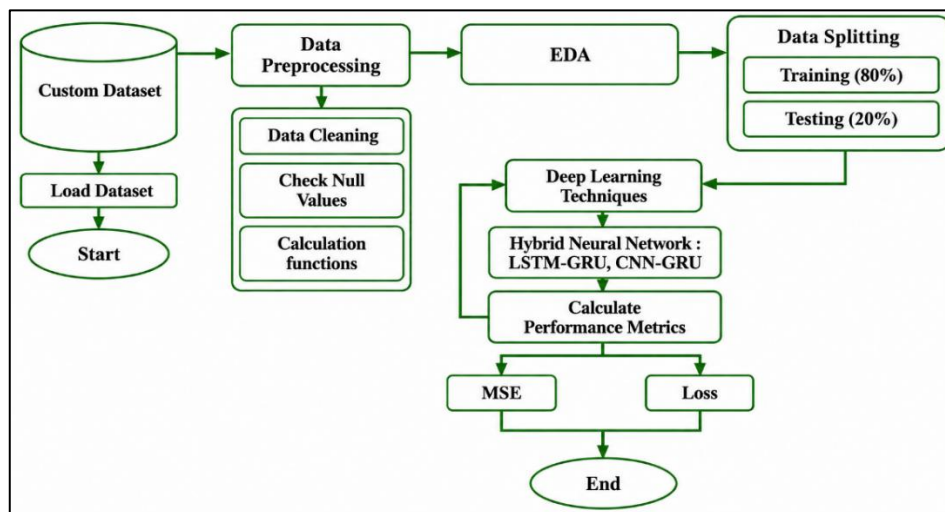


Figure 2. Proposed diagram showing the workflow

3.1. Data collection

This wide scope of information ensured that the models would take into consideration various factors that may affect the success of the students. Academic records normally contain grades, attendance to classes and other institutional indicators. Personal habits that included time spent studying and the method to study were collected through surveys whereas family background information included socio-economic status and the level of parental education.

During the research, we collected student data under twenty-one major fields including health status which included both physical and psychological health. Other factors include how many hours of self-study students do each day, the time they spend on extracurricular activities, and how they consistently use virtual learning platforms. Do they smoke or drink on a regular basis? And do they use social media? The factors that were used to find out how engaged the students are and what kind of lifestyle they follow thereby compromising academics directly or indirectly was very helpful and decisive towards our research approach. Additionally, socio-demographic data such as the number of siblings, parental education levels, their occupation, and data on family income was also included to assess the role of family background in shaping student academic achievement.

Referring to Table 2, it is evident that the fundamental goal of the dataset is to highlight the different elements that affect student academic performance, which can be harnessed to construct accurate models for forecasting educational success. The data were gathered in an effective, systematic and complete way through personal surveys in order to give a holistic picture of both the academic and personal life of the students and further to work upon the data using ML. The data would help in making more accurate predictions about the student performance and connecting the idea of academic performance and dependence on the environmental conditions.

3.2. Dataset overview and validation

In this research, we gathered our dataset through a structured survey using Google Forms, reaching out to various higher education institutions both in India and internationally. After a thorough process of data cleaning and validation, we selected 959 student records for our analysis. As illustrated in Table 3, the dataset comprises 21 predictive attributes that encompass academic performance, behavioral traits, socio-economic status, family details, health conditions, self-study practices, extracurricular activities, social media engagement, and participation in virtual learning. We collected responses from students enrolled in a diverse

range of undergraduate and postgraduate programs from over 30 universities and colleges, ensuring a rich mix of academic backgrounds and institutional settings. This diversity enhances the generalizability and strength of the prediction models we created.

Table 2. Summary of the primary categories of attributes that were examined in the study

Category	Representative attributes
Academic attributes	Class X marks, Class XII marks, CURRENT CGPA, and last semester SGPA
Behavioral attributes	Self-study hours, travel time, and social media exposure
Engagement attributes	Virtual learning platform usage and extracurricular activities
Health attributes	Health status, smoking habit, and alcohol consumption
Family and socio-economic attributes	Family income, number of siblings, parents' education, and parents' employment status

Table 3. Summary of dataset characteristics and validation procedures

Dataset characteristic	Value
Total student records	959
Total collected fields	26
Predictive attributes used	21
Number of institutions	>30
Data collection method	Google Form survey
Data type	Academic, behavioral, socio-economic, and health
Missing value treatment	Mean/median/mode imputation
Outlier handling	Z-score and IQR
Class balancing	SMOTE-ENN
Feature scaling	Min-Max normalization

3.3. Dataset distribution

The data show a great deal of variation as far as academic performance, family environments, and patterns of behaviour. This type allows for the models to be effective in capturing various aspects of students' performance. Before running into the model training, we made sure to perform the necessary transformations with categorical variables, namely label encoding, on numerical features, namely min-max scaling. To validate and ensure quality of the data, we adopted a series of checks as part of data validation and quality assurance. We eliminated any duplicate answers to the survey questions, and records with incomplete or inconsistent answers were carefully reviewed and corrected, where possible. Missing numerical values were filled in using mean or median statistics, while we tackled missing categorical values through mode imputation.

Other methods used for detecting outliers were Z-score and interquartile range (IQR) method. Logical consistency checks enabled us to identify any invalid entries or unusual answers. We used EDA to check the distribution of the attributes, its anomalies and relations between the variables. We used synthetic minority over-sampling technique-edited nearest neighbors (SMOTE-ENN) hybrid resampling technique to overcome the problem of class imbalance before training the models. Each of the validation and processing steps guaranteed that the final dataset was accurate, well-balanced, and suitable for the creation of efficient DL models designed to predict students' performance.

3.4. Experimental setup and reproducibility

In order to make our SAP supported attention-based hybrid CNN–GRU framework reproducible and transparent, we carried out all experiments with Python programming and popular DL libraries. The implementation was written using Python 3.10 and executed using the TensorFlow–Keras framework. The study was based on a preprocessed dataset with focus on student academics that included academic, behavioral and demographic elements.

We implemented feature selection using SAP, data normalization, and split the data into a training set (80%) and a test set (20%). Using Adam optimizer, we trained the attention-based CNN-GRU model for 100 epochs and evaluated the model using parameters such as MSE, RMSE, mean absolute error (MAE), mean absolute percentage error (MAPE), and R-squared (R^2). To judge our improvements, we compared the outcomes of our model with some of the available ML and DL models to check for any improvement in the performance.

Being important for us to ensure the reliability and reproducibility of the proposed model, we established an experimental environment with latest, well-defined software, appropriate hardware, and various training configurations. The following sub-sections clearly describes the computational resources, software tools, and training parameters that were use to implement and evaluate our proposed approach.

- Software environment. The model development and experimentation as shown in Table 4 involved the use of the following software libraries.

- b. Hardware configuration. The experiments were carried out on a workstation set up with following system specifications as indicated in Table 5.
- c. Training configuration. As shown in Table 6, the following hyperparameter settings were applied during the training of the DL models.

Table 4. Software and library configuration

Software/library	Version
S/w: Python	3.10
S/w Lib: TensorFlow	2.15
S/w Lib: Keras	2.15
Lib: NumPy	1.26
Lib: Pandas	2.1
Lib: Scikit-learn	1.4
Lib: Matplotlib	3.8
Lib: Seaborn	0.13

Table 5. System specifications

Component used	Specification
C.P.U.	Intel Core i-7/AMD Ryzen-7
R.A.M.	16 GB DDR4
G.P.U.	NVIDIA RTX 3060 (12 GB)
Storage used	512 GB SSD
Operating system used	Windows 11 (64-bit)

Table 6. Model hyperparameters

Parameter	Value
Optimizer	ADAM
Set learning rate	0.001
Size of the batch	32
Number of epochs	100
Loss-function	MSE
Validation-split	20%
Random-seed	42

To be fair, all baseline and hybrid models were processed and trained using identical preprocessing methods and conditions.

- d. Dataset accessibility. The academic and behavioural data is gathered from students in various universities by means of surveys. Due to concerns about privacy and ethics surrounding student information, the complete dataset isn't available for public distribution. Data will be promptly made available, however, for samples of anonymized data and information on implementation from the corresponding author for research or verification purposes.
- e. Reproducibility measures. Fixed random seeds were used in the partitioning of the data as well as in the initialisation of the model. Furthermore, each model was trained using the same train-test split, the same preprocessing pipeline, feature encoding strategy, normalization method and the same evaluation metrics, which facilitates a fair and unbiased comparison among the different approaches.

3.5. Data pre-processing

In preparation for training a model, the data was pre-processed after systematic collection. Missing values would be imputed and for the numeric data the mean or median would be used whereas the mode would be used for the categorical data. The Z-score and the inter quartile range (IQR) were used to identify extreme values. To ensure that the model's output was free of bias, we modified or eliminated these tactics. To convert categorical variables such as family status, study habits, and student's gender into coded variables, we used one-hot encoding, also known as label encoding. Scaling of numerical features (e.g., grades and hours of study) was done to minimize min-max method, so that no single feature (e.g., grade) would have a disproportionate influence on the learning process. Ensuring the data is well-organized is a crucial step in preprocessing that allows for the construction and testing of the model with confidence and novelty. By utilizing a general-purpose function to compute both MSE and loss simultaneously, we may simplify the assessment procedure. This technique can be used recursively in each row to expose which data points are important to the model's output by examining them closely. Through this comprehensive analysis, the strengths, weaknesses and trends of the model can be better understood.

3.6. Augmenting sequential attribute predictor based feature selection in the model

One of the widely used techniques for data dimensionality reduction and improving the efficiency of a predictive model is to perform attribute/feature selection using sequential attribute/feature selection (SAS) method with a SAP approach. In this implementation, as detailed in Algorithm 1, the sequential feature selection (SFS) from mlxtend's feature selection tools is combined with the RF classifier, which serves as the main estimator to gauge the importance of the features.

Algorithm 1. SAP-based feature selection using sequential forward selection

```

Input:
Feature matrix (X), target variable (Y)
Output:
Optimized feature subset (X_{SAP})
Steps:
Initialize Base Classifier
Select a RF Classifier being the primary estimator for its robustness and ability to
handle nonlinear feature interactions.
Configure SAP
Initialize a Sequential Feature Selector with the following settings:
Selection strategy: Sequential Forward Selection
Feature subset size: Automatically determined (best performing subset)
Evaluation metric: Classification accuracy
Cross-validation: 5-fold
Parallel processing enabled for computational efficiency
Feature Selection Process
Iteratively add features to the model that maximize cross-validated accuracy until no
further improvement is observed.
Model Fitting
Train the SAP model using the input feature matrix (X) and target variable (Y).
Extract Selected Features
Retrieve the optimal subset of selected features identified by SAP.
Dimensionality Reduction
Reduce the original feature matrix to include only the selected attributes, forming
(X_{SAP}).
Return Optimized Feature Set
Output the reduced feature matrix for subsequent DL model training.

```

The algorithm used by the SAP is SFS and uses cross-validation to incrementally add a feature that improves the model's predictive ability. Assessment with a 5-fold cross-validation (cv=5) guarantees a robust assessment and parallel processing (jobs= -1) can accelerate the calculations. This algorithm is capable of detecting the optimum subset of features that can maximize classification accuracy.

The selected features are 'Program of Study', 'Aggregate % Marks in Class-X', 'Aggregate % Marks in Class-XII', 'CGPA', 'Self Study Routine', 'Timewise Engagement in Extracurricular Activities', 'Smoking Habits', 'Father Qualification', 'Family's Annual Income', and 'Students' Ongoing Health Status'. These are a comprehensive and balanced set of academic, personal and socio-economic factors, indicative of their importance in predicting outcome. By compressing the original data, SAP not only makes the model easier to interpret, but also speeds up the computation time and limits the potential for overfitting while retaining the most informative variables.

3.7. Data inspection and visualization

EDA or data exploration and its analysis have been done to explore the data in order to find linkages and trends and an overall data examination. Summary statistics were used to compute key numerical variables in which cumulative grade point average (CGPA), semester grade point average (SGPA) and aggregate marks were to be summarized. To gain a deeper insight into the connection between student's academic achievement and factors like family income, study habits, and participation in extracurricular activities, a correlation analysis was systematically performed. In order to make a well-informed choice, EDA contributes to the development of forecasting models such as HNNs, which include the LSTM-GRU architecture. With these models, we can properly evaluate and forecast students' performance based on data. Figures 3-6 show the graphs that were constructed with the help of EDA. Figure 3 shows how the study programs are spread out and also demonstrates that technical programs are the most popular. Scatter plot in Figure 4 shows the relationship between variable and CGPA and helps us to show how students are distributed across different ranges of SGPA and CGPA. It also shows patterns of diverse student performance.

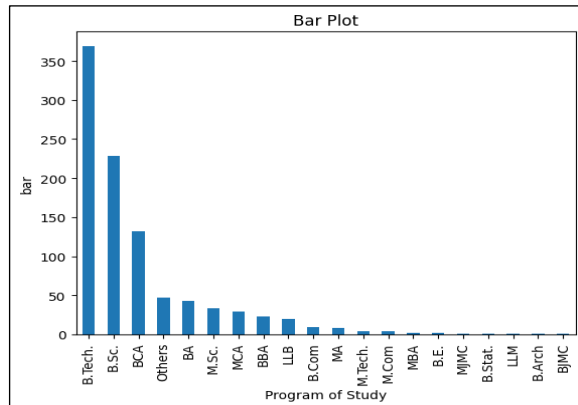


Figure 3. Program-specific bar chart representation

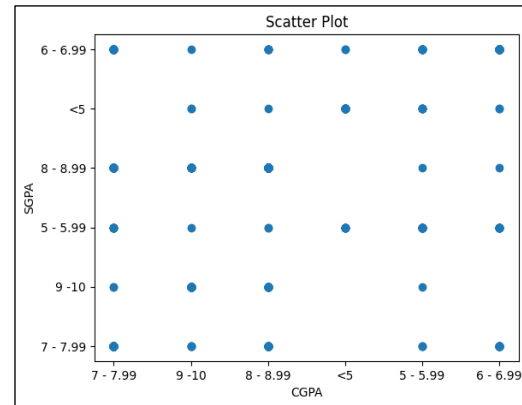


Figure 4. CGPA distribution scatter plot

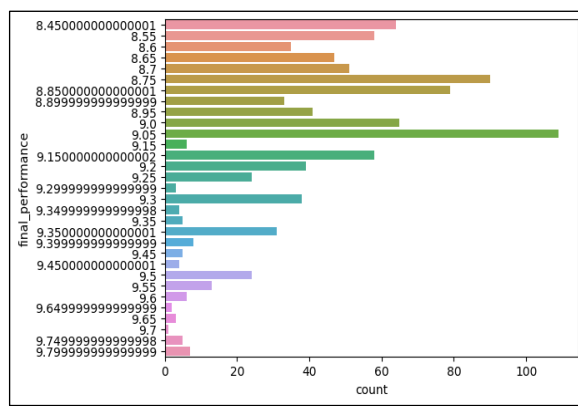


Figure 5. Count chart depicting final performance

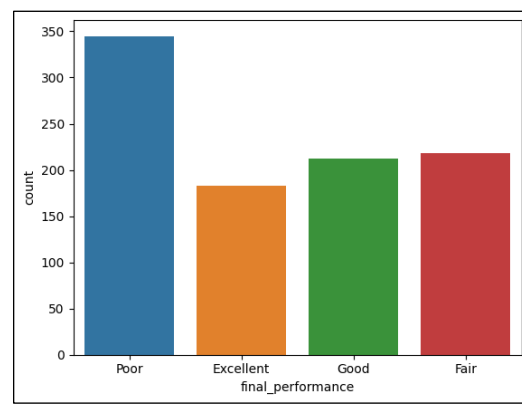


Figure 6. Final performance overview via bar chart

Figure 5 displays a count plot that illustrates how the ultimate performance of the students is distributed. The graph clearly shows student distribution as per their final performance scores. From this graph, it can be noted that majority of students lie within the 8.5-9.5 performance range. Also, only few students reach extremely high or lower performance scores. The graph in Figure 6 is a bar plot graph of final performance. It indicates skewness of overall student performance towards lower category. It can be observed that the majority of the group falls under 'Poor' since comparatively few students only fit under 'Excellent', 'Good' or 'Fair'. Figure 7 is a correlation matrix which shows how the variables that were analyzed relate to each other. Most correlations are found weak. This clears the fact that no single factor strongly influences others.

3.8. Data-driven deep learning and formulation of predictive model

ML's DL subfield employs several nonlinear processing layers in an effort to discover data hierarchies automatically. By utilizing deep neural networks (DNNs) with numerous hidden layers, the models are able to effectively discover patterns in large datasets that are complex and abstract. This article builds DL models in Python using the Kera TensorFlow framework. This is achieved by using a sequential design with a loss function that ensures optimization and convergence during learning, such as the Adam optimizer or MSE. Both models undergo 100 epochs of training to achieve optimal stability or minimal prediction error. While CNNs are great at learning about structures and locations, models like LSTM or GRU are great at learning about timing and sequences. The wide range of test scores provided by the combination of these models makes this a great tool for studying students' learning and performance.

3.8.1. Precision-driven long short term memory

To aid in the discovery of longer-span time correlations in sequentially dependent data, the LSTM model is layered with two additional components. Additionally, sequential architecture maintains time when changing the input (samples, timesteps, and features). LSTM units use their input gates, forget gates, and output gates to control and preserve the continuous movement of information. Using a Dense layer is the last stage in predicting using learned representations. The training of the model is done for 100 epochs or opting for the

Adam optimizer, both of which contribute to a reduction in loss, as assessed by the MSE. Processes where strong predictions depend on long-term connections benefit greatly from this design style.

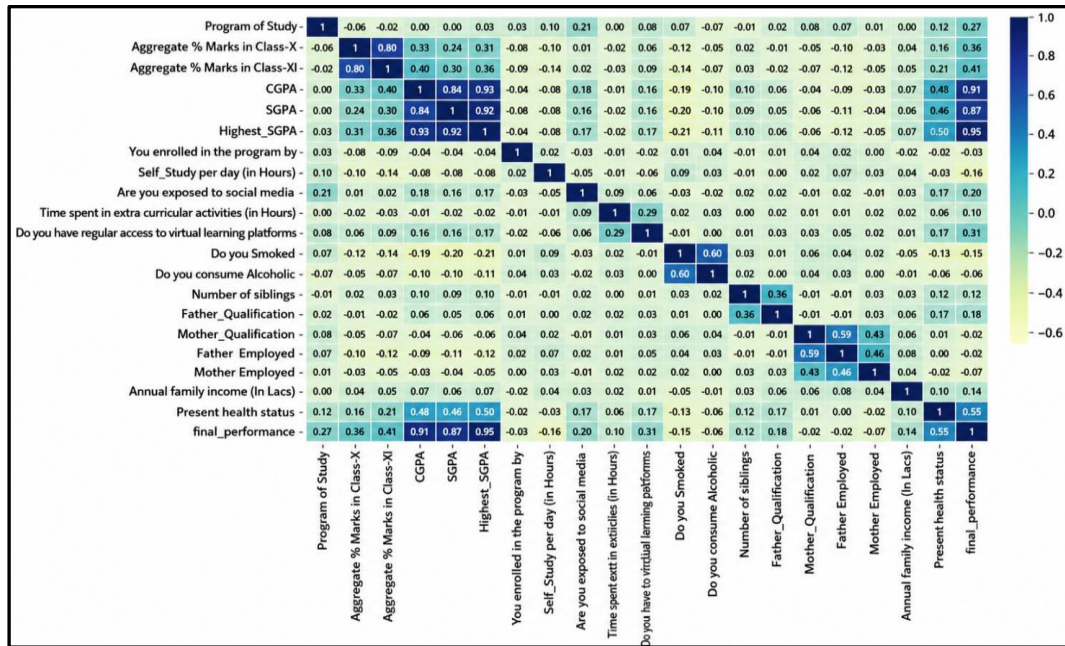


Figure 7. The correlation matrix

3.8.2. High-efficiency gated recurrent unit

Owing to its streamlined single-layer design, the GRU is both simpler and more efficient to compute than LSTM. To restore control over the information flow and address the issue of the gradient vanishing, it resets and refreshes the gates. It takes many thick layers piled on top of the GRU layer to get the result. Over the course of the hundred training iterations, the adaptive moment estimation (ADAM) optimizer, or the loss function is represented by the MSE. The fundamental design of GRU is able to successfully mimic short- and medium-term associations in sequential data. Furthermore, it converges at a faster rate.

3.8.3. Convolutional neural network

There are 64 filters in the CNN model, along with four layers: one-dimensional convolution layer (Conv1D), MaxPooling1D, Flatten, and Dense. The model also has a 3-kernel size. The max-pooling layer reduces the data's dimensionality and eliminates noise, allowing for more precise computations, while the convolution process identifies local spatial patterns in the input data. A one-dimensional vector is generated from the gathered feature maps following the Flatten layer. The last step before creating a prediction is to go through Dense layers. The model applies both ReLU activation functions that provide nonlinearity and Adam optimizer that optimizes the model efficiently and uses gradients. CNN learns sequential data both in spatial and locality and it is trained to be exposed to over 100 epochs, as it learns hierarchical representations of features. This enhances its general performance on prediction and to some extent, it is an effective methodology of modelling intricate data patterns.

3.8.4. Long short-term memory-gated recurrent unit hybrid model

A hybrid LSTM-GRU network combines the best attributes of LSTM and GRU in order to create a more powerful DL model. The model combines the good points of LSTM, which is good at capturing long-term dependencies, and GRU, which is fast and light in computation. This synergy has two aspects, increasing the model's ability to process and teaching it to detect more complex patterns in the sequential inputs. The next step is to reshape the incoming data into the format (samples, timesteps, features). Stacked layers of recurrent cells use LSTM or GRU memory or gating operations to carry the processed information from one layer to the next. The final stage is the Dense layer that makes a prediction of the output from the learned representations.

The model is composed and trained with the Adam optimizer over 100 epochs to maximize the speed of convergence and minimize the loss from the function MSE as shown in Algorithm 2. This hybrid design improves

the temporal and learning depth of the model by leveraging the strengths of LSTM, which has long-term memory, and the adaptability of the GRU framework, which has short and medium-term memory. This means that it is more stable, can make better predictions, and can be used to make predictions on a wide range of time series data.

Algorithm 2. LSTM–GRU hybrid model

```

Input : Dataset D with features X, target Y
Output : Predicted Output as  $\hat{Y}$ 
1. Import required libraries (TensorFlow/Keras, NumPy, pandas)
2. Load dataset D and preprocess:
   a. Handle missing values and categorical encoding
   b. Normalize features using MinMaxScaler or StandardScaler
3. Split the data into training and then testing sets: (X_train, X_test, Y_train, Y_test)
4. Reshape input data to 3D format (samples, timesteps, features)
5. Initialize Sequential model:
   Model=Sequential()
6. Add layers:
   a. Add LSTM(units=64, return_sequences=True, activation='relu')
   b. Add GRU(units=64, activation='relu')
   c. Add Dense(units=1, activation='linear')
7. Compile model:
   Model.compile(optimizer='adam', loss='mse', metrics=['mse'])
8. Train model:
   Model.fit(X_train, Y_train, validation_data=(X_test, Y_test),
           epochs=100, batch_size=32, verbose=1)
9. Evaluate model performance:
   Compute training MSE, validation MSE, loss, and validation loss
10. Return predicted output  $\hat{Y}$  and performance metrics

```

3.9. Proposed convolutional neural network–gated recurrent unit hybrid model

3.9.1. Model architecture

The proposed framework combines SAP, CNN, attention mechanism, and GRU to accurately predict academic performance and identify at-risk students at an early stage. SAP selects the most influential attributes, CNN extracts spatial feature patterns, attention highlights critical learning factors, and GRU captures temporal learning behavior. The final prediction layer classifies students into performance categories and generates early warnings. Figure 8 shows step by step flow diagram of our proposed framework. The well tested hybrid CNN-GRU model enhances its understanding regarding spatial and temporal data by utilizing both convolutional and recurrent network components. A Conv1D has been applied to extract local or spatial features.

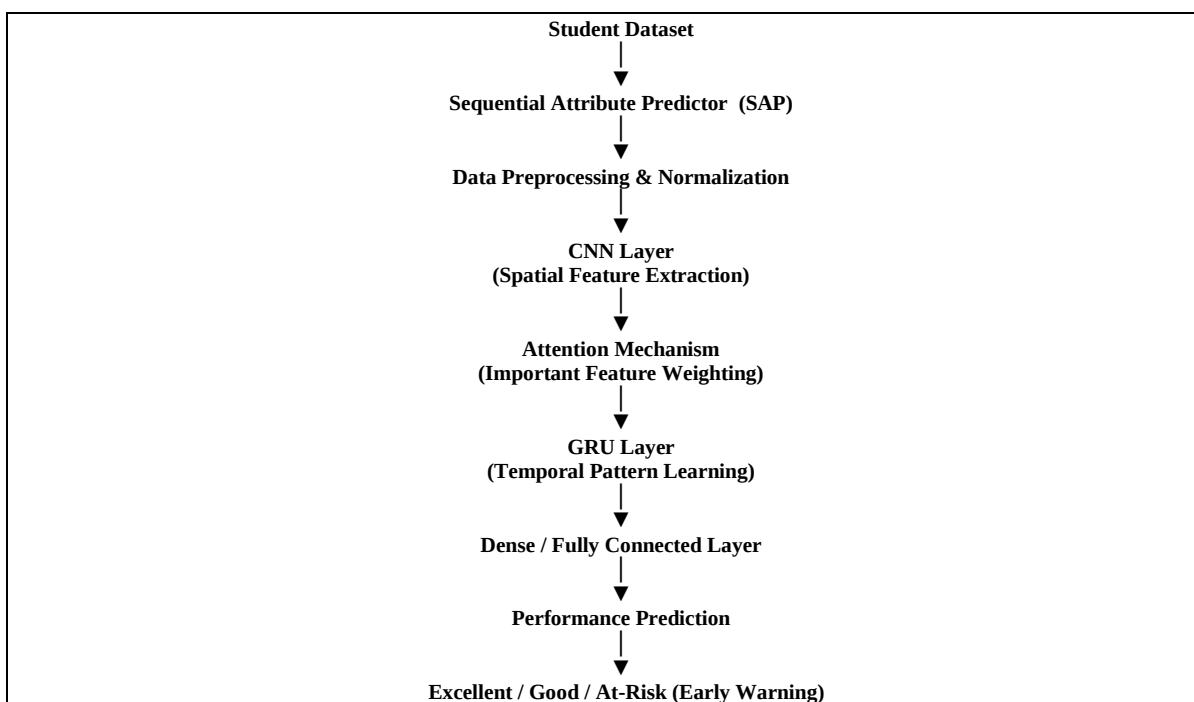


Figure 8. Flow diagram of the proposed framework and model

The extraction has been made from the input data in the first layer. In order to retain the most crucial properties while reducing the number of dimensions, the second layer employs Max Pooling1D. Next, a GRU layer receives these feature maps and uses them to train itself to understand the data's temporal dynamics and relationships.

As shown in Algorithm 3, Dense layers then use the learned representations to generate output values during the prediction stage. A less linear approach is used in the model to accelerate learning using the activation function for rectified linear unit (ReLU) in the hidden layers. Furthermore, it uses the Adam optimizer or the MSE-loss function to simplify optimization thereby reducing errors.

After 100 training epochs, the CNN-GRU hybrid was able to merge the best features of the two networks by drawing on GRU's temporal dependent learning and CNN's spatial feature analysis. The model's improved prediction accuracy and generalizability in situations involving complex sequential or time-dependent inputs are both made possible by this complementary combination.

Algorithm 3. CNN-GRU hybrid model

```

Input: Dataset D with features X, target Y
Output: Predicted output as  $\hat{Y}$ 
1. Import required libraries (TensorFlow/Keras, NumPy, pandas)
2. Load dataset D and preprocess:
   a. Handle missing values and encode categorical variables
   b. Normalize features for numerical stability
3. Split data into training and testing sets: (X_train, X_test, Y_train, Y_test)
4. Reshape input data to 3D format (samples, timesteps, features)
5. Initialize Sequential model:
   Model=Sequential()
6. Add layers:
   a. Add Conv1D(filters=64, kernel_size=3, activation='relu',
input_shape=(timesteps, features))
   b. Add MaxPooling1D(pool_size=2)
   c. Add GRU(units=64, activation='relu')
   d. Add Dense(units=1, activation='linear')
7. Compile model:
   Model.compile(optimizer='adam', loss='mse', metrics=['mse'])
8. Train model:
   Model.fit(X_train, Y_train, validation_data=(X_test, Y_test),
epochs=100, batch_size=32, verbose=1)
9. Evaluate model performance:
   Compute training MSE, validation MSE, loss, and validation loss
10. Return predicted output  $\hat{Y}$  and model evaluation results

```

3.9.2. Augmenting the attention-enhanced layer mechanism

To boost the learning ability, the attention layer of DL model has been introduced, which focuses more on the most important features discovered during training. The model is fed a subset of data variables after preprocessing and feature selection, which includes SFS using SAP. In Algorithm 4, the attention mechanism takes each of the hidden representations and passes them through a trainable Dense layer and SoftMax activation to obtain a learnable weight. The weights of the attention are used to indicate the relative importance of each feature in predicting the target variable.

Algorithm 4. The self-attention mechanism

```

1. Initialize model parameters
   a. Define input sequence X with shape (T, features)
   b. Set attention units d
2. Input Processing
   a. Receive sequential input data X
   b. Normalize input if required
3. Feature Encoding
   a. Pass X through an encoder layer (LSTM / GRU / CNN / Dense)
   b. Obtain hidden states H={h1, h2, ..., hT}
4. Attention Score Computation
   a. For each hidden state hi in H:
       Compute attention score ei using:
        $ei = f(hi \cdot W + b)$ 
       where W and b are trainable parameters
5. Attention Weight Normalization
   a. Apply softmax function to attention scores:
        $\alpha_i = \text{softmax}(ei)$ 
   b. Ensure  $\sum \alpha_i = 1$ 
6. Context Vector Formation
   a. Compute weighted sum of hidden states:
        $Z = \sum (\alpha_i \times h_i)$ 
7. Self-Attention Layer Integration

```

- a. Apply keras-self-attention layer:
 $Z = \text{SelfAttention}(H)$
8. Output Generation
 - a. Pass Z to subsequent layers (Dense / Output layer)
 - b. Generate final prediction Y
9. Model Optimization
 - a. Compute loss between predicted Y and true labels
 - b. Update parameters using backpropagation
10. End

The aggregated weighted feature representations are then used to form a context vector to represent the most important information that is required to classify the pattern. This context vector is passed on to the subsequent Dense layers to make the final prediction. The attention layer introduces the idea of highlighting critical factors, like academic scores, self-study hours, extracurriculars, and health-related factors, dynamically. The attention layer brings a novel concept: dynamically highlighting of critical factors, such as academic scores, self-study hours, extracurricular involvement, and health-related variables. This makes it easier to interpret, increases the accuracy of the predictions and improves the model's capacity to make generalizations that highlight patterns that most strongly classify student performance.

3.9.3. Comparison of CNN-GRU model with classical machine learning models

ML in the classical sense for training, models like RF, XGBoost, and SVM work faster. They are also speedier when tested on different engineered features. However, considering CNN-GRU, the architecture excels in handling raw high-dimensional sequential or spatio-temporal data, such as raw sensor inputs, audio, and image sequences, where the learning of features is crucial. Therefore CNN-GRU model outperforms other ML models.

It is important to note that Table 7 distinctly illustrates that each model possesses unique strengths based on the requirements for spatial features, temporal memory, or a combination of both. Basic models such as GRU and CNN are efficient and quick but only capture a single type of pattern, either temporal or spatial. LSTM is well suited to long term dependencies, but it also is a bit heavier on the parameters. The LSTM-GRU combination effectively represents various levels of memory for sequences, and the CNN-GRU model combines both local and temporal information. The CNN-GRU architecture is indeed the most feature-rich and computationally efficient one, as CNNs excel at capturing powerful features and GRUs excel at capturing sequence dynamics.

Table 7. Hyperparameters used in each DL model (using batch size=32, Epochs=100, learning rate=0.001, and optimiser=ADAM)

Model	Layers used	Activation function(s)	Loss function	Dropout rate	Information on additionally parameters
LSTM	2 LSTM Layers+1 Dense output layer	ReLU (hidden), linear (output)	MSE	0.2	Input shape (20, 1), return sequences equals true, for first LSTM
GRU	1 GRU Layer+1 Dense layer	ReLU (hidden), linear (output)	MSE	0.2	Input shape (20, 1), fewer parameters than LSTM
CNN	1 Conv1D+1 MaxPooling 1D+Flatten+2 Dense layers	ReLU	MSE	---	Filters equals 64, Kernel size equals 3, pool size equals 2
LSTM-GRU hybrid model	1 LSTM+1 GRU+1 Dense layer	ReLU (hidden), linear (output)	MSE	0.2	LSTM captures long-term, GRU short-term dependencies
CNN-GRU hybrid model	1 Conv1D+1 MaxPooling1D+1 Dense layer GRU+1	ReLU (hidden), linear (output)	MSE	–	Filters equals 64, Kernel equals 3, pool equals 2, GRU units equals 64

4. RESULTS AND DISCUSSION

As seen in Table 7, we used TensorFlow-Keras sequential API to develop and train all the models. We selected the Adam optimizer with a learning rate of 0.001 throughout all the trials. This was the reason we needed this stability and efficiency for gradient updating in training. In fact, we chose MSE as our loss function because it can be used to quantify and minimize the errors of all the models. With the 100-epoch training period, the models were able to stabilize and prevented overfitting due to the limitations imposed on training. To preserve the sequential data temporal dependencies, we also converted the input driven data with (samples, timesteps, and features)=(n, 20, 1) for recurrent/hybrid architecture.

4.1. Hyperparameter explanation

In this study, we used Adam optimizer with a learning rate of 0.001 and a batch size of 32 for training our model SAP-attention-CNN-GRU. The CNN part of the model was designed with 64 filters and a kernel size of 3 to effectively pull-out spatial features. Meanwhile, the GRU layer was made up of 128 units to help capture the dynamics of temporal learning. To emphasize the most important features, we also added an

attention layer to the model with 64 dimensions. For reduced overfitting, we used 0.2 dropout rate, and the model was trained for 100 epochs and had an 80:20 train-test split. For each prediction task, either MSE or cross-entropy loss was used to enhance performance.

Significance: the hyperparameters of the SAP-attention-CNN-GRU network significantly affect the model's learning process, its ability to handle complexity, and its generalization. Through the appropriate tuning, these variables can be optimized for better convergence, avoid overfitting phenomena, learning of features and, in the end, predictions of academic performance can be more accurate and reliable.

4.2. Appropriateness of MSE and loss functions for evaluating regression-based predictive models

We aimed to develop a regression-based DL system that can predict the students' performance rating continuously rather than categorizing the students into different groups. If the output variable is a numeric representation of progress, classification metrics such as F1-score, recall, precision, or accuracy are not meaningful to evaluate the model's performance. We therefore chose to take our measure to MSE, training loss or validation loss, as this is a rather regression-driven task. These can be summarized as measures of the frequency of mispredictions. The ADAM optimizer was also used to improve all the proposed models with MSE as the loss function (LSTM, GRU, CNN, and hybrid models). Reporting MSE and loss is in line with methodology and reflects the fact that the goal is to assist students in learning and thus maximize the convergence between their actual and expected results. Categorization addition measures would result in unnecessary introduction of bias and would set thresholds of continuous variables, decreasing scientific accuracy.

The overall analysis outcomes that can be obtained from these assessment tools reveal that the HNN can generate reliable and precise forecasts. This steady decrease of error and loss across different epochs emphasizes the robustness and versatility of the proposed architecture, making it a more effective predictive model than conventional architectures.

4.2.1. Predictive mean square error

One of the simplest and popular measures for assessing the performance of a model, especially in regression operations and image processing, is the MSE. This technique is used by statisticians to compute the mean of the squared differences between predicted values and observations. A positive feature of MSE is that it penalizes larger discrepancies more. When images are generated or restored, it is essential that the MSE is low; this means that the predictions are extremely accurate and the reconstruction loss is insignificant. MSE in a broad sense can be considered to be a risk function, it is said to be as the squared difference between the estimator and the true underlying parameter. However, in reality, the MSE will never be zero because there is a stochastic noise, variability of the data and a limit on the estimator in general. The non-zero values are evidence of remaining prediction errors, whether because of random oscillations or because of model restrictions that are unable to perfectly generalize these. So, the minimization of MSE is one of the main objectives of optimization.

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

4.2.2. Model loss function

The loss in predictive modeling is a way to quantify how wrong a model's prediction was by comparing the predicted outcome to the actual outcome. When the model makes bad predictions, the loss is bigger, which means that the prediction is less accurate. When the model makes good predictions, the loss is smaller, which means that the prediction is more accurate. In this sense, minimizing the loss is the most essential goal while training a model. It is through the process of iterative optimization that the model will continue to adjust internal parameters (biases and weights) to achieve the lowest possible average loss. This model is important because it ensures that the model generalizations are made between the invisible data and as well as it enhances reliability and power of the model.

$$Loss = -\frac{1}{m} \sum_{i=1}^m Y_i \cdot \log(Y_i)$$

The findings reported in Table 8 reveal that the hybrid CNN-GRU has the lowest training error and validation error out of all the models evaluated. This means that it has the best predictions of all the others. Combining all the above figures, we see a unique learning dynamic for both individual and HNN models in 100 epochs. Importantly, the MSE of the CNN-GRU model is the lowest of 0.1948, with the validation set having a respectable MSE of 0.2122. This is lower than the MSE values obtained by the LSTM, GRU, and CNN networks including the hybrid LSTM-GRU network. Combining convolutional and recurrent layers results in easier generalization, easier description of features, and more accurate predictions, as suggested by

these results. Conclusively, CNN-GRU hybrid model is much more efficient in predicting and understanding the tendency in student learning behavior.

Table 8. Comparative performance of DL models

Model description	MSE	Value of MSE	Loss	Value of loss
LSTM	1.0608	1.0927	1.0608	1.0927
GRU	0.3905	0.4897	0.3905	0.4897
CNN	0.2104	0.2412	0.2104	0.2412
Hybrid LSTM-GRU	0.4031	0.4432	0.4031	0.4432
Hybrid CNN-GRU	0.1948	0.2122	0.1948	0.2122

Figures 9 and 10, together, these figures highlight the distinct learning dynamics of individual and HNN models across 100 epochs, allowing for a comparative evaluation demonstrating which achieves better stability, faster convergence, and improved training performance. Table 9 offers a comparison between the models that have been reported earlier and the hybrid CNN-GRU model that has been developed in this study. The previous methods such as linear ridge regression (LRR) by Korchi *et al.* [34], with an error rate of 1.81, Machine learning-based models by Dessain [35], with an error rate of 0.70, and DNN by Li and Liu [36], with an error rate of 0.78, all had significantly higher error rates. The proposed SAP+attention hybrid CNN-GRU model, on the other hand, produced a significantly lower MSE of 0.0434, indicating a significant improvement in the prediction accuracy.

Epoch	Progress (Epoch/100)	Time/step	Loss	MSE	Val_Loss	Val_MSE
1/100	1%	4s 36ms/step	2.5517	2.5517	1.3688	1.3688
2/100	2%	4s 27ms/step	1.4142	1.4142	1.2888	1.2888
3/100	3%	0s 26ms/step	1.4322	1.4322	1.3075	1.3075
4/100	4%	0s 27ms/step	1.2977	1.2977	1.3188	1.3188
5/100	5%	1s 26ms/step	1.3414	1.3414	1.3153	1.3153
6/100	6%	1s 31ms/step	1.2944	1.2944	1.3389	1.3389
7/100	7%	1s 49ms/step	1.2755	1.2755	1.2702	1.2702
8/100	8%	1s 45ms/step	1.2760	1.2760	1.2851	1.2851
9/100	9%	1s 49ms/step	1.2862	1.2862	1.2884	1.2884
10/100	10%	1s 42ms/step	1.2039	1.2039	1.2954	1.2954
11/100	11%	0s 27ms/step	1.2782	1.2782	1.2812	1.2812
12/100	12%	0s 26ms/step	1.2941	1.2941	1.3389	1.3389
13/100	13%	0s 26ms/step	1.2569	1.2569	1.2446	1.2446
...	...	...	...	...	...	...
99/100	99%	1s 20ms/step	0.4615	0.4615	0.3040	0.3040
100/100	100%	1s 28ms/step	0.4031	0.4031	0.4432	0.4432

Figure 9. Training log details trace hybrid (LSTM+GRU) model progress across hundred epochs

Epoch 1/100	16/16	6s 38ms/step - loss: 3.1842 - mse: 3.1842 - val_loss: 1.4836 - val_mse: 1.4836
Epoch 2/100	16/16	0s 14ms/step - loss: 1.4406 - mse: 1.4406 - val_loss: 1.4435 - val_mse: 1.4435
Epoch 3/100	16/16	0s 16ms/step - loss: 1.3708 - mse: 1.3708 - val_loss: 1.3432 - val_mse: 1.3432
Epoch 4/100	16/16	0s 14ms/step - loss: 1.3149 - mse: 1.3149 - val_loss: 1.3356 - val_mse: 1.3356
Epoch 5/100	16/16	0s 13ms/step - loss: 1.2947 - mse: 1.2947 - val_loss: 1.3045 - val_mse: 1.3045
Epoch 6/100	16/16	0s 14ms/step - loss: 1.3413 - mse: 1.3413 - val_loss: 1.2883 - val_mse: 1.2883
Epoch 7/100	16/16	0s 14ms/step - loss: 1.3211 - mse: 1.3211 - val_loss: 1.2692 - val_mse: 1.2692
Epoch 8/100	16/16	0s 14ms/step - loss: 1.2329 - mse: 1.2329 - val_loss: 1.2568 - val_mse: 1.2568
Epoch 9/100	16/16	0s 14ms/step - loss: 1.2676 - mse: 1.2676 - val_loss: 1.2442 - val_mse: 1.2442
Epoch 10/100	16/16	0s 14ms/step - loss: 1.1998 - mse: 1.1998 - val_loss: 1.2028 - val_mse: 1.2028
Epoch 11/100	16/16	0s 14ms/step - loss: 1.2813 - mse: 1.2813 - val_loss: 1.1983 - val_mse: 1.1983
Epoch 12/100	16/16	0s 14ms/step - loss: 1.1043 - mse: 1.1043 - val_loss: 1.1546 - val_mse: 1.1546
Epoch 13/100	...	...
Epoch 99/100	16/16	0s 14ms/step - loss: 0.1842 - mse: 0.1842 - val_loss: 0.2636 - val_mse: 0.2636
Epoch 100/100	16/16	0s 16ms/step - loss: 0.1948 - mse: 0.1948 - val_loss: 0.2122 - val_mse: 0.2122

Figure 10. Training log details trace hybrid (CNN+GRU) model progress across hundred epochs

4.3. Comparative performance evaluation of deep learning models

We conduct a comprehensive comparative analysis with a range of DL architectures in order to evaluate the effectiveness of the proposed architecture. These consisted of CNN, LSTM, GRU, CNN-LSTM, CNN-GRU, and their SAP-enhanced versions. All the models were trained and tested on the same data set

while keeping the experimental conditions the same so that the models' performance evaluation can be considered fair and consistent. The performance of each model was assessed by the two widely used metrics for regression problems: MSE and loss values. The lower the MSE and loss values the better the accuracy of prediction and the generalization.

Table 9. Comparative analysis between existing and proposed model

Author	Models	Parameters
Korchii <i>et al.</i> (2023) [34]	Long-range regression	MSE=1.81
Dessain (2022) [35]	ML-based models	MSE=0.7
Li and Liu (2021) [36]	DNN	MSE=0.78
Our proposed	SAP+attention+hybrid CNN-GRU	MSE=0.0434

As can be seen in Table 10 results, the different architectures make different predictions. The MSE values for the traditional recurrent models such as LSTM and GRU were 0.2880 and 0.1081, respectively, with relatively higher prediction errors. However, the CNN model achieved a better performance on the educational dataset with an MSE of 0.1325, indicating that it was able to learn the various interactions between the features contained in the data. The hybrid LSTM-GRU model achieved a moderate performance of MSE 0.2893, indicating that the combination of all models is not necessarily the most accurate prediction model. On the other hand, our proposed model SAP+attention+hybrid CNN-GRU model outperformed others with the best MSE value of 0.0434 and minimum loss value of 0.0284.

Table 10. A comparative look at how DL models perform with the SAP+attention+hybrid CNN

Model	MSE	Val of MSE	Loss	Val of loss
LSTM	0.2880	0.3064	0.3321	0.3064
GRU	0.1081	0.2289	0.3711	0.2289
CNN	0.1325	0.1262	0.1325	0.1262
Hybrid LSTM-GRU	0.2893	0.3375	0.3048	0.3375
Hybrid CNN-GRU (proposed)	0.0434	0.0284	0.0434	0.0284

Using five regression performance metrics, Table 11 clearly presents a comparative evaluation of various DL models. It is to be noted that MSE, RMSE, MAE, and MAPE Lower values demonstrate reduced prediction errors. The higher R^2 value clearly confirms and reflects a better fit between the predicted and actual outcomes. In combination, the above shown additional metrics beside MSE provide a comprehensive assessment of the predictive accuracy and each model's robustness.

Table 11. Performance comparison of the proposed and baseline DL models using regression evaluation metrics with the SAP+attention+hybrid CNN

Model	MSE	RMSE	MAE	MAPE %	R^2
LSTM	0.2880	0.5366	0.1440	10.83	0.7758
GRU	0.1081	0.3287	0.2330	20.15	0.9159
CNN	0.1325	0.3608	0.2577	16.87	0.9699
Hybrid LSTM-GRU	0.2893	0.5379	0.3714	31.85	0.7747
Hybrid CNN-GRU (proposed)	0.0434	0.1774	0.0931	7.50	0.9746

4.3.1. Mean squared error comparison of deep learning models

Figure 11 shows a comparison of the MSE values of the DL models tested. It also shows that the higher R^2 value of the proposed model clearly confirms and reflects a better fit between the predicted and actual outcomes. MSE is a metric to show the deviation of our predictions from the actual values and the smaller the MSE, the more accurate our predictions and the more trustworthy the models are. This number allows us to make a direct comparison of the MSE values of different DL architectures. The model of SAP+attention+hybrid CNN-GRU learning is really impressive, as it recorded the smallest MSE value of 0.0434. Figure 11 also shows the MSE values of the models, where the GRU model and the LSTM model had higher MSE values of 0.1081 and 0.2880, respectively, which means that these models had larger prediction errors. Although the hybrid LSTM-GRU model had a slight improvement of 0.2893 for MSE. However, the CNN model exhibited much greater performance, with its MSE dropping to 0.1325, highlighting its effectiveness in learning intricate patterns within the field of education. The best result using the lowest MSE value of 0.0434 was obtained by the SAP+attention+hybrid CNN-GRU model among all the architectures that we evaluated.

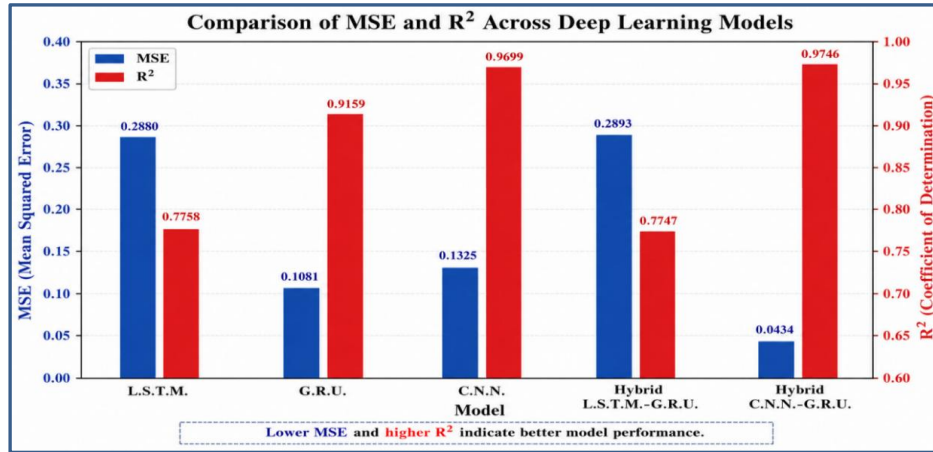


Figure 11. Comparative analysis of MSE and R^2 values across different DL architectures

Figures 12 and 13 shows the performance of the proposed LSTM-GRU and CNN-GRU models, respectively, during the training phase on the basis of the MSE and loss values. Figure 12(a) is a MSE curve which clearly illustrates how the prediction error of the model changes during the training and also clears how well the model is converging. Figure 12(b) is a loss curve showing the change in training loss across the epochs. The figure clearly provides an indication of how effectively the LSTM-GRU model learns and finally reaches a stable levelled state.

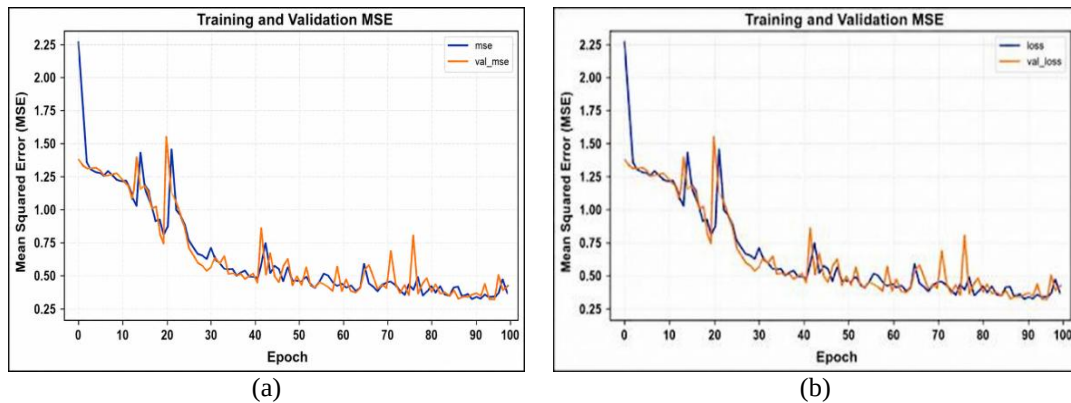


Figure 12. Training performance characteristics of the LSTM-GRU model: (a) MSE curve and (b) loss curve

The trend of decreasing MSE and loss values as the model learns more is clear in models, indicating convergence and successful training. Interestingly, CNN-GRU model is significantly better with faster convergence rate and lower error values, indicating its improved ability to recognize intricate patterns within the student performance data.

Figure 13(a) presents how the CNN-GRUs model performance changes during the training. It gives an overall view of its learning progress. Figure 13(b) illustrates the changes in loss during training and shows how smoothly the model learns and approaches the convergence.

4.4. Discussion of results and technical interpretation

As shown in the experimental results presented in Tables 7 and 8, the proposed SAP+attention+hybrid CNN-GRU model outperforms the stand-alone LSTM, GRU, CNN, and hybrid LSTM-GRU models, with the lowest MSE value (0.0434) and loss value (0.0284). The synergistic contribution of three applied techniques is responsible for this performance improvement. The LSTM model and the GRU model have relatively higher MSEs (0.2880 and 0.1081, respectively), indicating that using only recurrent architectures is not sufficient to adequately capture the complex multidimensional relationships between a variety of factors, such as academic performance, behavioral patterns, socioeconomic status, and lifestyle decisions. These models will be able to capture temporal dependencies but they are not able to capture the local feature interactions of the different attributes of students.

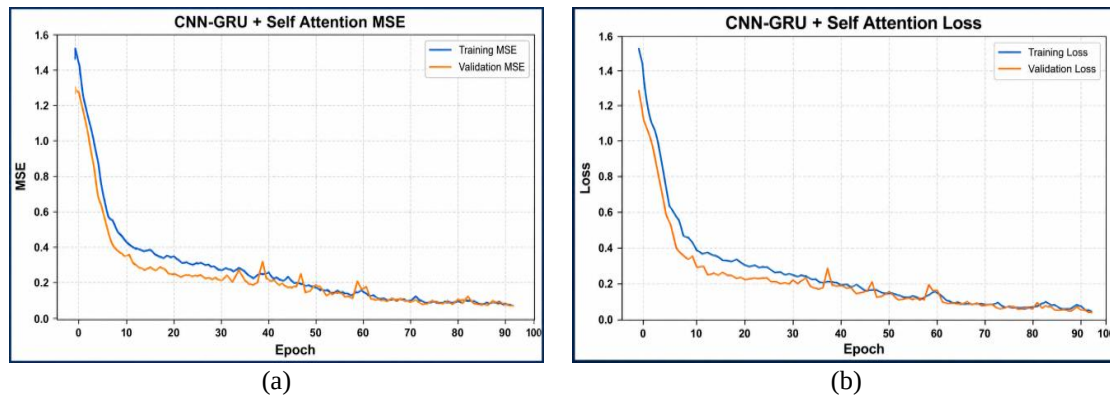


Figure 13. Training performance characteristics of the CNN-GRU model: (a) MSE curve and (b) loss curve

Also, the SAP feature adds additional enhancements. SAP ranks and selects the most impactful attributes prior to creating the model, thus reducing the impact of irrelevant and redundant variables. The attention layer is designed to focus the model home on the most relevant academic and behavioral indicators for each prediction, rather than taking all the same. The training MSE and loss has been decreased from 0.1948 to 0.0434 which is a substantial and appreciable 77.7% reduction, while the validation MSE and validation loss has been decreased from 0.2122 to 0.0284, which is also a favorable 86.6% reduction confirming the accuracy and strength of our proposed model. This truly demonstrates the power of having feature prioritization, attention, and HDL in a single architecture.

4.5. Early academic risk identification and promotion of a socially sustainable education system

Referring carefully to Tables 10 and 11, the hybrid CNN-GRU model was found to be better than the other models. It is an extremely practical model. The predictions are more accurate because it is able to follow complex patterns over time; this means that higher education systems can intervene and provide support earlier. It also aids in the development of improved educational systems to cater for the individual student, maintaining engagement and reducing drop-out rates. In summary, this model assists Colleges and Universities with making decisions based on actual data, so that they can effectively promote student success in an early manner rather than merely taking a reactive approach after a student is already behind.

4.6. Summary of key findings

Our research is on making regression predictions for continuous student performance ratings, but in the case of DL, it is possible to use multiple evaluation measures. Hence, it is not suitable to use classification-based measures such as recall, precision, accuracy, and F1 score for this investigation. The model-based metrics, model loss and MSE are beneficial since they will measure the difference between the expected and observed values, and are congruent with the objective of optimization for the Adam-MSE training method used for all of the models. There is always a risk of bias and a risk of altering the performance of the models if new criteria were added, as this would involve converting continuous scores into classes.

Therefore, the best and most scientifically correct method to measure the accuracy of this regression task is to closely examine the MSE and loss. In addition, we have included sequential attribute/feature selection as provided by the SAP approach, which is one of the most effective dimensionality reduction and predictive modeling efficiency methods. In our model, we do include an attention layer so that our model will not give equal attention to all features. Instead, it smartly emphasizes important factors like academic scores, hours spent studying independently, involvement in extracurricular activities, and health-related aspects.

4.7. Generalization and limitations of the proposed model

Generalization is considered important because it shows whether a model is able to make accurate predictions on new, unseen student data, making its results more acceptable and reliable in the real-world educational scenario. SAP-attention-CNN-GRU model truly excels in generalization tasks. It works really well at recognizing fine-grained spatial and temporal relationships in educational data. The hybrid design allows it to easily incorporate different datasets of students across different educational contexts, which makes it a reliable approach to predicting academic success and identifying students who may be in danger early on.

Limitations show where the model is likely to face challenges or may not be able to perform equally well in every situation. Understanding these limitations clearly assists to explain the results clearly and identify the crucial areas for future improvement. Therefore, the model's success largely depends on the quality and

completeness of the input data. It also demands greater computational power and longer training periods when compared to conventional ML methods. Also, if you use it on data sets that are very different from the ones it was trained on, you may experience a drop in performance, and you may need to retrain or tune the hyperparameters to achieve optimal results.

5. CONCLUSION

This research was conducted to analyze the performance of DL and HNN architectures for academic performance prediction of students. Various models such as LSTM, GRU, CNN, hybrid LSTM-GRU, and hybrid CNN-GRU models were tested and compared. The results showed that the proposed model SAP and attention-enabled hybrid CNN-GRU model presented excellent predictive performance, the lowest training MSE (0.0434) and the lowest validation MSE (0.0284) among all models evaluated. By applying SAP+attention mechanism to hybrid CNNGRU model, the training MSE and loss has been decreased from 0.1948 to 0.0434 which is a substantial and appreciable 77.7% reduction, while the validation MSE and validation loss has been decreased from 0.2122 to 0.0284, which is also a favourable 86.6% reduction confirming the novelty of our research and the accuracy and strength of proposed model. A comparatively higher coefficient of determination value ($R^2=0.9746$) of the proposed model clearly confirms and reflects a better fit between the predicted and actual outcomes.

The primary novelty of this work is the design of an innovative framework, named as SAP-Attention-CNN-GRU. The framework is based on SAP, CNN is used to extract features, attention mechanism is used to weight features and GRU is used to learn over time. The hybrid architecture effectively combines these features, making it more accurate in predicting the intricate spatial-temporal relationships embedded in educational data while being more robust than traditional and hybrid baseline architectures.

The framework offered is a great tool to identify students who could be at risk of not succeeding in Academics early on, and then provide early intervention and tailor learning. This added value in predictive power helps educators, administrators and policy makers make informed decisions based on data and can impact student retention, student success, and overall effectiveness speciall in higher education system. Moreover, the study underscores the exciting possibilities that advanced DL techniques bring to the development of smart educational analytics systems.

Future studies aim to delve into incorporating adpative SAP, BiGRU, and MutiHead attention methods to further boost the model's accuracy. Further, incorporating transformer-based architectures, multi-modal learning techniques and emotional signals, can further enhance the correctness of predictions, adaptability, and real-world applicability across diverse learning ecosystems.

ACKNOWLEDGMENTS

Our thanks to all those colleagues who provided us motivation and the timely resources to carry out this research work in a result-oriented manner.

FUNDING INFORMATION

Authors state no funding involved

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Taneja Sanjay	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Devkishan														
Sanjay Kumar Singh		✓				✓		✓	✓	✓	✓	✓		
Ajay Kumar Bharti	✓		✓	✓			✓			✓	✓	✓	✓	✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : **O** : Writing - **O**riginal Draft

E : **E** : Writing - Review & **E**ding

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article [and/or its supplementary materials]. The data that support the findings of this study will also be made available from the corresponding author, upon reasonable request.

REFERENCES

- [1] W. Gonzales, Z. Cordero, C. D. Abanto-Ramírez, E. T. S. Ramírez, H. Iftikhar, and J. L. López-Gonzales, "A hybrid AI approach for predicting academic performance in RBE students," *Frontiers in Artificial Intelligence*, vol. 8, 2025, doi: 10.3389/frai.2025.1651100.
- [2] K. O. Adefemi and M. B. Mutanga, "A robust hybrid CNN–LSTM model for predicting student academic performance," *Digital*, vol. 5, no. 2, May 2025, doi: 10.3390/digital5020016.
- [3] A. S. K. AL-Hurdi and N. M. A. Munassar, "Enhancing students' academic performance classification in e-learning using hybrid model (random forest and deep neural network)," *Journal of Science and Technology*, vol. 30, no. 7, pp. 86–98, Jun. 2025, doi: 10.20428/jst.v30i7.2948.
- [4] K. O. Adefemi, M. B. Mutanga, and V. Jugoo, "Hybrid deep learning models for predicting student academic performance," *Mathematical and Computational Applications*, vol. 30, no. 3, May 2025, doi: 10.3390/mca30030059.
- [5] G. Al-Tameemi, J. Xue, I. H. Ali, and S. Ajit, "A hybrid machine learning approach for predicting student performance using multi-class educational datasets," *Procedia Computer Science*, vol. 238, pp. 888–895, 2024, doi: 10.1016/j.procs.2024.06.108.
- [6] A. A. Alharbi and J. Allohbi, "A new hybrid classification algorithm for predicting student performance," *AIMS Mathematics*, vol. 9, no. 7, pp. 18308–18323, 2024, doi: 10.3934/math.2024893.
- [7] M. Viveka and D. N. S. Priya, "An efficient hybrid deep learning framework for predicting student academic performance," in *Salud, Ciencia y Tecnología - Serie de Conferencias*, vol. 3, Jan. 2024, doi: 10.56294/sctconf2024759.
- [8] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced intrusion detection with LSTM-based model, feature selection, and SMOTE for imbalanced data," *Applied Sciences*, vol. 14, no. 2, Jan. 2024, doi: 10.3390/app14020479.
- [9] A. Kala, O. Torkul, T. T. Yildiz, and I. H. Selvi, "Early prediction of student performance in face-to-face education environments: A hybrid deep learning approach with XAI techniques," *IEEE Access*, vol. 12, pp. 191635–191649, 2024.
- [10] R. Parkavi, P. Karthikeyan, and A. S. Abdullah, "Predicting academic performance of learners with the three domains of learning data using neuro-fuzzy model and machine learning algorithms," *Journal of Engineering Research*, vol. 12, no. 3, pp. 397–411, Sep. 2024, doi: 10.1016/j.jer.2023.09.006.
- [11] S. Umirzakova, M. Abdullaev, S. Mardieva, N. Latipova, and S. Muksimova, "Simplified knowledge distillation for deep neural networks bridging the performance gap with a novel teacher–student architecture," *Electronics*, vol. 13, no. 22, Nov. 2024, doi: 10.3390/electronics13224530.
- [12] L. Vives *et al.*, "Prediction of students' academic performance in the programming fundamentals course using long short-term memory neural networks," *IEEE Access*, vol. 12, pp. 5882–5898, 2024, doi: 10.1109/ACCESS.2024.3350169.
- [13] X. Wen and H. Juan, "Early prediction of students' performance using a deep neural network based on online learning activity sequence," *Applied Sciences*, vol. 13, no. 15, Aug. 2023, doi: 10.3390/app13158933.
- [14] H. Chavez, B. Chavez-Arias, S. Contreras-Rosas, J. M. Alvarez-Rodríguez, and C. Raymundo, "Artificial neural network model to predict student performance using nonpersonal information," *Frontiers in Education*, vol. 8, Feb. 2023, doi: 10.3389/educ.2023.1106679.
- [15] B. Holicza and A. Kiss, "Predicting and comparing students' online and offline academic performance using machine learning algorithms," *Behavioral Sciences*, vol. 13, no. 4, Mar. 2023, doi: 10.3390/bs13040289.
- [16] B. Venkatachalam and K. Sivanraju, "Predicting student performance using mental health and linguistic attributes with deep learning," *Revue d'Intelligence Artificielle*, vol. 37, no. 4, pp. 889–899, Aug. 2023, doi: 10.18280/ria.370408.
- [17] N. Abuzinadah *et al.*, "Role of convolutional features and machine learning for predicting student academic performance from MOODLE data," *PLOS ONE*, vol. 18, no. 11, Nov. 2023, doi: 10.1371/journal.pone.0293061.
- [18] A. S. Aljaloud *et al.*, "A deep learning model to predict student learning outcomes in LMS using CNN and LSTM," *IEEE Access*, vol. 10, pp. 85255–85265, 2022, doi: 10.1109/ACCESS.2022.3196784.
- [19] Y. Baashar *et al.*, "Toward predicting student's academic performance using artificial neural networks (ANNs)," *Applied Sciences*, vol. 12, no. 3, Jan. 2022, doi: 10.3390/app12031289.
- [20] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: Hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022, doi: 10.1109/ACCESS.2022.3206425.
- [21] S. Poudyal, M. J. Mohammadi-Aragh, and J. E. Ball, "Prediction of student academic performance using a hybrid 2D CNN model," *Electronics*, vol. 11, no. 7, Mar. 2022, doi: 10.3390/electronics11071005.
- [22] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learning Environments*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40561-022-00192-z.
- [23] X. Niu, "Deep-learning-guided student intelligent classroom management system," *Applied Bionics and Biomechanics*, vol. 2022, pp. 1–6, Aug. 2022, doi: 10.1155/2022/1961631.
- [24] M. Kamal *et al.*, "Metaheuristics method for classification and prediction of student performance using machine learning predictors," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–5, Jul. 2022, doi: 10.1155/2022/2581951.
- [25] Y. Tan, "Application research on face image evaluation algorithm of deep learning mobile terminal for student check-in management," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, Aug. 2022, doi: 10.1155/2022/3961910.
- [26] S. Sarwat *et al.*, "Predicting students' academic performance with conditional generative adversarial network and deep SVM," *Sensors*, vol. 22, no. 13, Jun. 2022, doi: 10.3390/s22134834.
- [27] Y. Zhang, T. Liu, and B. Ru, "Effect evaluation and student behavior design method of moral education in colleges and universities under the environment of deep learning," *Scientific Programming*, vol. 2022, pp. 1–9, Aug. 2022, doi: 10.1155/2022/5779130.
- [28] J. Waldeyer *et al.*, "A moderated mediation analysis of conscientiousness, time management strategies, effort regulation strategies,

- and university students' performance," *Learning and Individual Differences*, vol. 100, Dec. 2022, doi: 10.1016/j.lindif.2022.102228.
- [29] T. T. Mai, M. Bezbradica, and M. Crane, "Learning behaviours data in programming education: Community analysis and outcome prediction with cleaned data," *Future Generation Computer Systems*, vol. 127, pp. 42–55, Feb. 2022, doi: 10.1016/j.future.2021.08.026.
- [30] R. Alshabandar, A. Hussain, R. Keight, and W. Khan, "Students performance prediction in online courses using machine learning algorithms," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Glasgow, UK, pp. 1–7, Jul. 2020, doi:10.1109/IJCNN48605.2020.9207196.
- [31] F. D. Pereira, E. H. T. de Oliveira, D. Fernandes, and A. Cristea, "Early performance prediction for CS1 course students using a combination of machine learning and an evolutionary algorithm," in *2019 IEEE 19th International Conference on Advanced Learning Technologies (ICALT)*, Maceio, Brazil, pp. 183–184, Jul. 2019, doi:10.1109/ICALT.2019.00066.
- [32] M. R. Rimadana, S. S. Kusumawardani, P. I. Santosa, and M. S. F. Erwianda, "Predicting student academic performance using machine learning and time management skill data," in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, Dec. 2019, pp. 511–515, doi: 10.1109/ISRITI48646.2019.9034585.
- [33] 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, Dec. 2019, doi: 10.1186/s41239-019-0160-3.
- [34] A. Korchi, F. Messaoudi, A. Abatal, and Y. Manzali, "Machine learning and deep learning-based students' grade prediction," *Operations Research Forum*, vol. 4, no. 4, Oct. 2023, doi: 10.1007/s43069-023-00267-8.
- [35] J. Dessain, "Machine learning models predicting returns: Why most popular performance metrics are misleading and proposal for an efficient metric," *Expert Systems with Applications*, vol. 199, Aug. 2022, doi: 10.1016/j.eswa.2022.116970.
- [36] S. Li and T. Liu, "Performance prediction for higher education students using deep learning," *Complexity*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/9958203.

BIOGRAPHIES OF AUTHORS



Taneja Sanjay Devkishan    holds M.C.A., M.Tech. (IT) degree from Prestigious Indian Universities. He is an assistant professor and a research scholar working with Amity University Lucknow, India, since 2007, authored several articles in leading journals and IEEE conference proceedings. He can be contacted at email: sdtaneja@lko.amity.edu.



Dr. Sanjay Kumar Singh    working as the Deputy Director and Professor at Amity University, Lucknow, India, holds a Ph.D. in "Computer-Oriented Analysis and Computational Model Development for the Dynamic Behavior of the Universe." With over three decades of teaching experience, he has authored more than 85 research papers published in Scopus-listed and other reputed indexed journals. He can be contacted at email: sksingh1@amity.edu.



Dr. Ajay Kumar Bharti    is working as a Professor at Ambalika Institute of Management and Technology, Lucknow, India. He has brought in over 21 years of extensive experience in research and academia and has more than 50 publications in reputed journals and conference proceedings. He can be contacted at email: ajaykbharti@gmail.com.