INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Enhanced Student Performance Prediction Using Data Augmentation with Ensemble Generative Adversarial Network

Bakyalakshmi Venkatachalam[1]*, **Kanchana Sivanraju**[2]

1 Department of Computer Science, PSG College of Arts and Science, Coimbatore, 641014, India
2 Department of Software Systems, PSG College of Arts and Science, Coimbatore, 641014, Tamil Nadu, India

OPEN ACCESS

Received: 02-09-2024
Accepted: 24-10-2024
Published: 02-12-2024

Citation: Venkatachalam B, Sivanraju K (2024) Enhanced Student Performance Prediction Using Data Augmentation with Ensemble Generative Adversarial Network. Indian Journal of Science and Technology 17(44): 4619-4632. https://doi.org/10.17485/IJST/v17i44.2833

* **Corresponding author**.

vbakyalakshmiphd@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: To solve the issue of inadequate student data and increase the precision of performance prediction. **Methods:** This paper presents an innovative Ensemble Generative Adversarial Network (EGAN) model to augment student data and improve the accuracy of predicting student performance. The EGAN integrates two different GAN models: (i) Divergence GAN (DivGAN) and Success-aware GAN (SucGAN). The DivGAN reduces the difference between latent variables and observed data. The SucGAN uses Gumbel-Softmax Relaxation (GSR) to approximate the categorical distribution for creating high-quality data, which solves the imbalance ratio between raw and synthetic data in multiple classes (e.g., Pass, Fail, Distinction, and High Distinction). In contrast, only concentrating on data quality leads to mode collapse issue. So, a Tuna Swarm Optimization (TSO) is employed to stabilize the SucGAN training process and ensure a balance between data quality and diversity while training SucGAN. Thus, the synthetic student data is generated by EGAN and combined with the original dataset to form an augmented dataset. Moreover, this dataset is used by the Student Accomplishment prediction model using the Distinctive Deep Learning (SADDL) model to accurately predict student performance. **Findings:** The experimental findings illustrate that the EGAN-SADDL attains 94.71% accuracy and 0.0529 Root Mean Square Error (RMSE) in predicting student performance compared to the existing DL models. **Novelty:** This work expands the dataset by creating synthetic student data using EGAN. By increasing the amount of data available for training the SADDL model, this data augmentation strategy also boosts the model's accuracy and generalization abilities.

Keywords: Student's performance prediction; SADDL; GAN; Divergence; Tuna swarm optimizer; Gumbel-Softmax relaxation

1 Introduction

Learning performance is a crucial measure of academic success, as low grades can lead to stress, despair, and harmful behavior. Students may face obstacles such as mental health issues, family or social problems, or a lack of guidance that causes them to miss classes [1]. As a result, their academic progress is at risk, and educational institutions must identify early. Educational Data Mining (EDM) [2] has emerged as a vital field of study, aiming to predict students' learning performance and prevent negative outcomes like low grades or dropouts. Identifying learners with poor grades early on allows the investigation to improve their performance and ensure their success. High school and college students are particularly suitable for this research field. High school students, in particular, are an ideal target group because their academic success greatly impacts their future educational prospects. By analyzing a dataset that includes socioeconomic and educational information about high school students, it is possible to identify students with poor performance. Establishing specific criteria can help identify students likely to fail in the final semester. However, assessing the school performance of all learners is difficult for teachers and educational institutions because of the large number of learners and limited resources available.

Numerous scholars in the literature have proposed various DL algorithms as a solution to this problem. Fields such as identifying at-risk students, predicting final test results, forecasting placement rates, and detecting early failures have widely used these algorithms [3]. Liu et al. [4] developed a new student performance prediction model based on the belief rule base with automated construction. They used the IF-THEN rule-based model with evidential reasoning. The parameters were adjusted using the Projected Covariance Matrix Adaptive Evolution Strategy (P-CMA-ES). They also utilized the Akaike Information Criterion (AIC) to balance the accuracy and complexity of the model. Vives et al. [5] suggested the Long Short-Term Memory (LSTM) network for student performance prediction. Fazil et al. [6] developed a new Attention-aware Stacked convolutional Stacked Bidirectional LSTM (ASIST) model to predict student's academic performance. Baniata et al. [7] used the Gated Recurrent Unit (GRU) network to predict students' academic performance. Kukkar et al. [8] developed a new model using Recurrent Neural Network (RNN) and LSTM with Random Forest (RF) to predict student's academic performance from Open University Learning Analytics Dataset (OULAD) and self-generated emotional dataset. Liu et al. [9] developed a student performance prediction framework called the MCAG, which relied on the Maximum Information Coefficient (MIC), CNN, attention strategy, and GRU. The MIC was used to determine the correlation between influencing factors and student performance. The CNN-attention model was then used to extract high-dimensional features and assign attention weights to different features. The GRU was used to capture temporal features. Finally, features from both CNN and GRU were combined to predict student performance.

However, most existing models often focus on academic and demographic factors for student performance prediction. It is necessary to explore the relationship between student's mental health and academic performance by extracting the attributes from their posts or comments on social media platforms. As a result, the SADDL model [10] has been developed, which uses various student attributes, including physiological, demographic, and academic factors. In this model, student data was collected from Twitter, Facebook, and Instagram API services and preprocessed using Natural Language Processing (NLP) techniques. Linguistic Inquiry and Word Count (LIWC) and Latent Dirichlet Allocation (LDA) schemes were used for feature extraction related to mental health and attitude variations. Such qualities, which are common aspects related to student performance, were fed to the DDL, which combines LSTM, DCNN, and Multi-Layer Perceptron (MLP) algorithms. The LSTM extracts time series attributes, which were transformed into a feature tensor and inputted into the multidimensional DCNN to obtain attribute correlations. The temporal, correlation, and student demographic characteristics were merged to get a unified attribute vector. Moreover, the MLP classifier predicts students' academic performance.

1.1 Research Gap

Existing models realized low accuracy in predicting student performance owing to inadequate data regarding student's academics, demographics, and emotions. This limitation hinders the development of robust predictive models as they necessitate massive amounts of data. Gathering more data to solve this issue was tedious and time-consuming since mostly performed by manual entry, data cleaning, and validation methods. In this regard, this study focuses on generating a large volume of student data using a deep adversarial model. The idea behind it would be that the deep adversarial model can create synthetic data, hence augmenting the dataset available. It can learn student data's intricate features, including the complex dependencies between academic, demographic, and social media attributes. This model provides an advanced approach to data augmentation as compared with conventional methods.

1.2 Major Contributions

This article proposes a new EGAN-SADDL model for predicting student performance. The major contributions of this study are the following:

1. First, the EGAN model is developed by combining two distinct GANs to augment student data.

- The first GAN, called DivGAN, aims to minimize the divergence between a limited subgroup of latent elements and observed data. The Jensen-Shannon (JS) divergence is used to determine the divergence in this GAN. By generating samples that minimize divergence, DivGAN can learn disentangled and interpretable representations. This allows for assessing the impact of varying specific factors on the generated data.
- The second GAN, called SucGAN is a success-aware data generation framework. It calculates the imbalance ratio between raw and synthetic data for all classes and then minimizes the difference gap to create high-quality data. For the case of discrete data, it uses the GSR technique to avoid complex learning techniques in updating SucGAN. Generally, direct sampling from a discrete distribution is non-differentiable in conventional GANs, making it complicated for the generator to train successfully. This problem can be solved by the GSR, which provides a smooth, differentiable approximation to the sampling task. So, the SucGAN can balance the number of data instances related to multiple classes in the generated data. Moreover, TSO is used to prevent the mode collapse issue in this GAN. In training a SucGAN, TSO stabilizes the learning task and permits a trade-off between data quality and diversity.
- In the ensemble model, both GANs are integrated to create and augment student data.

2. Then, the augmented dataset is given to the SADDL to predict student performance. It identifies at-risk students in early phases, thus preventing students from failing grades.

The below sections are structured as follows: Section 2 explains the EGAN-SADDL model, and Section 3 evaluates its efficacy with previous models. Section 4 abridges the findings with possible improvements.

2 Methodology

This section details the EGAN-SADDL for predicting student performance, which involves the steps illustrated in Figure 1. The initial step is data gathering and preprocessing. The second step uses the EGAN for data augmentation. The augmented dataset is divided into training and test sets. Moreover, physiological features of both training and test sets are extracted using the LDA algorithm. Then, all the factors like academic, demographic, and physiological features are learned by the DDL model. Afterward, the trained DDL model predicts student performance as Distinction, Fail, High Distinction, and Pass. Finally, those predicted outcomes are considered to evaluate the efficacy of the model.

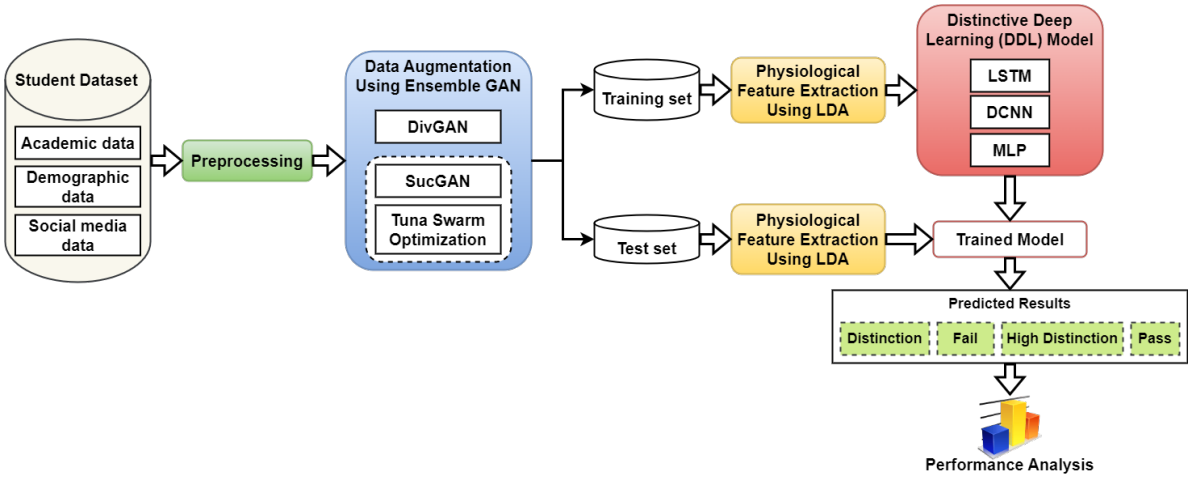

Fig 1. Proposed Model for Student Performance Prediction

2.1 Student Dataset Collection and Preprocessing

A dataset of 50,000 instances (12500 instances for each class) is created by gathering student records from government and self-financed engineering colleges in Coimbatore, Tamil Nadu, from June 2022 to December 2022. The dataset has 39 attributes, including students' names, age, gender, courses, college category (medicinal/engineering and government/self-financed), location, family category (nuclear/joint), family aspects such as career and academic skills of family members, time spent watching TV, college aspects, economic aspects, social aspects, electronic gadgets, individual aspects, and educational aspects. Locality, for instance, pertains to the geographical position of students' residences, educational institutions, and universities, regardless of whether they are in a rural, urban, or semi-urban region. Educational resources such as lecture notes and books, instructional techniques, class sizes, regulations on smartphone usage, and other related aspects are integral components of university life. Social factors include parental supervision for homework, social circle size, and academic achievement.

Furthermore, an additional dataset is created that comprises diverse information shared by students regarding their academic skills on Twitter, Facebook, and Instagram. This dataset comprises 50,000 posts/comments (12500 for each class). Various NLP techniques in [10] are applied to preprocess this dataset to eliminate redundant information. Such methods include converting text to lowercase, expanding contractions, deleting stopwords with an NLTK dictionary, removing punctuation, emoticons, unique typescripts, superfluous spacing, and numeric values. They also involve lemmatizing words with WordNet to their root forms. After preprocessing, the suggested EGAN augments both datasets, as described in the below section.

2.2 Student Dataset Augmentation with Ensemble Generative Adversarial Network

The augmentation of a student dataset is accomplished by the ensemble GANs. In GAN, the generator $G(z)$ creates artificial data by sampling from a latent distribution $z \sim Z$. An ensemble of T number of GANs is denoted by $\widehat{G}_T = \sum_{t=1}^{T} p_t G_t$, where p_t denotes weights for all GANs G_t, with the restraint $\sum_t p_t = 1$. To create data from $\widehat{G}_T$, each instance is obtained from G_t with a chance p_t, resulting in a dataset with $p_t N_g$ instances from G_t, where N_g refers to the sum of artificial instances.

For classification, where labels $k \in [1, K]$ are exist, bagging is applied by learning each GAN for all labels across each iteration. This certifies that the GANs include various segments of the data distribution and maximize mode coverage. In all iterations, K GANs $G_{(t,k)}$ are learned, resulting in an overall ensemble $\widehat{G}_T = \frac{1}{K}\sum_{k=1}^{K}\sum_{t=1}^{T} p_t G_{(t,k)}$. Accordingly, an absolute ensemble has KT GANs, but for ease, this study focuses on the iterations T.

This paper concentrates on autonomous ensembles, where all GANs, represented G_t, is learned independently. The benefit of employing several GANs comes from their stochastic nature, which permits all GANs to concentrate on various perspectives of the data distribution. All GANs G_t are allocated to a similar weight $p_t = \frac{1}{T}$ in the ensemble, confirming that all GANs contribute alike to the created data.

To maintain consistency with the actual dataset size, the number of synthetic instances created is kept constant. For example, if the real dataset contains N instances, then each GAN G_t in an ensemble of T generators will produce $\frac{N}{T}$ instances. This confirms that the sum of artificial instances from the ensemble equals the sum of actual instances. This study utilizes 2 kinds of GANs in the ensemble (i.e., $T = 2$): DivGAN and SucGAN, as portrayed in Figure 2. Each kind of GAN has unique features, which are explained in the following sections.

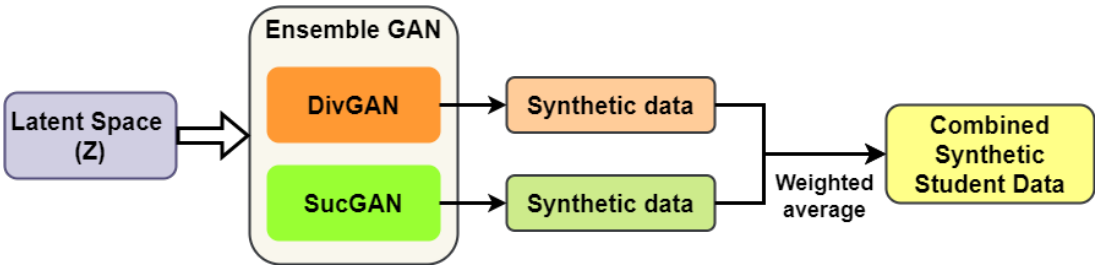

Fig 2. Structure of Ensemble GAN Model

Typically, GAN aims to train a generator G to create synthetic data that counterparts the distribution of actual data $P_{data}(x)$, i.e., G learns the distribution $P_G(x)$ such that it approximates $P_{data}(x)$. A basic GAN consists of a generator G and discriminator D. G creates synthetic data $G(z)$ from a noise variable z sampled from a noise distribution $P_{noise}(z)$. D evaluates whether a given sample comes from the real data distribution $P_{data}(x)$ or the synthetic distribution $P_G(x)$. The D's task is to distinguish between real and fake data.

The GAN uses a minimax game, where G tries to maximize D's error while D tries to minimize it. The objective function is defined as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim P_{data}}[logD(x)] - E_{z\sim P_{noise}}[log(1-D(G(z)))] \quad (1)$$

In Equation (1), E represents the probability, and $D(x)$ is the chance that D allocates to the instance x being actual.

2.2.1 Design of DivGAN

Conventional GANs utilize a particular latent variable z that guides G in a complex way, creating it difficult to extract significant characteristics. Rather, this study decomposes z into 2 portions:

i. Incompressible noise z: It is a random noise that can assist in creating diverse data.

ii. Latent code c: It signifies significant semantic features of $P_{data}(x)$. The latent code c comprises multiple restrained latent elements $c_1, \ldots, c_L$.

Here, $c_1, \ldots, c_L$ are considered to follow a factored distribution:

$$P(c_1, \ldots, c_L) = \prod_{i=1}^{L} P(c_i) \quad (2)$$

This facilitates to model of the data distribution efficiently by integrating semantic traits. G is adapted to consider both z and c as inputs, signified by $G(z,c)$. It assists G to use c appropriately that encodes significant characteristics.

Mutual Information (MI)

MI determines how much recognizing one variable Y minimizes the uncertainty regarding the other variable X. It is computed by

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \quad (3)$$

In Equation (3), H is entropy. High MI signifies that X and Y are strictly linked. To confirm that c is efficiently utilized by G and the created data holds the traits of c. This is accomplished by increasing the MI between c and $G(z,c)$. Thus, the adapted objective function has a regularization term λ that penalizes low MI between c and $G(z,c)$:

$$\min_{G}\max_{D} V_I(D,G) = V(D,G) - \lambda I(c;G(z,c)) \quad (4)$$

In Equation (4), $V(D,G)$ is the actual value function from Equation (1), $V_I(D,G)$ is the adapted value function, and $I(c;G(z,c))$ determines the MI between c and $G(z,c)$. To combine the MI term into the DivGAN efficiently, a variational scheme is employed. Directly maximizing MI $I(c;G(z,c))$ is difficult as it necessitates access to the posterior distribution $P(c|x)$, which is not obtainable in training. As an alternative, a variational lower bound is utilized to approximate this MI.

An auxiliary distribution $Q(c|x)$ is defined to approximate the true posterior distribution $P(c|x)$. It supports approximating the MI term $I(c;G(z,c))$. A variational lower bound $L_I(G,Q)$ of the MI is determined by

$$L_I(G,Q) = E_{x\sim G(z,c)}\left[E_{c'\sim P(c|x)}\left[\log Q\left(c'|x\right)\right]\right] + H(c) \quad (5)$$

Equation (5) approximates the MI between c and $G(z,c)$ by the variational distribution $Q(c|x)$. $H(c)$ is the entropy of c. The DivGAN integrates this $L_I(G,Q)$ into the standard GAN objective function. Therefore, the total objective function for DivGAN is defined by

$$\min_{G,Q}\max_{D} V_{DivGAN}(D,G,Q) = V(D,G) - \lambda L_I(G,Q) \quad (6)$$

The DivGAN architecture is portrayed in Figure 3.

According to the probability distributions, MI is defined by the JS divergence, which is a symmetrized and smoothed type of the KL divergence. It is utilized to determine the variance between probability distributions P and Q as:

$$D_{JS}(P(Q) = \frac{1}{2}D_{KL}\left(P \parallel \frac{P+Q}{2}\right) + \frac{1}{2}D_{KL}\left(Q \parallel \frac{P+Q}{2}\right)$$

$$= \frac{1}{2}\int_R p(x)\,log\left(\frac{2p(x)}{p(x)+q(x)}\right) + q(x)\,log\left(\frac{2q(x)}{p(x)+q(x)}\right) + d\lambda(x) \quad (7)$$

So, the JS divergence in DivGAN determines the variance between the created and actual data distributions (i.e., called MI), enhancing alignment and preserving semantic characteristics in the created data. This tradeoff between diversity and fidelity confirms that the created data is both varied and representative, enabling the DivGAN a powerful model to create accurate and relevant data, specifically in student data augmentation.

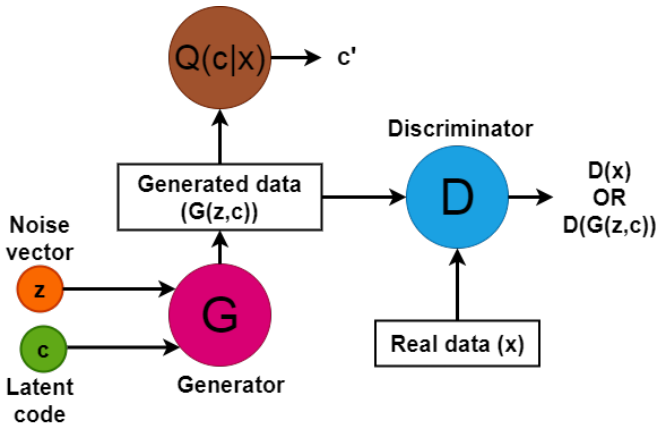

Fig 3. Structure of DivGAN Model

2.2.2 Design of SucGAN

The SucGAN is developed to generate class-specific data. Consider a dataset with k classes. All classes denote a particular tag or kind of information to be created. For e.g., in an academic theme, such classes can define multiple topics. The generator G_θ, parameterized by θ, can produce the text information. If information with a particular class c is required for generation, then G_θ is learned to create a series of text information $Y_\theta^c = (y_1, \ldots, y_t, \ldots, y_T)$. Every y_t in the series refers to a token (like a word or term) that is part of the created text. The tokens y_t are elected from a dictionary V that has each probable token (words, phrases, or symbols) related to the academic details. As well, the discriminator D_ϕ, parameterized by ϕ, can evaluate the quality of the text created by G_θ. D_ϕ offers feedback to G_θ by differentiating between the actual and created information. Explicitly, once the complete series Y_θ^c is created, D_ϕ evaluates it and gives a learning data $D(Y_\theta^c)$ to G_θ. This feedback supports G_θ to modify its parameters θ; thus, it generates precise and class-specific text in consecutive iterations.

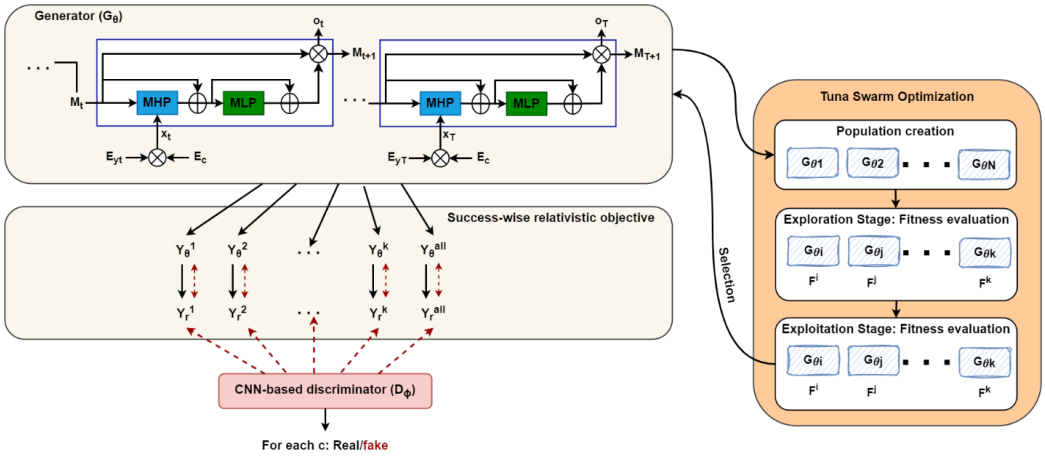

Fig 4. Structure of SucGAN Model

Figure 4 displays the complete SucGAN model, including G_θ and a CNN-based D_ϕ. The success-aware model utilizes G_θ to create data with a specific c, to deceive D_ϕ trained to differentiate between actual and created examples for c. This is accomplished using the success-wise relativistic objective, with the assistance of Gumbel-Softmax Relaxation (GSR) allowing D_ϕ to send gradients back to G_θ directly. The standard GAN trains D_ϕ to estimate the possibility that input information is original using ground-truth labels. However, this method does not provide useful feedback to modify G_θ. Hence, this study suggests a new learning goal according to the relative relationship between actual and created examples in c. Consider Y_r^c and Y_θ^c are the actual information sampled from distribution P_r^c and the created information sampled from distribution P_θ^c on c, correspondingly. The relativistic goal is categorized by the total label loss and the improved actual counterfeit loss. The goal of

D_ϕ is stated by Equation (8).

$$l_{D_\phi}^{SucRa} = \sum_{c=1}^{k} L^{Ra}\left(Y_r^c, Y_\theta^c\right) + L^{Ra}\left(Y_r^{all}, Y_\theta^{all}\right) \tag{8}$$

In Equation (8), Y_r^{all} and Y_θ^{all} are sampled from P_r^{all} and P_θ^{all} on c, correspondingly. The first term on the right-hand side calculates the distance between the actual and created information for c, whereas the 2nd expression calculates the total distance across c. Like RaGAN[11], L^{Ra} is described in Equation (9).

$$L^{Ra}\left(Y_r, Y_\theta\right) = -E_{Y_r \sim P_r}\left[log\left(\bar{D}_\phi\left(Y_r\right)\right)\right] - E_{Y_\theta \sim P_\theta}\left[log\left(1 - \bar{D}_\phi\left(Y_\theta\right)\right)\right] \tag{9}$$

The relativistic relation is determined in Equation (11).

$$\bar{D}_\phi\left(Y\right) = \begin{cases} sigmoid\left(D_\phi\left(Y\right) - E_{Y_\theta \sim P_\theta}\left[D_\phi\left(Y_\theta\right)\right]\right), & if\ Y\ is\ real \\ sigmoid\left(D_\phi\left(Y\right) - E_{Y_r \sim P_r}\left[D_\phi\left(Y_r\right)\right]\right), & or\ else \end{cases} \tag{10}$$

The relativistic relation measures the difference in probabilities of real and generated samples being real. G_θ aims to minimize this difference for c, making the created information as accurate as the actual ones. However, D_ϕ aims to maximize the possibility that actual information is more accurate than the created ones. The goal of G_θ is assigned to $l_{G_\theta}^{SucRa} = -l_{D_\phi}^{SucRa}$. The success-wise relativistic objective is more efficient than the standard GAN objective for training the SucGAN model in student data generation.

In this SucGAN model, a relational memory core is employed as the generator G_θ. Its fundamental concept is to use a fixed set of memory slots that interact through a self-attention strategy, enabling the model to effectively manage and utilize memory. This enhanced memory capacity significantly increases the generator's expressive power and its ability to capture and represent label-specific information. When a new vocabulary token y_t is observed at period t, it is transformed into an embedded token representation E_{y_t}. Simultaneously, an embedded label representation E_c is constructed to encode and control the label information. The generator's input vector x_t is then derived by applying a linear transformation W_x to the fusion of E_{y_t} and E_c:

$$x_t = \left[E_{y_t}; E_c\right] W_x \tag{11}$$

In Equation (11), [;] denotes the row-wise concatenation. In the SucGAN, a memory matrix M_t is restructured over an interval as new data is fused. Figure 4 portrays the procedure, where M_{t+1} is determined from M_t by fusing the input x_t at t based on the multi-head dot product attention policy. Explicitly, a rational memory core with H heads has H groups of linear conversion weights for the query $M_t W_q$, key $[M_t; x_t] W_k$, and value $[M_t; x_t] W_v$. Afterwards, M_{t+1} is restructured by Equation (12).

$$\widetilde{M}_{t+1} = \sigma\left(\frac{M_t W_q \left([M_t; x_t] W_k\right)^T}{\sqrt{d_k}}\right)[M_t; x_t] W_v \tag{12}$$

In Equation (12), $\sigma(\bullet)$ is the row-wise softmax function, and d_k denotes the column size of $[M_t; x_t] W_k$. The M_{t+1} and G_θ output o_t are obtained by

$$M_{t+1} = \psi_1\left(\widetilde{M}_{t+1}, M_t\right), o_t = \psi_2\left(\widetilde{M}_{t+1}, M_t\right) \tag{13}$$

In Equation (13), $\psi_1(\bullet)$ and $\psi_2(\bullet)$ are the relations between $\widetilde{M}_{t+1}$ and M_t by using residual links, MLP, and gated functions.

Directly sampling the successive token y_{t+1} from the multinomial distribution $\sigma(o_t)$ leads to a non-differentiability issue. To resolve this, the GSR is implemented to G_θ to approximate the sampling task. The Gumbel-Max trick and the softmax function $\sigma(\bullet)$ are considered to sample discrete texts and approximate the argmax function, respectively. Explicitly, y_{t+1} at t is sampled by

$$y_{t+1} = \underset{1 \le d \le |V|}{argmax}\left(o_t^{(d)} + g_t^{(d)}\right) \tag{14}$$

In Equation (14), $o_t^{(d)}$ is the value of the d^{th} size of o_t and $g_t^{(d)}$ is sampled from the Gumbel distribution, determined by $g_t^{(d)} = -log\left(-logU_t^{(d)}\right)$ where $U_t^{(d)} \sim Uniform(0,1)$. The differentiable approximation of argmax is accomplished by

$$\hat{y}_{t+1} = \sigma\left(\tau\left(o_t + g_t\right)\right) \tag{15}$$

In Equation (15), τ is the temperature variable that fine-tunes the bias and variance in approximation. A higher τ leads to a lower bias and higher difference, causing larger diversity in the created data with lower quality. To further enhance the diversity and quality of the created data, the TSO is proposed to train the SucGAN due to its ability to balance exploration and exploitation compared to conventional optimization algorithms like Particle Swarm Optimization (PSO), Grey Wolf Optimization (GWO), etc. Also, TSO ensures sustained diversity in solutions, especially significant in avoiding mode collapse in SucGANs. Its inherent randomness in selecting between spiral and parabolic policies makes it more robust, minimizing the risk of fluctuations and instability that other optimization algorithms may encounter during SucGAN training. The TSO involves a population of generators $G_\theta = \{G_{\theta 1}, G_{\theta 2}, \ldots, G_{\theta N}\}$ within the SucGAN by employing various mutation policies within a particular atmosphere $\left(D_\phi\right)$. Once the learning task is finished, the best-performing G_θ is chosen in the SucGAN to create realistic data for the given c.

2.2.2.1 Tuna Swarm Optimization Algorithm. Tuna, or tunnini, is a predatory fish that lives in the sea and has a unique swimming strategy called the fishtail form. It is an experienced hunter that feeds on multiple surfaces and midwater organisms[12]. On the other hand, it cannot equal the lightning-fast responses of small fish, thus it frequently involved in a group hunting to get its prey. It utilizes cunning to situate and get its prey utilizing 2 primary hunting policies: spiral hunting, where it swims in a spiral form to trap fish in shallow waters, and parabolic hunting, where it creates a curved line around its prey.

Initialization : It is commenced by arbitrarily creating initial tuna populations in the hunt zone as Equation (16):

$$S_i^{ini} = rand \bullet (u_l - l_l) + l_l, i = 1, \ldots, NP \tag{16}$$

In Equation (16), S_i^{ini} denotes i^{th} tuna, u_l and l_l are the upper and lower boundaries of the hunt zone, N represents the population size and $rand$ is random value from 0 to N. All tunas in the swarm represent a candidate solution. These tunas consist of a set of N-dimensional information. In each iteration, the fitness function (SucGAN loss in Equation (8)) is calculated for all tunas in the hunt zone. Genetic operators like crossover and mutation are used to balance exploration and exploitation by generating a new population in all iterations. The tunas' positions are modified using 2 distinct hunting policies.

1. Spiral hunting : Most tuna swim aimlessly while hunting for food, but a small group may lead the way. The rest of the tuna will follow this group as they chase their prey, eventually forming a spiral pattern to catch their target. When the tuna swarm uses a spiral hunting approach, they communicate to identify the best individuals to follow. If a leader fails, the tuna will choose a random member to follow instead. This policy is stated in Equation (17).

$$S_i^{(t+1)} = \begin{cases} \alpha_1 \bullet ({S^t}_{rand} + \beta \bullet |S^t rand - {S^t}_i|) + \alpha_2 \bullet {S^t}_i, i = 1 & if\ rand < \dfrac{t}{t_{max}} \\ \alpha_1 \bullet ({S^t}_{rand} + \beta \bullet |{S^t}_{rand} - {S^t}_i|) + \alpha_2 \bullet {S^t}_{i-1}, i = 2, 3 \ldots N & \\ \alpha_1 \bullet ({S^t}_{best} + \beta \bullet |{S^t}_{best} - {S^t}_i|) + \alpha_2 \bullet {S^t}_i, i = 1 & if\ rand \geq \dfrac{t}{t_{max}} \\ \alpha_1 \bullet ({S^t}_{best} + \beta \bullet |{S^t}_{best} - {S^t}_i|) + \alpha_2 \bullet {S^t}_{i-1}, i = 2, 3 \ldots N & \end{cases} \tag{17}$$

$$\alpha_1 = \alpha + (1 - \alpha) \bullet \frac{t}{t_{max}} \tag{18}$$

$$\alpha_2 = (1 - \alpha) - (1 - \alpha) \bullet \frac{t}{t_{max}} \tag{19}$$

$$\beta = e^{bl} \bullet cos(2\pi b) \tag{20}$$

$$l = e^{3cos\left(\left(\left(t_{max} + \frac{1}{t}\right)^{-1}\right)\pi\right)} \tag{21}$$

In Equations (17), (18), (19), (20) and (21), S_i^{t+1} refers to the i^{th} tuna in $t+1$ iteration produced by the genetic functions, S_{best}^t denotes the current optimal individual, S_{rand}^t is the reference spot randomly elected, α_1 is the weight factor to manage the tuna whirling to S_{best}^t or S_{rand}^t, α_2 is the weight factor to manage the tuna whirling to the individual in front of it, β is the

distance to manage the disparity between the tuna and S_{best}^{t} or S_{rand}^{t}, α is a constant to regulate the level of tuna following, t indicates the current iteration, t_{max} is the maximum iteration and b is the random integer between 0 and 1.

2. Parabolic hunting : Tunas work together to feed efficiently by creating spiral and parabolic patterns. They use prey as a reference point to form a parabolic form and scan their environment for food. Both methods are employed concurrently with a 50% chance of success for each. This policy is stated in Equation (22).

$$S_i^{t+1} = \begin{cases} S_{best}^{t} + rand \bullet (S_{best}^{t} - S_i^{t}) + \gamma \bullet p^2 \bullet (S_{best}^{t} - S_i^{t}), & if\ rand < 0.5 \\ \gamma \bullet p^2 \bullet S_i^{t}, & if\ rand \geq 0.5 \end{cases} \tag{22}$$

Where

$$p = \left(1 - \frac{t}{t_{max}}\right)^{\left(\frac{t}{t_{max}}\right)} \tag{23}$$

In Equation (22), γ represents the arbitrary integer between 1 and -1. In each iteration, tuna can arbitrarily choose between spiral or parabolic hunting policies. Tuna in the population is continuously updated until the termination criteria are satisfied. Thus, the optimal SucGAN is chosen for improving the training efficiency of the SucGAN.

Algorithm 1: TSO for Training SucGAN

Input:itr_{max} and the number of G_θ in SucGAN $(G_{\theta 1}, G_{\theta 2}, \ldots, G_{\theta N})$

Output: Best-performing G_θ for SucGAN

1. **Begin**
2. Create an initial population of tunas S_i^{ini}\;(i=1,\dots,NP , i e , \left(G_{\theta1},G_{\theta2},\dots,G_{\theta N}\right\} arbitrarily;
3. Set free parameters a and z;
4. $while\ (t < itr_{max})$
5. Determine the fitness f of each tuna according to the SucGAN loss;
6. Change the place of $S^{t}{}_{best}$;
7. $for\ (each\ tuna)$
8. Change α1,α2, p;
9. $if\ (rand < z)$
10. $Change\ S_i^{t+1}$;
11. $else\ if\ (rand \geq z)$
12. $if\ (rand < 0.5)$
13. Change the place S_i^{t+1};
14. $else\ if\ (rand \geq 0.5)$
15. Change the place S_i^{t+1};
16. ***end if***
17. ***end if***
18. ***end for***
19. ***end while***
20. Identify the optimal tuna S_{best} (best-performing G_θ in SucGAN) and its fitness $(f(S_{best}))$;
21. **End**

Using TSO, the SucGAN discovers multiple solutions and adapts high-quality networks to improve G_θ's efficiency in creating accurate and relevant data. At last, the EGAN determines the weighted mean of created data from DivGAN and SucGAN to generate absolute artificial student data. Accordingly, this paper facilitates EGAN to produce a large volume of artificial data to augment available datasets.

2.3 Student Performance Prediction

Once data augmentation is finished, the augmented social media dataset, comprising students' online posts and interactions, is processed by the LDA algorithm to extract physiological features (e.g., mental health and mood changes)[10]. Those features along with the augmented student dataset, comprising academic and demographic attributes are then given to the SADDL model[10], which encompasses the following 3 major modules:

i. LSTM network: It learns temporal dependencies to comprehend how student factors alter over the period, extracting patterns and trends in data.

ii. DCNN: It learns relationships between demographic, academic, and physiological factors.

iii. MLP: It categorizes and predicts student performance utilizing features learned by the LSTM and DCNN.

3 Results and Discussion

This part displays the efficacy of the EGAN-SADDL and compares it with the previous models, such as SADDL [10], LSTM [5], GRU [7], and RNN+LSTM+RF [8] in MATLAB 2019b. To realize a fair analysis, proposed and existing models were implemented with the parameters listed in Table 1 . They were tested utilizing the dataset described in Section 2.1. This dataset is augmented by the proposed EGAN as follows:

- Creating 20,000 artificial instances (2500 in each class) for academic and demographic attributes.
- Generating 20,000 artificial instances (2500 in each class) for social media data.

So, by integrating the created instances with the actual datasets, the augmented datasets are now obtained as follows:

- The academic and demographic dataset has 70,000 instances (17500 in each class).
- The social media dataset has 70,000 instances (17500 in each class).

This indicates that the EGAN augments the dataset by approximately 2.5 times for both student and social media datasets. So, the entire dataset consists of 1,40,000 instances for student academic performance prediction. In this experiment, 80% of the instances are taken for training (totally 112000 instances (28000 in each class)) and 20% of the instances are taken for testing (totally 28000 instances (7000 in each class)). Also, 2-fold cross-validation is used to obtain the performance results.

Table 1. Hyperparameter Settings

Models	Parameters	Range
LSTM [5]	Number of layers	3
	Number of neurons in LSTM encoder	64
	Number of neurons in LSTM decoder	32
	Activation function	Rectified Linear Unit (ReLU)
	Optimizer	Adam
	Loss	Categorical cross-entropy
GRU [7]	Number of hidden layers	2
	Number of neurons in each hidden layer	256
	Recurrent dropout	0.23
	Batch size	90
	Learning rate	0.01
	Activation function	Sigmoid function
	Optimizer	Adam
	Number of epochs	300
	Loss function	Binary cross-entropy
RNN+LSTM+RF [8]	Number of stacked RNN layers	3
	Number of hidden units in each RNN layer	64
	Number of LSTM layers	3
	Number of hidden units in each LSTM layer	64
	Number of trees in the forest	100
	Maximum depth of each tree	10
	Minimum samples split	2
	Minimum samples leaf	1
	Batch size	128
	Number of iteration	200
	Learning rate	0.0001
	Dropout rate	0.5

Continued on next page

Table 1 continued

	Optimizer		Adam
	Loss function		Mean Square Error (MSE)
	Activation function		ReLU
SADDL[10], and **Proposed EGAN-SADDL**	EGAN	Optimizer	Adam
		Learning rate for generator	0.001
		Learning rate for discriminator	0.01
		λ	1
		a in TSO	0.7
		z in TSO	0.05
	LSTM	Learning rate	0.0001
		Batch size	64
		Loss function	Weighted cross-entropy
		Number of LSTM units	12
		Number of epochs	120
		Optimizer	Adam
		Dropout	0.1
	DCNN	Learning rate	0.001
		Dropout	0.5
		Weight decay	0.0001
		Momentum	0.9
		Batch size	64
		Number of epochs	120
		Activation function	ReLU
		Loss function	Weighted cross-entropy
	MLP	Number of hidden layers	1
		Number of neurons in hidden layer	20
		Activation function	Softmax
		Number of neurons in output layer	4
		Batch size	25
		Learning rate	0.01
		Maximum iteration	100
		Momentum	0.9
		Dropout	0.5
		Optimizer	Adam

The performance evaluation metrics include the following:

- **Accuracy:** It defines the fraction of a precise prediction of a student's performance to the sum of cases analyzed, as computed in Equation (24).

$$Accuracy = \frac{True\ Positive\ (TP) + TrueNegative\ (TN)}{TP + TN + False\ Positive\ (FP) + False\ Negative\ (FN)} \tag{24}$$

TP is the total +ve cases (e.g., distinction, high distinction, and pass) exactly predicted as themselves, TN is the total -ve cases (e.g., fail) exactly predicted as themselves, FP is the total -ve cases inexactly predicted as +ve, and FN is the total +ve cases inexactly predicted as -ve.

- **Precision:** It is computed in Equation (25) .

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

- **Recall:** It is determined by Equation (26) .

$$Recall = \frac{TP}{TP + FN} \tag{26}$$

- **F-measure:** It is determined as Equation (27) .

$$F - measure = 2 \times \frac{Precision \bullet Recall}{Precision + Recall} \tag{27}$$

- **RMSE:** It is the discrepancy between the actual grade (y_g) and the predicted grade (p_g).

$$RMSE = \sqrt{\sum (y_g - p_g)^2 / N} \tag{28}$$

In Equation (28) , N is the total number of actual performance values.

hows the confusion matrix of the EGAN-SADDL using 20% test dataset. It shows the number of exactly and inexactly predicted grades of the students. Performance indicators are determined by this matrix.

Table 2. Confusion Matrix for 20% Test Dataset

		Actual			
	Class	Distinction	Fail	High-distinction	Pass
	Distinction	6623	80	155	142
Predicted	Fail	152	6615	155	78
	High-distinction	130	150	6610	110
	Pass	95	155	80	6670

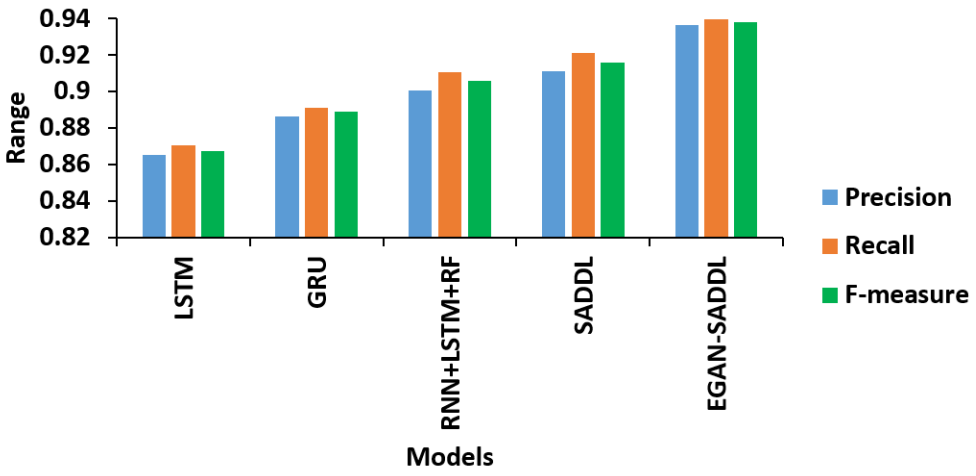

Fig 5. Scrutiny of Precision, Recall, and F-Measure for Various Student's Performance Prediction Models

Figure 5 portrays the precision, recall, and F-measure for different models applied for student performance prediction. The EGAN-SADDL increases the precision by 8.29%, 5.69%, 4.03%, and 2.82% compared to the LSTM, GRU, RNN+LSTM+RF, and SADDL, respectively. The recall is improved by 7.96%, 5.44%, 3.17%, and 1.98% compared to the LSTM, GRU, RNN+LSTM+RF, and SADDL, correspondingly. The F-measure is 8.18%, 5.55%, 3.58%, and 2.44% higher than the LSTM, GRU, RNN+LSTM+RF, and SADDL models respectively. So, these enhancements are realized by integrating the EGAN to augment students' academic and demographic data, as well as social media data by generating a huge volume of artificial data.

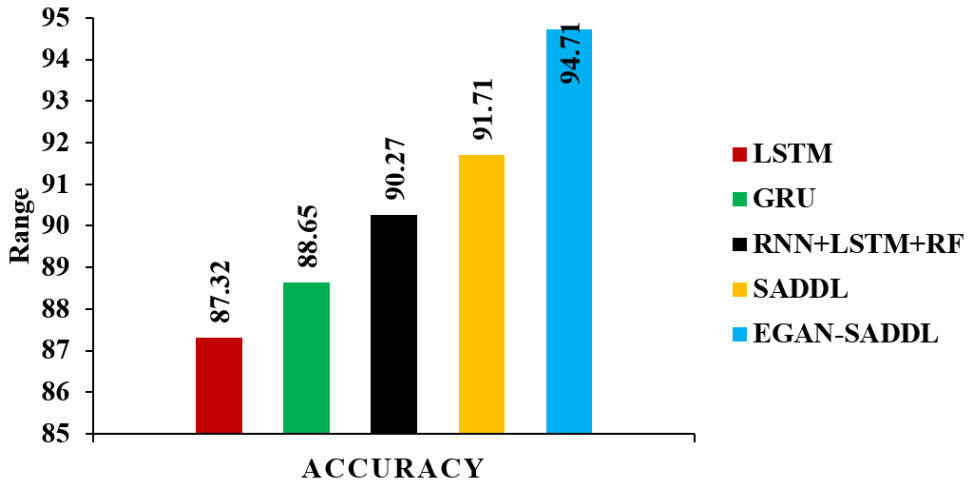

Fig 6. Comparison of Accuracy for Various Student's Performance Prediction Models

Figure 6 displays the accuracy of different student performance prediction models. The EGAN-SADDL model increases the accuracy by 8.46%, 6.84%, 4.92%, and 3.27% compared to the LSTM, GRU, RNN+LSTM+RF, and SADDL, respectively. This can be realized by training a huge volume of student datasets, which are generated by the EGAN, while other models rely on a limited dataset.

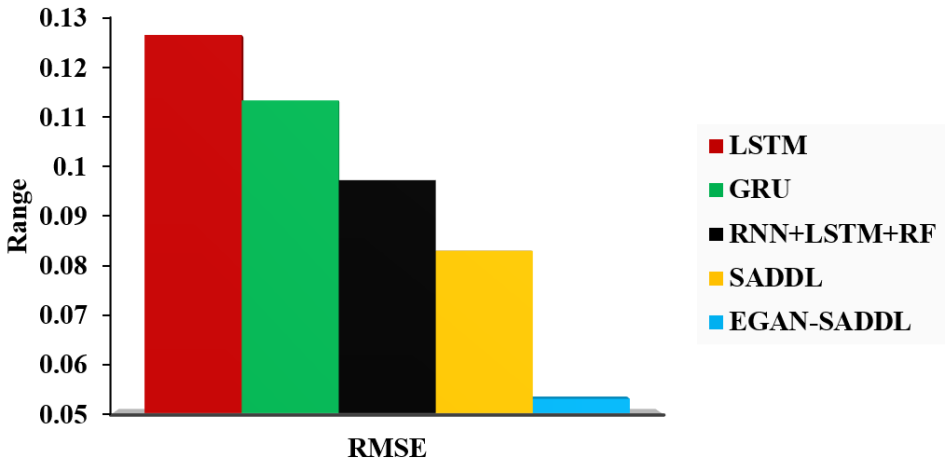

Fig 7. Comparison of RMSE for Various Student's Performance Prediction Models

An evaluation of RMSE for different models for predicting student performance is shown in Figure 7 . The results indicate that the EGAN-SADDL model reduces the RMSE by 58.28%, 53.39%, 45.63%, and 36.19% compared to the LSTM, GRU, RNN+LSTM+RF, and SADDL, respectively. This suggests that the EGAN-SADDL model successfully boosts the accuracy of predicting student performance by leveraging a huge volume of student datasets.

4 Conclusion

This study presents the EGAN-SADDL model to predict student performance. The ensemble of DivGAN and SucGAN augmented the student dataset efficiently to increase the accuracy of prediction. The TSO was used in the SucGAN to balance data quality and diversity. The artificial data created by EGAN was merged with the actual data to get an augmented dataset. Then, this dataset was used to extract the physiological features and predict the student performance using LDA and SADDL,

respectively. Finally, test outcomes proved that the EGAN-SADDL on the augmented student dataset (e.g., academic and demographic dataset, and social media dataset) has 94.71% accuracy and 0.0529 RMSE, compared to the previous models. However, considering heterogeneous data, i.e., academic data, demographic data, and social media data in the EGAN-SADDL model poses a significant challenge since it might fail to identify correlations between them. Therefore, future work will focus on developing a multi-view learning-based EGAN-SADDL model, which can learn correlation features from heterogeneous data independently to improve accuracy in predicting student performance. It can leverage the strengths of each data type when reducing redundancy.

References

1) Al-Maskari A, Al-Riyami T, Kunjumuhammed SK. Students academic and social concerns during COVID-19 pandemic. *Education and Information Technologies*. 2022;27:1–21. Available from: https://doi.org/10.1007/s10639-021-10592-2.
2) Batool S, Rashid J, Nisar MW, Kim J, Kwon HY, Hussain A. Educational data mining to predict students' academic performance: a survey study. *Education and Information Technologies*. 2023;28:905–971. Available from: https://doi.org/10.1007/s10639-022-11152-y.
3) E A. Student performance prediction using machine learning algorithms. *Applied Computational Intelligence and Soft Computing*. 2024;2024(1):1–15. Available from: https://onlinelibrary.wiley.com/doi/epdf/10.1155/2024/4067721.
4) Liu M, He W, Zhou G, Zhu H. A new student performance prediction method based on belief rule base with automated construction. *Mathematics*. 2024;12(15):1–23. Available from: https://www.mdpi.com/2227-7390/12/15/2418.
5) Vives L, Cabezas I, Vives JC, Reyes NG, Aquino J, Cóndor JB, et al. Prediction of students' academic performance in the programming fundamentals course using long short-term memory neural networks. *IEEE Access*. 2024;12:5882–5898. Available from: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10381724.
6) Fazil M, Rísquez A, Halpin C. A novel deep learning model for student performance prediction using engagement data. *Journal of Learning Analytics*. 2024;11(2):23–41. Available from: https://learning-analytics.info/index.php/JLA/article/view/7985.
7) Baniata LH, Kang S, Alsharaiah MA, Baniata MH. Advanced deep learning model for predicting the academic performances of students in educational institutions. *Applied Sciences*. 1963;14(5). Available from: https://www.mdpi.com/2076-3417/14/5/1963.
8) Kukkar A, Mohana R, Sharma A, Nayyar A. A novel methodology using RNN+ LSTM+ ML for predicting student's academic performance. *Education and Information Technologies*. 2024;29:14365–14401. Available from: https://link.springer.com/article/10.1007/s10639-023-12394-0.
9) Liu Y, Hui Y, Hou D, Liu X. A novel student achievement prediction method based on deep learning and attention mechanism. *IEEE Access*. 2023;11:87245–87255. Available from: https://ieeexplore.ieee.org/document/10216940.
10) Venkatachalam B, Sivanraju K. Predicting student performance using mental health and linguistic attributes with deep learning. *Revue d'Intelligence Artificielle*. 2023;37:889–899. Available from: https://doi.org/10.18280/ria.370408.
11) Jolicoeur-Martineau A. The relativistic discriminator: a key element missing from standard GAN. *arXiv*. 2018;p. 1–25. Available from: https://arxiv.org/pdf/1807.00734.
12) Xie L, Han T, Zhou H, Zhang ZR, Han B, Tang A. Tuna swarm optimization: a novel swarm-based metaheuristic algorithm for global optimization. *Computational Intelligence and Neuroscience*. 2021;2021:1–22. Available from: https://onlinelibrary.wiley.com/doi/epdf/10.1155/2021/9210050.