Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.028659
Article

Tech Science Press

A Feature Learning-Based Model for Analyzing Students' Performance in Supportive Learning

P. Prabhu[1], P. Valarmathie[2,*] and K. Dinakaran[3]

[1]Department of Computer Science and Engineering, K. S. R College of Engineering, Thiruchengode, Tamil Nadu-637215, India
[2]Centre for Artificial Intelligence, Chennai Institute of Technology, Chennai, Tamil Nadu-600069, India
[3]Department of Computer Science and Engineering, S. A. Engineering College, Chennai, Tamil Nadu-600077, India
*Corresponding Author: P. Valarmathie. Email: valarmathiep@citchennai.net
Received: 15 February 2022; Accepted: 10 June 2022

Abstract: Supportive learning plays a substantial role in providing a quality education system. The evaluation of students' performance impacts their deeper insight into the subject knowledge. Specifically, it is essential to maintain the baseline foundation for building a broader understanding of their careers. This research concentrates on establishing the students' knowledge relationship even in reduced samples. Here, Synthetic Minority Oversampling TEchnique (SMOTE) technique is used for pre-processing the missing value in the provided input dataset to enhance the prediction accuracy. When the initial processing is not done substantially, it leads to misleading prediction accuracy. This research concentrates on modelling an efficient classifier model to predict students' performance. Generally, the online available student dataset comprises a lesser amount of sample, and k-fold cross-validation is performed to balance the dataset. Then, the relationship among the students' performance (features) is measured using the auto-encoder. The stacked Long Short Term Memory ($s-LSTM$) is used to learn the previous feedback connection. The stacked model handles the provided data and the data sequence for understanding the long-term dependencies. The simulation is done in the MATLAB 2020a environment, and the proposed model shows a better trade-off than the existing approaches. Some evaluation metrics like prediction accuracy, sensitivity, specificity, AUROC, F1-score and recall are evaluated using the proposed model. The performance of the $s-LSTM$ model is compared with existing approaches. The proposed model gives 89% accuracy, 83% precision, 86% recall, and 87% F-score. The proposed model outperforms the existing systems in terms of the earlier metrics.

Keywords: Student performance; quality education; supportive learning; feature relationship; auto-encoder; stacked LSTM

1 Introduction

Numerous colleges and universities suffer from the poor performance of students in today's world. Even though the latest standard is raised to higher education outcome-based education [1], about 40% of college

students joined college for a bachelor's degree graduate. Meanwhile, the college dropout rate attained a massive rate of 40% in 2018. Many research studies tried to expand the automated models to conquer the consequences of the circumstances that help predict students' academic performance in higher education [2]. The forecast about the performance of the students on time provides countless advantages, which include earlier detection of students fighting to pass their courses and the students, who are at the threat of dropping out from education, pathways to select the system, and also the features which impact the rates of student retention and behaviours [3]. This kind of brilliant perception empowers educational leaders to conceive and execute the correct interference to take care of academic advising, lead the way during changes and advancement in the curriculum, and decide the risks in the program [4]. Moreover, the selection of suitable machine learning techniques is used to evaluate the performance of students correctly that endure a complicated attempt [5].

The infinite number of attributes generally influences students' academic achievements that differ from non-academic to educational features [6] and findings to identify them. Developing a complicated predictive model is essential, having the variability of the attributes. The individual learning modes are proven by the ensemble learning model, a hybrid learning model, which outperforms student academic performance prediction accuracy [7]. However, the latest research shows that about 50% of the researchers use supervised learning approaches. At the same time, about 5% of the studies employ unsupervised learning algorithms [8]. It is important to note that a supervised machine learning algorithm contributes adequate prediction accuracy [9]. Moreover, it is notable that increasing the supervised learning approaches will generate better accurate predictions than believing an unsupervised model with some commonly defined errors. Moreover, the prevailing model combines the effects of unsupervised and supervised learning algorithms.

The accessible algorithms and techniques enhance accuracy in predicting students' performance [10]. There is a shortage in producing the descriptive examination of the available attributes that are variables that causes the execution of observed student. In addition, depending on the single technique, if this technique is non-linear or linear, that can be inadequate because of the challenges in obtaining multiple numbers of attributes in a single predictor technique [11]. Student performances are affected by the features that frequently vary between students and academic semesters for a similar set of students. Single linear modes usually suffer from the data that are under fitted to eliminate the ambiguity on the trained data that consists of multiple overlapping behaviours among the students leading to a greater rate of wrong predictions. At the same time, with non-linear models, the fake predictions will become high likelihood that the risk of overfitting problem is the factor of the data on that they are qualified [12]. This model will remember a few characteristics of behaviour only.

However, a solution in ensemble machine learning does not contain the contribution weighted dynamically to predict the students' performance using participating techniques. Furthermore, restrictions concern the misuse of the training dataset or utilizing a single data set to approve the model [13]. The limits, as mentioned earlier, are addressed by contributing deep learning techniques to identify the enabled attributes and the factors which block the performance in the academic courses [14]. However, few models concentrate on the accomplishment of first-year students only on prediction (for example, [15]). Above 50% of the researchers utilized Artificial Neural Network (ANN) and Support Vector Machine (SVM) approaches to predict the performances. However, many associated techniques are available to confine in predicting the grades of future courses and are not related to the critical attributes that lead to the attained performance of students. Pupils' accomplishments should be improved, and reducing dropout challenges from education is achieved by understanding the effect of enabled and interdicting attributes based on our perception. The suggested technique is eminent by the clustering attributes, which requires understanding the related features that lead to predicting future course grades. A proposed approach helps boost the accuracy of prediction on the same depending on the grades and the

identification of necessary attributes that can cause the accomplishments of observed students. The strengths and weaknesses of the program are initiated using the group of features and the environments that are assumed to get latent or direct consequences in the results of academic students. Specifically, this research offers the major benefaction that includes:

- Here, an online accessible dataset of students is taken and considered an input to the predictor model.
- The imbalanced dataset with (minority or majority samples) is balanced using the pre-processing step known as SMOTE. Later, the features are analyzed with word embedding and auto-encoders, where the learnt features play a substantial role in enhancing the prediction accuracy.
- Finally, the classification process is done with a stacked LSTM model, where the network model learns the initial state to measure the performance. The simulation is done with MATLAB 2020a environment, and various metrics like precision, accuracy, recall and F1-score are evaluated and compared with multiple prevailing approaches.

The exhaustive performance evaluation is performed with the help of various metrics of the paradigm instead of the prediction approaches for the performance of initial students with seven standard illustration data sets. The experimental outcome establishes the efficiency of performance and our predictive models with efficient advantages.

2 Related Works

We discussed the associated work from two essential viewpoints since our study concentrates on computer-based predictive analysis. The first step introduces the fundamental ideas that describe academic performance. The second step explores the modern techniques which help predict and explain those.

Commonly, the data analytics on the academic information evolves two crucial aspects: learning analytics and predictive analytics [16]. Learning analytics helps gather and analyze the learning information, and the circumstance needs to enhance the learning process results. Moreover, predictive analytics helps student learning prediction and finds the failure rates performance, and insists that the course can obtain better outcomes in the future [17]. However, data mining techniques resolve the association between attributes and learning [18]. For instance, the cumulative grade point scores of the student are not reflected by ethnicity. At the same time, the requirements of cognitive university admission do not accurately explain the performance, proposing the non-academic attributes that provide an essential part of learning. Our study tries to enhance the performance prediction and describe the attained forecast, applying predictive and explanatory modelling. The current study explains performance as a proficiency scale in future programs [19]. It is important to note that estimation of excellence is a remarkable aim to conquer continuing problems in higher education. This aim includes low academic grades, failure and dropouts from education have been increased and extended time of graduation between students. The past semester grades and present coursework tests like midterms, assignments, projects, and final exams are considered to determine the achievements. Moreover, consecutive tasks are examined the influence of non-academic features like demographics of students and status of socio-economic on the successes of students [20]. The tests utilize the learning results less often despite their significance in estimating student performance. Previous conclusions denote the various entwined attributes that influence the performance and the accurate prediction with the enhancement of the refined techniques.

Prediction in higher education is a worthy task to achieve strategic advantages like the advancement of quick warning and the recommended methods to select the course path, the identification of unfortunate behaviour of students, and the automation of education program assessments [21]. Moreover, the exact

prediction of student academic accomplishments is a complex study requiring a deep knowledge of all characteristics and the environments surrounding the student and their learning circumstance [9]. However, student performance prediction evolves to discover their activities, and the choice and different powerful academic and non-academic attributes are considered [22]. Moreover, the present summaries are not satisfactory; even now, it is noted that (i) single learning techniques provide low prediction accuracy; the attributes that lead to the noticed academic performance are recognized or determined insufficiently [23]. This study focuses on interconnecting the main difference.

For example, few studies have gone above predicting course grades to identify endangered students. Moreover, exact predictive modelling in academics is still tricky because of the data sparsity and exponentiality issues and influential classifiers like SVM. To manage, this illustrates the example that the last-mentioned provocation for the SVM [24] used a standard multivariate technique and the vector transformations to minimize the approach training period. Even though this approach has a minimum training time of about 59% approximately, the optimized algorithm attained promised accuracy of about 93% to recognize the most susceptible students with lack of success. The author in [25] expanded the productive consensual network-based deep support vector machine model in the consecutive task that denotes ICGAN-DSVM that manages the small training datasets and accurately predicts the performances. The outcomes demonstrated that family mentoring that integrates with school coaching enhances performance. Even though integrating the existing techniques such as CGAN improved the prediction by about 29%, small confirmation data sets are utilized to check the model's performance. Many scientists indicated that learning univariate analysis and SVM produced better classification outcomes in project grades while the small datasets are trained like in the scenario of postgraduate courses [26]. Moreover, the research concentrates on predicting the students who may fail in future programs and does not fully describe the attributes that lead to the failure.

The researchers proposed a technique for genetic programming to find underperformed students, especially those who feel challenged by socio-economic demerits. Students' data are gathered from different origins in this technique, which strengthens the response commended to the decision-makers. Moreover, the suggested architecture does not find the attributes that lead to the predicted performance. The author in [27] offers a genetic algorithm that becomes a part of an earlier warning method for identifying early dropouts from programs. The warning methods do not support the aims again beyond the feasible dropouts of students. The achievements which are predicted correctly is a complicated work involving brilliant innovative techniques that recognize the developing attributes and environments that affect student academic performance [28]. The effect of these attributes and environments might vary from one group of students to another batch and from one course to another course. The exhaustive analysis of the needed tasks exposed the gaps concerning the below areas [29]:

- Due to the inadequate use of hybrid techniques, this technique integrates the benefits of unsupervised learning and supervised learning techniques to optimize and automate the performance of student academic prediction accuracy.
- The existing techniques provide the inflexibility to examine myriad academic and non-academic factors that need to be considered to impact the quality of student education. Few methods help students' accomplishments predict without relating them using the enabled attributes or feasible demerits. Meanwhile, only small subsets of efficient features are considered by others.
- The hybrid techniques are comprised of models that do not modify the benefaction to estimate the predictions dynamically under the student's environments.

Most prediction techniques are approved using the single dataset as an error to the approach's viability [30–35]. The most related works are summarized in Table 1, which helps predict students' achievements and represents their weaknesses by associating them with the research gaps.

Table 1: Comparison of various existing approaches

Methods	Focus	Computation	Dataset	Observations
Matrix factorization and Linear regression model	Predicts the student's results in their courses under the chosen degree pathways	RMSE = [0.63, 0.72] Precision = [26.68%]	Minnesota university, USA, Private data set comprising (2k undergraduate students, 2k various courses, 75k student course grades and 2 Majors)	• The research concentrated on the prediction of grade letters. • A course related subgroup of information provided better predictions. • Performance varies remarkably among various departments and is dependent on prior courses. • There is no prediction on student marks, only grade prediction. • The student course-specific techniques gave fewer predictions.
Attention graph convolutional network model	They predict students' consecutive semester grades for courses and detect the in-danger students at risk of dropping out or failing.	MAE = [0.30, 0.54] Precision = [80.21%, 93.23%]	USA, George Mason University, Private dataset comprising (43490 undergraduate students, 185 courses, 385505 grades, 5 Majors)	• Prediction on rank. • Tests are implemented in two various semesters. • The earlier programs describe the prediction that is utilized by this technique. • Dependency among programs was considered (evolution of student knowledge). • The performance of a model differs throughout majors. • The description has been integrated into available courses alone; extra features have been eliminated. • MAE is considerably large in a few majors.
Five models compared: KNN (k-Nearest Neighbour), Learning Discriminant Analysis (LDA), ANN, SVM, Naive Bayes (NB)	They predict the student's performance at the postgraduate level by small data sets.	Precision = [58.1%, 69.7%]	Emirates, British University in Dubai. Private dataset, comprising (50 postgraduate students, nine courses, 311 instances, 1 Major)	• This technique can be used for training and show smaller data sets suitable for postgraduate studies. • For small datasets, Key predictors are determined. • A heat map provides the best performance indicators, which can be wrong. • Five student features alone are utilized in predicting performance; subsequent variables are avoided. • Four encoding labels alone (grades) are utilized.
Bayesian deep learning approaches (LSTM and MLP (Multi-Layer Perceptrons))	Prediction of student grades. Evaluating the unreliability that is related to performance. Predicting to identify the basic courses for student success.	MAE = [0.253, 0.588] Precision = [79.32%, 92.62%]	USA, George Mason University, Private dataset comprising (28717 undergraduate students, 182 courses, 249716 grades, 5 Majors)	• This model observes the preceding semester's courses; then, the students acquire the knowledge. • It insists on in-danger students and describes predictions on performance in earlier critical studies, which leads to the failure of students. • Remarkable variable to predict the performance among majors. • The supremacy of suggested models achieves no statistical testing.

(Continued)

Table 1 (continued)

Methods	Focus	Computation	Dataset	Observations
Matrix Factorization	They predict the next term's grade depending on latent features like tutors for courses and academic level.	MAE = [0.615, 0.654] Precision = [63.8%, 67.0%]	USA, George Mason University, Private dataset comprising (11027 undergraduate students, 1318 courses, 140259 grades, 8 Majors)	• This technique integrates extra latent features with matrix factorization. • Different majors achieve experiments. • The chosen courses are affected by the features. • This technique employs better in the specific majors alone. • This model focuses on four latent factors alone.
Discriminative and Generative and classification models: C4.5, SVM, NB, and CART Bayes Network	Prediction on the completion of students' degrees using the personalized attributes like the expenses in the family.	Precision = [71%, 86.7%]	Pakistan, different universities Private data set comprising 776 student data.	• This research analyzed 23 attributes; like expenses in the family, students' success can be predicted using their personal information. • This technique integrates attributes in predicting the student's performance. • Student grades are not predicted; they predict whether it is success or failure.

3 Methodology

This section provides a detailed analysis of the anticipated stacked LSTM model for exercising cooperative learning. Here, an online dataset known as students' performance in the Exam dataset is considered for validation from Kaggle [23]. The student's performance is measured with the stacked LSTM using the pre-processing steps like SMOTE, word embedding, and auto-encoding for the feature learning process are analyzed. The simulation is done in the MATLAB 2020a environment, and various performance metrics like precision, accuracy, recall and F1-score are evaluated and compared with multiple existing approaches.

3.1 SMOTE

The samples are imbalanced with diverse classes by examining the sample distribution. In the worst-case scenario, the number of samples with majority classes is ten times higher than the minority classes. However, some samples are nearer to the classification boundaries. These factors increase the complexity of the classification task and influence the model performance. Thus, data augmentation is considered a vital factor. Here, SMOTE is used as a pre-processing approach that generates synthetic data of the minority classes. However, it does not consider the significant factors related to adjacent majority classes while synthesizing the minority class data. Therefore, the classes are overlapped, and to resolve this issue, the borderlines of the data samples are given greater attention to evaluating the nearby points of the available minority class. If the minority class is labelled first, the nearest neighbours are extracted from those available samples with the minority class. Then, the set of chosen minority classes is related to the majority class. The chosen neighbours are selected and multiplied based on the distance among the samples. The nearest neighbours range from 0 to 1. These values are included with the available data samples. Thus, the synthetic models are generated based on Eq. (1):

$$synthetic_j = p_j + r_j^* diff_j; \quad j = 1,\ 2,\ \ldots,\ s \tag{1}$$

Here, p_j specifies the distance among the neighbours, r_j specifies the random number $\in [0,\ 1]$, $diff_j$ specifies the distance among the sample's nearest neighbours. After the data augmentation process (all

subjects), the number of samples in every class is closer to the number in the minority class. Thus, the problem related to imbalanced data is resolved and helps to enhance the classification process.

3.2 Auto-Encoding for Feature Analysis

The auto-encoding part of NN is split into two diverse parts: encoder and decoder. It is mathematically provided as in Eqs. (2)–(4):

$$\phi = \chi \rightarrow \quad \mathcal{F} \quad (encoder) \tag{2}$$

$$\psi: \mathcal{F} \rightarrow \quad \chi \ (decoder) \tag{3}$$

$$\phi, \quad \psi = \arg\min_{\phi, \psi} ||X - (\psi \quad o \quad \phi)||^2 \tag{4}$$

The network encoding part is specified with a function passed via bias parameter b, activation function σ and latent dimension z. It is shown in Eq. (5): the encoder part ϕ marks the provided original data χ towards the latent space $\mathcal{F}$ for dimensionality reduction. Subsequently, decoder function ψ needs to map latent and reduced output space. Here, the output is the same as input data χ, where the encoder and decoder pair intend to reconstruct the data and shape after capturing and performing specific generalized non-linear data transformation.

$$z = \sigma \ (Wx + b) \tag{5}$$

It is a related way of providing the NN's decoding part, and it is represented with diverse activation functions, weight, and bias. It is expressed as in Eq. (6):

$$x' = \sigma' \ (W'z + b') \tag{6}$$

The loss function L for the provided NN is expressed using the encoding and decoding network function. It is expressed as in Eq. (7):

$$L(x, \quad x') = ||x - x'||^2 = ||x - \sigma'(W' \ (\sigma(Wx + b)) + b')||^2 \tag{7}$$

The objective of an auto-encoder is to choose suitable encoder and decoding functions with minimal information encoded and re-generated using the decoder with a minimal loss function. Based on the provided Eq. (7), the loss function L is used for training the NN via the standard back-propagation process. This method facilitates supervised learning by constructing cluster labels (sign and voice) using k-means clustering and the generated tags for a different purpose. The following is the step-by-step process:

- Initially, capture the meta-data descriptive and characteristics as features and construct the feature vectors as $<f_1, \quad f_2, \ \ldots, \ f_n)$ for all the data.
- Apply the traditional k-means for feature vector clustering and predict the cluster group.
- Consider the class groups and corresponding identifications (tags) as labels;
- Fed the input data with corresponding feature vectors and generated labels for successive stages.

Then, construct the auto-encoder model based on NN with specific hidden neurons and layers, i.e., nodes.

- The number of nodes over the inner layers specifies the number of clusters;
- The number of nodes over the input layer specifies the feature size and vectors;
- The nodes over the output layer specify the probabilistic values for the provided two datasets representing the cluster labels.
- Then, partition the constructed data into testing/training datasets.
- Train auto-encoder based stacked LSTM with the training dataset.
- Predict and cluster the testing dataset labels with the trained network model.

The encoding part is accountable for predicting the sign or voice data's most influencing or essential features. However, the encoder and decoder decrease the feature space, and the chosen features are used for clustering. The encoder then diminishes the full features from the most critical input data components. Subsequently, the decoder considers the diminished set of influencing features and intends to reconstruct initial values devoid of losing the information. The encoding pair forms the mechanism for diminishing the data dimensionality for clustering the clustered data. The objective of knowledge tracing relies on the students' past status. While considering status, which is not connected with time series; however, it varies based on the learning ability. Generally, students learn gradually; therefore, while tracing students' knowledge state, the consequences of time series are also considered. The learning level is updated constantly as it understands the related knowledge concept within a specific time and forgets it. Thus, the stacked LSTM model analyses the student's sequence.

3.3 Embedding Features

The concept of determining the multiple features with SMOTE measures the learning performance. Initially, to eliminate the unit restriction of every feature and transform it to a dimensional less and the numerical value of every sequential feature $s(i, j) = \{f_1, f_2, \ldots, f_n\}$ is normalized. The feature sequence is converted as in Eq. (8):

$$f_i' = \frac{f_i - \bar{f}}{m}; \quad \bar{f} = \frac{1}{n}\sum_{i=1}^{n} f_i, \quad m = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - \bar{f})^2} \tag{8}$$

Here, $\bar{f}$ specifies mean, m specifies standard deviation, and f_i' specifies new features after pre-processing. Some new sequences $s'_{(i,t)} = \{f_1', \ldots, f_n'\}$ and feature matrix.

$$S_i' = [s'_{(i,1)}, \ldots, s'_{(i,t)}]^T \in R^{t*n} \tag{9}$$

The covariance matrix D is expressed as in Eq. (10):

$$D = \frac{1}{t} S_i' S_i^T \in R^{n*n} \tag{10}$$

The eigenvectors and their corresponding values are evaluated via D. The eigenvalues are sorted from the largest to the smallest. Thus, the corresponding eigenvectors are sorted, and this method chose the initial k highest eigenvalues and related eigenvalues to form the matrix $P \in R^{n*k}$. At last, the final matrix $X_i = S_i' P, \ S_i \in R^{t*k}$ is computed. Here, SMOTE is exploited for reducing the feature dimensionality and captures the matrix form of $X_i = x_1, x_2, \ldots, x_t$ that specifies the student's record from time $1 \rightarrow t$ where $x_i = Dp_i, \ i = 1, 2 \ldots, t, \ l = 1, 2 \ldots, k$ and p_l defines the l^{th} eigenvalue of D. After attaining every feature representation from the embedded features $x_1, \ldots, x_t$ specifies the input to stacked LSTM for training purpose, provisioning students response prediction y_{ij} at $t + 1$. It is expressed as in Eqs. (11) to (16):

$$i_t = \sigma\left(w_i x_t + u_i h_{t-1} + b_i\right) \tag{11}$$

$$c_t = \tan(w_c x_t + u_c h_{t-1}, \ \ bc) \tag{12}$$

$$f_i = \sigma\left(w_f x_t + u_f h_{t-1} + b_f\right) \tag{13}$$

$$c_t = i_t \tilde{c}_t + f_t c_{t-1} \tag{14}$$

$$o_t = \sigma(w_0 x_t + u_0 h_{t-1} + v_o c_t + b_0) \tag{15}$$

$$h_t = o_t \tanh(c_t) \tag{16}$$

Here, i_t, f_t, o_t specifies the input, forget and output gate, $\tilde{c}_t$ specifies the cell state where the input feature vectors move via the input gate at 't' time, and c_t defines the cell state integration with essential information forget and input gate. w_i, w_c, w_f, w_0, u_i, u_c, u_f, u_0, v_0 specifies weight coefficients and b_i, b_c, b_f, b_0 specifies bias. Various model parameters, and $\sigma(x)$ and $\tan(x)$ define the non-linear activation function. Initially, parameters are initialized; the model evaluates the hidden state (students). With $t+1$, the students' form specifies the weighted sum aggregation of historical conditions during the training process. In the successive step $t+1$, the student's attention state vector is depicted as in Eq. (17):

$$h_{attention} = \sum_{j=1}^{t} \alpha_j h_j \alpha_j = softmax\ (h_j) \tag{17}$$

Here, h_j specifies the hidden state at j and softmax is known as the activation function, α_j specifies the attention score for evaluating the features. After analyzing the attention state of students $h_{attention}$ at $t+1$, merging the present input x_{t+1} to the output of students response r_{t+1}. It is expressed as in Eq. (18):

$$y_{t+1} = \sigma\ (w_h x_{t+1},\ \ w_l h_{attention} + b_t) \tag{18}$$

$$r_{t+1} = softmax\ (y_{t+1}) \tag{19}$$

Here, y_{t+1} specifies the overall prediction process at *the* $t+1$ exercise step. $\{W_h,\ \ W_l,\ \ b_l\}$ sets some parameters. It is highly solicited to convert the actual label to a 2D vector using one-hot encoding to make a probable comparison. The values $[0,\ 1]$ specify that the response is appropriate, and $[1,\ 0]$ specifies the response is inappropriate. The left-side element specifies the probability of the wrong response, and the right side element specifies the appropriate response from the students. The output data using the actual label is used for evaluating the loss function. It is depicted using the cross-entropy, and the expression is provided as in Eq. (20):

$$\mathcal{L} = \quad - \quad \sum_{t=1}^{T} [r'_t . \log r_t + (1 - r'_t) . \log(1 - r_t)] \tag{20}$$

At $t-$ time, r'_t specifies the actual score and r_t specifies the anticipated score of the expected model. The proposed model reduces the loss function using the prevailing optimizers. Here, the dataset attributes are considered features, and the labels are exam scores (math, reading and writing scores).

3.4 Stacked LSTM

The stacked LSTM is efficiently used for resolving the gradient explosion with the set of memory units as in Fig. 1. It facilitates the network to learn the trust value of cooperative neighbourhood nodes and when to forget the prior network information of the memory unit (it holds the essential information) and provides the fact regarding when to update the memory unit with further new information. The memory unit preserves all the historical network information (pattern analysis, traffic flow, source and destination nodes, related information, cooperative details, previous network connection, and further connection establishment). All three gates manage it. This model is well suited for incoming data analysis and previously available datasets. The relationship and the dependencies among the incoming data are analyzed in time steps. To perform this function, any input dataset is considered. The dataset is partitioned into training and validation sets with $(_{Regular\ data\ (Abnormal}\ and\ D_{valid})$ and holds some abnormal data $(D_{abnormal})$. In the real-time environment, anomalous samples are relatively lesser in number. The stacked LSTM model predominantly uses regular data to train hyper-parameter determined by the validation set. The prediction outcomes of the normal and abnormal data are attained concurrently. The difference between the real and predicted data is made, and the errors are identified. Consider error at every point in the test samples as the attributes of those error datasets. Here, the error dataset is partitioned into a training and testing set. The labels specify '0' as normal flow without any error or interruption, and '1' determines the abnormal functionality, identifies the feature and fails to provide the prior students learning details.

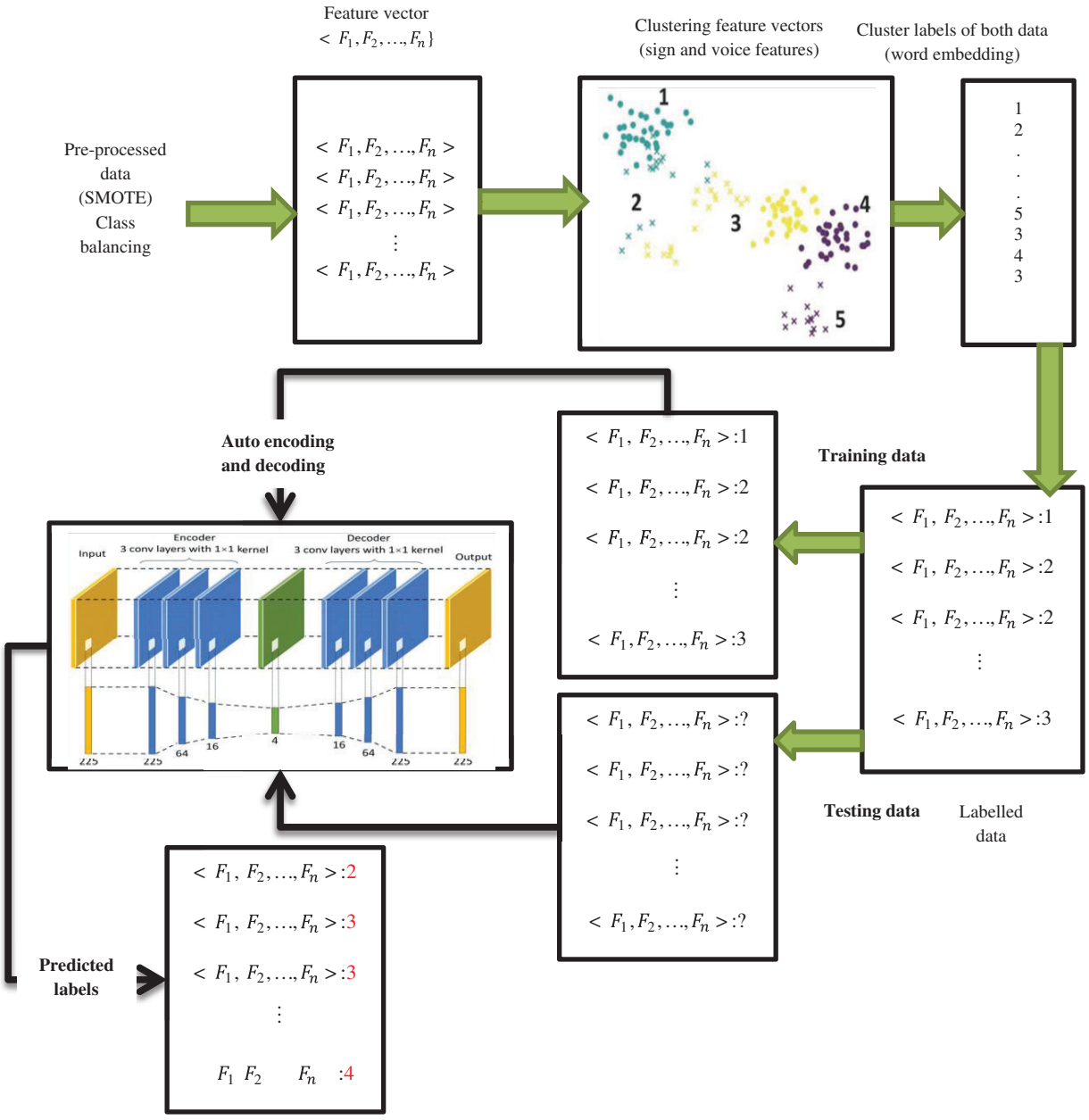

Figure 1: Block diagram of stacked LSTM model for feature learning

The fault or error over the incoming data flow is subjected to the Gaussian distribution. However, the storage assumptions are highly efficient and provide robust outcomes. Here, Gaussian probability distribution is used to identify the attributes in the presence of a specific class label. It is expressed as $p(x|y = 1)$, where 'x' and 'y' specify the samples and corresponding labels. The LSTM generated a sequence vector and was used as an input to the successive layers of LSTM. The previous step feedback captures the routing details (from the memory)/feature patterns. This hierarchical and stacked network is used to handle the complex representation of the dataset information at different network scaling perspectives. The dropout layer of the network excludes 5% of neurons to avoid the under-fitting and over-fitting issues. The proposed stacked LSTM model ingests essential information and extracts the

hidden patterns from the available variables, and efficiently identifies the establishment factors. The proposed stacked model has the competency of dealing with long and short term dependency based on the network lifetime (validate the active and passive nodes over the network). The convergence rate is based on input ti_t, output o_t, and forget f_t gate. It is expressed as in Eqs. (21)–(25):

$$f_t = g(W_f.x_t + U_f.h_{t-1} + b_f) \tag{21}$$

$$i_t = g(W_i.x_i + U_i.h_{t-1} + b_i) \tag{22}$$

$$c_t = f_t.c_{t-1} + i_t.k_t \tag{23}$$

$$o_t = g(W_0.x_t + U_o.h_{t-1} + b_0) \tag{24}$$

$$h_t = o_t.\tanh(c_t) \tag{25}$$

Here, i_t is the input vector; g is the activation function; W is a weighted vector, and C_t is a memory cell.

4 Numerical Results and Discussion

Based on the above methodology, it is known that the anticipated model is composed of two diverse sub-tasks known as feature learning and classification. Here, 70% of data is considered for training and 30% of data is used for testing. The anticipated model adopts gradient descents for training the stacked LSTM model with a 0.01 learning rate and 100 epochs (mini-batches). The loss of function and accuracy need to be monitored. The training loss is reduced, and the accuracy is increased for all the epochs (refer to the Figs. 2 to 4). The accuracy and epochs are observed at the peak during the successive epochs. After the 10^{th} epochs, the proposed stacked LSTM model initiates the training data optimization process. The anticipated model is trained from the beginning and evaluated during the testing process to avoid over-fitting issues. Here, cross-entropy is considered as the loss function, and it is expressed as in Eq. (26):

$$E(y,\ \ y') = -\sum y\ (l) \log y' l \tag{26}$$

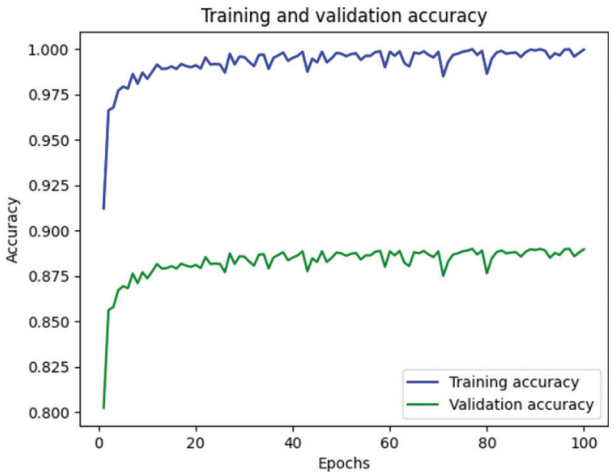

Figure 2: Training and validation accuracy

While validating the multi-classification model, the proposed stacked LSTM model needs to produce the probability of every class where the target class possesses the highest probability. Here, y and y' specify the expected and predicted possibility for the given label 1. The softmax function is utilized as an activation function over the stacked LSTM layers. The significant causes of using the softmax functions are to

produce the probability range as an output from 0 to 1, and the sum of probabilities needs to be 1. The model alike of the softmax function makes the output in diverse ranges and aligns the result ranges from 0 to 1 while predicting the target class. The softmax function is provided at the output layer, and it is expressed as in Eq. (27):

$$Softmax\ (h_t') = \frac{\exp(h_t')}{\sum_{k=1}^{k} \exp(h_t)} \tag{27}$$

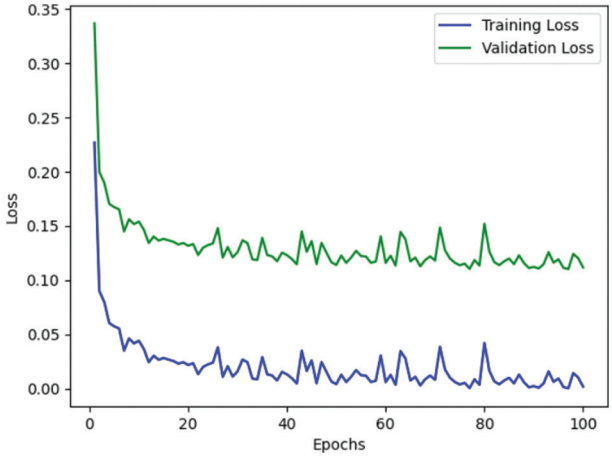

Figure 3: Training loss and validation loss

Here, h_t' specifies the final hidden state (stacked LSTM output) later that ranges from $-\infty$ to $+\infty$. The stacked LSTM works better than individual LSTM in layered performance for the next tasks. To compute the model's performance, some essential metrics are evaluated, and it is mathematically expressed as in Eqs. (28)–(31):

$$Precision = \frac{TP}{TP + FP} \tag{28}$$

$$Recall = \frac{TP}{TP + FN} \tag{29}$$

$$F1\text{–}score = \frac{2 * precision * recall}{(precision + recall)} \tag{30}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{31}$$

The outcomes of the various word embedding model are depicted in Table 2. The anticipated model provides reasonable classification accuracy when word embedding is utilized as evaluated to pre-train the word. The initial layer output of feature extraction is shown in Table 2. The evaluated F1-score for feature extraction is predicted with the recall and precision of 86% and 83%, respectively. The outcomes are demonstrated for all categories. It is observed that the model shows a lesser F1-score in the evaluation category as the training instances for evaluation are lesser in number and because stacked LSTM has not learned the aspect label category. Therefore, the model provides more training samples for all the categories. Thus, the score is improved further. The performance of the successive layers during the prediction process with various recall and precision are considered for the targeted categories. The outcomes are mentioned in Table 3. The F1 values range from 87% to 90%, with an average value of

88%. The model performance is compared with some baseline investigations performed over the academic domain, as depicted in Tables. 2 and 3.

Table 2: Overall performance comparison with deep learning approaches

Existing approaches	Prediction accuracy	Precision	Recall	F1-score
LSTM + embedding process	77%	90%	62%	73%
LSTM + attention process	76%	82%	85%	83%
Fused LSTM	75%	89%	83%	85%
Aspect attention + GRU	84%	93%	90%	91%
Layered LSTM	85%	87%	88%	87%
Proposed stacked LSTM + auto-encoder	89%	83%	86%	87%

Table 3: Overall performance comparison with the machine and deep learning approaches

Existing approaches	Prediction accuracy	Precision	Recall	F1-score
Sentiment analyzer	–	69%	76%	72%
Supervised SVM	78%	–	–	–
Naïve bayes	89%	–	–	–
Naïve bayes + Lexicon	80	–	–	–
Bi-directional SVM	80	–	–	–
Proposed stacked LSTM + auto-encoder	89%	83%	86%	87%

The model outperforms the baseline classifiers in feature learning and classification tasks, where the model attains 89% accuracy during the detection task and 90% accuracy in feature extraction. Some investigations evaluate the model over the standard dataset, including restaurant and laptop reviews. To validate the performance of the proposed model, it is applied over various domains by slight variation using the input and output parameters. Table 3 depicts the parameter evaluation of the standard set. It is observed that the dataset shows five diverse labelled categories and classification outcomes where the output parameters are changed based on feature learning and classification from layers 5 to 4 and 4 to 3 layers. The parameters are maintained alike in Table 3. However, some pre-processing steps (SMOTE) are applied using the benchmark dataset before feeding it to the stacked LSTM model as without these pre-processing steps. The model performance is degraded. Table 2 depicts the performance of the anticipated stacked LSTM model using the standard dataset with the various existing approaches. The model shows superior performance using the common dataset with 89% accuracy during the detection process and an F1 score of 87% during feature analysis. From all these observations (See Figs. 4–7), it is known that the model works over various feature selection and detection processes effectually.

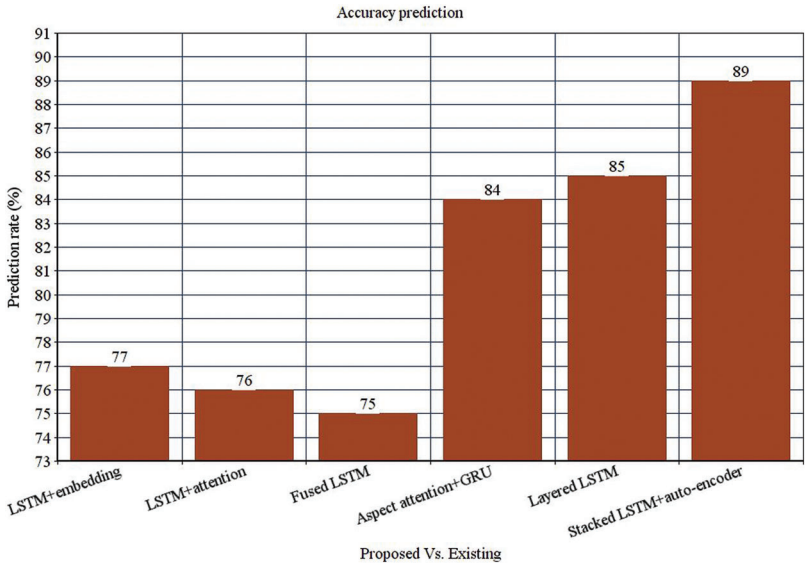

Figure 4: Accuracy prediction

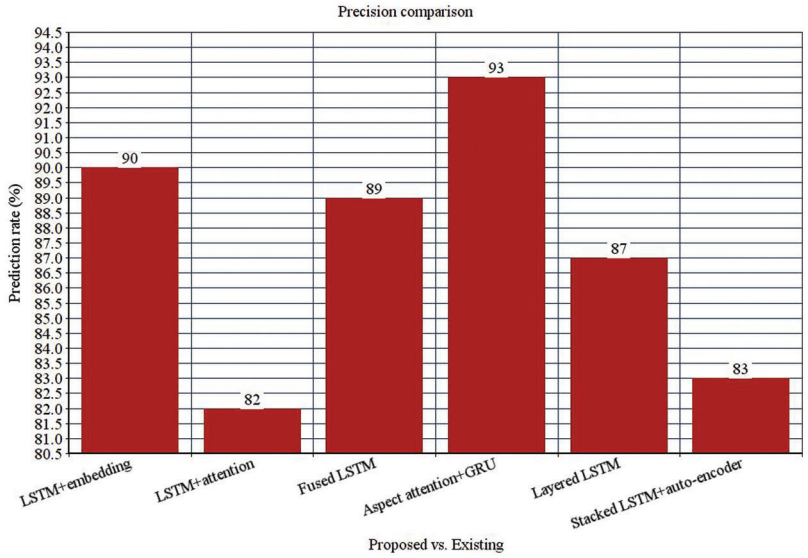

Figure 5: Precision comparison

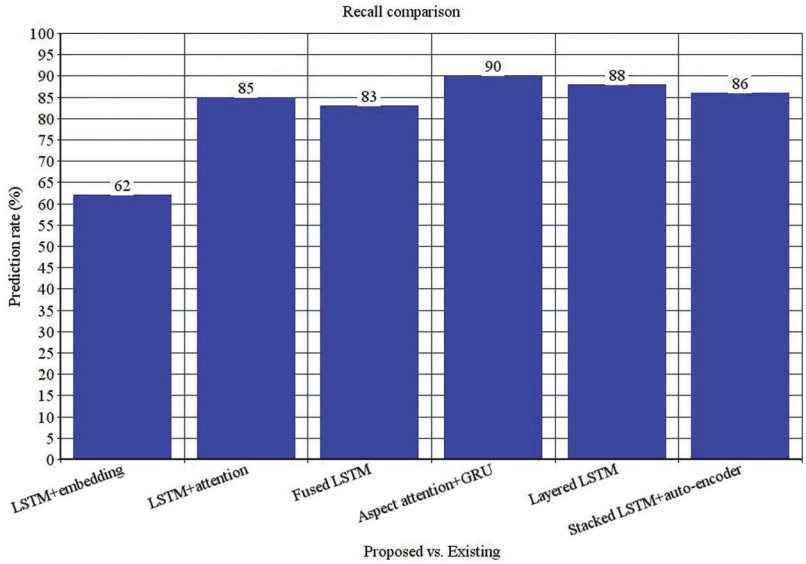

Figure 6: Recall comparison

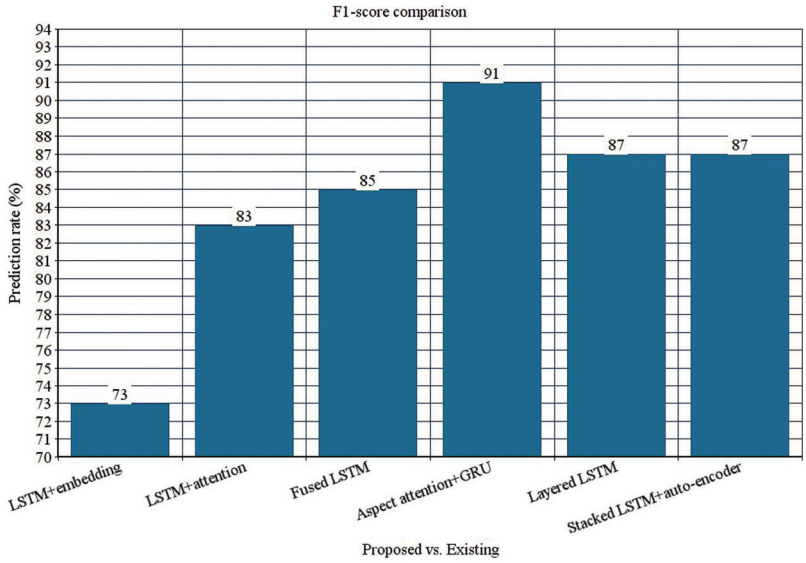

Figure 7: F1-score comparison

5 Conclusion

The evaluation of students' performance is done with various aspects, where some manual computation is also done in the real-time evaluation process. A novel stacked LSTM model is used for the automatic feature learning process with the available data and classification. The prediction framework applies SMOTE for pre-processing and stacked LSTM for the classification process. The features are learnt using the auto-encoder concept to measure the influencing features over the supportive learning process. The simulation is done in MATLAB 2020a environment and various metrics like accuracy, precision, F1-score and recall. The proposed model gives 89% accuracy, 83% precision, 86% recall, and 87% F-score. The proposed model offers satisfactory outcomes compared to various existing approaches. However, the

model encounters constraints like data acquisition with constant labels as the classification criteria may change among the standard dataset. In the future, the construction of a real-time dataset is highly solicited to boost the performance of the proposed classifier model.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Altrabsheh, M. Cocea, S. Fallahkhair and K. Dhou, "Evaluation of the SA-E system for analysis of students," in *Proc. IEEE 17th Int. Conf. on Advanced Learning Technologies (ICALT)*, Timisoara, Romania, pp. 60–61, 2017.

[2] A. Mukherjee and B. Liu, "Aspect extraction through semi-supervised modeling," in *Proc. 50th Annual Meeting Association for Computational Linguistics*, Jeju Island, Korea, vol. 1, pp. 339–348, 2012.

[3] A. Cahyadi and M. L. Khodra, "Aspect-based sentiment analysis using convolutional neural network and bidirectional long short-term memory," in *Proc. 5th Int. Conf. on Advanced Informatics: Concept Theory and Applications (ICAICTA)*, Krabi, Thailand, pp. 124–129, 2018.

[4] A. Ortigosa, J. M. Martín and R. M. Carro, "Sentiment analysis in facebook and its application to e-learning," *Computers in Human Behavior*, vol. 31, pp. 527–541, 2014.

[5] L. C. Yu, C. W. Lee, H. I. Pan, C. Y. Chou, P. Y. Chao *et al.,* "Improving early prediction of academic failure using sentiment analysis on self-evaluated comments," *Journal of Computer Assisted Learning*, vol. 34, no. 4, pp. 358–365, 2018.

[6] F. F. Balahadia, M. C. G. Fernando and I. C. Juanatas, "Teacher's performance evaluation tool using opinion mining with sentiment analysis," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, Bali, Indonesia, pp. 95–98, 2016.

[7] G. S. Chauhan, P. Agrawal and Y. K. Meena, "Aspect-based sentiment analysis of students' feedback to improve teaching–learning process," in *Proc. Information and Communication Technology for Intelligent Systems*, Singapore, pp. 259–266, 2019.

[8] T. Young, D. Hazarika, S. Poria and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Computational Intelligence Magazine*, vol. 13, no. 3, pp. 55–75, 2018.

[9] Y. Ma, H. Peng, T. Khan, E. Cambria and A. Hussain, "Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis," *Cognitive Computation*, vol. 10, no. 4, pp. 639–650, 2018.

[10] Z. Toh and W. Wang, "Dlirec: Aspect term extraction and term polarity classification system," in *In Proc. 8th Int. Workshop on Semantic Evaluation*, Dublin, Ireland, pp. 235–240, 2014.

[11] Y. Ding, J. Yu and J. Jiang, "Recurrent neural networks with auxiliary labels for cross-domain opinion target extraction," in *Proc. AAAI Conf. on Artificial Intelligence*, Austin, Texas, vol. 31, no. 1, pp. 3436–3442, 2017.

[12] M. Carlsson, G. B. Dahl, B. Öckert, and D. O. Rooth, "The effect of schooling on cognitive skills," *Review of Economics and Statistics*, vol. 97, no. 3, pp. 533–547, 2015.

[13] S. J. Ritchie, T. C. Bates and I. J. Deary, "Is education associated with improvements in general cognitive ability, or in specific skills?," *Developmental Psychology*, vol. 51, no. 5, pp. 573, 2015.

[14] C. Granberg and J. Olsson, "ICT-Supported problem solving and collaborative creative reasoning: Exploring linear functions using dynamic mathematics software," *The Journal of Mathematical Behavior*, vol. 37, pp. 48–62, 2015.

[15] D. Tod, C. Edwards, M. McGuigan and G. Lovell, "A systematic review of the effect of cognitive strategies on strength performance," *Sports Medicine*, vol. 45, no. 11, pp. 1589–1602, 2015.

[16] I. Fister, P. N. Suganthan, S. M. Kamal, F. M. Al-Marzouki, M. Perc *et al.,* "Artificial neural network regression as a local search heuristic for ensemble strategies in differential evolution," *Nonlinear Dynamics*, vol. 84, no. 2, pp. 895–914, 2016.

[17] A. Daud, N. R. Aljohani, R. A. Abbasi, M. D. Lytras, F. Abbas *et al.,* "Predicting student performance using advanced learning analytics," in *Proc. 26th Int. Conf. on World Wide Web Companion*, Perth, Australia, pp. 415–421, 2017.

[18] K. R. Koedinger, A. T. Corbett and C. Perfetti, "The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning," *Cognitive Science*, vol. 36, no. 5, pp. 757–798, 2012.

[19] A. M. Shahiri, W. Husain and N. A. Rashid, "A review on predicting student's performance using data mining techniques," *Procedia Computer Science*, vol. 72, pp. 414–422, 2015.

[20] J. McLurkin, J. Rykowski, M. John, Q. Kaseman and A. J. Lynch, "Using multi-robot systems for engineering education: Teaching and outreach with large numbers of an advanced, low-cost robot," *IEEE Transactions on Education*, vol. 56, no. 1, pp. 24–33, 2013.

[21] J. L. M. Nunez, E. T. Caro and J. R. H. Gonzalez, "From higher education to open education: Challenges in the transformation of an online traditional course," *IEEE Transactions on Education*, vol. 60, no. 2, pp. 134–142, 2017.

[22] J. Y. Chung and S. Lee, "Dropout early warning systems for high school students using machine learning," *Children Youth Services Review*, vol. 96, pp. 346–353, 2019.

[23] J. Kuzilek, M. Hlosta, D. Herrmannova, Z. Zdrahal and A. Wolff, "OU analyse: Analysing at-risk students at the open university," *Learning Analytics Review*, vol. 8, pp. 1–16, 2015.

[24] Y. Cui, F. Chen and A. Shiri, "Scale up predictive models for early detection of at-risk students: A feasibility study," *Information and Learning Sciences*, vol. 121, no. 3–4, pp. 97–116, 2020.

[25] T. Soffer and A. Cohen, "Students' engagement characteristics predict success and completion of online courses," *Journal of Computer Assisted Learning*, vol. 35, no. 3, pp. 378–389, 2019.

[26] R. Baker, B. Evans, Q. Li and B. Cung, "Does inducing students to schedule lecture watching in online classes improve their academic performance? an experimental analysis of a time management intervention," *Research in Higher Education*, vol. 60, no. 4, pp. 521–552, 2019.

[27] J. M. Lim, "Predicting successful completion using student delay indicators in undergraduate self-paced online courses," *Distance Education*, vol. 37, no. 3, pp. 317–332, 2016.

[28] S. Lee and J. Y. Chung, "The machine learning-based dropout early warning system for improving the performance of dropout prediction," *Applied Sciences*, vol. 9, no. 15, pp. 3093, 2019.

[29] A. Behr, M. Giese and K. Theune, "Early prediction of university dropouts—A random forest approach," *Jahrbücher für Nationalökonomie und Statistik*, vol. 240, no. 6, pp. 743–789, 2020.

[30] A. Ortigosa, R. M. Carro, J. Bravo-Agapito, D. Lizcano, J. J. Alcolea *et al.,* "From lab to production: Lessons learnt and real-life challenges of an early student-dropout prevention system," *IEEE Transactions on Learning Technologies*, vol. 12, no. 2, pp. 264–277, 2019.

[31] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021. https://doi.org/10.1109/TITS.2021.3130403

[32] W. Sun, L. Dai, X. R. Zhang, P. S. Chang, and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, pp. 1–16, 2021. https://doi.org/10.1007/s10489-021-02893-3

[33] P. Sushmitha, "Face Recognition Framework based on Convolution Neural Network with modified Long Short Term memory Method," *Journal of Computational Science and Intelligent Technologies*, vol. 1, no. 3, pp. 22–28, 2020. https://doi.org/10.53409/mnaa.jcsit20201304

[34] M. B. Sudhan, T. Anitha, M. Aruna, G. C. P. Latha, A. Vijay *et al.,* "Weather forecasting and prediction using hybrid C5.0 machine learning algorithm," *International Journal of Communication Systems*, vol. 34, no. 10, pp. e4805, 2021. https://doi.org/10.1002/dac.4805

[35] R. Khilar, K. Mariyappan, M. S. Christo, J. Amutharaj, T. Anitha *et al.,* "Artificial Intelligence-based security protocols to resist attacks in Internet of Things," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1440538, pp. 1–10, 2022. https://doi.org/10.1155/2022/1440538