

Optimized ensemble modeling approach for student cumulative grade point average prediction using regression models

Hemalatha Gunasekaran¹, Rex Macedo Arokiaraj¹, Angelin Gladys Jesudoss¹, Deepa Kanmani²

¹College of Computing and Information Sciences, University of Technology and Applied Sciences, Ibri, Sultanate of Oman

²Department of Information Technology, Sri Krishna College of Engineering and Technology, Coimbatore, India

Article Info

Article history:

Received Jun 10 2024

Revised Jun 25, 2025

Accepted Jul 10, 2025

Keywords:

Bayesian optimization

Ensemble models

GridSearchCV

Predictive model

Regression model

Student grade prediction

ABSTRACT

This research focuses on developing models to accurately predict student's cumulative grade point average (CGPA) in the early stages of their study to tackle the problem of dropout rates in educational institutions. The state-of-the-art methods address CGPA prediction as a classification problem, providing only an approximate prediction where precise prediction is essential. In this research, six regression models, namely linear regression, support vector regression (SVR), decision tree (DT), random forest (RF), lasso regression (LR), and ridge regression (RR) are developed without optimization and later fine-tuned using Bayesian optimization (BO) and GridSearchCV. BO efficiently searches the hyper-parameter space using probabilistic distribution's function, whereas GridSearchCV exhaustively searches the hyper-parameter space. These techniques significantly improved the model's performance; SVR achieved an R^2 score of 94.11% through BO. Ensemble techniques, such as stacking, voting, and boosting, can further enhance the predictive capability of the model. The stacking ensemble model achieved the highest R^2 score of 94.45%, providing a 0.50% improvement in the R^2 score. The findings of this study suggest that advanced optimization and ensemble techniques can substantially enhance the predictive capability of the model, thus enabling institutions to support students at risk of academic probation proactively.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Hemalatha Gunasekaran

College of Computing and Information Sciences, University of Technology and Applied Sciences

Ibri, Sultanate of Oman

Email: hemalatha.david@utas.edu.om

1. INTRODUCTION

Regularly analyzing and monitoring student's performance in any educational institution is crucial, as the reputation and growth of the institution largely depend on the quality and success of its students. Both the students and the institution are interrelated as they influence each other's growth. Students typically exhibit high levels of motivation and engagement at the commencement of the degree program, but some students experience a decline in focus as time progresses. This decline in motivation often results in low academic performance, which can lead to a decline in their cumulative grade point average (CGPA) [1], [2].

In the Middle East, most of the colleges and universities often implement academic probation policies. When a student's CGPA falls under 2.0, they are placed on academic probation. During this probationary period, students are assigned a reduced workload and given a fixed number of attempts to improve their CGPA. However, if a student is unable to demonstrate significant improvement and fails to meet the probationary requirements after the predetermined number of attempts, they may face expulsion from the college. The attrition rate of students due to academic probation and subsequent expulsion will have

a significant impact on educational institutions, affecting reputation, funding, and overall performance metrics. To mitigate the potential negative impact of a high attrition rate, institutions should strive to monitor and predict student performance closely, thereby proactively identifying those at risk of academic probation or expulsion. By leveraging machine learning algorithms and predictive analytics, educators and administrators can accurately predict students' CGPA in advance.

Considerable research has been conducted in machine learning [3]–[13] and, mostly through classification methods, to track the CGPA of students by classifying students' academic performance into categorical values that could be classified as high, medium, and low-level. Classification methods primarily relied on prior grades, attendance, and engagement of students to identify students at risk of probation. However, it provides merely approximate estimations of the students' performance by indicating categorical level whereas, CGPA is a numeric component. A limited amount of studies use regression models [14]–[16] to generate exact predictions of CGPA. However, the use of ensemble techniques to enhance the predictive performance of the model remains underutilized.

This paper focuses on developing a predictive model for student's academic performance, specifically their CGPA. The following are the objectives of this research work: i) to predict students' CGPA accurately using various regression models, including linear regression, support vector regression (SVR), decision tree (DT), random forest (RF), lasso regression (LR), and ridge regression (RR); ii) to optimize the model's hyper-parameters using Bayesian optimization (BO) and GridSearchCV to improve predictive accuracy; iii) to use ensemble techniques to boost the predictive capability of the model; and iv) to evaluate the effectiveness of ensemble models in forecasting the CGPA accurately. The research methodology will involve the careful selection and preprocessing of the dataset, followed by the implementation of a series of regression modeling techniques, including linear regression, support vector classifier (SVC), DT, RF, LR, and RR. Enhancing the model performance through ensemble methods, including voting, stacking, bagging, and boosting. The ensemble technique combines multiple individual models to improve the predictive capability and robustness of the model. By aggregating the strengths of different models, the ensemble technique can reduce errors and increase reliability. Overall, this research endeavors to provide valuable insights into the optimal predictive modeling approach for CGPA prediction, shedding light on the most effective techniques and methodologies for forecasting student's academic performance.

The rest of the study is structured as follows. Section 2 lists relevant literature. Section 3 introduces the dataset and research technique. Section 4 contrasts the outcomes of different models. Lastly, section 5 concludes and outlines future directions.

2. RELATED WORKS

Yağcı [3] employed data mining techniques to predict students' academic performance. The author has used midterm grades to predict final grades. Predictions are made using machine learning algorithms such as RF, k-nearest neighbor (KNN), support vector machine (SVM), logistic regression, and naïve Bayes, and their performance is compared. The author has utilized three types of features: midterm marks, departmental data, and faculty data to identify the target variable (CGPA). The author achieved an accuracy of 70-75% for the proposed model.

Baashar *et al.* [4] predicted the CGPA of postgraduate students using various machine learning algorithms. The author used a real dataset of 635 students from a private university in Malaysia. Among the six different machine learning models, namely artificial neural network (ANN), least squares regression, SVM, DT, Gaussian process regression (GPR), and ensemble model, ANN achieved the best R^2 score of 89%. In contrast, GPR achieved 71%.

Bujang *et al.* [5] predicted the final grade of 1st-year students using six machine learning models. The author created two multi-class machine learning models, one with and one without synthetic minority oversampling technique (SMOTE), incorporating feature selection. The author found that the RF algorithm has the highest F1-score of 99.5% after applying SMOTE on the imbalanced dataset.

Said *et al.* [6] predict the final grade of students using nine machine learning classification models. The author used a real student dataset from Saudi University and implemented a majority voting (MV) algorithm. The extra tree (ET) algorithm achieved an accuracy of 82.8%, and the MV model achieved 92.7% and outperformed all the other models.

Nachouki and Naaj [7] proposed a CGPA prediction model (CPM) that predicts students' CGPA using second- and third-year courses. The author utilized the RF machine learning model on 105 student records, achieving an accuracy of 92.87%. Alangari and Alturki [8] predicted the performance of the students (GPA) using 15 classification algorithms. Among 15 classification models, the naïve Bayes and Hoeffding tree obtained the highest accuracy of 91%.

Ibrahim and Ahmed [9] predicted the CGPA of students in two scenarios: one with the first three years' grades and another with the first two years' grades. The dataset consisted of student data collected

from Comboni College of Science and Technology, Sudan, between 2007 and 2015. The author has utilized the J48 algorithm, achieving an accuracy of 83.333% for the first scenario and 81.0345% for the second scenario.

Korchi *et al.* [10] predicted the performance of the students using machine learning models such as DT, RF, linear regression, KNN, extreme gradient boosting (XGBoost) and deep neural network. The author used 1,000 students' records, which included the marks in maths, reading, and writing. The deep neural network outperformed other models, achieving a determination coefficient of 99.97%.

Iqbal *et al.* [11] predicted students' grades in various courses using collaborative filtering (CF), matrix factorization (MF), and restricted Boltzmann machine (RBM). The dataset used is real-world data collected from Information Technology University (ITU), Pakistan. The author found that the RBM technique outperformed other prediction techniques.

Most research in the field of student CGPA prediction approaches it as a classification problem [6]–[8], [12], [13] rather than a regression problem [14]–[16]. The existing models categorize the students into different groups based on the predicted CGPA, rather than providing a precise numerical prediction of their actual CGPA. By treating the CGPA prediction as a classification problem, the models are making approximate predictions rather than accurate ones. This is because they are not considering the continuous nature of CGPA values and are instead focusing on placing students into predefined categories. While classification models may still provide valuable insights and help in identifying trends and patterns in CGPA data, they may not be the most effective approach for accurately predicting individual student CGPA scores. Regression models, on the other hand, are specifically designed to provide precise numerical predictions and may be more suitable for this purpose. The summary of existing methods is given in Table 1.

Table 1. Comparison between CGPA predictive techniques

| Reference | Features used | Methods | No. of records | Best performance | Limitation |
|---------------------------|---|--|----------------|------------------|---|
| Yağcı [3] | Mid-term marks, departmental data, and faculty data | RF, NN, LR, SVM, NB and KNN | 1854 | Accuracy- 74% | An accurate prediction of CGPA was not done only range was predicted. |
| Baashar <i>et al.</i> [4] | Gender, race, program name, sponsorship, attendance | ANN, LSR, SVR, DT, GPR, and Bagged trees ensemble models | 635 | Accuracy-89% | The dataset used is very small. |
| Bujang <i>et al.</i> [5] | Continuous assessment marks | DT (J48), SVM, NB, KNN, LR, and RF | 1282 | F1-99.5% | CGPA was treated as a categorical value |
| Said <i>et al.</i> [6] | Demographic, pre-admission, and academic. | ET and MV | | Accuracy-92.7% | CGPA was treated as a categorical value |
| Nachouki and Naaj [7] | CGPA for second- and third-year and high school average | RF | 105 | Accuracy-94.29% | This study is conducted using a smaller dataset. |
| Alangari and Alturki [8] | Semester CGPA, course grade | NB and Hoeffding tree | 530 | Accuracy-91% | CGPA was treated as a categorical value |
| Ibrahim and Ahmed [9] | Course grades of different subjects | DT (J48) | 522 | 83.333% | CGPA was treated as a categorical value |
| Korchi <i>et al.</i> [10] | Marks in maths, reading, and writing | Deep neural network | 1000 | F1-99.87% | CGPA was treated as a categorical value |
| Iqbal <i>et al.</i> [11] | Age, gender, high school exam scores, region, CGPA | Logistic regression | | Accuracy-83.5% | CGPA was treated as a categorical value |

3. METHODOLOGY

This research aims to predict the final CGPA of undergraduate students at three different stages: after the first-, second-, and third-year of study. To achieve this, we developed three models:

- Model 1: predicts final CGPA using grades from first-, second-, and third-year subjects.
- Model 2: predicts final CGPA using grades from first-, second-year subjects.
- Model 3: predicts final CGPA using grades from first-year subjects.

Each model is evaluated based on the performance metrics to determine its efficiency in predicting the CGPA. Predicting student's performance at the end of every year is essential to analyze the performance of the students and also to identify the students on the borderline and take necessary measures to prevent the dropout rate. Figure 1 illustrates the fundamental steps involved in developing a predictive model.

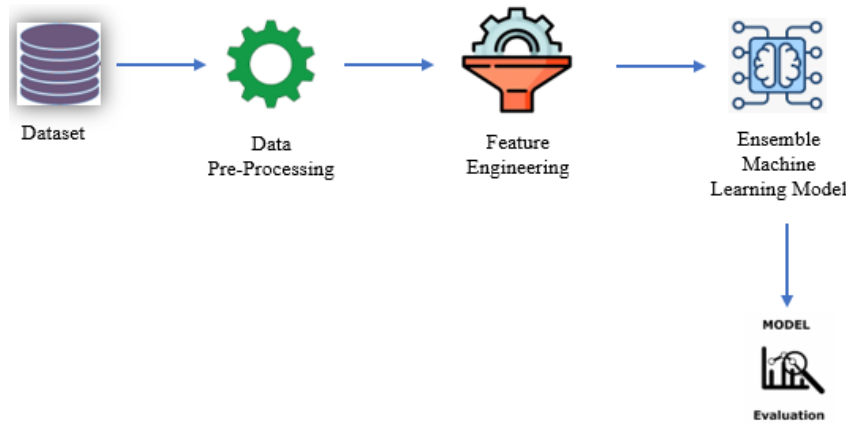


Figure 1. Steps in building a predictive model

3.1. Data collection

The dataset used in this research comprises a collection of academic records from undergraduate students spanning the academic years 2016-2020. It was obtained from the University of Technology and Applied Sciences, Ibri in Oman and includes subject marks of a four-year degree program for 1,259 students. The dataset includes students who have completed their degrees and those who are still registered, as shown in Table 2. Only students with the status "degree completed" are considered for predictive modeling. To graduate, students must complete 128 credit hours. Specialization courses are 3 credits each, and a few introductory courses are 2 credits each. As the university operates on a credit-based system, students at the same level may pursue different subjects. Some first-year courses are introductory courses, and the grades obtained in those courses will not affect the CGPA; those courses are not included in the predictive modeling.

Table 2. Sample dataset

| Student id | Status | CGPA | Sub-1 | Sub-2 | Sub-3 | . | . | . | Sub-43 | Sub-44 |
|------------|------------------|------|-------|-------|-------|---|---|---|--------|--------|
| 2011592001 | Degree completed | 3.04 | B+ | B+ | C+ | . | . | . | B- | A- |
| 2011592002 | Degree completed | 2.74 | C | B | C+ | . | . | . | C- | B |
| 2011592003 | Degree completed | 2.79 | A- | B+ | C- | . | . | . | D | B |
| 2011592004 | Degree completed | 2.28 | C+ | C+ | D | . | . | . | B | C |
| 2011592005 | Degree completed | 2.71 | B- | B- | B | . | . | . | C+ | B+ |
| 2011592008 | Degree completed | 2.75 | B | B+ | C+ | . | . | . | B- | B- |
| 2011592009 | Degree completed | 3.18 | B | A- | C | . | . | . | C+ | B+ |
| 2011592010 | Degree completed | 2.34 | D+ | C- | D | . | . | . | B- | C+ |

3.2. Data preprocessing

The first step in building any machine learning model is to preprocess the raw data. Data pre-processing techniques, such as feature engineering, handling missing values, and data conversion, are applied to the raw data to make it suitable for predictive modeling. The grades of the subjects are converted from categorical values to numerical values based on the criteria given in Table 3.

Table 3. Credit score for grades

| Grade | Scale |
|-------|-------|
| A | 4.0 |
| A- | 3.7 |
| B+ | 3.3 |
| B | 3.0 |
| B- | 2.7 |
| C+ | 2.3 |
| C | 2.0 |
| C- | 1.7 |
| D+ | 1.3 |
| D | 1.0 |
| F | 0 |

3.3. Dataset filtering

Only the records of graduated students are considered for predictive modeling. The students with the status "registered" are excluded from model creation. In the dataset, the column representing the CGPA is considered the target column or dependent column, while the subject grades are regarded as the independent features. The frequency distribution of the CGPA column is shown in Figure 2.

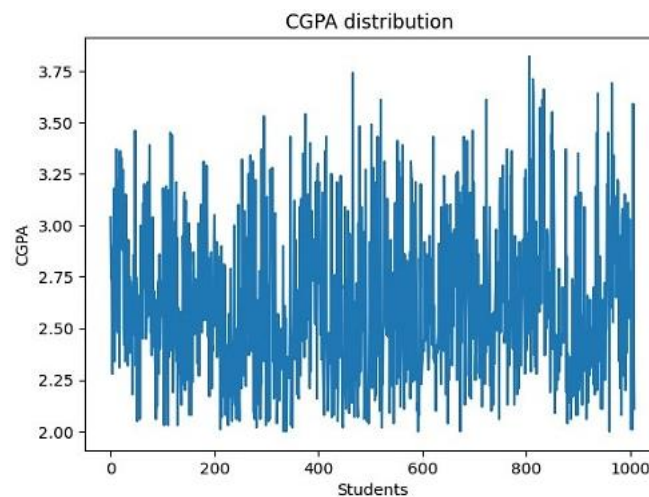


Figure. 2. CGPA distribution of graduated students

3.4. Handling missing values

Some elective courses are opted by very few students, so those columns will have more null values; such features are dropped and not included in predictive modeling. Some subjects will have few null values; those null values are replaced with the average grade of 'C' to avoid bias. This imputation strategy ensures that the model maintains robustness without disproportionately penalizing students who did not select certain electives.

3.5. Handling ordinal values

Each subject contains the grade that was scored in that subject. Grade is an ordinal attribute that is mapped to a numerical value. For example, if 'A' is the highest grade, it could be mapped to 4, and 'F' is the lowest grade, it could be mapped to 0. Replace the ordinal grades with their corresponding numerical values in the dataset to facilitate analysis and model building.

4. PREDICTIVE MODELS

4.1. Linear regression

Linear regression is a machine learning algorithm used to predict continuous variables [17]. It assumes a linear relationship between the target variable and the independent variable. The model estimates the slope and intercept of the line's best fit, as shown in Figure 3, which represents the relationship between the variables using the formula given in (1).

$$Y_i = \beta_0 + \beta_i X_i \quad (1)$$

Where Y_i is dependent variable, β_0 is constant/intercept, β_i is slope/intercept, and X_i is independent variable. The linear regression model is simple and easy to interpret. They require minimal computational resources, but it is very sensitive to the outlier and underperform with non-linear data.

4.2. Support vector regression

SVR is a supervised machine learning model used to predict the continuous target variable. The objective of SVR is to find a hyperplane in a higher-dimensional space that best represents the relationship between the input features and the target variable [18], as shown in Figure 4. The equation of the hyperplane is given in (2).

$$Y = WX + b \quad (2)$$

Where b is the bias term, W is the weight, and X and Y are the input feature and target value, respectively.

Different hyper-parameters can be tuned to improve the performance of the SVR, such as:

- Regularization parameter (C): this parameter controls the trade-off between maximizing the margin of tolerance and minimizing the error. A smaller value of C results in more deviations from the actual values, and a large value of C results in strict adherence to the actual values.
- Epsilon (ϵ): this parameter defines the margin of tolerance between the predicted values and the actual values. Larger values of ϵ result in more significant deviation between the actual and predicted values. Smaller values result in a smaller margin of deviation.
- Gamma: this parameter defines the influence of a single training example on the decision boundary. A higher gamma value yields a narrower range of influence and can lead to overfitting.

SVR performs well with high dimensional data and maintains robustness with numerous features. But it is highly sensitive to the choice of kernel and the C .

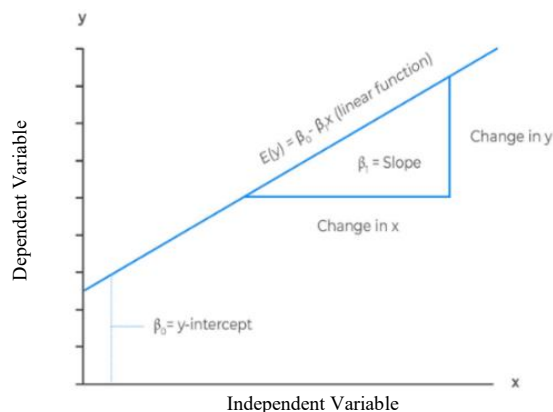


Figure 3. Linear regression

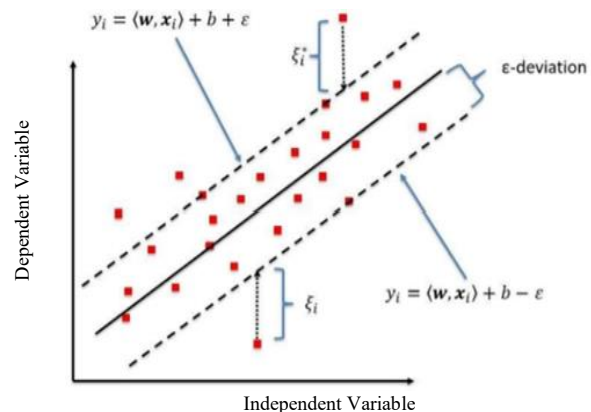


Figure 4. Support vector regression

4.3. Decision tree regression

The DT is a popular machine learning algorithm that can be used for both classification and regression tasks. It is a tree-based predictive model. DTs learn from data to approximate a sine curve, as shown in Figure 5, with a set of if-then-else decision rules [19], [20]. The deeper the tree, the more complex the decision rules and the fitter the model. The `max_depth` parameter controls the maximum depth of the tree; if it is set too high, the DT learns too many fine details, resulting in overfitting. However, a DT cannot effectively capture complex relationships, and it is prone to overfitting.

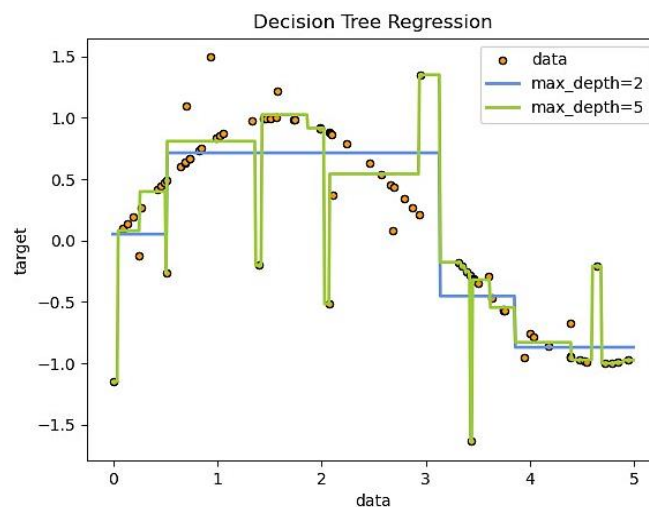


Figure 5. DT regression

4.4. Lasso regression

LR is a linear regression technique that incorporates a penalty term known as the L1 regularization [21]. It is used for feature selection and improving the interpretability of the model. In this method, the objective function is modified by adding the sum of the absolute values of the coefficients, multiplied by a tuning parameter λ . Mathematically, the LR model can be represented by the following formula, as shown in (3).

$$RSS + \lambda * \sum(|\beta_j|) \quad (3)$$

Where RSS is the residual sum of squares and β_j represents the regression coefficients. The λ parameter controls the amount of regularization applied, with higher values resulting in more shrinkage of the coefficients towards zero. The LR model is beneficial in scenarios where the number of predictors is large and a subset of essential variables needs to be selected. By introducing the penalty term, lasso encourages sparsity in the coefficient matrix, effectively shrinking some coefficients to zero and eliminating those variables from the model. This feature selection property makes lasso a valuable tool in situations where the interpretability and simplicity of the model are desirable. Additionally, the introduction of the λ parameter allows for tuning the model's complexity and balancing between bias and variance. However, the lasso model can over-shrink the coefficients, leading to underfitting.

4.5. Ridge regression

RR, also known as Tikhonov regularization, is a linear regression technique that mitigates the effects of multicollinearity in a dataset. It achieves this by adding a penalty term called the L2 regularization to the ordinary least squares objective function [22]. Mathematically, the RR model can be expressed as shown in (4).

$$RSS + \lambda * \sum(\beta_j^2) \quad (4)$$

Where RSS is the residual sum of squares and β_j represents the regression coefficients. The λ parameter controls the amount of regularization applied, with higher values resulting in more significant shrinkage of the coefficients towards zero. The RR model helps to overcome the issue of multi-collinearity, where predictor variables are highly correlated with each other. By adding the L2 regularization term, RR reduces the impact of correlated variables by shrinking their coefficients. This leads to a more stable and robust model. However, the coefficient can still be challenging to predict in high-dimensional space. Unlike LR, RR does not set coefficients to precisely zero. Instead, it shrinks them towards zero, but they remain non-zero. This property of RR allows all predictors to still contribute to the model, with a reduced influence. The λ parameter in RR provides a way to control the trade-off between model simplicity and fitting the training data, enabling a flexible approach to balance bias and variance.

4.6. Ensemble model

Ensemble models are models that combine the predictions of individual machine learning models using techniques such as averaging, voting, or stacking. The four main types of ensemble models are bagging, boosting, voting, and stacking. In bagging, multiple models are trained, each using a subset of the training data. The ensemble model combines the advantages of different models, thereby reducing overfitting. Boosting, on the other hand, also uses multiple models, but each model is trained on the same dataset. The difference is that incorrectly classified instances are given more weight, and subsequent models focus on correcting those errors. The voting ensemble combines the predictions of different predictive models using either soft voting (a weighted average of probabilities) or hard voting (a majority vote). This leverages the collective knowledge of diverse models for a more accurate prediction. Lastly, the stacking ensemble model combines the predictions of the base models using another machine learning model known as a meta-learner. This meta-learner learns how to combine the predictions of base models best. By combining the advantages of several models and mitigating their drawbacks, ensemble models are an effective means of enhancing prediction performance.

5. RESULTS AND DISCUSSION

Regression model fitness can be measured using the metrics R^2 coefficient of determination. The R^2 measures the percentage of variance in the target variable with respect to the independent variable, as shown in (5). The value can range between 0 and 100%. The higher the value, the better the model predicts the target variable. The mean squared error (MSE) calculates the average of the squared differences between the

target variable and the prediction, as shown in (6). The lower value of MSE indicates how the prediction is close to the true value of the variable. Since the difference between the prediction and actual value is squared, extreme values, such as outliers, will have a significant impact on the model's performance. To overcome this limitation, we utilize the root mean squared error (RMSE) metric. RMSE measures the average difference between values predicted by a model and the actual values, as shown in (7). The mean absolute error (MAE) metrics are measured as the average of the absolute error values as shown in (8).

$$\text{The F1 score (R2)} = \frac{2 \times P^R \times R^R}{P^R + R^R} \quad (5)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (t_i - y_i)^2 \quad (6)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (t_i - y_i)} \quad (7)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n (t_i - y_i) \quad (8)$$

Where, y_i is the predicted value of the model, t_i is the actual value and n is the number of samples in the test dataset.

6. MODEL EVALUATION

6.1. Hyper-parameter tuning methods

Hyperparameters are parameters that influence both the training process and the model's performance. The process of searching for the optimal hyper-parameter value combinations for a model is known as hyper-parameter tuning [23], [24]. Hyper-parameter optimization can be represented in the form as shown in (9).

$$x^* = \arg \min_{x \in X} f(x) \quad (9)$$

Where, $f(x)$ represents an objective function to minimize RMSE or MAE or MSE on the validation dataset, x^* is the set of hyper-parameters that yields the lowest value of score and X takes any value from set x^* .

There are many hyper-parameter tuning methods available, such as GridSearchCV, BO, random search, hill climbing, and simulated annealing, to improve the model performance, we proposed BO for tuning the regression models. We also compare BO with GridSearchCV. BO is a tuning method that relies on the Bayesian Gaussian theorem and is contingent upon a prior distribution. This approach combines the prior probability and the posterior probability of the function to evaluate the optimal point of the function. BO is a black box function specifically designed for global optimization techniques. It employs an acquisition function to strike a balance between exploration and exploitation when determining the next hyper-parameter value to evaluate [4].

GridSearchCV is particularly useful for tuning multiple hyperparameters simultaneously, providing an automated approach to model optimization. It exhaustively searches over a grid of hyper-parameters. The hyper-parameters for the six regression models were optimized using BO and GridSearchCV. The best hyper-parameters for each model are listed in Table 4.

Table 4. Hyper-parameters of machine learning models

| Model | Best hyper-parameter using BO | Best hyper-parameter using GridSearchCV |
|-------|------------------------------------|---|
| SVR | C=95.0701 gamma=0.001 ε=0.01 | C=10 ε=0.07411 gamma=0.001 |
| RR | alpha=0.9999 | alpha=10 |
| LR | Alpha=0.0 | Alpha=0.001 |
| RF | n_estimator=10 random_state=42 | n_estimator=300 random_state=42 |
| DT | max_depth=12 random_state=42 | max_depth=9 |

6.2. Predicting student's performance using three-year course grade

The experiment was conducted in Google Colab Pro using Python 3 and a Google Compute Engine backend (GPU-A100) with 40 GB of GPU RAM. Six machine learning models (linear regression, SVR, DT,

RF, LR, and RR) were trained on student's first three-year course marks to predict their final CGPA. The model's performance was evaluated using the metrics MSE, MAE, R^2 , and RMSE. The performance of the models, both with and without optimization using GridSearchCV and BO, is presented in Tables 5 to 7, respectively. Without optimization, linear regression, RR, and LR achieved the highest R^2 score of 93.98%.

Table 5. Performance of regression models using default parameters

| Model | MSE | MAE | R2 | RMSE |
|-------------------|--------|--------|--------|--------|
| Linear regression | 0.0092 | 0.0743 | 0.9398 | 0.0961 |
| RF | 0.0183 | 0.1096 | 0.8801 | 0.1356 |
| SVR | 0.0126 | 0.0890 | 0.9176 | 0.1124 |
| DT | 0.0589 | 0.1870 | 0.6161 | 0.2427 |
| RR | 0.0092 | 0.0743 | 0.9398 | 0.0960 |
| LR | 0.1547 | 0.3230 | -0.008 | 0.3933 |

Table 6. Performance of regression models using GridSearchCV

| Model | MSE | MAE | R2 | RMSE |
|-------------------|--------|--------|--------|--------|
| Linear regression | 0.0092 | 0.0743 | 0.9398 | 0.0961 |
| RF | 0.0185 | 0.1107 | 0.8768 | 0.1363 |
| SVR | 0.0090 | 0.0740 | 0.9400 | 0.0953 |
| DT | 0.0527 | 0.1755 | 0.6563 | 0.2296 |
| RR | 0.0092 | 0.0741 | 0.9400 | 0.0959 |
| LR | 0.0092 | 0.0744 | 0.9395 | 0.0963 |

Table 7. Performance of regression models using BO

| Model | MSE | MAE | R2 | RMSE |
|-------------------|--------|--------|--------|--------|
| Linear regression | 0.0092 | 0.0743 | 0.9398 | 0.0961 |
| RF | 0.0222 | 0.1206 | 0.8552 | 0.1490 |
| SVR | 0.0098 | 0.0742 | 0.9411 | 0.0950 |
| DT | 0.0604 | 0.1900 | 0.6064 | 0.2450 |
| RR | 0.0092 | 0.0743 | 0.9398 | 0.0960 |
| LR | 0.0092 | 0.0744 | 0.9395 | 0.0966 |

After hyper-parameter tuning with GridSearchCV, SVR outperformed the other models with an R^2 score of 94.07%. BO further improved SVR's performance, resulting in an R^2 score of 94.11% due to the tuning of hyper-parameter such as kernel type, C, and ϵ allowed the model to better capture the pattern in the data. SVR also exhibits lower MSE and MAE values, highlighting its strong predictive capability. The evaluation metrics of ensemble models after BO is given in Figure 6. The difference between the actual and the predicted value for the different machine learning models are given in the line chart in Figure 7. Among the different ensemble models, the stacked ensemble model created with SVR, RR, and LR in layer 1 and LR as the meta layer achieved the highest R^2 score of 94.45% as shown in Table 8. Different models make different error's by combining these models one model's strength can hide the other model's weakness. Moreover, stacking model learns to generalize better through meta layer.

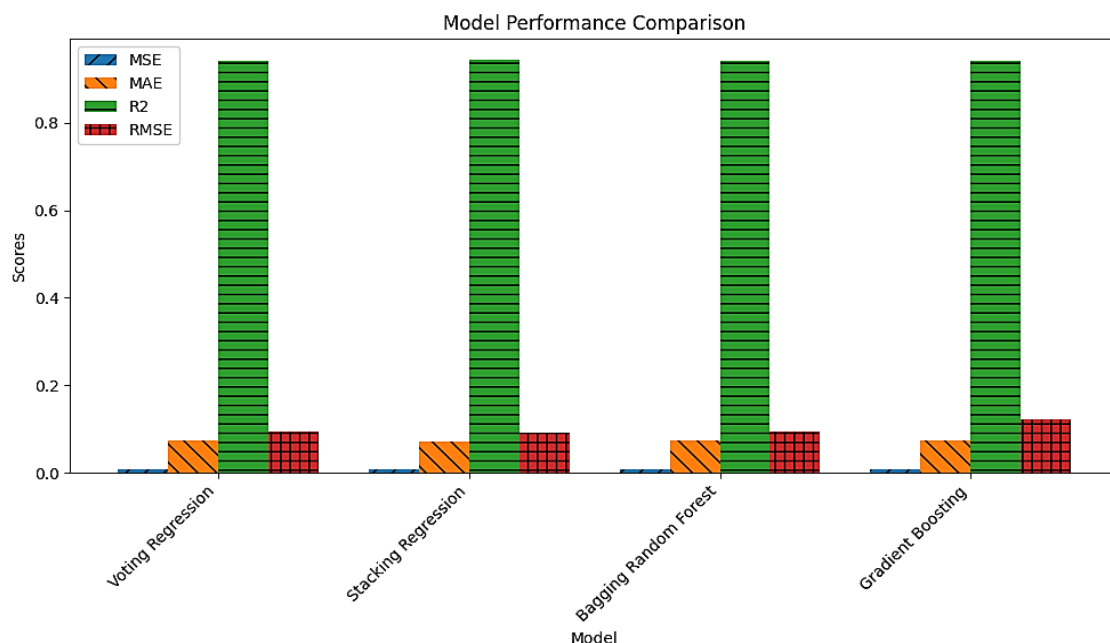


Figure 6. CGPA prediction model using BO

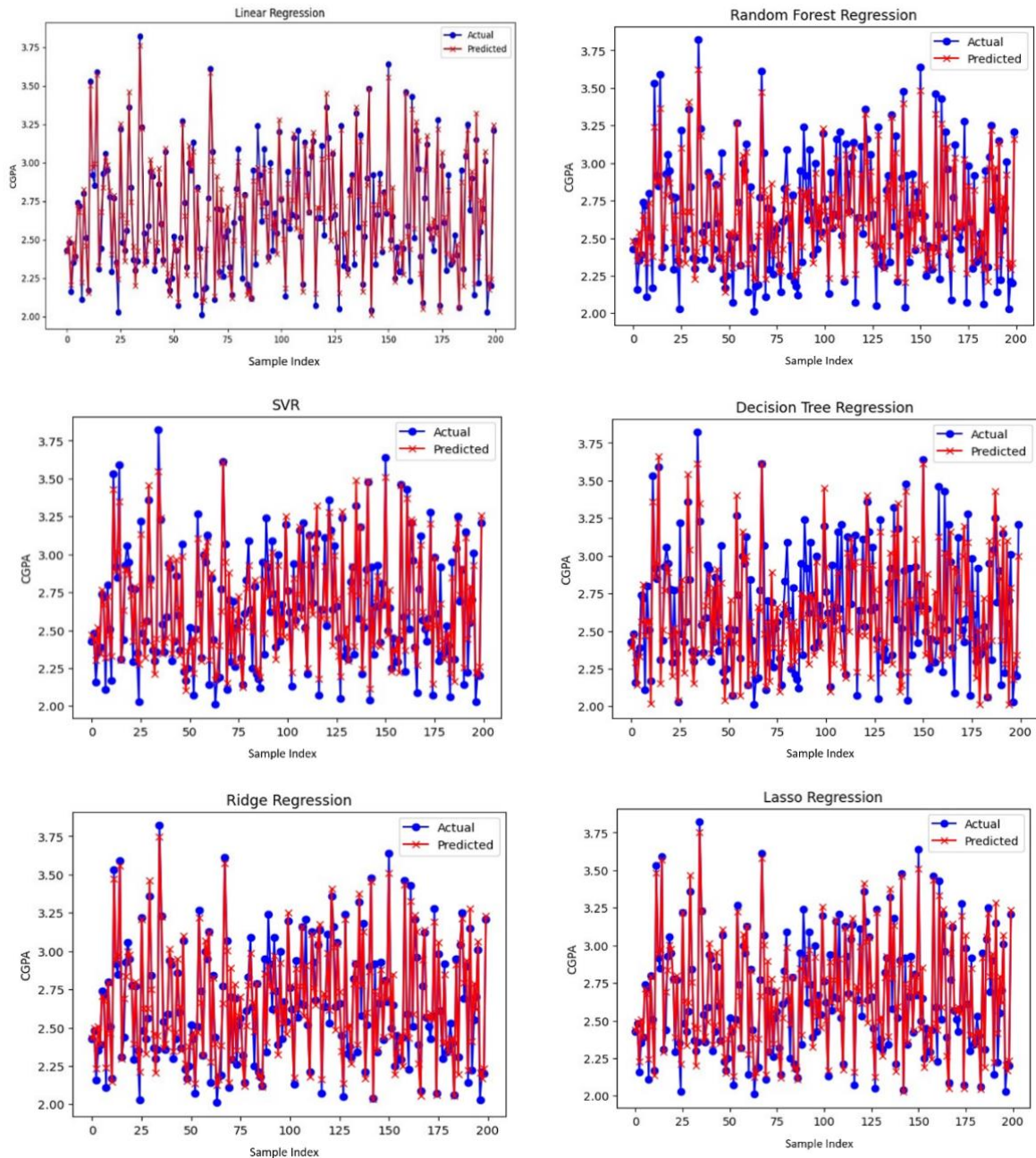


Figure 7. Line chart of CGPA prediction model after BO

Table 8. Ensemble predictive model performance using BO

| Model | MSE | MAE | R2 | RMSE |
|-----------------------|--------|--------|--------|--------|
| Voting regression | 0.0089 | 0.0735 | 0.9415 | 0.0947 |
| Stacked regression | 0.0086 | 0.0719 | 0.9445 | 0.0928 |
| Bagging random forest | 0.0088 | 0.0728 | 0.9422 | 0.0941 |
| Grading booting | 0.0148 | 0.0982 | 0.9034 | 0.1217 |

6.3. Predicting student's performance using first two years course grade

In model 2, only the first 2 years' marks of all subjects are included to train the predictive model. The model performance without hyper-parameter tuning is shown in Table 9. The model LR and SVC have the highest R^2 score of 86.62% when compared to the other models. The model performance using GridSearchCV and BO is shown in Tables 10 and 11, respectively. The model SVC obtained the highest R^2 score of 87.26 and 87.29% with GridSearchCV and BO, respectively. The evaluation metrics after BO are

given in Figure 8. Among the various ensemble models, the stacked ensemble model with SVR, RR, and LR in layer 1, and LR as the meta layer, achieved the highest R² score of 86.95%, as shown in Table 12. In terms of MSE/MAE, SVR obtained a lower error rate when compared to other individual models.

Table 9. Performance of regression models using default values

| Model | MSE | MAE | R ² | RMSE |
|-------------------|--------|--------|----------------|--------|
| Linear regression | 0.2053 | 0.1133 | 0.8662 | 0.1433 |
| RF | 0.0266 | 0.1304 | 0.8265 | 0.1631 |
| SVR | 0.0254 | 0.1262 | 0.8340 | 0.1595 |
| DT | 0.0796 | 0.2299 | 0.4813 | 0.2821 |
| RR | 0.0205 | 0.1133 | 0.8662 | 0.1432 |
| LR | 0.1547 | 0.3230 | -0.008 | 0.3933 |

Table 10. Performance of regression models with GridSearchCV

| Model | MSE | MAE | R ² | RMSE |
|-------------------|--------|--------|----------------|--------|
| Linear regression | 0.0205 | 0.1133 | 0.8662 | 0.1433 |
| RF | 0.0279 | 0.1347 | 0.8180 | 0.1671 |
| SVR | 0.0195 | 0.1115 | 0.8726 | 0.1398 |
| DT | 0.0689 | 0.2117 | 0.5509 | 0.2625 |
| RR | 0.0205 | 0.1134 | 0.8663 | 0.1432 |
| LR | 0.0205 | 0.1138 | 0.8659 | 0.1434 |

Table 11. Performance of regression models using BO

| Model | MSE | MAE | R ² | RMSE |
|-------------------|--------|--------|----------------|--------|
| Linear regression | 0.0205 | 0.1133 | 0.8662 | 0.1433 |
| RF | 0.0324 | 0.1471 | 0.7884 | 0.1802 |
| SVR | 0.0195 | 0.1103 | 0.8729 | 0.1396 |
| DT | 0.7562 | 0.2198 | 0.5073 | 0.2750 |
| RR | 0.0205 | 0.1133 | 0.8662 | 0.1432 |
| LR | 0.0205 | 0.1138 | 0.8659 | 0.1434 |

Table 12. Ensemble predictive model performance using BO

| Model | MSE | MAE | R ² | RMSE |
|-----------------------|--------|--------|----------------|--------|
| Voting regression | 0.0201 | 0.119 | 0.8686 | 0.1420 |
| Stacking regression | 0.0200 | 0.1112 | 0.8695 | 0.1415 |
| Bagging random forest | 0.0195 | 0.1087 | 0.8737 | 0.1398 |
| Gradient boosting | 0.0263 | 0.1288 | 0.8280 | 0.1624 |

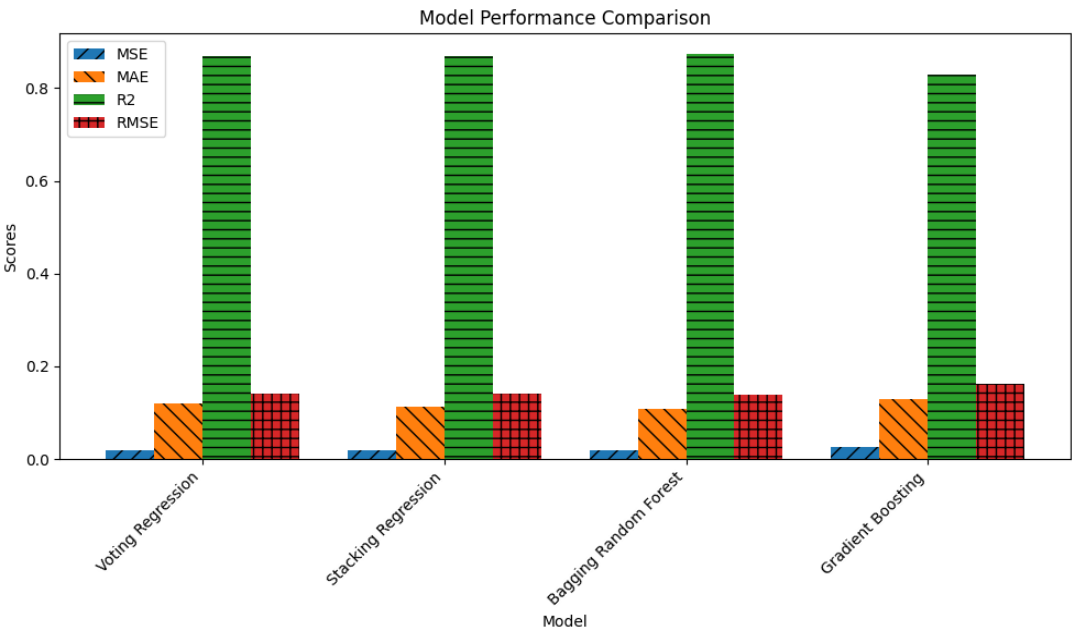


Figure 8. CGPA prediction model using BO

6.4. Predicting student's performance using first-year course grade

Model 3 focuses on only including the marks of the first-year courses to train the predictive model. The model performance without hyper-parameter tuning is shown in Table 13. The model RF has the highest R^2 score of 70.63% when compared to the other models. RF also shows lower MSE, MAE, and RMSE values, indicating reduced error rates. The model performance after optimization with GridSearchCV and BO is shown in Tables 14 and 15, respectively. The SVR model obtained the highest R^2 scores of 72.62 and 72.39% using GridSearchCV and BO, respectively. To further improve the model performance, ensemble models are created with these individual models. The stacking ensemble model, which combines SVR, RR, and LR in layer 1 and LR in the meta layer, performed better than SVR with BO. The R^2 score improved by 0.08% from SVR to the stacking ensemble model as shown in Table 16.

Table 13. Performance of regression models using default values

| Model | MSE | MAE | R^2 | RMSE |
|-------------------|--------|--------|---------|--------|
| Linear regression | 0.0453 | 0.1702 | 0.7046 | 0.2129 |
| RF | 0.0450 | 0.1687 | 0.7063 | 0.2123 |
| SVR | 0.0459 | 0.1732 | 0.7003 | 0.2144 |
| DT | 0.1127 | 0.2532 | 0.2656 | 0.3351 |
| RR | 0.0453 | 0.1702 | 0.7047 | 0.2129 |
| LR | 0.1547 | 0.3230 | -0.0081 | 0.3933 |

Table 14. Performance of regression models using GridSearchCV

| Model | MSE | MAE | R^2 | RMSE |
|-------------------|--------|--------|--------|--------|
| Linear regression | 0.0453 | 0.1702 | 0.7046 | 0.2129 |
| RF | 0.0446 | 0.1684 | 0.7092 | 0.2112 |
| SVR | 0.0420 | 0.1629 | 0.7262 | 0.2049 |
| DT | 0.0778 | 0.2171 | 0.4928 | 0.2790 |
| RR | 0.0452 | 0.1702 | 0.7050 | 0.2127 |
| LR | 0.0453 | 0.1703 | 0.7047 | 0.2128 |

Table 15. Performance of regression models using BO

| Model | MSE | MAE | R^2 | RMSE |
|-------------------|--------|--------|--------|--------|
| Linear regression | 0.0453 | 0.1702 | 0.7046 | 0.2129 |
| RF | 0.0455 | 0.1696 | 0.7031 | 0.2134 |
| SVR | 0.0443 | 0.1681 | 0.7239 | 0.2104 |
| DT | 0.0978 | 0.2433 | 0.3625 | 0.3128 |
| RR | 0.0452 | 0.1703 | 0.7054 | 0.2126 |
| LR | 0.0453 | 0.1703 | 0.7047 | 0.2128 |

Table 16. Ensemble predictive model performance using BO

| Model | MSE | MAE | R^2 | RMSE |
|-----------------------|--------|--------|--------|--------|
| Voting regression | 0.0422 | 0.1656 | 0.7245 | 0.2056 |
| Stacking regression | 0.0422 | 0.1647 | 0.7247 | 0.2055 |
| Bagging random forest | 0.0436 | 0.1680 | 0.7158 | 0.2088 |
| Gradient boosting | 0.0471 | 0.1716 | 0.6927 | 0.2171 |

7. COMPARATIVE ANALYSIS

This section compares the results of the proposed model with the existing studies as shown in Table 17. Most of the studies [6]–[8], [25] considered CGPA prediction as a classification problem and obtained an accuracy of 92.7, 94.29, and 91% respectively. Whereas [2] predicted CGPA using regression models and obtained the highest R^2 score of 90% for LR and Bayesian regression. Essayad and Abdella [26] used gradient boost regression and obtained an R^2 score of 78.98%. The proposed stacked ensemble model was created with base learners such as SVC, RR, and LR with linear regression in the meta layer obtained the highest R^2 score of 94.45%. The accuracy and R^2 score cannot be directly comparable but still the higher R^2 score indicates better model performance.

Table 17. Comparison with related works on academic performance prediction

| Previous studies | Model | R^2 /accuracy | Features used |
|---------------------------|--------------------------|-----------------|--|
| Bhushan <i>et al.</i> [2] | Linear regression | R^2 -87% | Academic, social media interaction, and attendance |
| | RR | R^2 -87% | |
| | LR | R^2 -90% | |
| | Bayesian regression | R^2 -90% | |
| Said <i>et al.</i> [6] | ET, MV | Accuracy-92.7% | Academic |
| Nachouki and Naaj [7] | RF | Accuracy-94.29% | Academic |
| Alangari and Alturki [8] | NB Hoeffding tree | Accuracy-91% | Academic |
| Chen and Zhai [25] | RF | Accuracy-89% | Academic |
| Essayad and Abdella [26] | Gradient boost regressor | R^2 -78.98% | Academic |
| Proposed model | Stacking ensemble model | R^2 -94.45% | Academic |

8. CONCLUSION

This research paper focused on optimizing regression and ensemble models to predict student's CGPA to prevent drop-out rates in educational institutions. Six regression models, namely linear regression, SVR, DT, RF, RR, and LR models, were tested, with and without optimization. SVR achieved the highest R^2 score of 94.11%, along with the lowest MAE, RMSE, and MSE values when compared to the other regression models. By integrating ensemble methods, mainly stacking, the predictive capability of the model was significantly improved, achieving an R^2 score of 94.45%. Thus, the study demonstrated the effectiveness of ensemble modeling and BO in enhancing the accuracy of CGPA prediction models. These findings are valuable in improving the early identification of students at risk of dropping out and taking proactive measures to support their academic success. However, the main challenge of this research work is the use of a real-time dataset, given the student's credit-based academic system. Students enroll in different subjects across various semesters, and predicting their CGPA is further complicated by this variability. Students on probation typically take 4 subjects, while regular students take 5 subjects, and those students with CGPA greater than 3.5 have the option to take on additional courses beyond the standard five subjects. As a result, we must address the presence of null values in the database due to these differences. Furthermore, the dataset was limited to subject marks and CGPA, excluding other potential factors that could influence CGPA. The models do not account for non-academic factors that may influence student performance. Future work can concentrate on other relevant factors that influence the performance of the students, such as attendance, internal assessment marks, extracurricular activities, and socio-economic background to develop a deep learning model for CGPA prediction. The data can be normalized and standardized before model creation as it involves different features. Techniques such as principal component analysis (PCA) can be used to select the most relevant features required to create the model. Models, such as convolutional neural network (CNN) or recurrent neural network (RNN), can be used for training, and the model's performance can be evaluated using metrics like MAE or RMSE.

FUNDING INFORMATION

This research project was funded by the University of Technology and Applied Sciences through the Internal Research Funding Program (04-IRFP-IBRI-2023).

AUTHOR CONTRIBUTIONS STATEMENT

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

| Name of Author | C | M | So | Va | Fo | I | R | D | O | E | Vi | Su | P | Fu |
|-------------------------|---|---|----|----|----|---|---|---|---|---|----|----|---|----|
| Hemalatha Gunasekaran | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | ✓ | | | ✓ | |
| Rex Macedo Arokiaraj | | ✓ | | | | ✓ | | ✓ | ✓ | | ✓ | ✓ | | |
| Angelin Gladys Jesudoss | ✓ | | ✓ | ✓ | | ✓ | ✓ | | | ✓ | ✓ | | ✓ | ✓ |
| Deepa Kanmani | | | ✓ | | ✓ | | | | ✓ | ✓ | | | | |

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflict of interest for this work.

DATA AVAILABILITY

Derived data supporting the findings of this study are available from the corresponding author [HG] on request.

REFERENCES

- [1] D. Ying and J. Ma, "Student performance prediction with regression approach and data generation," *Applied Sciences*, vol. 14, no. 3, Jan. 2024, doi: 10.3390/app14031148.




- [2] M. Bhushan, S. Vyas, S. Mall, and A. Negi, "A comparative study of machine learning and deep learning algorithms for predicting student's academic performance," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 6, pp. 2674–2683, Dec. 2023, doi: 10.1007/s13198-023-02160-3.
- [3] M. Yağcı, "Educational data mining: prediction of students' academic performance using machine learning algorithms," *Smart Learning Environments*, vol. 9, no. 1, Dec. 2022, doi: 10.1186/s40561-022-00192-z.
- [4] Y. Baashar *et al.*, "Evaluation of postgraduate academic performance using artificial intelligence models," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9867–9878, Dec. 2022, doi: 10.1016/j.aej.2022.03.021.
- [5] S. D. A. Bujang *et al.*, "Multiclass prediction model for student grade prediction using machine learning," *IEEE Access*, vol. 9, pp. 95608–95621, 2021, doi: 10.1109/ACCESS.2021.3093563.
- [6] M. B. Said, Y. H. Kacem, A. Algarni, and A. Masmoudi, "Early prediction of student academic performance based on machine learning algorithms: a case study of bachelor's degree students in KSA," *Education and Information Technologies*, vol. 29, no. 11, pp. 13247–13270, Aug. 2024, doi: 10.1007/s10639-023-12370-8.
- [7] M. Nachouki and M. A. Naaj, "Predicting student performance to improve academic advising using the random forest algorithm," *International Journal of Distance Education Technologies*, vol. 20, no. 1, pp. 1–17, Mar. 2022, doi: 10.4018/IJDET.296702.
- [8] N. Alangari and R. Alturki, "Predicting students final gpa using 15 classification algorithms," *Romanian Journal of Information Science and Technology*, vol. 23, no. 3, pp. 238–249, 2020.
- [9] S. K. A. Ibrahim and M. A. Ahmed, "Prediction of students' cumulative grade point averages (CGPAS) at graduation: a case study," *International Journal of Computer Applications*, vol. 174, no. 24, pp. 35–44, Mar. 2021, doi: 10.5120/ijca2021921149.
- [10] A. Korchi, F. Messaoudi, A. Abatal, and Y. Manzali, "Machine learning and deep learning-based students' grade prediction," *Operations Research Forum*, vol. 4, no. 4, Oct. 2023, doi: 10.1007/s43069-023-00267-8.
- [11] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran, "Machine learning based student grade prediction: a case study," *arXiv-Computer Science*, pp. 1-22, Aug. 2017.
- [12] S. Al-Sudani and R. Palaniappan, "Predicting students' final degree classification using an extended profile," *Education and Information Technologies*, vol. 24, no. 4, pp. 2357–2369, Jul. 2019, doi: 10.1007/s10639-019-09873-8.
- [13] U. Frederick and C. c. Okezie, "Evaluation of data mining classification algorithms for predicting students performance in technical trades," *International Journal of Engineering and Computer Science*, vol. 5, no. 8, pp. 17593–17601, Aug. 2016, doi: 10.18535/ijecs/v5i8.29.
- [14] A. Kumar, K. K. Eldhose, R. Sridharan, and V. V. Panicker, "Students' academic performance prediction using regression: a case study," in *2020 International Conference on System, Computation, Automation and Networking*, IEEE, Jul. 2020, pp. 1–6, doi: 10.1109/ICSCAN49426.2020.9262346.
- [15] M. Arifin, W. Widowati, F. Farikhin, and G. Gudnanto, "A regression model and a combination of academic and non-academic features to predict student academic performance," *TEM Journal*, vol. 12, no. 2, pp. 855–864, 2023, doi: 10.18421/TEM122-31.
- [16] L. Falát and T. Piscová, "Predicting GPA of university students with supervised regression machine learning models," *Applied Sciences*, vol. 12, no. 17, Aug. 2022, doi: 10.3390/app12178403.
- [17] J. Lederer, "Linear regression," in *Fundamentals of High-Dimensional Statistics*, Cham, Switzerland: Springer, 2022, pp. 37–79, doi: 10.1007/978-3-030-73792-4_2.
- [18] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing – Letters and Reviews*, vol. 11, no. 10, pp. 203-224, 2007.
- [19] H. Blockeel, L. Devos, B. Frénay, G. Nanfack, and S. Nijssen, "Decision trees: from efficient prediction to responsible AI," *Frontiers in Artificial Intelligence*, vol. 6, Jul. 2023, doi: 10.3389/frai.2023.1124553.
- [20] M. A. Al-Barrak and M. Al-Razgan, "Predicting students final gpa using decision trees: a case study," *International Journal of Information and Education Technology*, vol. 6, no. 7, pp. 528–533, 2016, doi: 10.7763/IJiet.2016.V6.745.
- [21] S. Kwon, S. Han, and S. Lee, "A small review and further studies on the LASSO," *Journal of the Korean Data and Information Science Society*, vol. 24, no. 5, pp. 1077–1088, Sep. 2013, doi: 10.7465/jkdi.2013.24.5.1077.
- [22] N. S. M. Shariff and H. M. B. Duzan, "An application of proposed ridge regression methods to real data problem," *International Journal of Engineering & Technology*, vol. 7, no. 4.30, pp. 106–108, Nov. 2018, doi: 10.14419/ijet.v7i4.30.22061.
- [23] S. Albahli, "Efficient hyperparameter tuning for predicting student performance with bayesian optimization," *Multimedia Tools and Applications*, vol. 83, no. 17, pp. 52711–52735, Nov. 2023, doi: 10.1007/s11042-023-17525-w.
- [24] A. M. Elshewey, M. Y. Shams, N. El-Rashidy, A. M. Elhady, S. M. Shohieb, and Z. Tarek, "Bayesian optimization with support vector machine model for Parkinson disease classification," *Sensors*, vol. 23, no. 4, Feb. 2023, doi: 10.3390/s23042085.
- [25] Y. Chen and L. Zhai, "A comparative study on student performance prediction using machine learning," *Education and Information Technologies*, vol. 28, no. 9, pp. 12039–12057, Sep. 2023, doi: 10.1007/s10639-023-11672-1.
- [26] A. Essayad and K. M. Abdella, "Predicting baccalaureate student result to prevent failure: a hybrid model approach," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 764-774, Mar. 2024, doi: 10.11591/ijai.v13.i1.pp764-774.

BIOGRAPHIES OF AUTHORS






Dr. Hemalatha Gunasekaran    holds a Bachelor of Engineering (B.E.) in Computer Science and Engineering, Master of Engineering (M.E.) in Computer Science and Engineering, Ph.D. in Information and Communication Engineering, besides several professional certificates and skills. She is currently lecturing with the Department of Information and Technology at University of Technology and Applied Sciences, Ibri, Oman. She has more than 19 years of teaching and research experience and her area of interest is deep learning in the health care sector, natural language processing and big data analytics. She has published more than 25 papers in SCI and Scopus journals. She can be contacted at email: hemalatha.david@utas.edu.om or hemalatha2107@gmail.com.






Dr. Rex Macedo Arokiaraj    holds Bachelor of Science (BSc.) in Mathematics, Master of Computer Applications (MCA), Master of Engineering (M.E) in Computer Science and Engineering and Ph.D. in Information and Communication Engineering. He is currently lecturing with the Department of Information and Technology at University of Technology and Applied Sciences, Ibri, Oman. She has more than 25 years of teaching and research experience and her area of interest is network and security, IoT, and machine learning. He has published more than 7 papers in various journals. He can be contacted at email: rex.mecedo@utas.edu.om.



Ms. Angelin Gladys Jesudoss    is a Lecturer in the Department of Information Technology at the University of Technology and Applied Sciences, Ibri, Oman. With 24 years of teaching experience, she holds a Master's in Computer Applications from Bharathidasan University and a Master's in Computer Science and Engineering from Anna University. Her research focuses on compiler design and operating systems, contributing significantly to publications in research papers and books. She can be contacted at email: angelin.gladys@utas.edu.om.



Dr. Deepa Kanmani    holds a Bachelor of Engineering (B.E.) in Computer Science and Engineering, Master of Engineering (M.E.) in Computer Science and Engineering, Ph.D. in Computer Science and Engineering. She is currently working as Associate Professor in the Department of Information Technology in Sri Krishna College of Engineering and Technology, Coimbatore, Tamil Nadu, India. She has more than 16 years of teaching and research experience. Her area of interests includes in database, machine learning and deep learning. She has made significant contributions such as SCI, Scopus, and WoS Publications. She can be contacted at email: deepakanmanis@skcet.ac.in.