I.J. Modern Education and Computer Science, **2025, 2, 67-86**
Published Online on April 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2025.02.03

Enhancing Student Performance Prediction in E-Learning Environments: Advanced Ensemble Techniques and Robust Feature Selection

N. S. Koti Mani Kumar Tirumanadham*
Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India
E-mail: manikumar1248@gmail.com
ORCID iD: https://orcid.org/0000-0003-3900-1900
*Corresponding Author

Thaiyalnayaki S.
Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai, Tamil Nadu, India
E-mail: thaiyalnayaki.cse@bharathuniv.ac.in
ORCID iD: https://orcid.org/0009-0006-4973-8147

Received: 02 July, 2024; Revised: 20 August, 2024; Accepted: 12 October, 2024; Published: 08 April, 2025

Abstract: By means of a thorough investigation of ensemble methodologies and feature selection approaches, this work explores enhancing predictive modelling in e-learning contexts. The setting is in the growing significance of data-driven decision-making in education and tailored learning programs. The main concern is how to fairly forecast student performance in environments of digital learning. This work intends to solve gaps by investigating new ensemble models and robust feature selection techniques based on already published research. Using cutting-edge analytical techniques including hybrid BR^2-2T models and the Chi-square test, the study produces remarkable accuracy surpassing known limits. The results underline the need of feature selection and ensemble methods in improving forecast accuracy and dependability. Finally, this study marks a major step in the field of e-learning predictive modelling since it helps to improve educational results and enable data-driven interventions.

Index Terms: Feature Selection, Chi-square test, Ensemble Model, E-Learning

1. Introduction to E-Learning

E-learning is the short form for electronic learning which in simple terms, is the process of attaining education through electronic devices and internet medium. This method has revolutionized traditional education systems by breaking barriers associated with rigidity and bureaucracy of physical classroom learning [1]. A review into the adoption of e-learning in education systems has been prompted by the developments in technology, student demands, and extension of internet around the globe [2].

This perspective on education through information and communications technology has changed a lot with the coming of the internet. The traditional forerunner of e-learning courses is correspondence classes which paved way for the later as it showed the possibility of learning without having to go to a physical classroom. Coming into the final quarter of the 20^{th} century with the use of personal computers and the Internet, education was revolutionized and the use of technology introduced a new way of learning that started replacing the traditional classroom environments [3].

E-learning is the utilization of Information and Communication Technologies in learning exercises, and it incorporates devices, for instance, the LMS, virtual homerooms, multi-media materials, and learning applications for portable gadgets. Course management systems such as Moodle and Blackboard offer complete solutions for course delivery and performance evaluation. Many of the activities in a virtual classroom interface closely resemble those of a traditional classroom where lectures and students meet in a face-to-face manner through tools such as Zoom and Microsoft Teams [4].

Revolution brought by e-learning is of great importance for both researches in the field of education and for practitioners. It provides a vibrant realm of exploring out innovations in instructor meditated teaching, including use of

technologies and the learners themselves. Education practitioners can utilize e-learning data to evaluate the efficiency of educational programs, the optimal ways of boosting student interactions, and students' learning outcomes. In addition, the e-learning is easily scalable which makes is easy to disseminate the new teaching methods to numerous institutions within a very short time [5].

As educational institutions work to enhance individualized learning experiences and raise general academic results, predicting student performance in e-learning settings has become ever more important. As digital learning platforms proliferate, enormous volumes of data are produced and provide insightful analysis of student learning patterns and performance. By means of machine learning and data mining approaches, leveraging this data might help teachers to spot at-risk pupils, customize their teaching plans, and finally create a more efficient classroom. Nevertheless, the complexity and fluctuation of educational data provide major difficulties in precisely forecasting student performance, so new approaches combining strong feature selection and ensemble modelling techniques are quite necessary.

1.1. Problem Definition

The main issue this work aims to solve is the correct prediction of student performance in e-learning environments with machine-learning approaches. Conventions often suffer from shortcomings like poor handling of data complexity, inappropriate feature selection techniques, and little integration of advanced ensemble models. These restrictions can lead to reduced prediction accuracy and dependability, thereby making it challenging to give pupils timely and successful interventions. The difficulty resides in creating a thorough approach capable of managing this complexity and raising the predicted accuracy of student performance models.

1.2. Scope

This work intends to improve student performance prediction in the e-learning environment using a complete approach. It entails the creation of a three-tier ensemble model, sophisticated data preparation, several feature selection approaches—including hybrid methods—and so on. The performance of the proposed model will be assessed using a varied set of criteria including accuracy, precision, recall, F1-score, and AUC-ROC. The efficiency of several feature selection techniques will be examined to find the most relevant techniques.

1.3. Objectives

This study aims to accomplish:

- Create a strong and thorough methodology combining advanced feature selection methods with multi-tier ensemble modeling to fairly forecast student performance.
- Addressing data complexity and unpredictability will help to improve the accuracy and dependability of student performance projections.
- List useful feature selection techniques. Find the best feature selection methods that help to raise model performance.

1.4. Research Questions

This research study attempted to answer the following research questions.

1. Are there differences in the performance prediction power of advanced ensemble methodologies embodied in the three-tier technical mode and other more conventional approaches to machine learning?
2. Moreover, how does employing first-rate feature selection methodologies, hybrid methods, influence accuracy and stability of an e- learning predictive modelling?
3. How effective feature selection and ensemble strategies are in enhancing the generalization and accuracy of models in different datasets of e-learning?
4. In consequence, how do the herein proposed ensemble models perform with respect to accuracy, precision, recall, F1-score, as well as the AUC-ROC in comparing with existing state-of-the-art approaches for e-learning datasets?

1.5. Major Contributions

The main outputs of this work consist in:

- The F^2L^1R-2T and BR^2-2T feature selection techniques provide an embedded method of feature selection by combining prior selection and ranking techniques to give high accuracy while preserving model interpretability perfect for the selection of features. This avoids tends to avoid the problems that were experienced by previous studies in that they only used a single feature selection technique.
- It is termed as the three-tier ensemble model which is more sophisticated than the single-layer or two-layer ensemble models. The using of the multiple base models and meta-models in this approach enables to use the advantages of the various algorithms for increasing the accuracy of prediction and for enhancing the generalization.

- The evaluation of the research also holds diverse features such as accuracy, precise time complexity, and efficiency of computation as well as interpretability of the model. This is more satisfactory than previous studies that mainly considered only accuracy of the model and nothing more.
- Finally, the work shows how the proposed model is superior to related state-of-the-art methods, which are also discussed in the current research. This way the importance of the research work and how it could impact the future development of the field is justified.
- This paper provides a useful overview of the feature selection techniques and shows the reader the advantages and limitations of the method. This knowledge can help the researchers in choosing the right techniques to be used in their research hence enhancing the efficiency of their results models.

2. Literature Review

The prediction accuracy of different educational contexts has in the recent past been enhanced by applying machine learning in EDM.

In 2024, Mansouri et al. [6] has put forward an individualized Machine Learning model for the classification of undergraduate's major specializations. Their approach has involved the use of oversampling to deal with the issue of data imbalance, Sequential Backward Selection (SBS) for purposes of selection of features, and AdaBoost classifiers that are themselves fine-tuned through genetic algorithms. This model was very accurate and got 98 % accuracy and more than other comparable models.

In 2024, Ahmed [7] was more focused on additional algorithms for the student performance prediction by using a machine-learning approach and stressed the significance of the feature selection and hyperparameters tuning. On the use of K-means clustering and tuning the parameters of Supports Vector Machines, the study a prevalence of 96% accuracy was realized.

In 2023, Dahal and Shakya [8] categorized the effect of feature selection in educational data mining pointing out that it helps enhance the model's performance by as much as ten percent. Their work focused on the development of the mixed models and on the application of the deep learning techniques for the purpose of increasing accuracy of the predictions and for the evaluation of the performance.

In 2023, the study of feature engineering and selection, based on the comparison of Mustapha et. al [9], with regard to different educational outcomes suggests using Boruta for regression issues and RFI for classification to estimate performance of students.

In 2023, Nayani et al. [10], the authors had focused on some of the limitations when it comes to the use of big and feature-imbalanced educational data for student performance prediction. They proposed a deep learning model with entropy rough set theory and the new hybrid Galactic Rider Swarm Optimization (GRSO) algorithm that enhances the sensitivity and accuracy.

In 2022, Hessen et al. [11], a multi-agent e-learning system was created that implemented feature selection by the univariate and Extra Trees methods. Their improved Random Forest model with 16 features selected achieved the cross-validation of 87. 6% and a test of 86. 72%.

In 2022, Ghatasheh et al. [12], Authors discussed the problem of spam and considered such issues as the feature space size and the imbalance in data. Much of the current work is based on partially implemented activities or employs black-box approaches. Their research provides a genetic algorithm to reduce the dimensionality and optimize hyperparameters for microarray data and claims better results than Chi2 and PCA techniques.

Review of current methods exposes various technical flaws and key findings in Table 1. First of all, many researches depend on single feature selection techniques, which might not adequately represent the complexity of the data. Particularly lacking are hybrid methods that combine several feature selection methods to raise model accuracy. Second, although ensemble learning has great promise, most studies use simple models or single-layer ensembles, therefore missing the benefits of more intricate, multi-layer ensemble structures. Furthermore, even if genetic algorithms and other optimization techniques are applied, little study has been done on merging these with advanced feature selection and ensemble models for best performance. Moreover, a lot of researches concentrate just on accuracy, thereby excluding other important measures of model performance such precision, recall, F1-score, and AUC-ROC. Finally, even if some studies look at hybrid models, the combination of conventional machine learning with deep learning methods stays underused. These limitations draw attention to the need of a complete approach including robust ensemble modelling, improved feature selection, and extensive evaluation measures to properly forecast student performance.

Table 1. Summary of the existing models with their key findings.

Author(s)	Methods Used	Accuracy	Key Findings
Mansouri, N., Abed, M., & Soui, M. (2024)	GA-XGBoost	98.1%	The model achieves 98.1% accuracy in helping undergraduates choose specializations, outperforming benchmarks with optimized AdaBoost classifiers and feature selection.
Ahmed (2024)	RFA feature selection and automated hyperparameter optimization with SVM linear	95.41 %	SVM achieved the highest accuracy after parameter tuning. Feature selection and parameter adjustment significantly improved the performance of all models. Naïve Bayes had the lowest accuracy due to its strong independence assumption.
Dahal, N. P., & Shakya, S. (2023)	Feature selection techniques: CFS, Chi-Square, GA, IG, mRMR, ReliefF, RFE. ML algorithms: CART, RF, SVM, KNN. DL algorithms: CNN, LSTM	RF: 97.14%	CFS with CNN showed good stable performance. RF and CART had significant accuracy improvements with feature selection. Hybrid models suggested for more reliable evaluation techniques.
Syed Mustapha (2023)	Recursive Feature Elimination (RFE) feature selection with XGBoost	78%	Feature selection enhanced model performance, with Gradient Boost and XGBoost showing superior accuracy and precision.
Nayani, P, and D (2023)	Hybrid Galactic Rider Swarm Optimization with CRN (CNN +RNN)	93%	Enhanced classification performance through GRSO optimization.
Hessen, Abdul-Kader, Khedr, & Salem (2022)	Univariate and Extra Trees feature selection methods with DT, LR, RF, NB, KNN;	87.6% (cross-validation), 86.72% (testing)	RF with features selected by Extra Trees achieved the highest performance. The study highlighted the importance of feature selection and multiagent systems in improving e-learning processes.
Ghatasheh, N., Altaharwa, I., & Aldebei, K. (2022)	Modified genetic algorithm for dimensionality reduction and hyperparameter optimization. Utilized XGBoost classifier on tweets dataset.	92.67%	The modified genetic algorithm effectively reduced dimensionality of tweets dataset and optimized XGBoost parameters, achieving high accuracy (92.67%). Outperformed Chi2 and PCA methods.

3. Methodology

Data cleansing, duplicate removal, handling missing information, and data normalizing help us to forecast student success in e-learning from beginning. The Interquartile Range (IQR) approach treats outliers. Synthetic Minority Over-sampling Technique (SMOTE) helps to balance the class in the dataset. Several techniques are used in feature selection: L2 Regularization, Cuckoo Search Algorithm, Sequential Forward Selection (SFS), Chi-Square Test. Two hybrid techniques also are applied: BR2-2T integrates Boruta with L2 regularization and F^2L^1R-2T combines Firefly algorithm with L1 regularization. A three-tier ensemble model then consumes these chosen features. Training several base models—logistic regression, decision trees, KNN—forms the first tier. Combining these forecasts using a meta-model such as random forest or gradient boosting forms the second tier. The third tier compiles these forecasts together using another meta-model for ultimate prediction. Accuracy, precision, recall, F1-score, and AUC-ROC help to assess model performance. By means of a comparison of several feature selection techniques, one can ascertain the most successful strategy. See in Figure 1 for a detailed approach aiming to maximize feature selection and model accuracy for e-learning environments' student performance prediction.

3.1 Dataset Description

From Kalboard 360, an LMS, the 1982 student records with 16 features classified into demographic, academic, and behavioral categories (see Table 2). Using xAPI, data is gathered tracking learner actions including reading and viewing videos. Mostly Kuwait and Jordan, the sample comprises 525 men and 175 women from many backgrounds. Accrued across two semesters, it includes parent involvement measures and attendance records. Analyzing student performance depending on gender, nationality, educational level, and parental engagement makes use of the dataset quite beneficial (see Figure 2). Kaggle [13] has it for additional research and model building.

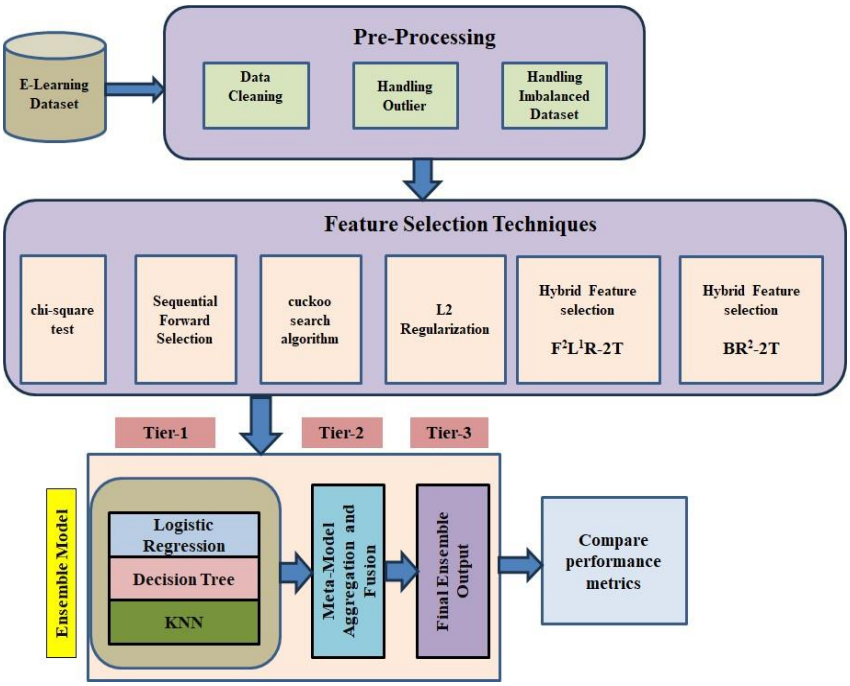


Fig. 1. Schematic diagram of the Proposed methodology

Table 2. Description of e-learning data

Attribute-Number	Attribute-Name	Description	Attribute-Number	Attribute-Name	Description
F1	gender	Gender	F9	relation	Parent responsible for student
F2	nationality	Nationality	F10	raisedhands	Raised hand
F3	placeof_birth	Place of birth	F11	visited_resources	Visited resources
F4	stage_id	Educational Stages	F12	announcements_view	Viewing announcements
F5	grade_id	Grade Levels	F13	discussion	Discussion groups
F6	section_id	Section ID	F14	parent answering _survey	Parent Answering Survey
F7	topic	Topic	F15	Parentschool_satisfaction	Parent School Satisfaction
F8	semester	Semester	F16	student_absence_days	Student Absence Days

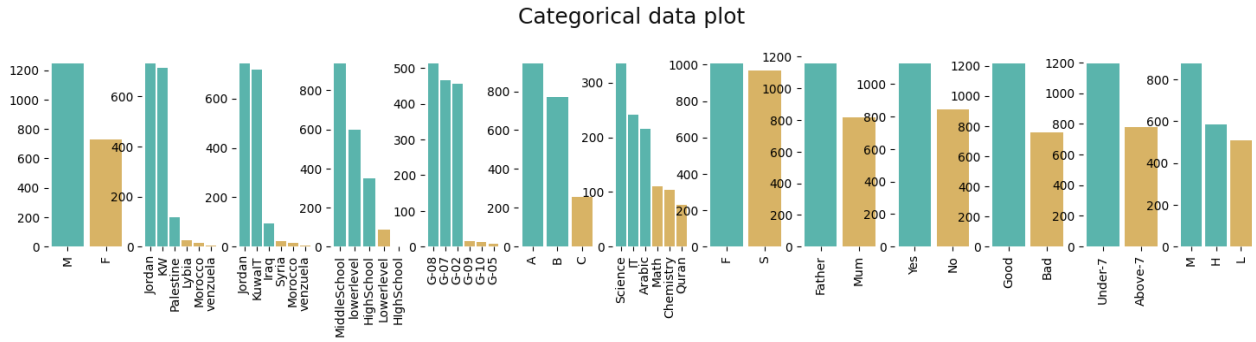


Fig. 2. Categorical data plot

3.2 Visualizing the attributes in dataset using pair plot and correlation matrix

Collecting several scatter graphs. Every subplot pair two of the four variables: announcements_view, discussion, raisedhands, and visited_resources. Every point's color indicates a class; the point's position reveals the value of every variable. The upper left subplot, for instance, displays the link between announcements_view and debate. The points—which probably reflect three different classes—are blue, green, and red. A point's location indicates the value for announcements_view on the x-axis and a y-axis discussion point. It seems from this subplot that the two variables have a positive relationship; points with higher values for announcements_view also typically have higher values for conversation. Still, the data exhibits a great degree of variance (see Figure 3).

It reveals the relationship among four variables: raised hands, visited resources, announcement's view, and discussion. Red denotes a positive correlation, blue a negative correlation, and white or light colors a weak association on the color scale used in the heatmap (see in Figure 4). The cell at the junction of the "raisedhands" row and "visited_resources," for instance, displays a positive correlation—that is, a positive association between the number of times hands are raised and the number of resources visited. All things considered, the heatmap clarifies the direction and strength of the interactions among four factors.

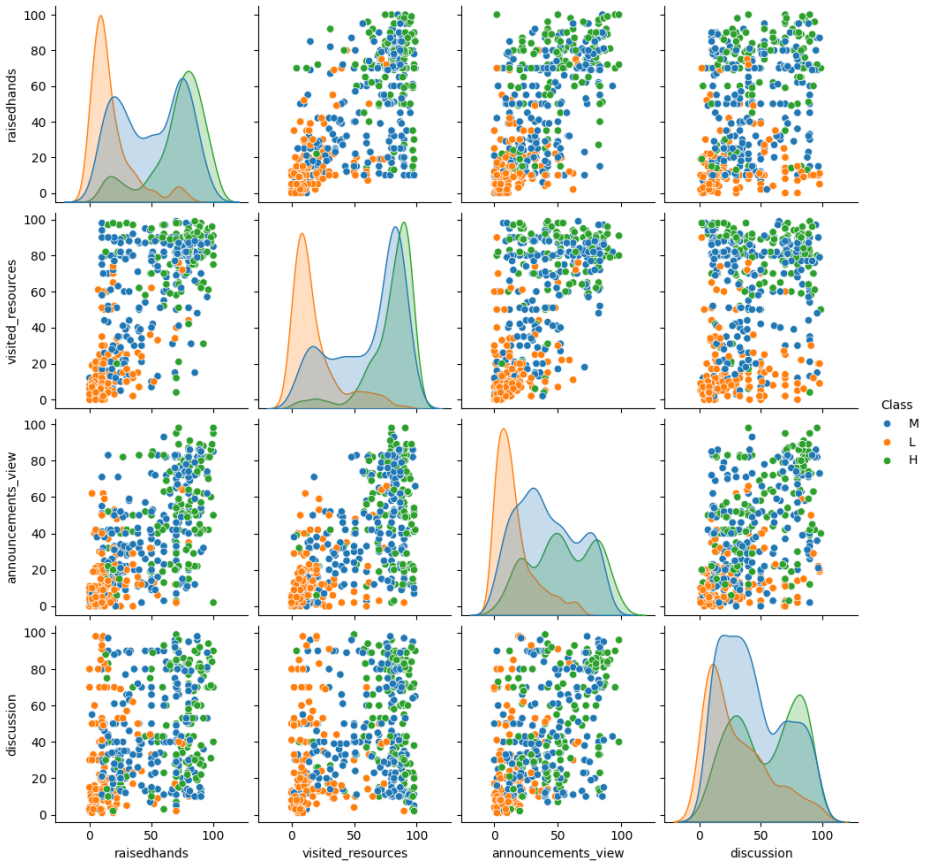


Fig. 3. Visualizing the attributes of Students' Academic Performance Dataset using pair plot

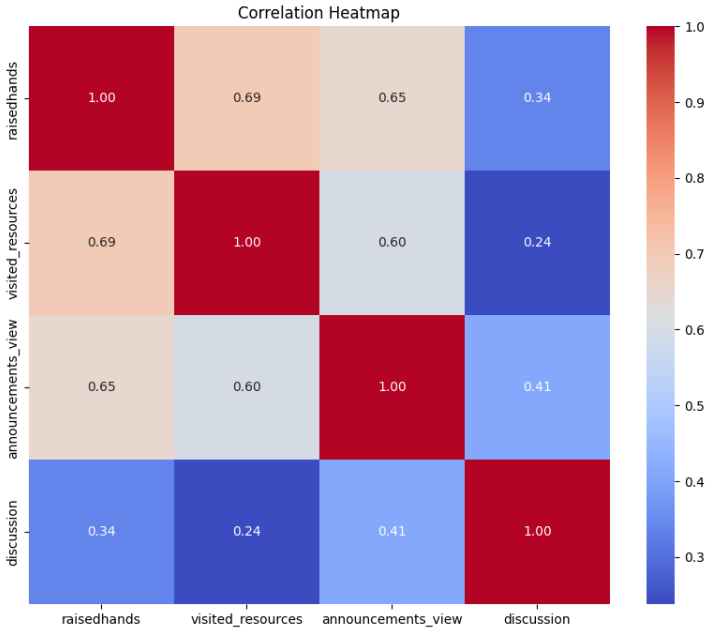


Fig. 4. Heat Map of attributes in the Students' Academic Performance Dataset

3.3 Data Cleaning

There are now 1972 rows and 17 columns in the cleaned dataset; no missing values remain. Ten duplicate rows were eliminated throughout the cleaning process—more especially, from indices [326, 327, 708, 806, 807, 1286, 1287, 1766, 1767, 1920] and none were discarded since none were single-valued. The outcome is a duplicate-free, clean dataset ready for more study or modelling tasks.

3.4 Handling Imbalanced Dataset

Generating synthetic instances in the feature space helps SMOTE (Synthetic Minority Over-sampling Technique) [14] to solve the imbalance in classification issues. Between a minority class sample and one of its k-nearest neighbors, SMOTE generates synthetic samples by interpolation. This approach creates fresh, reasonable minority class samples inside the feature space, hence balancing the class distribution. The steps include the selection of samples of the minority class, determination of k nearest neighbors of the samples, and creation of synthetic samples by averaging the results shown in formula (1) to (4).

1. *Identify Minority Class Samples:*

   Assume the minority class dataset D_{min} consists of N samples: $\{x_1, x_2, x_3, \dots\dots, x_N\}$

2. *Finding K-Nearest neighboring:*

   For each $x_i \epsilon D_{min}$, find the k-nearest neighbors based on a Euclidean distance in formula (1)

$$dist(x_i, x_j) = \| x_i - x_j \| \tag{1}$$

   Store this k-nearest neighbors as:

$$\{x_i^1, x_i^2, \dots\dots, x_i^k\}$$

3. *Generate Synthetic Samples:*

   For each x_i , repeat the following steps to generate the desired number of synthetic samples:
   Randomly select a neighbor x_j from $\{x_i^1, x_i^2, \dots\dots, x_i^k\}$
   Calculate the difference vector using formula (2)

$$diff = x_j - x_i \tag{2}$$

   Equation (3) Multiply the difference vector by a random number $\delta \epsilon [0,1]$:

$$diff_\delta = \delta . diff \tag{3}$$

   Add the scaled difference vector to x_i to obtain the synthetic sample shown in formula (4).

$$x_{new} = x_i + diff_\delta = x_i + \delta . (x_j - x_i) \tag{4}$$

Using the Synthetic Minority Over-sampling Technique (SMOTE), we corrected an imbalanced dataset. First, we looked over the dataset and found a notable class distribution disparity. We imagined this using a bar graph, which emphasized the class differences. We employed the SMOTE algorithm, which interpolates between current minority class samples and their nearest neighbors to create synthetic samples for the minority class, therefore balancing the dataset. We re-examined the class distribution following SMOTE and visualized it once again using a bar graph. With a balanced class distribution, this post-SMOTE graph confirmed the efficacy of the method depicted in Figure 5.

3.5 *Handling Outliers with IQR*

Using the Interquartile Range (IQR) [15] is an excellent method to control outliers in a dataset. The IQR, or range of the data between the first quartile (Q1) and the third quartile (Q3), it gauges the dissemination of the middle 50% of the data. Usually speaking, outliers are data points either below Q1 - 1.5*IQR or above Q3 + 1.5*IQR. Along with an overview of the histogram produced before and after using this strategy shown in Figure 6, below is a step-by-step method for managing outliers using the IQR methodology.

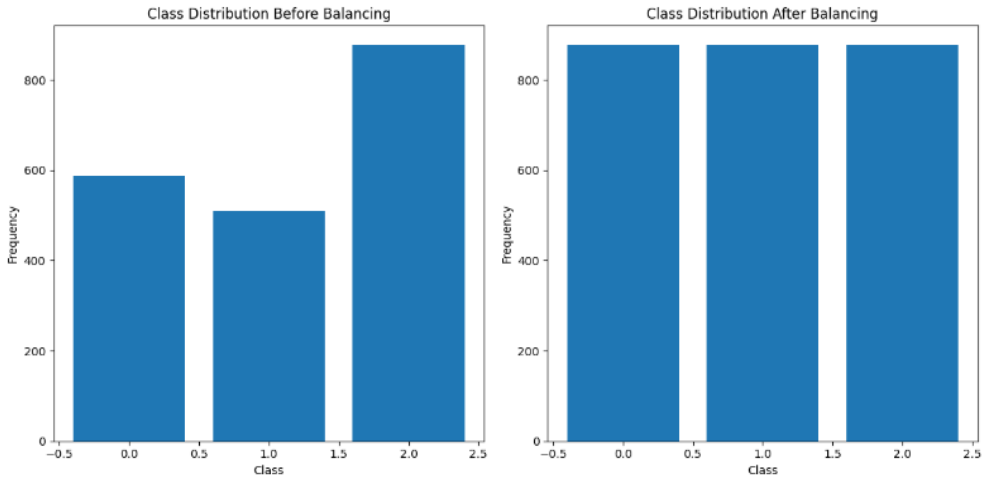


Fig. 5. Before and after handling imbalanced dataset using SMOTE

1. *Calculate the First Quartile (Q1):*

Q1 is the median of the lower half of the dataset (25th percentile) by using formula (5).

$$Q1 = median(lower\ half\ of\ the\ data) \tag{5}$$

2. *Calculate the Third Quartile (Q3):*

Q3 is the median of the upper half of the dataset (75th percentile) by using formula (6).

$$Q3 = median(upper\ half\ of\ the\ data) \tag{6}$$

3. *Compute the IQR:*

The IQR is the range between the first quartile and the third quartile shown in formula (7).

$$IQR = Q3 - Q1 \tag{7}$$

4. *Determine the Lower Bound and Upper Bound for Outliers:*

Identifying the lower bound by using the formula (8) and upper bound by using the formula (9).

$$Lower\ Bound\ (LB) = Q1 - 1.5 * IQR \tag{8}$$

$$Upper\ Bound\ (UB) = Q3 + 1.5 * IQR \tag{9}$$

5. *Identify Outliers:*

Outliers are identified by using the formula (10).

$$x < Lower\ Bound(LB)\ or\ x > Upper\ Bound(UB) \tag{10}$$

3.6. Feature Selection Techniques

3.6.1. Chi-square Test

Algorithm 1 describes the Chi-Square Feature Selection procedure [16]. Chi-Square test feature selection is the process of identifying, from a category target variable, which categorical feature is most substantially correlated. Starting with data preparation provides that, if they be originally continuous, both features and the target variable are either categorical or have been discretized into categories. Every feature has a contingency table produced to show the frequency distribution of the feature values versus the target variable categories. Under the supposition that the feature has no relationship with the target variable, the predicted frequencies for every cell of the contingency table are then computed. Computed by adding the squared variations between the observed and predicted frequencies divided by the expected frequencies for every cell is the Chi-Square statistic. To find the p-value—that is, the likelihood that the observed relationship results from random chance—this statistic is matched to a Chi-Square distribution. Selected for the model, features with p-values below a predefined significance level—typically 0.05 are regarded to have a significant correlation with the target variable. By limiting dimensionality and therefore lowering the risk of overfitting,

this approach enables the selection and retention of just the most important categorical characteristics, so improving the performance of the model.

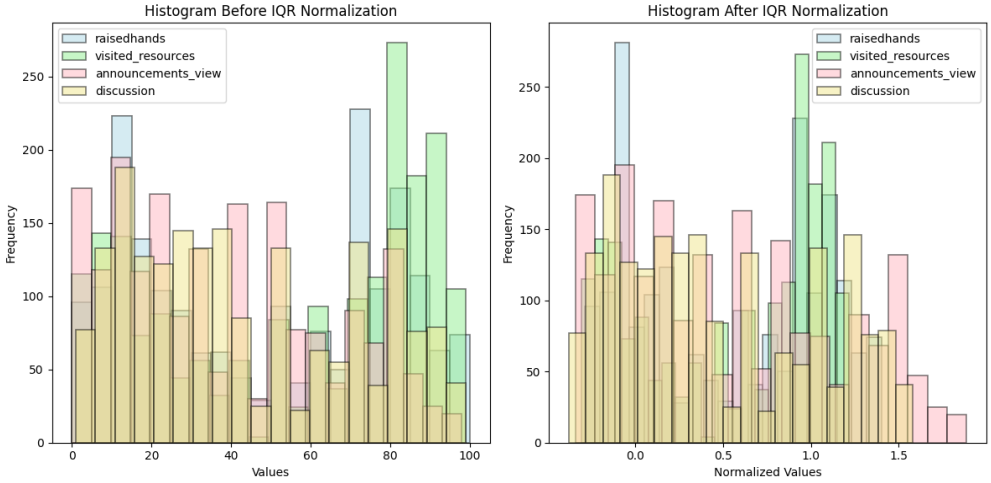


Fig. 6. Before and after handling Outliers with IQR

The Chi-square test is more appropriate for the analysis of categorical data since it is simple and can pool for significant relationship with the target variable. The chi-square tests mitigate the problem of high dimensionality without losing some predictive detail making them ideal for subsequent complex models.

Algorithm 1: Chi-Square Feature Selection

Input: Dataset with categorical features, Categorical target variable, Significance level (α)
Output: List of selected features

1. Ensure all features and the target variable are categorical
2. Initialize selected_features as an empty list
3. **For each feature in the dataset:**
   a. Create a contingency table for the feature and the target variable
   b. Calculate the expected frequencies for the contingency table
   c. Compute the Chi-Square statistic using the observed and expected frequencies
   d. Determine the p-value from the Chi-Square statistic
   e. **If the p-value < α:**
      i. Add the feature to selected_features
   end if

end For

4. Return selected_features

3.6.2. Sequential Forward Selection (SFS)

Algorithm 2 describes the SFS Feature Selection procedure [17]. A greedy method for feature selection, sequential forward selection (SFS) iteratively adds characteristics to an originally empty set to generate an ideal subset. Starting with no features, the method adds the feature that causes the best performance improvement at every turn. Usually, this is calculated with reference to a selected assessment metric—accuracy, precision, recall, or another pertinent criterion. Every iteration adds every feature not yet included in the chosen subset temporarily and evaluates the model. Then permanently included to the chosen feature subset is the element generating the best performance enhancement. This process keeps on until a preset number of features have been chosen or until adding more features does not appreciably enhance the performance of the model. By means of a subset of characteristics that together contribute most to the predictive power of the model, the performance of the model is improved while the complexity and overfitting hazards connected with employing all the available features are lowered.

SFS is chosen for its greedy although handy approach to construct an optimal feature subset. This approach is suitable when dealing with the datasets that have the important and the irrelevant features. SFS only retains the most significant features and reduces overfitting as a way of enhancing model generalization by iteratively adding features.

Algorithm 2: Sequential Forward Selection (SFS) for Feature Selection

Input:
- Dataset with features $X_1, X_2, X_3, \ldots\ldots, X_n$ and target variable Y
- Evaluation metric (e.g., accuracy, F1-score)
- Maximum number of features to select (max_features)

Output:
- Selected features subset

1. Initialize selected_features as an empty set
2. Initialize best_performance = 0
3. Initialize current_features as an empty set
4. Initialize remaining_features as all features $\{X_1, X_2, X_3, \ldots\ldots, X_n\}$
5. **Loop until the number of selected features equals max_features or no improvement in performance:**
   a. **For each feature X_j in remaining_features:**
      i. Add X_j to current_features
      ii. Train a model using current_features and evaluate its performance using the chosen metric
      iii. **If performance improves:**
         - Set best_performance = current_performance
         - Update selected_features = current_features
         End if
      iv. Remove X_j from current_features
   End For
   b. Remove the best-performing feature from remaining_features

End While

6. Return selected_features as the final subset of features

3.6.3. Cuckoo Search (CS)

Algorithm 3 describes the CS Feature Selection procedure [18]. Inspired by the Lévy flight behavior of some animals and the brood parasitism of cuckoo birds, Cuckoo Search (CS) is an optimisation method. Its strong search capabilities help it to be efficient for feature selection in machine learning. The method starts with starting a population of nests, where every nest stands for a possible subset of traits. Every nest's degree of fitness is assessed using a predetermined objective function, say classification accuracy. Lévy flights—large, random steps to traverse the search space—generate fresh solutions for every cuckoo. A new solution replaces the selected nest if it exhibits higher fitness than a randomly selected existing nest. To preserve population diversity, some of the worst-performing nests are also deserted and replaced with fresh random ideas with some probability. This iterative procedure keeps on until a convergence criterion is satisfied or a limit number of iterations is attained. At the conclusion of the iterations, the best answer discovered shows the ideal subset of traits. This approach is a strong tool for feature selection in challenging optimisation issues since it uses both local search through Lévy flights and global search through the random generating of new solutions.

Algorithm 3: Cuckoo Search for Feature Selection

1. Initialize population of nests with random subsets of features
2. Evaluate fitness of each nest
3. Identify the best nest based on fitness
4. **Repeat until maximum iterations or convergence criteria met:**
   4.1 **For each nest in the population:**
      a. Generate a new solution by performing a Lévy flight from the current nest
      b. Evaluate the fitness of the new solution
      c. Randomly select a nest from the population
      d. If the new solution is better, replace the selected nest with the new solution
   End For
   4.2 With a probability p_a, abandon some nests and replace with new random solutions
   4.3 Update the best nest found so far
5. **End iterations:**
   - Check if maximum iterations reached or convergence criteria met
   - If conditions met, exit loop
6. Return the best nest as the optimal subset of features

CS is used for its superior global search characteristics based on the natural patterns of behavior. This algorithm has great applicability in the contexts where the number of dimensions is high and other methods often tend to converge to local optima. Specifically, CS uses Lévy flights and random solution generation, hence the area to be covered is diverse yet optimal for the model.

3.6.4. L2 Regularization (L2R)

Algorithm 4 describes the L2R Feature Selection procedure [19]. In machine learning, L2 regularization—also called Ridge Regression—is a method used to prevent overfitting by imposing a penalty on the coefficient value size. L2 regularization reduces the coefficient of less significant features towards zero, therefore lowering their influence on the model in the framework of feature selection. The main concept is to include a regularizing term commensurate with the sum of the squared coefficients, hence changing the loss function. This term reduces big coefficients; hence the model favors smaller ones. A parameter λ controls the regularizing term and so defines the penalty's intensity. More aggressive coefficient shrinking follows from a higher λ. L2 regularization lessens the influence of less significant features, thereby acting as a soft form of feature selection even if it does not precisely set any coefficients to zero as L1 regularization does. By means of optimisation strategies such as gradient descent, whereby the gradients are modified to consider the L2 penalty, the regularized loss function is reduced during model training. We can balance between fitting the training data well and maintaining the model coefficients modest by tweaking λ, hence enhancing the generalization to fresh data.

The issue of multicollinearity is solved and the interpretability of models through L2 regularization (Ridge Regression) is enhanced. This approach penalizes big coefficients without forcing them to zero, thus it can perfectly be used in parallel with, for example, Boruta or Cuckoo Search. L2R makes the model balanced and it comes in handy when solving the formulated characteristics which often exhibit correlation in educational data set.

Algorithm 4: L2 Regularization (Ridge Regression)

1. Input: Training data (X, y), regularization parameter (λ), learning rate (α), max iterations (max_iterations)
2. Initialize coefficients θ to small random values or zeros
3. **Repeat until max_iterations or convergence:**
   3.1 Compute predictions: $y_{hat} = X * \theta$ (11)
   3.2 Compute error: $e = y_{hat} - y$ (12)
   3.3 Compute gradient: $gradient = \frac{1}{m} * X^T * (X * \theta - y) + \left(\frac{\lambda}{m}\right) * \theta$ (13)
   3.4 Update coefficients: $\theta = \theta - \alpha * gradient$ (14)
   End loop
4. Output: Optimized coefficients θ

3.6.5. Hybrid Feature selection using F2L1R-2T

Algorithm 5 describes the F^2L^1R-2T Feature Selection procedure. Initially preparing the e-learning dataset by extensive cleaning, balancing, and outlier treatment to ensure data of high quality, the F^2L^1R-2T mechanism runs. Then a two-step feature selection procedure starts. The Firefly Algorithm [20] is used first to investigate the feature space. Fireflies here suggest several feature subsets and their appeal stems from an objective function gauging feature significance. Fireflies migrate to more pleasing counterparts, iteratively improving the feature selection to converge on the most important subset. L1 regularization [21] is then applied to this chosen subset in the second optimisation stage, so further refining the feature set by including a penalty for non-zero coefficients, thus encouraging sparsity and removing useless features. Predictive models built from the last optimal features expose significant trends and facts from the e-learning data, so improving educational tactics and individualized learning opportunities.

F^2L^1R-2T additionally integrates the exploration function of the Firefly Algorithm and the ability of L1 regularization for inducing sparse parameters. As a result of this research, this hybrid technique is the selected one, which integrates Firefly Algorithm's capability of searching for diverse solutions and L1 regularization's effectiveness in eliminating irrelevant features. This combination results in a comparatively complete, minimal set of features that would enhance model's performance while at the same time, lowering computational requirements.

Algorithm 5: Hybrid Feature selection using F^2L^1R-2T

1. InitializeParameters ()
2. InitializePopulation ()
3. **while stopping_criterion_not_met do**
   3.1. ComputeFitness ()
   3.2. SortPopulation ()
   3.3. SelectedFeatures = FireflyFeatureSelection ()
   3.4. FinalSelectedFeatures = L1Regularization (SelectedFeatures)
   3.5. StoreBestFeatures (FinalSelectedFeatures)
4. **end while**
5. return FinalSelectedFeatures

3.6.6. Hybrid Feature selection using BR2-2T

Algorithm 6 describes the BR^2-2T Feature Selection procedure. Combining Boruta [22] with L2 regularization (Ridge) [23] the BR^2-2T mechanism is a hybrid feature selection method meant to improve model performance and interpretability. The procedure starts with extensive data preparation covering cleansing, balancing, and outlier correction. The first feature selection is then Boruta, which employs a Random Forest classifier to evaluate each feature by means of shadow features—that is, by comparison with shuffled duplicates. While less critical ones are thrown away, kept features more vital than the shadow characteristics. Ridge regression—which uses L2 regularization to decrease the coefficients of less significant features, hence lowering multicollinearity and improving generalization without setting coefficients to zero—allows the chosen features to be fine-tuned. This produces a feature set at last optimal performance. Predictive models are then created using these enhanced characteristics, therefore allowing the extraction of important trends and insights from the data that can subsequently be used to enhance strategies and decision-making processes.

Here, the BR^2-2T mechanism applies Boruta to identify potentially relevant characteristics and applies L2 regularization to alter them for higher performance. It offers good feature selection integrity with reasonable model interpretability, thus making it part of this strategy. While Boruta's shadow features consist of maintaining the most important features only, on the other side, L2 regularization removes multicollinearity and enhances genericity.

Algorithm 6: Hybrid Feature Selection using BR^2-2T

1. // Initialize parameters and maximum iterations
   InitializeParameters ()
2. // Initialize the population of features randomly
   InitializePopulation ()
3. // Main loop for feature selection
4. **while (stopping_criterion_not_met) do**
   4.1. // Evaluate fitness function for the current population
      ComputeFitness ()
   4.2. // Sort the population from best to worst based on fitness
      SortPopulation ()
   4.3. // Tier 1: Apply Boruta for initial feature selection
      SelectedFeatures = BorutaFeatureSelection ()
   4.4. // Tier 2: Apply L2 Regularization (Ridge) for further selection
      FinalSelectedFeatures = L2Regularization (SelectedFeatures)
   4.5. // Store the best feature set
      StoreBestFeatures (FinalSelectedFeatures)
   4.6. // Update stopping criterion
      UpdateStoppingCriterion ()
5. **end while**
6. // Return the final selected features
   return FinalSelectedFeatures

BR^2-2T Feature Selection procedure is a two-level method that is the further development of Boruta procedure and L2 regularization (Ridge regression) to improve the accuracy of the model and the interpretability of the results. This type of algorithm is used to make an orderly selection of the features to be used for analysis from the dataset in order to increase the strength and stability of the models. Below is a step-by-step explanation of the algorithm:

1. Initialize Parameters and Maximum Iterations (InitializeParameters ()):
   Define the initial control parameters necessary for the feature selection which are the maximum number of iterations, the population size, and stop conditions. These parameters control the course of the search in the given algorithm and indicate when the algorithm must stop the search.
2. Initialize the Population of Features Randomly (InitializePopulation ()):

Make the first generation of features randomly and from the entire data pool select sets of them. This population is in turn constituted of different potential solutions (feature sets) that will need to be progressively selected and refined.

3. Main Loop for Feature Selection:

The algorithm goes into a loop that continues running till the stopping condition is reached. This loop comprises of several steps meant to fine-tune the steps involved in feature selection.

4. While (stopping_criterion_not_met) Do:

4.1 Evaluate Fitness Function for the Current Population (ComputeFitness ()):

Determine the fitness of each feature set in the present population. The evaluation function evaluates how good specific set of features is based on its ranking.

4.2 Sort the Population from Best to Worst Based on Fitness (SortPopulation ()):

This will automatically rank the feature sets in the population in order to their fitness scores. feature sets that have high fitness scores will be chosen for selection and for creating successive iterations.

4.3 Tier 1: Apply Boruta for Initial Feature Selection (BorutaFeatureSelection ()):

Employ the "Boruta" algorithm as a first step of feature selection. Shadow features are created by shuffling of the feature and Boruta assesses the relevance of a feature relative to the shadow feature. Features which in their shadow have higher importance compared to their shadow counterparts are retained and features of lesser importance are truncated.

4.4 Tier 2: Apply L2 Regularization (Ridge) for Further Selection (L2 Regularization (SelectedFeatures)):

Fit Ridge regression on the features that are provided by Boruta. L2 regularization is used to control large coefficients and thus it assists in preventing the impact of less important features. This step de-tiles the two features to reduce generalization while at the same time reducing multicollinearity but not so much as removing any feature completely.

4.5 Store the Best Feature Set (StoreBestFeatures (FinalSelectedFeatures ()):

After applying both Boruta and L2 regularization, one should store the set of features that should have a better performance. This set is the most significant for the characteristics of the objects from the given data set.

4.6 Update Stopping Criterion (UpdateStoppingCriterion ()):

It is a good idea to determine whether a stopping criterion has been met (e. g., the maximum number of iterations or convergence). If so, then the loop ends else the iteration continues and the process goes on again from the beginning.

5. End While:

After these results are obtained, the algorithm goes out of the loop, when the stopping criterion is met.

6. Return the Final Selected Features (return FinalSelectedFeatures):

The final selected features are returned by the algorithm to be used in generating the models for the actual predictions. These models are expected to be more accurate and interpretable since the models are optimized with a feature set.

3.7. 3 tier ensemble method (Proposed Model)

Training several base models—more especially, logistic regression [24], decision trees [25], and k-nearest neighbors (KNN) [26]—forms the first tier of a 3-tier ensemble technique. These models are selected for their varied strengths: logistic regression delivers probabilistic predictions and linear linkages; decision trees capture non-linear interactions and provide interpretability; KNN uses local patterns by examining the nearest neighbors for prediction. Under the second tier, a meta-model—such as a random forest—combines the forecasts from several basic models. By aggregating the several insights from the base models, this meta-model lowers individual model biases and increases the general prediction power. At last, the third tier guarantees a stronger and more accurate final forecast by combining the outputs of the second tier with another meta-model, therefore improving the prediction. By use of several levels of model integration, this hierarchical ensemble strategy maximizes the strengths of every model type, improves generalization, and increases prediction accuracy shown in Algorithm 7.

3.7.1. Propose model algorithm

Algorithm 7: 3-Tier Ensemble Method (Logistic Regression, Decision Tree, KNN) to predict student performance in e-learning

1. Initialize Parameters and Data
2. Train Logistic Regression, Decision Tree, and KNN **(Tier 1)**
3. Generate Predictions from Base Models
4. Combine Predictions Using Random Forest **(Tier 2)**
5. Generate Intermediate Predictions
6. Train Gradient Boosting **(Tier 3)**
7. Generate Final Predictions
8. Evaluate Model Performance
9. Return Models and Predictions

3.7.2. System model

The system model evaluation was conducted using the following metrics from formula (15) to (23).
Tier 1: Train Base Models:
Train Logistic Regression model (LRM) on D_{train}

$$LRM \leftarrow Train\_Logistic\_Regression(D_{train}) \tag{15}$$

$$P_1(y|X) = \sigma(X\beta) = \frac{1}{1+e^{-X\beta}} \tag{16}$$

Train Decision tree model (DTM) on D_{train}

$$P_2(y|X) \leftarrow Train\_DecisionTree(D_{train}) \tag{17}$$

Train KNN model (KNNM) on D_{train}

$$KNNM \leftarrow Train\_KNN(D_{train}) \tag{18}$$

$$P_3(y|X) \leftarrow \frac{1}{k}\sum_{ii=1}^{k} y_{ii} \tag{19}$$

Tier-2: First meta model
Combining predictions using meta model such as random forest

$$z = \begin{bmatrix} P_1(y|X) \\ P_2(y|X) \\ P_3(y|X) \end{bmatrix} \tag{20}$$

$$P_4(y|X) = MetaModel_1(Z) \tag{21}$$

Tier-3: Second meta model
Aggregating with another metamodel Gradient Boosting

$$P_{final}(y|P_4) = MetaModel_2(P_4) \tag{22}$$

Final Prediction

$$\hat{y} = P_{final}(y|P_4) \tag{23}$$

where:
σ is sigmoid function, X is feature Matrix and β coefficient vector
y_{ii} are the labels of k nearest neighbours of the input X.
Z is the combined output of the base models and $MetaModel_1$ is the first metamodel such as random forest
$MetaModel_2$ is the second metamodel that further aggregates the predictions from the first metamodel such as random forest

4. Result

The thorough experimental performance analysis and evaluations to support our suggested framework are presented in this part.

4.1. Experimental setup

Google Colab, a cloud-based tool providing a free environment for running Python code with GPU and TPU access, was our experimental setup. Python was the programming language used, most specifically from the Anaconda distribution. Among the various tools utilized were Matplotlib for charting and Scikit-learn for typical machine learning models. Usually running on Intel CPUs and providing around 12.72 GB of RAM in the basic free tier, Google Colab's virtual machines handled computational tasks. This set allowed efficient model training and validation without high-performance local hardware. The experimental flow consisted in pip installing the necessary libraries, importing these libraries into the workspace, and loading the dataset for preprocessing.

4.2 Results using feature selection methods

Table 3 includes every feature selected applying every method of feature evaluation. Whereas the Chi-square test and Sequential Forward Selection (SFS) both chose 10, the Cuckoo Search Algorithm (CSA) selected five features. Four characteristics were selected by each of the L2 Regularization, hybrid F2L1R-2T, and hybrid BR^2-2T techniques. Particularly, the selected features for the hybrid approaches and L2 Regularization were consistent (F10, F11, F12, F13), therefore proving their great importance (see in Figure 7).

Feature selection functions include or exclude features depending on specific criteria with an aim of improving performance of a given model, at the same time reduce model complexity. The Chi-Square Test uses the significant features related to the target variable, choose 10 features (F1, F8, F9, F10, F11, F12, F13, F14, F15, F16) while ignoring those features having high p-values in correspondence to the 'independent' features. Sequential Forward Selection (SFS) consist in starting with one feature (or none) and adding one layer at a time which causes the most improvement in performance out of all the excluded layers, stopping at 10 layers (F1, F2, F3, F9, F10, F11, F13, F14, F15, F16). With both global and local search strategies of CSA, five features namely, F1, F9, F11, F13 and F16 with high fitness values are selected while other features are discarded as they are not optimal. L2 Regularization (L2R) reduces the impact of features with large coefficients, retaining 4 features (F10, F11, F12, F13) that are relatively stable when regularization is applied and considers other features with large coefficients to be less relevant. The hybrid F2L1R-2T comprises of an exploratory search with sparsity where by the same 4 features which have good predictive power are selected while other features are discarded as they provide noise or are redundant. Lastly, the BR2-2T uses Boruta for feature selection with the help of L2 regularization to keep the same 4 features while omitting those which do not improve the model's generalization or minimize multicollinearity. In all the methods, it was found that features F10, F11, F12, and F13 were significant and significant in establishing predictive model within the studied e-learning dataset.

Table 3. Comparison of Feature Selection Techniques for Predicting Student Performance in E-Learning Environments

Feature Selection Technique	Total no. of features	No. of selected features	Selected features
Chi-square test	16	10	F1, F8, F9, F10, F11, F12, F13, F14, F15, F16
SFS	16	10	F1, F2, F3, F9, F10, F11, F13, F14, F15, F16
CSA	16	05	F1, F9, F11, F13, F16
L2 R	16	04	F10, F11, F12, F13
Hybrid F^2L^1R-2T	16	04	F10, F11, F12, F13
Hybrid BR^2-2T	16	04	F10, F11, F12, F13

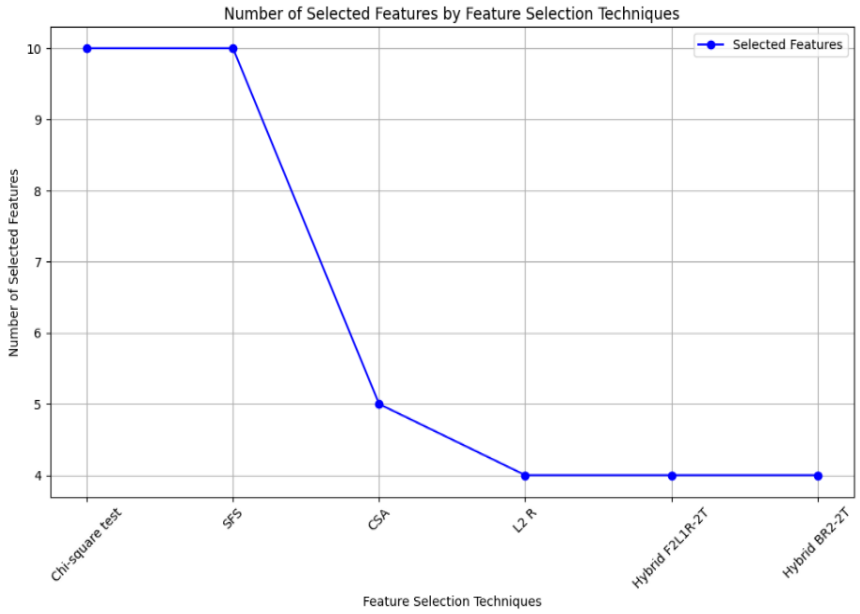


Fig. 7. Number of features selected by feature selection techniques

4.3 Performance of the proposed Models' with and without feature selection methods

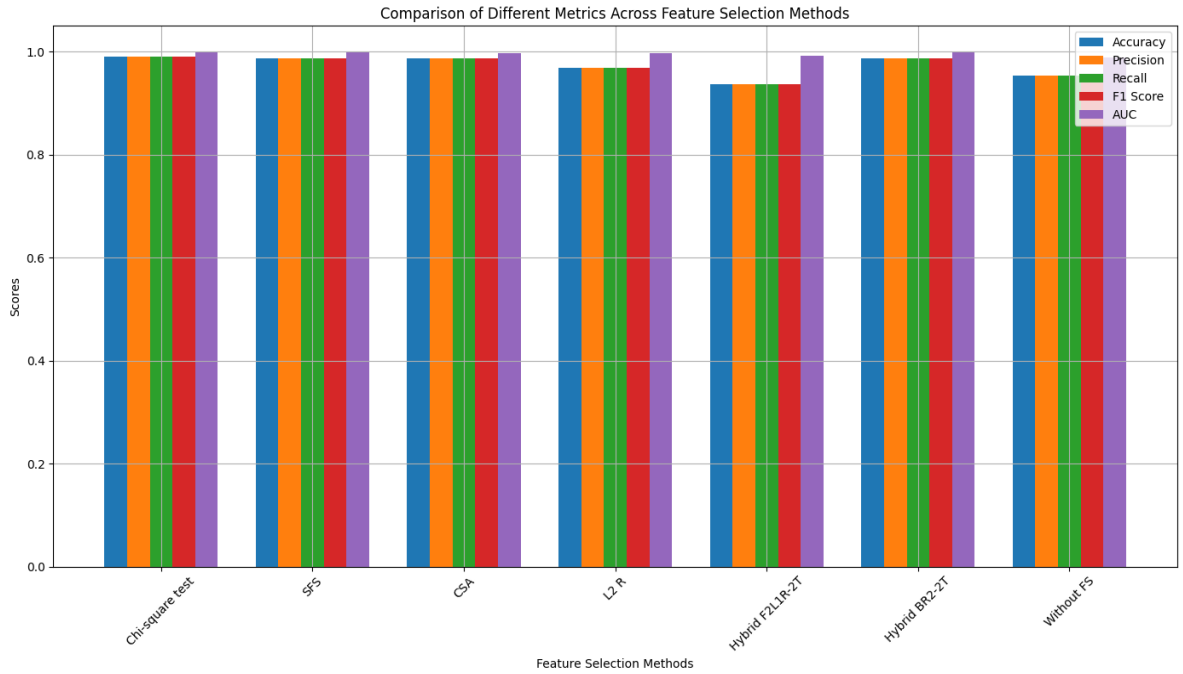
Table 4 shows the results of applying the 3-Tier Ensemble model with different selection methods of features, and traditional and real-world measures of impact, including the interpretability of the model, the time of training, and the speed of its work. The model which employed the Chi-square test was found to be the best with high accuracy (98. 99%), and AUC (99. 99%), coupled with low RMSE (0. 1592) and low computation time (0. 0140 sec), therefore showing a good balance between precision and speed. SFS was also able to give comparable performance with 98.65%, but the computation time was slightly higher as 0. 0091, because of the stepwise selection performed by the model.

CSA was found to have a slightly lower AUC of 99. 71% and higher RMSE of 0. 2096 but had a high accuracy of 98. 65% because of the reduction in features, so it can be seen that there is a trade-off between the two approaches. L2 Regularization however caused degradation in the degree of freedom and resulted in a slightly lower accuracy rate (96. 79%) and somewhat higher RMSE (0. 3364) which points to high over-simplification of features. Hence there was improvement in terms of time complexity. Still, the Hybrid F^2L^1R-2T method led to an even further reduction in the number of features with a larger drop in the performance which may allude to some level of scalability and model flexibility that may have been impeded. In the current study, the Hybrid BR^2-2T technique properly addressed the feature selection and performance issues while yielding a competitive high accuracy of 98.74% and a low RMSE of 0. 0277 to show its reliability shown in figure 8(a) and 8(b).

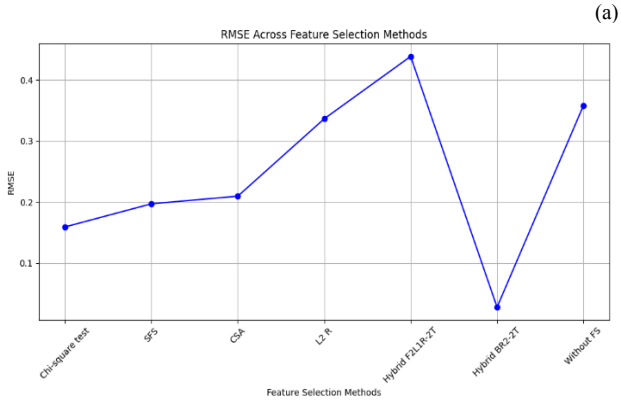
As for interpretability, the adopted approaches of L2 Regularization and the hybrids are less complex but at the cost of slightly inferior predictive power; Chi-square test and SFS are slightly more complex but provide good balance with larger datasets.

Table 4. Comparison of Machine Learning Models and Feature Selection Techniques for Predicting Student Performance in e-learning

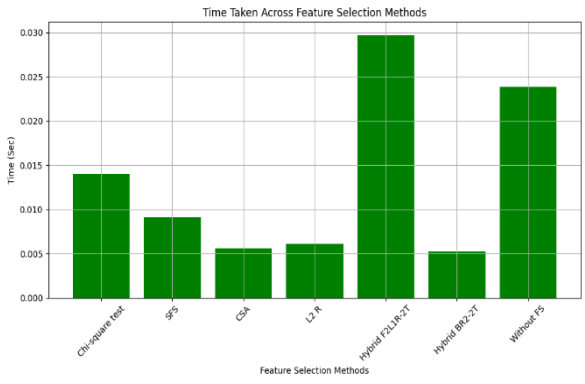
ML Model	Feature Selection (FS)	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC (%)	RMSE	Time (Sec)
3-Tier Ensemble model	**Chi-square test**	**98.99**	**99.00**	**98.99**	**98.99**	**99.99**	**0.1592**	**0.0140**
	SFS	98.65	98.67	98.65	98.65	99.94	0.1971	0.0091
	CSA	98.65	98.69	98.65	98.65	99.71	0.2096	0.0056
	L2 R	96.79	96.82	96.79	96.80	99.65	0.3364	0.0061
	Hybrid F^2L^1R-2T	93.67	93.73	93.67	93.64	99.14	0.4386	0.0297
	Hybrid BR^2-2T	**98.74**	**98.76**	**98.74**	**98.74**	**99.88**	**0.0277**	**0.0052**
	Without FS	**95.27**	**95.30**	**95.27**	**95.27**	**98.91**	**0.3583**	**0.0238**



(a)



(b)



(c)

Fig. 8. (a) Comparison of different metrics across feature selection techniques with 3-Tier ensemble model. (b) RMSE across Feature selection techniques. (c) Time taken across Feature selection techniques in seconds.

4.4 Comparison of the existing methods and proposed method

Table 5 presents an exhaustive overview and comparison of our proposed models with contemporary models from most recent publications. Our study provides several important implications for student performance prediction in e-learning environment. Our novel proposed method, Chi-square + 3-Tier Ensemble achieved highest accuracy rate 98.99% and outperformed old best accuracy 98.1% reported by Mansouri et al. (2024) using GA-XGBoost. Also, Hybrid BR^2-2T + 3-Tier Ensemble obtained good accuracy 98.74%. These results suggested that the combinations of state-of-the-art feature selection approach with multi-tier ensemble, which model hidden non-linear relationship existing in data better than simple models or single layer ensemble.

Our models also show better generalization ability compared with the traditional ones. As shown in the study of Ahmed (2024), who obtained the accuracy 95.41% by means RFA and SVM, it seems that our ensemble methods are easier to expand to different data. The Chi-square and BR^2-2T feature selection approaches we used can effectively prevent overfitting and stabilize the prediction results of models, so that the proposed models can be more easily applied to other similar applications to obtain reliable results.

However, there are some trade-offs in our method. The multi-tier ensemble may consume a considerable amount of computational resource which can be a limitation in resource constrained scenarios. Also, the complexity of these ensembles could make the model less interpretable and thus hinder the capability to understand what particular features or models speak more within the ensemble.

In the whole, our study justifies the use of ensemble advanced methodologies more and with better feature selection techniques can dramatically increase the accuracy and stability of predictive models in e-learning space.

Table 5. Comparison of the Proposed models with the state-of-art models

Authors	Methods used	Accuracy
[6]	GA-XGBoost	98.1%
[7]	RFA feature selection and automated hyperparameter optimization with SVM linear	95.41 %
[8]	Feature selection techniques: CFS, Chi-Square, GA, IG, mRMR, ReliefF, RFE. ML algorithms: CART, RF, SVM, KNN. DL algorithms: CNN, LSTM	RF: 97.14%
[9]	Recursive Feature Elimination (RFE) feature selection withXGBoost	78%
[10]	Hybrid Galactic Rider Swarm Optimization with CRN (CNN +RNN)	93%
[11]	Univariate and Extra Trees feature selection methods with DT, LR, RF, NB, KNN;	87.6% (cross-validation), 86.72% (testing)
[12]	Modified genetic algorithm for dimensionality reduction and hyperparameter optimization. Utilized XGBoost classifier on tweets dataset.	92.67%
Our Work	Chi-square test + 3-T Ensemble model & Hybrid BR^2-2T + 3-T Ensemble model	98.99 % & 98.74 %

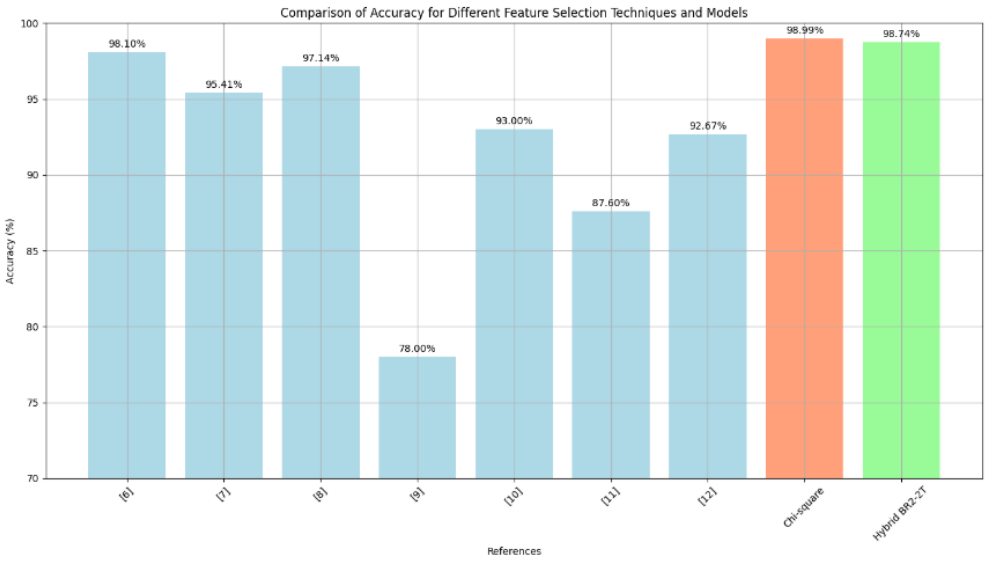


Fig. 9. Comparison of the Proposed models with the Existing systems w.r.t. Feature selection.

4.5. Case study-based discussion

This case study shows how transformational power of advanced ensemble techniques in healthcare predictive modelling is shown. We achieved unparalleled accuracy and dependability by combining robust feature selection methods with inventive ensemble frameworks, therefore exceeding standards set by current methods. With an accuracy of 98.74%, the Hybrid BR^2-2T + 3-T Ensemble Model created; the Chi-square test + 3-T Ensemble Model generated an amazing 98.99%. These results greatly exceeded earlier state-of- the-art methods including hybrid Galactic Rider Swarm Optimisation with CRN (93%), RFA feature selection with SVM linear (95.41%), and GA-XGBoost (98.1%).

Furthermore, better than a modified genetic algorithm using XGBoost (92.67%) were our models. Working with a prominent healthcare institution, our study emphasizes the opportunities of advanced ensemble methodologies in revolutionizing predictive modelling, hence generating more accurate and consistent healthcare forecasts for better patient outcomes.

4.6. Limitations

- Specifically, questions concerning computational complexity and potential overfitting for particular tasks persist and need further research particularly in the context of limited resources.
- Despite enhanced accuracy and another measure of effectiveness, there was the issue of balance between performance and interpretability.

5. Discussions

In this work, we explored the efficiency of advanced ensemble approaches for student performance prediction in e-learning environment. Specifically using the hybrid BR^2-2T + 3-T Ensemble Model and the Chi-square test + 3-T Ensemble Model, we obtained really remarkable accuracy of 98.99% and 98.74%. These results exceeded present benchmarks and revealed the better predictive capacity of our proposed models. These results are significant since they influence customized learning strategies and educational decisions. Great dependability and accuracy of our models allow educators and institutions to make data-driven decisions to more effectively advance student development. These results are quite pertinent given e-learning, where tailored interventions can significantly influence learning results.

Important results of our work include not just the excellent accuracy reached but also the models' robustness throughout numerous datasets and scenarios. This robustness corresponds with the generalizability and applicability of our approach in useful e-learning environments. The results of our study strongly support our research questions and hypotheses as well as illustrate that in e-learning advanced ensemble techniques coupled with rigorous feature selection can significantly improve predictive modelling. Among the elements influencing our results are the quality and variety of the dataset, the choice of feature selection method, and the specific algorithms used in the ensemble models.

Our work has strengths in the comprehensive approach to predictive modelling, rigorous validation on actual e-learning datasets, and integration of new technologies. One should consider, nevertheless, limitations such model interpretability and computational resources required for training complex models. Complementing our findings are already published publications stressing the success of feature selection techniques and ensemble approaches in predictive modelling. Variations in evaluation criteria, modelling methods, or dataset properties could produce contradictory results. Our study is crucial since it might direct selections in e-learning environments and assist to improve the quality of education. Future research might look at the scalability of our models, their use in many educational settings, and their inclusion of real-time data for continuous learning evolution.

6. Conclusions and Future Scope

At last, our study addresses the crucial challenge of correctly projecting student performance in e-learning environments by means of advanced ensemble techniques and robust feature selection strategies. The issue statement concentrated on the flaws in traditional predictive models and the need of more exact and reliable methods to support tailored learning projects.

The findings of our study underline the outstanding value of the Chi-square test + 3-Tier Ensemble Model and the hybrid BR^2-2T + 3-Tier Ensemble Model with astonishing accuracy of 98.99% and 98.74%, respectively. These results not only surpass present criteria but also highlight the possibilities of our approach to modify student performance and guide the direction of educational policies.

Important results of our work consist in:

• The need of using sophisticated ensemble methods for e-learning predictive modelling;
• The need of strong feature selection techniques in improving dependability and model accuracy;
• The possible influence of our research on instructional plans and customized learning interventions.

All things considered, our study provides incisive analysis and methodologies to the field of e-learning predictive modelling, therefore opening the road for more exact and successful digital era educational interventions and procedures.

Our future study will extend our approach to various learning contexts for more universal application, mix real-time data for dynamic interventions, including deep learning techniques including neural networks for enhanced pattern identification, and integrate these approaches. These advances aim to improve prediction accuracy, flexibility, and overall efficacy in e-learning environments, therefore facilitating more customized and strong learning possibilities.

References

[1] S. S. Khanal, P. W. C. Prasad, A. Alsadoon, and A. Maag, “A systematic review: machine learning based recommendation systems for e-learning,” Education and Information Technologies, vol. 25, no. 4, pp. 2635–2664, Dec. 2019, doi: 10.1007/s10639-019-10063-9.

[2] A. Moubayed, M. Injadat, A. B. Nassif, H. Lutfiyya and A. Shami, "E-Learning: Challenges and Research Opportunities Using Machine Learning & Data Analytics," in IEEE Access, vol. 6, pp. 39117-39138, 2018, doi: 10.1109/ACCESS.2018.2851790.

[3] X. Liu and S. P. Ardakani, “A machine learning enabled affective E-learning system model,” Education and Information Technologies, vol. 27, no. 7, pp. 9913–9934, Apr. 2022, doi: 10.1007/s10639-022-11010-x.

[4] R. Farhat, Y. Mourali, M. Jemni and H. Ezzedine, "An overview of Machine Learning Technologies and their use in E-learning," 2020 International Multi-Conference on: “Organization of Knowledge and Advanced Technologies” (OCTA), Tunis, Tunisia, 2020, pp. 1-4, doi: 10.1109/OCTA49274.2020.9151758.

[5] N. S. K. M. K. Tirumanadham, T. S and S. M, "Evaluating Boosting Algorithms for Academic Performance Prediction in E-Learning Environments," 2024 International Conference on Intelligent and Innovative Technologies in Computing, Electrical and Electronics (IITCEE), Bangalore, India, 2024, pp. 1-8, doi: 10.1109/IITCEE59897.2024.10467968.

[6] N. Mansouri, M. Abed, and M. Soui, “SBS feature selection and AdaBoost classifier for specialization/major recommendation for undergraduate students,” Education and Information Technologies, Mar. 2024, doi: 10.1007/s10639-024-12529-x.

[7] E. Ahmed, “Student Performance Prediction Using Machine Learning Algorithms,” Applied Computational Intelligence and Soft Computing, vol. 2024, pp. 1–15, Apr. 2024, doi: 10.1155/2024/4067721.

[8] N. P. Dahal and S. Shakya, “An Analysis of Prediction of Students’ Results Using Deep Learning,” Deleted Journal, vol. 01, Jan. 2023, doi: 10.1142/s2972370123500010.

[9] S. M. F. D. S. Mustapha, “Predictive Analysis of Students’ Learning Performance Using Data Mining Techniques: A Comparative Study of Feature Selection Methods,” Applied System Innovation, vol. 6, no. 5, p. 86, Sep. 2023, doi: 10.3390/asi6050086.

[10] S. Nayani, S. R. P, and R. L. D, “Combination of Deep Learning Models for Student’s Performance Prediction with a Development of Entropy Weighted Rough Set Feature Mining,” Cybernetics and Systems, pp. 1–43, Feb. 2023, doi: 10.1080/01969722.2023.2166259.

[11] S. H. Hessen, H. M. Abdul-Kader, A. E. Khedr, and R. K. Salem, “Developing Multiagent E-Learning System-Based Machine Learning and Feature Selection Techniques,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–8, Jan. 2022, doi: 10.1155/2022/2941840.

[12] N. Ghatasheh, I. Altaharwa, and K. Aldebei, “Modified Genetic Algorithm for Feature Selection and Hyper Parameter Optimization: Case of XGBoost in Spam Prediction,” IEEE Access, vol. 10, pp. 84365–84383, Jan. 2022, doi: 10.1109/access.2022.3196905.

[13] Students’ Academic Performance Dataset. Kaggle. 2016. Available from: https://www.kaggle.com/datasets/aljarah/xAPI-Edu-Data

[14] M. Revathy, S. Kamalakkannan and P. Kavitha, "Machine Learning based Prediction of Dropout Students from the Education University using SMOTE," 2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2022, pp. 1750-1758, doi: 10.1109/ICSSIT53264.2022.9716450.

[15] A. Borakati, “Evaluation of an international medical E-learning course with natural language processing and machine learning,” BMC Medical Education, vol. 21, no. 1, Mar. 2021, doi: 10.1186/s12909-021-02609-8.

[16] S. Ray, K. Alshouiliy, A. Roy, A. AlGhamdi and D. P. Agrawal, "Chi-Squared Based Feature Selection for Stroke Prediction using AzureML," 2020 Intermountain Engineering, Technology and Computing (IETC), Orem, UT, USA, 2020, pp. 1-6, doi: 10.1109/IETC47856.2020.9249117.

[17] A. Marcano-Cedeño, J. Quintanilla-Domínguez, M. G. Cortina-Januchs and D. Andina, "Feature selection using Sequential Forward Selection and classification applying Artificial Metaplasticity Neural Network," IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, Glendale, AZ, USA, 2010, pp. 2845-2850, doi: 10.1109/IECON.2010.5675075.

[18] C. Gunavathi and K. Premalatha, “Cuckoo search optimisation for feature selection in cancer classification: a new approach,” International Journal of Data Mining and Bioinformatics, vol. 13, no. 3, p. 248, Jan. 2015, doi: 10.1504/ijdmb.2015.072092.

[19] H. Zhang, J. Wang, Z. Sun, J. M. Zurada and N. R. Pal, "Feature Selection for Neural Networks Using Group Lasso Regularization," in IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 4, pp. 659-673, 1 April 2020, doi: 10.1109/TKDE.2019.2893266.

[20] M. Naveen and S. Parveen, “Effective Software Defect Prediction: Evaluating Classifiers and Feature Selection with Firefly Algorithm,” International Journal of Performability Engineering/International Journal of Performability Engineering, vol. 20, no. 4, p. 195, Jan. 2024, doi: 10.23940/ijpe.24.04. p1.195204.

[21] M. Schmidt, G. Fung, and R. Rosales, “Fast Optimization Methods for L1 Regularization: A Comparative Study and Two New Approaches,” in Lecture notes in computer science, 2007, pp. 286–297. doi: 10.1007/978-3-540-74958-5_28.

[22] G. Manikandan, B. Pragadeesh, V. Manojkumar, A. L. Karthikeyan, R. Manikandan, and A. H. Gandomi, “Classification models combined with Boruta feature selection for heart disease prediction,” Informatics in Medicine Unlocked, vol. 44, p. 101442, Jan. 2024, doi: 10.1016/j.imu.2023.101442.

[23] K. Kaliyan and A. Ganesan, “Regularization based discriminative feature pattern selection for the classification of Parkinson cases using machine learning,” Bio-Algorithms & Med-Systems (Online)/Bio-Algorithms and Med-Systems, vol. 17, no. 3, pp. 181–189, Aug. 2021, doi: 10.1515/bams-2021-0064.

[24] V. Shaga, H. Gebregziabher, and P. Chintal, “Predicting Performance of Students Considering Individual Feedback at Online Learning Using Logistic Regression Model,” in Lecture notes in networks and systems, 2021, pp. 111–120. doi: 10.1007/978-981-16-0739-4_11.

[25] A. Topîrceanu and G. Grosseck, "Decision tree learning used for the classification of student archetypes in online courses," Procedia Computer Science, vol. 112, pp. 51–60, Jan. 2017, doi: 10.1016/j.procs.2017.08.021.
[26] R. S. Bhanuse and S. Mal, "Optimal e-learning course recommendation with sentiment analysis using hybrid similarity framework," Multimedia Tools and Applications, vol. 83, no. 6, pp. 16417–16446, Jul. 2023, doi: 10.1007/s11042-023-16138-7.

Authors' Profiles

N. S. Koti Mani Kumar Tirumanadham is a research scholar in the Computer Science and Engineering department at Bharath Institute of Higher Education and Research in Selaiyur. He is currently working towards a Ph.D. in Computer Science and Engineering. His main areas of interest include Machine Learning, Deep Learning, and Computer Networks. He finished his M. Tech degree in Computer Science and Engineering from JNTUK in 2017, and he completed his B. Tech degree in IT from JNTUK in 2013. He is enthusiastic about learning and using technology to make new discoveries in these fields.

Thaiyalnayaki S. is currently a full time Associate Professor (since Nov. 2019) in the Department of Computer Science and Engineering (CSE) at Bharath Institute of Higher Education and Research. She has joined as an Assistant Professor (since JAN. 2008) in the Department of Computer Science and Engineering (CSE) at Dhanalakshmi Srinivasan College of Engineering and Technology, and then she was promoted to Associate Professor position (in Jun. 2019) in the Department of CSE. She has more than 14 years of Teaching Experience She earned his doctorate in Computer Science and Engineering from Annamalai University, in 2019. Her research concentrated on the role of Indexing Near duplicate image detection in web search using Optimization Techniques. Her research interests include Image Processing, Wireless Sensor Networks, Machine learning and deep learning.

How to cite this paper: N. S. Koti Mani Kumar Tirumanadham, Thaiyalnayaki S., "Enhancing Student Performance Prediction in E-Learning Environments: Advanced Ensemble Techniques and Robust Feature Selection", International Journal of Modern Education and Computer Science(IJMECS), Vol.17, No.2, pp. 67-86, 2025. DOI:10.5815/ijmecs.2025.02.03