



OPEN Employing artificial bee and ant colony optimization in machine learning techniques as a cognitive neuroscience tool

Kajal Mahawar¹, Punam Rattan¹, Ammar Jalamneh²,
Mohd Shukri Ab Yajid³, Omar Abdeljaber⁴, Raman Kumar^{5,6}, Ayodele Lasisi⁷ &
Muhammad Imam Ammarullah^{8,9}✉

Higher education is essential because it exposes students to a variety of areas. The academic performance of IT students is crucial and might fail if it isn't documented to identify the features influencing them, as well as their strengths and shortcomings. The student academic prediction system needs to be enhanced so that teachers can forecast their students' performance. Numerous studies have been conducted to increase the prediction accuracy of IT students, but they encountered difficulties with unbalanced data and algorithm tuning. To address these issues, the study proposed different machine learning (ML) algorithms that handled imbalanced data by applying the synthetic minority oversampling technique (SMOTE) and employing hyperparameter tuning algorithms to enhance prediction during the training process. The ML models we used were decision tree (DT), k-nearest neighbor, and XGBoost. The models were fine-tuned by applying Ant colony optimization (ACO) and artificial bee colony optimization techniques. Subsequently, these optimization techniques further enhanced the performance of the models. After comparing them, the results showed that SMOTE and ACO combined with the DT model outperformed other models for academic prediction. Additionally, the study utilized the Kendall Tau correlation coefficient technique to analyze the correlation between features and identify factors that positively or negatively impact student success.

Keywords Student learning outcomes, Machine learning, Hyperparameter tuning, Decision tree, Artificial bee colony, Ant colony optimization, Kendall Tau

Indeed, one of the most challenging and extensively researched areas in machine learning (ML) revolves around modeling student performance¹. Predicting IT students' academic achievements is pivotal for educational planning and decision-making. Tailoring ML techniques to address the distinct challenges faced by these students offers promising avenues to enhance predictive accuracy and optimize educational outcomes. According to higher education studies, the high attrition rate demonstrates the ineffectiveness of the prior educational initiatives. Significant reforms in higher education are required to address the problem, increase student retention, and raise graduation rates. The crucial stage at which the research concentrated on the features significantly affecting the outcomes was performance prediction. In addition, the prediction models inside the designated domain experienced low efficacy and precision, necessitating modifications to yield superior outcomes suitable for real-time analysis. Nonetheless, for the decision-maker to effectively manage their student, academic prediction from the student needs to be more accurate².

¹Lovely Professional University, Phagwara 144411, Punjab, India. ²College of Arts and Science, Applied Science University, Al Ekir 3201, Kingdom of Bahrain. ³Management and Science University, Shah Alam 40100, Selangor, Malaysia. ⁴Hourani Center for Applied Scientific Research, Al-Ahliyya Amman University, Amman 19328, Jordan. ⁵Department of Mechanical and Production Engineering, Guru Nanak Dev Engineering College, Ludhiana 141006, Punjab, India. ⁶Jadara University Research Center, Jadara University, Irbid 733, Jordan. ⁷Department of Computer Science, College of Computer Science, King Khalid University, Abha 61421, Asir, Saudi Arabia. ⁸Department of Mechanical Engineering, Faculty of Engineering, Universitas Diponegoro, Semarang 50275, Central Java, Indonesia. ⁹Undip Biomechanics Engineering & Research Centre (UBM-ERC), Universitas Diponegoro, 50275 Semarang, Central Java, Indonesia. ✉email: imamammarullah@gmail.com

Despite significant progress in using ML for educational purposes, predicting student academic performance remains a pressing challenge due to the high complexity of factors influencing success and the limited effectiveness of existing prediction models³. Educational institutions face increasing pressures to identify at-risk students early, especially in IT programs with high attrition rates⁴. Integrating cognitive neuroscience, academic performance analysis, and machine learning opens the door to new avenues that can help improve the learning experience for students in IT. Cognitive neuroscience provides insights into the mechanisms of the brain when engaged in learning, memory, and problem-solving. Therefore, it allows a more nuanced understanding of how information gets processed and retained. Academic performance metrics are then associated with these insights to analyze novel strategies to optimize educational outcomes. This interdisciplinary approach is particularly relevant to IT education, where high cognitive demands and learning challenges often overlap with rapidly evolving technological content. Machine learning is now indispensable in education, providing robust methods to analyze complex datasets and uncover patterns that traditional approaches might miss. Predictive modeling allows for identifying at-risk students who may perform poorly, thus ensuring timely interventions. Machine learning algorithm-based personalized learning systems can adapt to individual student needs and provide them with tailored educational experiences. These applications benefit IT students, as their academic performance is affected by cognitive skills, technical aptitude, and study habits. Cognitive neuroscience integrated with machine learning in education is a high-end trend that can potentially transform education. Neuro-cognitive data, in the form of EEG or MRI measurements of brain activity, can hone ML models, providing insights into learning patterns with student precision⁵. These models will assess real-time cognitive load, supporting adaptive learning systems that respond according to the learner's mental state⁶. Applying machine learning to monitor emotional states, engagement, and motivation also improves learning experience personalization, increasing the effectiveness of education for students⁷. However, while such developments are promising, they pose challenges in actual implementation, particularly with issues on ethical grounds regarding the privacy of data, especially when working with sensitive neuro-cognitive data⁸. Moreover, collaboration between cognitive scientists, educators, and ML experts is necessary to deliver practical and scalable solutions⁹. Therefore, the potential benefits of integration in IT education are very high. By aligning the insights of cognitive neuroscience with the capabilities of machine learning, this work aims to advance the analysis of academic performance and create new strategies for improving educational results in IT. Current predictive models often lack precision or fail to address issues such as imbalanced datasets, underrepresenting influential features, and inefficient hyperparameter tuning methods. Motivated by the urgent need for accurate and actionable insights, this research aims to bridge these gaps by developing a robust prediction framework for IT students. The study seeks to empower educators with practical tools to enhance academic outcomes and inform targeted interventions, ultimately improving retention rates and academic achievements in higher education.

Academic success is affected by myriad intricate features, rendering ML particularly appealing given the abundance of educational datasets available. Educational Data Mining (EDM) seamlessly integrates data mining (DM) techniques to refine and predict learners' academic trajectories¹⁰. The EDM process helps educators and education researchers gather information by converting unprocessed data into understandable information. Using the EDM tools, student groups can employ classification techniques more successfully. Moreover, it impacts decision-making processes for administrators, aiming to yield high-quality outcomes¹¹. ML employs computational methods to analyze and visualize educational data. They can help identify problematic student behaviors and offer guidance. Such models support educators in student recruitment, feedback acquisition, and curriculum design¹². Data on education is sourced from various outlets, including surveys, heuristic evaluations, and online platforms. Several DM techniques are employed to tackle educational challenges, with EDM drawing upon various DM methodologies. For instance, classification emerges as a highly effective strategy for constructing predictive educational models, often augmented by optimization techniques to enhance model performance^{13,14}. According to¹⁵, the primary prediction is to analyze the datasets because systems built using unbalanced data failed real-time testing. Furthermore, an imbalanced dataset obscures the optimal features that may harm a student's performance. When imbalanced classes are handled, the model's prediction accuracy increases throughout the training phase. El-kenawy, et al.¹⁶ presented a Greylag Goose Optimization (GGO) algorithm based on a swarm metaheuristic inspired by the efficiency of geese's "V" flight formation. GGO was validated by experiments on UCI datasets and engineering benchmarks. It significantly outperformed other algorithms in terms of accuracy and reliability, as statistically certified by Wilcoxon's rank-sum and ANOVA tests.

In ML, optimization strategies are critical to improving and fine-tuning the effectiveness of predictive models. These methods seek to optimize forecast accuracy, reduce mistakes, and fine-tune model parameters. Standard optimization methods are commonly used in various ML applications, including gradient descent, Adam optimization, stochastic gradient descent, and evolutionary algorithms such as genetic algorithms and particle swarm optimization. These algorithms modify the model's parameters and progressively approach ideal values by evaluating a specified objective function iteratively. Practitioners can increase model efficiency, accelerate convergence, and improve predictive performance across various applications and domains by integrating optimization techniques into ML workflows. This study notably concentrates on two popular optimization techniques, ACO and ABC. The ABC algorithm aims for the best answers by imitating how honey bees forage. In ABC, potential solutions include bees searching the search space, assessing their fitness using a predetermined objective function, and communicating with one another via a waggle dance-like mechanism. This foraging behavior forms the basis of the ABC algorithm, particularly suited for discrete optimization problems, such as hyperparameter tuning for machine learning models. Figure 1 depicts the flowchart of the ABC technique. In this algorithm, the three main components are,

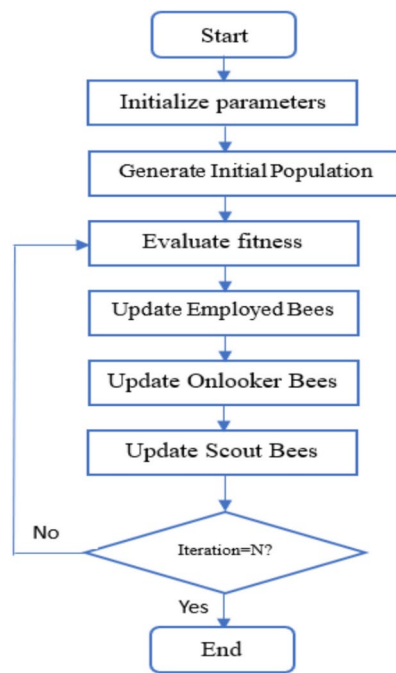


Fig. 1. ABC flowchart for hyperparameter optimization in machine learning models.

- **Employed bees:** The employed bees are tasked with exploring the search space by leveraging existing solutions and realizing new ones through localized searches.
- **Onlooker bees:** These bees select solutions based on the information obtained from employed bees and perform local searches to improve these solutions.
- **Scout bees:** Scout bees are responsible for randomly seeking fresh solutions, primarily when employed and onlooker bees have exhausted their search efforts.

As shown in the figure, the algorithm begins with the initialization of parameters (Step 1). The parameters define the problem's search space and include factors like population size, iteration limits, and other algorithm-specific parameters. Once the parameters are initialized, the algorithm generates an initial population (Step 2), where each solution corresponds to a set of potential hyperparameters for the machine learning model. The fitness of each solution is then evaluated (Step 3), which typically involves training the model with the given hyperparameters and evaluating its performance. Based on their fitness, the employed bees update their positions (Step 4) by exploring neighboring solutions to find better-performing solutions. The onlooker bees then update their solutions (Step 5) based on the fitness of the employed bees' solutions, selecting the best-performing ones. Next, the scout bees are employed (Step 6), searching for entirely new solutions when a specific solution has failed to improve after a certain number of iterations. After the bees have updated their positions and explored the search space, the algorithm checks whether the stopping criterion has been met (Step 7). The process ends if the algorithm has reached the predefined number of iterations or found a solution that meets the fitness threshold (Step 8). If the stopping condition is not satisfied, the algorithm returns to the fitness evaluation step (Step 3), continuing the search for optimal hyperparameters.

Another optimization technique, ACO, is inspired by the foraging behavior of ants and has been applied to various optimization problems, such as routing, scheduling, and combinatorial optimization. ACO is particularly effective for discrete optimization problems that involve large search spaces and complex constraints¹⁷. In nature, ants deposit a chemical substance called pheromone as they forage for food, which helps them remember and communicate the path to the food source. This behavior forms the basis of the ACO algorithm. Figure 2 depicts the flowchart of the ACO technique.

As shown in the figure, the ACO algorithm begins with the initialization of parameters (Step 1). Once the parameters are set, the algorithm generates random solutions (Step 2), where each solution represents a potential combination of hyperparameters for the machine learning model. The fitness of each solution is then evaluated (Step 3) based on model performance, typically measured by accuracy or another relevant metric.

Following this, the pheromone levels are updated (Step 4), reinforcing better solutions and guiding future iterations toward the optimal set of hyperparameters. The algorithm then applies a transition rule (Step 5) to decide whether to explore new solutions or exploit the best-performing ones. A new path is generated (Step 6), and the algorithm checks if the number of iterations has reached the predefined limit (Step 7). The process ends if the stopping criterion is met (Step 8). If not, the algorithm returns to the global random generation step (Step 2) to continue exploring the search space.

Both ABC and ACO algorithms are metaheuristic optimization techniques known for their ability to explore complex solution spaces and find near-optimal solutions efficiently. These algorithms have found applications

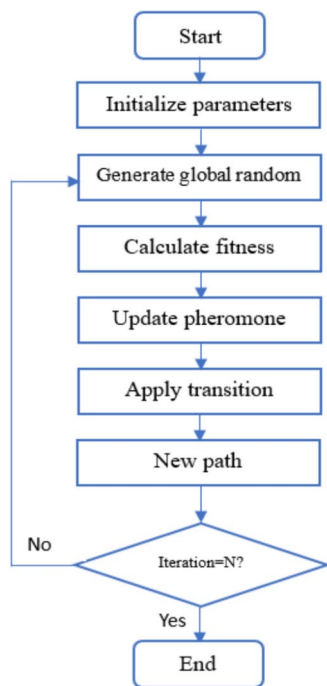


Fig. 2. ACO flowchart for hyperparameter optimization in machine learning models.

in various domains, including ML, where they are utilized to optimize model parameters, feature selection, and hyperparameter tuning, among others. Alongside optimization techniques in ML, various techniques are employed to understand the relationship between features, such as the Kendall tau correlation coefficient (τ)¹⁸. This statistical method is utilized to assess the correlation between two ordinal features. It evaluates the resemblance in ordering data points between the variables, irrespective of their specific values.

This paper presents a novel model for assessing student performance, incorporating a unique set of attributes. Employing diverse ML techniques, the model precisely scrutinizes the dataset to comprehend how students' attributes impact their academic success. Additionally, through the combination of ACO hyperparameter tuning and SMOTE for handling unbalanced datasets, this research study seeks to improve the academic prediction of students based on their performance. The analysis revealed that if the model parameters are appropriately adjusted and the data used is sufficiently balanced, the performance of the ML classifiers could improve. The Kendall-Tau correlation coefficient technique is also used in this study to evaluate the relationship between features and identify variables that are positively or adversely related to student progress. In light of earlier research, the following research objectives are laid for this study.

- To propose a systematic method for improving, developing, and refining ML models to accurately predict IT students' academic performance, aiding educators in identifying students' strengths and weaknesses early on in the ML classifier by incorporating distinct attributes.
- To assess the relationship between characteristics and determine features that positively or negatively connected with academic success using the Kendall Tau correlation coefficient technique and implement the SMOTE to manage and correct imbalanced datasets effectively.
- Hyperparameter tuning using ACO and ABC techniques will be applied, and the performance of various ML classifiers, including DT, KNN, and XGB, will be evaluated.
- To illustrate the superior performance of the ACO-optimized DT classifier, combined with SMOTE, in predicting students' academic outcomes and propose future research directions, including longer-term studies and incorporating additional features and advanced ML approaches.

The study addresses the crucial requirement for precise student academic performance prediction and has significant information for higher education, especially IT departments. With the implications from the study, teachers can more precisely predict student performance. Using the SMOTE, unbalanced datasets in educational data can be effectively managed, and predictions are made in a representative and trustworthy manner. ML performs better and is more successful in predicting academic results when hyperparameter optimization is incorporated using ACO and ABC approaches. The study offers a comparative analysis that identifies the best models for academic prediction by evaluating some ML classifiers, including DT, KNN, and XGB. Educators can better understand the elements that favorably or unfavorably affect academic success by applying the Kendall-Tau correlation coefficient to examine the connections between various features and student achievement. The study establishes the framework for future research to improve comprehension and student academic achievement forecasting by recommending new features, longer-term data collection, and investigating sophisticated ML

techniques. The study offers reliable techniques for forecasting student performance, advances educational data analytics, and helps teachers enhance students' academic progress.

After the introduction in Section "Introduction", this paper is organized as follows: Section "Related works" provides an overview of pertinent literature. Section "Proposed methodology" explores the study's proposed methodology. Section "Results and discussion" explains the experiment's discussion and results. Finally, Section "Conclusions, limitations, ethical and privacy considerations, and future work" summarizes the findings from the analysis and outlines potential directions for future research.

Related works

This section explores previous research endeavors that have investigated the learning performance of students using traditional ML algorithms and studies that have investigated the integration of optimization techniques and the Kendall Tau correlation coefficient.

Najieha et al.¹⁹ introduced a website system built using PHP and Laravel that uses the C4.5 data mining method to forecast students' academic performance. Using statistical patterns and reports protected by digital signatures helped lecturers monitor academic performance by predicting who may make the list and identifying students who might receive poor grades. Gunasinghe et al.²⁰ assessed how well the UTAUT-3 model explained how internet-based technology, such as e-learning, changes education in response to the model's inadequate instructional validity. To determine if one variable cause another, hypotheses were evaluated using a quantitative technique and a logical approach. Simple random selection was used to gather data, and 441 academics were given a self-administered questionnaire using Google Forms. Structural equation modeling was used to analyze the data. In employment education data processing, Fang²¹ integrated classifiers, K-means, and Apriori algorithms to harness data mining technology effectively. Cohausz et al.²² scrutinized the significance of demographic features in at-risk prediction models and assessed their necessity alongside study-related features. Verger et al.²³ introduced a novel metric, Model Absolute Density Distance, for analyzing model discriminatory behaviors independently of predictive performance, alongside visualization-based analysis for fine-grained human assessment of model discrimination between student groups. Alhazmi and Sheneamer¹ analyzed features and predicted students' GPA using clustering and classification algorithms, including the T-SNE algorithm for dimensionality reduction, aiming to provide insights into academic trajectories and enhance student outcomes. Bellaj et al.²⁴ aimed to improve the accuracy of ML algorithms by employing eight ML classifiers, which were optimized through hyperparameter tuning, including various correlation coefficient techniques. Ouyang et al.¹² combined learning analytics techniques with an AI prediction model to improve student learning outcomes in a cooperative learning environment. Chen and Ding¹¹ utilized 'black box' ML models enhanced with educational and socioeconomic data to forecast academic performance while mitigating the influence of logical associations, employing logistic regression, support vector machine, random forest, DT, and neural network techniques. Al-Alawi et al.²⁵ investigated factors adversely affecting academic performance among students using supervised ML techniques, employing the Information Gain algorithm to identify influential features and ensemble methods such as Vote, Bagging, and Logit Boost. Wang¹³ proposed a singular optimized machine-learning approach utilizing the Hybrid Cuckoo Search PSO to analyze factors influencing education. Nie and Ahmadi Dehrashid²⁶ introduced two innovative algorithms, the Harris Hawk's Optimizer, and the Earthworm Optimization Algorithm, to enhance student performance through a series of Adaptive Neuro-Fuzzy Inference System models.

In research that concentrated on ABC and ACO optimization techniques, Teodorović and Dell'Orco²⁷ examined the ABC metaheuristic, which is well-known for its suitability for combinatorial problems, especially uncertainty. The researchers emphasized the ABC algorithm's versatility and usefulness in resolving real-world issues and its handling of a range of optimization tasks. Karaboga and ÇETİNKAYA²⁸ presented a novel technique for creating adaptable finite and infinite impulse response filters using the ABC algorithm. To investigate noise cancellation, researchers ran simulations and evaluated the study approach's efficacy against well-known gradient and evolutionary-based techniques. An improved version of the ABC algorithm designed especially for optimization problems was presented in the work by²⁹. Deb's rule was integrated into this adaptation. The researchers then applied the updated algorithm to four traditional engineering benchmark issues that included continuous and discrete variables.

In addition, Zhang¹⁷ improved the ACO algorithm and ML classification approach by creating a model for student entrepreneurship. In Ye et al.³⁰ study, researchers proposed two novel approaches for selecting wrapper features by integrating hybrid rice optimization and ant colony optimization techniques. Based on ACO³¹, a framework for calculating the weight of each model within the ensemble of ML prediction models was devised, and Kendall tau was applied to analyze the features.

Numerous research endeavors have addressed challenges in predicting student academic achievement using ML and optimization techniques (refer to Fig. 3). However, only a few of these studies have incorporated techniques like ABC and ACO to enhance the learning process. Integrating these approaches aims to bolster the accuracy of results and yield more favorable outcomes.

Proposed methodology

The study aims to improve the ML model by using SMOTE to handle imbalanced datasets and ACO hyperparameter tuning to optimize performance and accuracy in student academic prediction. Three ML classifiers are used as the classification algorithms. The dataset used in this study was collected from three private colleges in Jabalpur, Madhya Pradesh state, India. A questionnaire was prepared and distributed to collect the data, with 1369 IT students responding. The questionnaire was designed using Google Forms for easy distribution and data collection. The dataset consists of 1369 records with 70 features previously. Using the

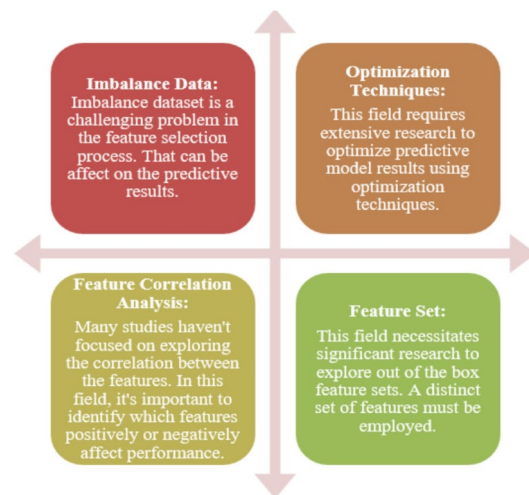


Fig. 3. Analysis of challenges in student academic prediction.

Feature name	Feature	Description
Gender	GEN	Student gender (female, male, or other)
Age	AGE	Student age (≤ 18 years, 19–25 years, > 25 years)
Number of children's	N_CHILD	Student's number of children (None, 1–2, 3–5, more than 5)
Admission type	ADM_TYPE	Student's admission types such as (Regular and Private)
Co-curricular activities in college	CLG_CURRACT	Participated in Co-curricular activities in the college (high, medium, and low)
Attendance	ATTEN	Student's attendance ($\geq 75\%$, 55–74%, $\leq 54\%$)
Self-confident	R_CONF	Student's self-confidence levels such as (1, 2, 3, 4, and 5)
Emotional stability	R_EMOT	Student's emotional stability level such as (1, 2, 3, 4, and 5)
Single parent–child	SIN_PAR	Are students being a single parent–child (yes and no)
Father academic qualification	FAT_ACDQUA	Student's father's academic qualification (None, less than graduation, graduation, post-graduation, higher than post-graduation)
Mother academic qualification	MOT_ACDQUA	Student's mother's academic qualification (None, less than graduation, graduation, post-graduation, higher than post-graduation)
Part-time job	PT_JOB	Are students doing any part-time jobs (yes and no)
Total income	TOT_INC	Student's total income status (< 50 k, 50 k–1 lakh, 1 lakh–2 lakh, 2 lakh–5 lakh, and more than 5 lakhs)
Father occupation	FH_OCCUP	Student's father's occupation (none, self-employed, private sector, government sector, business, and others)
Transportation problem	TRANS_DIFF	Are students facing transportation problems (yes and no)
Gender inequality	GEN_INQ	Are students facing gender inequality issues (yes and no)
Hours spent in coaching classes	H_COCHCL	Numbers of hours spent by the students in coaching classes such as (none, < 1 h, 1–2 h, 2–4 h, and > 4 h)
Hours spent in self-study	H_SELFST	Numbers of hours spent by the students in self-study such as (none, < 1 h, 1–2 h, 2–4 h, and > 4 h)
Hours spent on social sites	H_SOCSITE	Numbers of hours spent by the students on social sites such as (none, < 1 h, 1–2 h, 2–4 h, and > 4 h)
Hours spent playing video games	H_VIDGAM	Numbers of hours spent by the students playing video games such as (none, < 1 h, 1–2 h, 2–4 h, and > 4 h)
Hours spent in café, canteen, and campus	H_CANCAF	Numbers of hours spent by the students in café, canteen, and campus such as (none, < 1 h, 1–2 h, 2–4 h, and > 4 h)
Last year grade (target feature)	TT	Student's first year grade ($\geq 60\%$, $< 60\%$)

Table 1. IT students' dataset and their description.

Chi-square technique, 21 optimal features were identified (Table 1). The framework of the proposed approach is illustrated in Fig. 4, followed by the algorithm.

Algorithm 1: Proposed Method

- Step 1: Start
- Step 2: Execute Data Preprocessing Steps
- Step 3: Feature Correlation Analysis using Kendall Tau
- Step 4: Add SMOTE
- Step 5: Training DT Model with baseline methods;
- Step 6: Training KNN Model with baseline methods;
- Step 7: Training XGB Model with baseline methods;

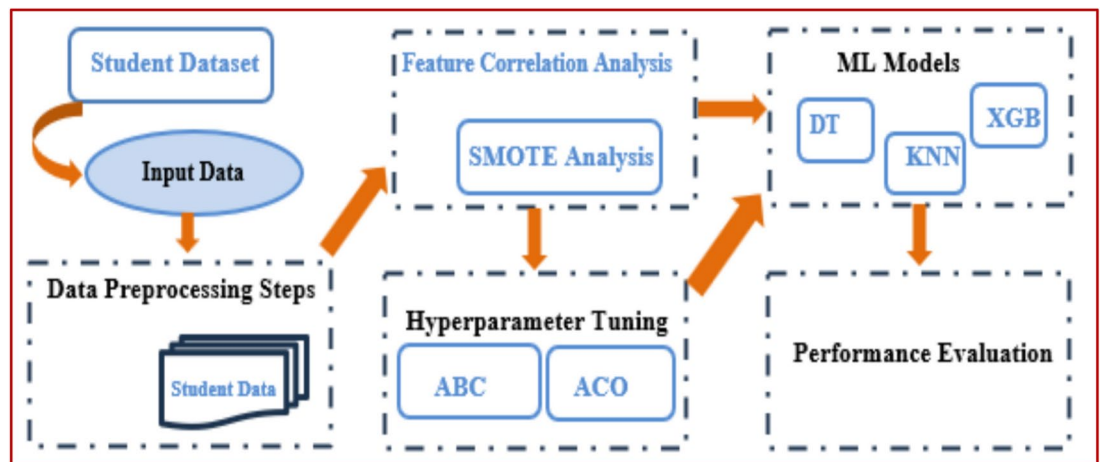


Fig. 4. The study framework.

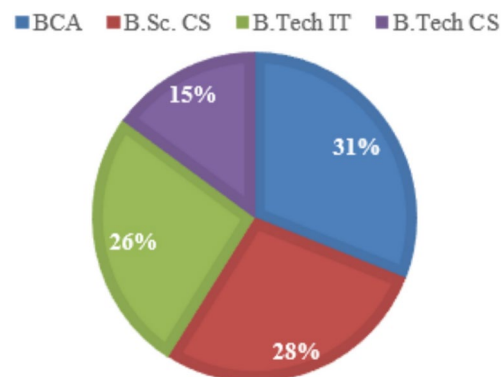


Fig. 5. Demographics by UG courses.

- Step 8: Repeat steps 4, 5, and 6 without and with SMOTE
- Step 9: Evaluate the performance
- Step 10: Tune DT hyperparameters with ABC and ACO without SMOTE
- Step 11: Tune DT hyperparameters with ABC and ACO with SMOTE
- Step 12: Tune KNN hyperparameters with ABC and ACO without SMOTE
- Step 13: Tune KNN hyperparameters with ABC and ACO with SMOTE
- Step 14: Tune XGB hyperparameters with ABC and ACO without SMOTE
- Step 15: Tune XGB hyperparameters with ABC and ACO with SMOTE
- Step 16: Evaluate the performance
- Step 17: End

Business and data understanding

This study focuses on second-year undergraduate IT students from three colleges. The courses included are BCA (Bachelor of Computer Applications) and B.Sc. CS (Bachelor of Science in Computer Science), B.Tech. IT (Bachelor of Technology in Information Technology), and B.Tech. CS (Bachelor of Technology in Computer Science). Figure 5 presents a chart illustrating the percentages of students from these courses who responded to the questionnaire. In the dataset, a grade of more than or equal to 60% indicates good performance, while a grade of less than 60% suggests unsatisfactory performance for a student.

Data preprocessing

Data preprocessing is the process of cleaning and preparing unprocessed data for analysis. This usually entails addressing missing information, standardizing formats, and eliminating duplicates³². This study had no missing values because all questions had to be answered, and any duplicates or outliers were identified and removed manually. The relevance and integrity of the dataset for analysis were also preserved by ensuring that only data from IT students was gathered. To enable seamless analysis, data transformation was carried out to guarantee consistency and compatibility across several platforms. For this, the Excel to CSV format conversion method was used, which streamlined data processing and improved accessibility for analytical needs.

The study utilized the Chi-square statistical technique to determine the optimal feature set that substantially impacts academic performance. This process ensures that only the most informative features are kept around for analysis. The Chi-square (χ^2) method establishes the correlation between two category features in a dataset. The first step in the Chi-square feature selection algorithm is finding each feature's Chi-square value. Higher Chi-square values for features signify a stronger correlation with the target variable and are therefore chosen for inclusion in the model, while lower values might be eliminated³³. Following are the steps of this technique,

Algorithm 2: Chi-square technique

- Step 1: Start
- Step 2: Create a contingency table for each feature that cross-tabulates it against the target feature. This table displays the frequency distribution of the two features and aids in analyzing their connection.
- Step 3: For each contingency table, determine the chi-square value using the Eq. (1).

$$\chi^2 = \frac{\sum (O_i - E_i)^2}{E_i} \quad (1)$$

where the expected frequency of a feature is represented by (E_i) and the observed frequency by (O_i) in the table.

- Step 4: Sort the features in descending order.
- Step 5: Choose the top-k features with the highest value as a final feature set.
- Step 6: End

Feature correlation analysis

Feature correlation analysis focuses on the relationships between the features to determine how various features in a dataset relate to one another. The Kendall Tau correlation coefficient was utilized to decide which features are positively or adversely connected with student success, as shown in Eq. (2).

$$\tau = \frac{(n_c - n_d)}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \quad (2)$$

Here, n_0 signifies the overall number of pairings, n_c indicates the count of discordant pairs and n_1 and n_2 indicate the number of tied values in feature 1 and feature 2, respectively.

Synthetic minority over-sampling technique (SMOTE)

After standardizing and normalizing the dataset, the study utilized SMOTE to find the imbalance in the data. Of the 1369 students, 456 had a success rate of less than or equal to 60%, while the remaining achieved above 60%, with 456 and 913 records, respectively. The dataset exhibited an imbalance in class distribution, and to address this problem, the SMOTE was used. By randomly increasing minority class samples through replication, SMOTE achieves class distribution balance. Through neighborhood exploration and the creation of new data points, this algorithm creates synthetic data. The unbalanced data was successfully managed by using SMOTE before being divided into segments for the ML models' training. Following are the steps of this method,

Algorithm 3: SMOTE steps

Let (X) represent the minority class record and (X_{nn}) represent one of its nearest neighbors.

- Step 1: Start
- Step 2: Calculate the Euclidean distance between X and all other minority class records to find its k nearest neighbors.
- Step 3: Randomly select one of the nearest neighbors, denoted as X_{nn}
- Step 4: Generate a synthetic record (X_{new}) using the Eq. (3).

$$X_{new} = X + rand(0, 1) \cdot (X_{nn} - X) \quad (3)$$

where $rand(0,1)$ is a random value between 0 and 1.

- Step 5: End

Table 2 shows the result of the dataset before and after the technique. The majority class possessed 913 records before SMOTE, compared to 456 records for the minority class. There are 913 records in each of the two classes

Class	Before SMOTE	After SMOTE
Majority (> 60%)	913 records	913 records
Minority ($\leq$ 60%)	456 records	913 records
Total	1369 records	1826 records

Table 2. Dataset before and after SMOTE.

after using SMOTE to balance them. Because synthetic samples were created for the minority class, the total number of records in the dataset increased from 1369 to 1826.

Data splitting and cross-validation

Splitting data into separate training and testing sets is essential for evaluating model performance in ML³⁴. The study used an 80:20 split ratio, setting aside 20% of the data for testing the predictive models and 80% of the data for training. eightfold cross-validation is used to assess the methods.

ML classifiers

Decision tree (DT)

A reliable predictive model that is frequently used to forecast student performance is the DT classifier. From a mathematical perspective, the DT algorithm divides the dataset into subsets based on Entropy or Gini impurity and iteratively chooses features until specific stopping conditions are satisfied³⁵. The Gini impurity is shown in Eq. (4), used to calculate the importance of the particular tree(node),

$$Important(p)node = SawL(p) \times ImpL(p) - SawR(p) \times ImpR(p) \quad (4)$$

where $Important(p)node$ is the importance of p th node, $SawL(p)$ is the weighted sample of the node, $ImpL(p)$ is the impurity value, and the left and right nodes after the split is denoted by Lp and $R(p)$. The primary hyperparameters of a DT are 'max_depth', 'min_samples_leaf', and 'min_samples_split'. Each of these parameters has specific values that were tuned for the proposed model.

K-nearest neighbor (KNN)

An algorithm that does not require predefined parameters is the KNN method. It uses the dominant class of the k nearest neighbors in the feature space to forecast the class of a given data point. The most common label among a new data point's k nearest neighbors determines its class, with Euclidean distance commonly used to compute the distance³⁶, as shown in Eq. (5).

$$y = \text{mode}(y_{i1}, y_{i2}, \dots, y_{ik}) \quad (5)$$

where y is the predicted class label, y_{ik} are the labels of the k nearest neighbors of x , and $\text{mode}()$ returns the most frequent label among the neighbors and has 'n_neighbors', 'weights', and 'metric' hyperparameters. It has a specific range of values that were tuned for the proposed model.

XGBoost (XGB)

XGB, short for Extreme Gradient Boosting, is an advanced ML algorithm that constructs an ensemble of weak prediction models, typically decision trees, to predict student outcomes. Optimizing the total of each weak learner's predictions, which are updated by gradient descent iterations, it minimizes a loss function³⁷. The XGB forecast for a given data point x_i can be expressed by Eq. (6).

$$y^i = \sum_{k=1}^K k(k=1) f_k(x_i) \quad (6)$$

where the predicted result for data point x_i is y^i , the number of weak learners is k , and the prediction result of the weak learner is represented as $f_k(x_i)$. The main hyperparameters are 'n_estimators', 'max_depth', and 'min_child_weight' and are tuned to specific values.

Artificial bee colony (ABC) optimization technique

To improve model performance, ML classifiers use the ABC approach, which was inspired by the foraging action of the honey bees. By using bees to represent potential solutions, the algorithm searches the search space and evaluates their suitability according to a predetermined objective function. The algorithm iteratively updates potential solutions through communication and information sharing, progressively moving closer to the best answers. Mathematically, the ABC algorithm involves searching through possible solution vectors, calculating fitness values based on the objective function, and selecting the best solutions for further iterations, refer to Eq. (7).

$$Fitness(x_i) = \frac{1}{(1 + f(x_i))} \quad (7)$$

where (x_i) is a candidate solution (bee) in the population, $Fitness(x_i)$ is the fitness value assigned to the solution, showing its quality relative to other solutions, and $f(x_i)$ is the objective function for the solution. This Equation calculates the fitness value by inversely scaling the objective function's value.

Ant colony optimization (ACO) technique

The ACO approach simulates the cooperative behaviour of ants in search of the best solutions when used in conjunction with ML classifiers to evaluate student datasets. Within ML workflows designed to predict student performance, ACO can be utilized to optimize feature selection, model selection, or parameter tuning. The features or parameters that have the biggest impact on predicting accuracy are chosen based on this algorithm's pheromone trail updating Eq. (8). Using performance input from each iteration, the system adjusts the pheromone levels on pathways that reflect various feature subsets or model configurations iteratively.

Pheromone concentrations rise along paths that lead to increasingly accurate predictions over time, directing further iterations toward better solutions.

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \Delta\tau_{ij} \tag{8}$$

where the pheromone level on the path between nodes i and j is represented by τ_{ij} , the amount of pheromone deposited on the path by ants is determined by $\Delta\tau_{ij}$, and the pheromone evaporation rate, ρ , controls the degree to which pheromone levels decrease over time. Equation (8) depicts the iterative procedure that updates pheromone levels in response to each ant’s performance, leading to better solutions in subsequent iterations.

Performance measures of ML model

To assess the efficacy and predictive capacity of the model, the performance metrics considered, which include ROC (Receiver Operating Characteristic) curve analysis³⁸ and it is a graph depicting True Positive Rate (TPR) versus False Positive Rate (FPR) across different threshold values and *Accuracy*, refer to Eq. (9)³⁹, *F₁ score*, refer to Eq. (10)⁴⁰, *Recall*, refer to Eq. (11)⁴¹, *Precision*, refer to Eq. (12)⁴², r^2 , refer to Eq. (13)⁴³.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{9}$$

$$F_1\text{ score} = \frac{(2 \times (precision \times recall))}{(precision + recall)} \tag{10}$$

$$Recall = \frac{(TP)}{(TP + FN)} \tag{11}$$

$$Precision = \frac{(TP)}{(TP + FP)} \tag{12}$$

$$r^2 = 1 - \frac{(SS_{tot})}{(SS_{res})} \tag{13}$$

where SS_{res} is the sum of squared residuals and SS_{tot} is the total sum of squares.

Computational resources and performance

In our experiments, we utilized Google Colab, which provides cloud-based CPU/GPU resources, enabling efficient execution of machine learning models and optimization techniques. However, the computational burden varied significantly depending on the dataset size and model complexity. Integrating SMOTE and optimization techniques such as ACO and ABC further impacted the performance, especially regarding processing time, memory usage, and computational complexity. Table 3 summarizes each experiment’s processing times, memory usage, and computational complexity. While smaller datasets were manageable within Google Colab’s resources, the added complexity of SMOTE and optimization techniques may introduce scalability challenges when dealing with large-scale datasets.

Model	Processing time	Memory usage	Computational complexity
DT without SMOTE	5–10 s	Moderate (2–3 GB RAM)	Low
KNN without SMOTE	5–10 s	Moderate (2–3 GB RAM)	Low
XGB without SMOTE	5–10 s	High (4–5 GB RAM)	High
DT with SMOTE	2 min	High (4–5 GB RAM)	Moderate
KNN with SMOTE	2.5 min	High (4–5 GB RAM)	Moderate
XGB with SMOTE	2.5 min	Very high (6 GB RAM)	Very high
ABC-DT without SMOTE	2.2 min	High (4–5 GB RAM)	High
ABC-KNN without SMOTE	2.5 min	High (4–5 GB RAM)	High
ABC-XGB without SMOTE	3 min	Very high (6 GB RAM)	Very high
ABC-DT with SMOTE	2.3 min	High (4–5 GB RAM)	High
ABC-KNN with SMOTE	2.4 min	High (4–5 GB RAM)	High
ABC-XGB with SMOTE	3 min	Very high (6 GB RAM)	Very high
ACO-DT without SMOTE	2.1 min	High (4–5 GB RAM)	High
ACO-KNN without SMOTE	2.2 min	High (4–5 GB RAM)	High
ACO-XGB without SMOTE	3 min	Very high (6 GB RAM)	Very high
ACO-DT with SMOTE	2.3 min	High (4–5 GB RAM)	High
ACO-KNN with SMOTE	2.4 min	High (4–5 GB RAM)	High
ACO-XGB with SMOTE	3 min	Very high (6 GB RAM)	Very high

Table 3. Processing time, memory usage, and computational complexity for each model.

Table 3 shows that processing times for models without optimization ranged from 5 to 10 s, while models with SMOTE or optimization techniques (ACO, ABC) required approximately 2 to 3 min. Memory usage increased when applying SMOTE and optimization algorithms, particularly for models like XGBoost, which demand more computational resources. The computational complexity, especially with hyperparameter optimization, added a significant load, particularly with the XGBoost model. The hyperparameter optimization in models like XGBoost, ABC, and ACO adds significant computational load due to iterative processes, leading to longer processing times and higher memory demands. SMOTE further increases memory usage by generating synthetic data. While the models work well on medium-sized datasets in Google Colab, scalability may be a concern with large datasets, as processing time and memory usage will likely increase, requiring more powerful resources or parallelization for efficient performance.

Parameter settings

The parameters for each algorithm were carefully tuned to optimize their performance⁴⁴. Parameters such as maximum depth and splitting criteria were considered for the Decision Tree. KNN was configured with a specific number of neighbors, while XGBoost utilized a learning rate and maximum depth based on prior experimentation. SMOTE was implemented with a predefined sampling strategy to address class imbalance. Optimization techniques, including ACO and ABC, were fine-tuned with specific values for the number of iterations, ants, and bees. A summarized table (Table 4) is provided below, detailing the baseline parameter settings for all models and techniques. These values were selected based on experimental trials to ensure consistent and reliable experiment comparisons.

Results and discussion

This study conducted eight experiments utilizing feature correlation analysis, feature relevance, SMOTE, ML classifiers, and hyperparameter tuning algorithms for student datasets. Three ML classifiers, DT, KNN, and XGB, were employed, and ABC and ACO techniques were also used as the hyperparameter tuning algorithms. All the experiments were conducted in Python, utilizing its libraries.

Experiment I: Feature correlation analysis

The degree of association between pairs of features was determined by computing correlation coefficients using the Kendall-Tau method. The analysis helps identify features that exhibit no connection, are negatively linked (move in opposite directions), or are positively associated (rise or decrease together). Table 5 presents the results of the analysis of each feature using the Kendall Tau method.

The observations regarding positive, negative, and no correlation among features are discussed in the following sub-sections.

Positive correlated features

The features with correlation coefficients greater than 0 in the Kendall Tau correlation matrix are categorized as positively correlated features and are depicted in Table 6. It includes the features that exhibit positive correlations with others.

The findings from the Kendall Tau correlation matrix, as shown in Table 6, reveal several key relationships between features,

- GEN and TOT_INC have a positive correlation of 0.11, suggesting that students from wealthier families may benefit from better access to educational resources, which can enhance their academic performance.
- The 0.197 correlation between AGE and PT_JOB indicates that older students, who are more likely to work part-time, may develop time management and financial independence skills contributing to academic success.
- A correlation of 0.16 between ADM_TYPE and CLG_CURRACT suggests that the type of admission process a student undergoes may influence their choice of curriculum, which can impact academic outcomes.

Model	Parameter	Value
DT	Criterion	Gini index
	Max depth	10
	Min samples split	2
KNN	Number of neighbors (k)	5
	Distance metric	Euclidean
XGB	Learning rate	0.1
	Max depth	6
SMOTE	Sampling strategy	0.5
ABC	Colony size	100
	Maximum cycles	500
ACO	Number of ants	50
	Number of iterations	100

Table 4. Parameter settings.

Features	GEN	AGE	PT_JOB	SIN_PAR	FAT_ACDQUA	MOT_ACDQUA	FH_OCCUP	TOT_INC	ADM_TYPE
GEN	1	−0.005	0.001	−0.006	0.081	−0.028	0.039	0.11	−0.079
AGE	−0.005	1	0.197	−0.073	0.037	0.016	0.056	0.009	0.113
PT_JOB	0.001	0.197	1	0.037	0.08	0.104	−0.1	0.068	0.16
SIN_PAR	−0.006	−0.073	0.037	1	0.117	0.125	−0.056	0.06	0.161
FAT_ACDQUA	0.081	0.037	0.08	0.117	1	0.455	−0.189	0.221	−0.043
MOT_ACDQUA	−0.028	0.016	0.104	0.125	0.455	1	−0.133	0.217	−0.016
FH_OCCUP	0.039	0.056	−0.1	−0.056	−0.189	−0.133	1	−0.042	−0.079
TOT_INC	0.11	0.009	0.068	0.06	0.221	0.217	−0.042	1	−0.083
ADM_TYPE	−0.079	0.113	0.16	0.161	−0.043	−0.016	−0.079	−0.083	1
CLG_CURRACT	−0.021	0.034	0.048	0.031	0.026	0.097	−0.045	0.157	−0.103
N_CHILD	0.049	0.104	0.026	−0.027	0.017	−0.011	−0.012	0.048	0.03
GEN_INQ	−0.111	0.125	0.156	0.11	0.038	0.063	−0.059	−0.019	0.206
TRANS_DIFF	−0.103	−0.043	0.205	0.134	−0.055	−0.05	−0.008	−0.043	0.132
H_SOCSITE	−0.034	0.03	0.022	0.009	0.096	0.097	−0.042	0	−0.055
H_VIDGAM	0.092	−0.021	0.02	0.062	0.147	0.108	0.018	0.059	−0.026
H_CANCAF	0.028	−0.02	−0.128	−0.023	0.109	0.066	0.042	−0.061	−0.106
H_COCHCL	−0.048	−0.12	−0.037	0.053	0.047	0.082	−0.065	0.107	0.012
H_SELFST	−0.065	0.037	−0.044	−0.105	0.016	0.031	0.084	−0.014	−0.059
R_CONF	−0.065	0.021	−0.022	−0.006	0.055	0.117	−0.055	−0.029	−0.029
R_EMOT	0.05	0.077	0.1	0.118	0.099	0.107	−0.067	0.005	0.118
ATTEN	−0.062	0.003	0.126	−0.037	0.071	0.134	−0.146	0.033	0.007
TT	−0.042	0.032	0.112	−0.009	0.037	0.044	−0.172	−0.013	0.154

Features	CLG_CURRACT	N_CHILD	GEN_INQ	TRANS_DIFF	H_SOCSITE	H_VIDGAM	H_CANCAF	H_COCHCL
GEN	−0.021	0.049	−0.111	−0.103	−0.034	0.092	0.028	−0.048
AGE	0.034	0.104	0.125	−0.043	0.03	−0.021	−0.02	−0.12
PT_JOB	0.048	0.026	0.156	0.205	0.022	0.02	−0.128	−0.037
SIN_PAR	0.031	−0.027	0.11	0.134	0.009	0.062	−0.023	0.053
FAT_ACDQUA	0.026	0.017	0.038	−0.055	0.096	0.147	0.109	0.047
MOT_ACDQUA	0.097	−0.011	0.063	−0.05	0.097	0.108	0.066	0.082
FH_OCCUP	−0.045	−0.012	−0.059	−0.008	−0.042	0.018	0.042	−0.065
TOT_INC	0.157	0.048	−0.019	−0.043	0	0.059	−0.061	0.107
ADM_TYPE	−0.103	0.03	0.206	0.132	−0.055	−0.026	−0.106	0.012
CLG_CURRACT	1	−0.016	0.002	−0.007	0.05	0.063	0.053	0.071
N_CHILD	−0.016	1	−0.069	−0.028	0.021	−0.003	−0.12	−0.003
GEN_INQ	0.002	−0.069	1	0.284	−0.06	−0.039	−0.099	0.013
TRANS_DIFF	−0.007	−0.028	0.284	1	−0.11	−0.014	−0.088	0.121
H_SOCSITE	0.05	0.021	−0.06	−0.11	1	0.068	0.107	−0.106
H_VIDGAM	0.063	−0.003	−0.039	−0.014	0.068	1	0.284	0.227
H_CANCAF	0.053	−0.12	−0.099	−0.088	0.107	0.284	1	0.186
H_COCHCL	0.071	−0.003	0.013	0.121	−0.106	0.227	0.186	1
H_SELFST	−0.063	0.053	−0.004	−0.035	0.074	−0.25	−0.037	−0.061
R_CONF	0.076	−0.023	0.006	−0.018	0.036	−0.067	0.04	0.191
R_EMOT	0.001	0.052	−0.023	0.01	0.062	−0.023	−0.069	−0.014
ATTEN	0.16	−0.046	0.041	0.09	−0.02	−0.04	−0.026	0.01
TT	0.125	−0.075	0.236	0.091	−0.06	0.069	0.083	0.09

Features	H_SELFST	R_CONF	R_EMOT	ATTEN	TT
GEN	−0.065	−0.065	0.05	−0.062	−0.042
AGE	0.037	0.021	0.077	0.003	0.032
PT_JOB	−0.044	−0.022	0.1	0.126	0.112
SIN_PAR	−0.105	−0.006	0.118	−0.037	−0.009
FAT_ACDQUA	0.016	0.055	0.099	0.071	0.037
MOT_ACDQUA	0.031	0.117	0.107	0.134	0.044
FH_OCCUP	0.084	−0.055	−0.067	−0.146	−0.172
TOT_INC	−0.014	−0.029	0.005	0.033	−0.013
ADM_TYPE	−0.059	−0.029	0.117	0.007	0.154
CLG_CURRACT	−0.063	0.076	0.001	0.16	0.125
Continued					

Features	H_SELFST	R_CONF	R_EMOT	ATTEN	TT
N_CHILD	0.053	−0.023	0.052	−0.046	−0.075
GEN_INQ	−0.004	0.006	−0.023	0.041	0.236
TRANS_DIFF	−0.035	−0.018	0.01	0.09	0.091
H_SOCSITE	0.074	0.036	0.062	−0.02	−0.06
H_VIDGAM	−0.25	−0.067	−0.023	−0.04	0.069
H_CANCAF	−0.037	0.04	−0.069	−0.026	0.083
H_COCHCL	−0.061	0.191	−0.014	0.01	0.09
H_SELFST	1	0.242	0.082	0.047	0.087
R_CONF	0.242	1	0.137	0.115	0.162
R_EMOT	0.082	0.137	1	0.216	0.042
ATTEN	0.047	0.115	0.216	1	0.28
TT	0.087	0.162	0.042	0.28	1

Table 5. Features correlation analysis.

Feature 1	Feature 2	Value	Feature 1	Feature 2	Value
GEN	TOT_INC	0.11	ADM_TYPE	CLG_CURRACT	0.16
AGE	PT_JOB	0.197	ADM_TYPE	GEN_INQ	0.205
PT_JOB	SIN_PAR	0.037	CLG_CURRACT	GEN_INQ	0.155
SIN_PAR	FAT_ACDQUA	0.116	GEN_INQ	TRANS_DIFF	0.283
FAT_ACDQUA	MOT_ACDQUA	0.455	TRANS_DIFF	ATTEN	0.089
MOT_ACDQUA	TOT_INC	0.216	ATTEN	TT	0.28
TOT_INC	CLG_CURRACT	0.157			

Table 6. Positive correlated features.

Feature 1	Feature 2	Value
GEN	AGE	−0.004
PT_JOB	SIN_PAR	−0.100
FAT_ACDQUA	FH_OCCUP	−0.189
FH_OCCUP	ADM_TYPE	−0.078
ADM_TYPE	TOT_INC	−0.083

Table 7. Negative correlated features.

- The 0.205 correlation between ADM_TYPE and GEN_INQ implies that certain admission types may be associated with gender-related challenges in the academic environment, affecting students’ academic engagement and performance.
- The strong correlation of 0.455 between FAT_ACDQUA and MOT_ACDQUA underscores the significant role of parental education in shaping student outcomes. Educated parents are likely to provide higher expectations and greater intellectual support, positively influencing their children’s academic success.
- The 0.216 correlation between TOT_INC and MOT_ACDQUA suggests that financial stability and higher parental education foster a supportive home environment that can enhance academic performance.
- A correlation of 0.28 between ATTEN and TT indicates that regular attendance and increased study time are positively related to academic success. Additionally, TRANS_DIFF correlates with ATTEN, suggesting that students facing transportation challenges may be more motivated to attend classes, driven by their determination to overcome these barriers.
- Lastly, the 0.283 correlation between GEN_INQ and TRANS_DIFF implies that students facing gender inequality in the academic environment may also experience transport-related challenges, reflecting broader social and logistical issues that can impact their academic journey.

Negative correlated features

Negatively correlated features in the Kendall Tau correlation matrix are those with correlation coefficients less than 0, as depicted in Table 7. These features exhibit negative correlations with others, suggesting that an increase in one feature is associated with a decrease in the other.

The negative correlations observed are,

- GEN and AGE have a negligible negative correlation of -0.004 , suggesting that there is almost no relationship between gender and age in this context.
- PT_JOB and SIN_PAR show a correlation of -0.100 , indicating that students with part-time jobs may be slightly less likely to come from single-parent households or vice versa.
- FAT_ACDQUA and FH_OCCUP have a negative correlation of -0.189 , suggesting that students whose fathers have higher academic qualifications may have fathers in higher-status or more specialized occupations, as opposed to lower-status jobs.
- FH_OCCUP and ADM_TYPE show a correlation of -0.078 , implying a minimal negative relationship between the father's occupation and the admission type, indicating that the two factors may not strongly influence each other.
- ADM_TYPE and TOT_INC correlate -0.083 , suggesting a slight negative relationship between the type of admission process and total family income, indicating that the admission process might have minimal influence on the family's income level.

No correlated features

It is observed that no features have a correlation coefficient of exactly zero, indicating that every feature is at least weakly correlated with others in the dataset. While some correlations may be very weak or close to zero, the lack of perfectly zero correlations suggests that all features contribute, to some extent, to the academic performance prediction model. Even weak correlations can provide valuable insights into subtle relationships that may not be immediately apparent. These weak correlations could become significant when combined with other features or in different modeling contexts. Therefore, all features should be considered for further analysis and inclusion in the predictive model.

Experiment II: Feature relevance and interpretation

This experiment focused on analyzing the relevance of individual features to understand their impact on students' academic performance. The analysis ranked features based on their relevancy scores, determined through the chi-square test, providing insights into their influence on the prediction outcomes. Figure 6 illustrates the top 21 features ranked by their relevancy scores.

As depicted in the figure, Gender (418) plays a significant role in academic performance, reflecting differences in academic engagement, learning preferences, and social dynamics. Age (402) may be related to maturity and learning pace, with older students possibly demonstrating greater self-discipline and focus. A part-time job (400) can affect the availability of study time, as students must balance work and academic responsibilities. Being a single-parent child (392) may influence academic performance due to varying levels of parental support and household dynamics. Self-confidence (389) directly impacts a student's ability to tackle challenges and persevere academically, while emotional stability (360) is critical for maintaining focus and resilience under academic pressure. Admission type (331) may affect the level of preparedness and alignment with academic expectations, and transportation issues (330) can affect attendance and energy levels, limiting a student's ability to perform consistently. Attendance (291) ensures regular exposure to instruction, which positively impacts learning outcomes.

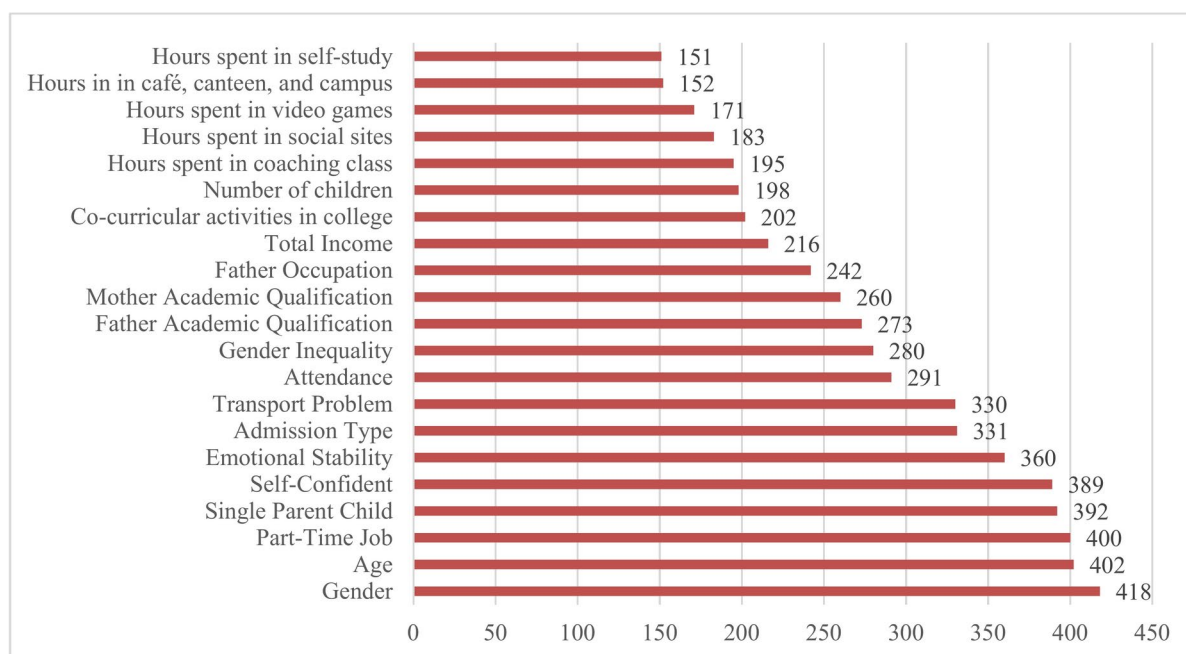


Fig. 6. Features relevance score.

Metrics	DT (%)	KNN (%)	XGB (%)
F_1 score	95.25	85.41	95.06
Accuracy	95.35	85	95
Precision	95.35	85.83	95.12
Recall	95.15	85	95
ROC_AUC	95	92	94
r^2	84	46.22	83.2

Table 8. Results of the base classifiers without SMOTE.

Metrics	DT (%)	KNN (%)	XGB (%)
F_1 score	98	86.15	96.06
Accuracy	96.53	86.18	96.11
Precision	96.52	86.21	96.12
Recall	96.50	86.10	96
ROC_AUC	96	93	95
r^2	84.4	47.32	84.66

Table 9. Results of the base classifiers with SMOTE.

Metrics	DT (%)	KNN (%)	XGB (%)
F_1 score	97.25	87.30	96.25
Accuracy	97.15	87.8	96.33
Precision	97.35	87.51	96.41
Recall	97.15	87.10	96.1
ROC_AUC	96	93	95
r^2	84.73	48.62	85.36

Table 10. Results of optimized base classifiers using ABC without SMOTE.

Perceived or real gender inequality (280) could influence confidence levels and opportunities for academic engagement. A father's academic qualification (273) and a mother's academic qualification (260) both shape the learning environment and encourage home. A father's occupation (242) may influence economic stability and access to educational resources, while household income (216) determines access to extracurricular resources, technology, and private coaching. Co-curricular activities (202) enhance overall learning but may distract from academics if not managed well. The number of children in a family (198) can impact resource allocation and parental attention. Time spent in coaching classes (195) directly correlates with academic support and performance, whereas excessive time on social media (183), gaming (171), and in cafés (152) may detract from productive study hours. Finally, self-study hours (151) are crucial for understanding and reinforcing concepts, directly impacting academic performance, as dedicated self-study allows students to master material at their own pace and gain deeper understanding outside of class hours.

Experiment III: ML baseline models without SMOTE

The results of the ML baseline models are presented in Table 8 when applied to the student dataset without SMOTE. Based on the performance metrics, the DT exhibited the best performance among the three classifiers, achieving an *Accuracy* of 95.35%, an F_1 score of 95.25%, *Recall* and *Precision* of 95.15% and 95.35%, ROC_AUC of 95%, and r^2 of 84%.

Experiment IV: ML baseline models with SMOTE

After Experiment II, the ML baseline models were analyzed with SMOTE. Based on the performance metrics in Table 9, once again, the DT exhibited the best performance among the three classifiers, achieving an *Accuracy* of 96.53%, an F_1 score of 98%, *Recall* and *Precision* of 96.50% and 96.52%, ROC_AUC of 96%, and r^2 of 84.4%.

Experiment V: ABC hyperparameter optimization of ML classifiers without SMOTE

In this experiment, all three base classifiers were optimized using the ABC technique but without SMOTE, refer to Table 10. The results showed that the optimized DT achieved the best *Accuracy* of 97.15%, an F_1 score of 97.25%, *Recall* and *Precision* of 97.15% and 97.35%, ROC_AUC of 96%, and *Precision* of 84.73%.

Metrics	DT (%)	KNN (%)	XGB (%)
F_1 score	97.51	88.29	96.39
Accuracy	97.56	88	96.49
Precision	97.46	88.37	96.48
Recall	97.56	88.21	96.3
ROC_AUC	97	94	96
r^2	84.84	48.66	86

Table 11. Results of optimized base classifiers using ABC with SMOTE.

Performance metrics	DT (%)	KNN (%)	XGB (%)
F_1 score	97.69	88.91	96.50
Accuracy	97.91	89	96.8
Precision	97.70	88.83	97
Recall	97.69	89	96
ROC_AUC	97	94	96
r^2	84.86	50.22	87

Table 12. Results of optimized base classifiers using ACO without SMOTE.

Metrics	DT (%)	KNN (%)	XGB (%)
F_1 score	98.10	89.01	97
Accuracy	98.15	89.05	97.2
Precision	98.09	88.98	97
Recall	98.10	89.05	97
ROC_AUC	98	95	97
r^2	85.75	52.34	88

Table 13. Results of optimized base classifiers using ACO with SMOTE.

Experiment VI: ABC hyperparameter optimization of ML classifiers with SMOTE

All three base classifiers were optimized using the ABC technique but with SMOTE in this experiment. From Table 11, the results showed that once again, the optimized DT achieved the best *Accuracy* of 97.56%, an F_1 score of 97.51%, *Recall* and *Precision* of 97.56% and 97.46%, ROC_AUC of 97%, and r^2 of 84.84%.

Experiment VII: ACO hyperparameter optimization of ML classifiers without SMOTE

For the second to last experiment, all three base classifiers were optimized using the ACO technique without SMOTE, refer to Table 12. The results showed that the optimized DT using ACO achieved the best performance, with an *Accuracy* of 97.91%, an F_1 score of 97.69%, *Recall* and *Precision* of 97.69% and 97.70%, ROC_AUC of 97%, and r^2 of 84.86%.

Experiment VIII: ACO hyperparameter optimization of ML classifiers with SMOTE

In the final experiment, all three base classifiers were optimized using the ACO technique with SMOTE. The results in Table 13 showed that the optimized DT using ACO with SMOTE achieved the best performance, with an *Accuracy* of 98.15%, an F_1 score of 98.10%, *Recall* and *Precision* both at 98.10% and 98.09%, respectively, a ROC_AUC of 98%, and an r^2 of 85.75%.

Optimized parameters of decision tree without and with SMOTE

Table 14 shows the optimized parameters of the DT without and with SMOTE, using both optimization techniques: ABC and ACO. The maximum depth of the tree, without SMOTE for ABC-DT, is 19; hence, it can rise to 19 levels. The model can identify more intricate patterns in the data. One sample is the bare minimum needed at a leaf node. The tree may include leaf nodes with only one sample, which, if improperly managed, may result in overfitting. Three samples are the minimum needed to separate an internal node, which helps avoid overfitting caused by splitting nodes with relatively few samples.

Optimized parameters of K-nearest neighbor without and with SMOTE

Table 15 shows the optimized parameters of the k-nearest neighbor without and with SMOTE, using both optimization techniques: ABC and ACO.

Techniques	Max_depth	Min_samples_leaf	Min_samples_split
Without SMOTE			
ABC-DT	19	1	3
ACO-DT	11	4	2
With SMOTE			
ABC-DT	17	2	6
ACO-DT	18	1	6

Table 14. Optimized parameters of DT.

Techniques	N_neighbors	Weights	Matric
Without SMOTE			
ABC-KNN	3	Uniform	Manhattan
ACO-KNN	4	Uniform	Euclidean
With SMOTE			
ABC-KNN	4	Distance_based	Euclidean
ACO-KNN	5	Uniform	Euclidean

Table 15. Optimized parameters of KNN.

Techniques	n_estimators	max_depth	min_child_weight
Without SMOTE			
ABC-XGB	3	100	4
ACO-XGB	4	100	2
With SMOTE			
ABC-XGB	5	100	1
ACO-XGB	6	100	1

Table 16. Optimized parameters of XGB.

Optimized hyperparameters of XGBoost without and with SMOTE

Table 16 shows the optimized parameters of the XGBoost classifier without and with SMOTE, using both optimization techniques: ABC and ACO.

The results effectively meet the study’s aim, which is to improve the prediction of IT students’ academic performance by ML techniques. The optimized settings and comparative performance measures highlight several vital conclusions. The study effectively illustrates how ML models’ predictive power is increased when SMOTE is used with optimization strategies like ABC and ACO. After concluding the experiment, the study determined that SMOTE and ACO can enhance students’ academic performance prediction. Among all the experiment results, the ACO-DT with SMOTE showed the best performance across all metrics compared to other models. The DT consistently performed best among the base models, followed by XGB and KNN. The optimal parameters for the ACO-DT with SMOTE were a max_depth of 18, a min_samples_leaf of 1, and a min_samples_split of 6, contributing to its robust performance in predicting student academic performance. This is consistent with the research goal of assessing and contrasting ML classifiers to predict academic achievement. The best settings for every model, with and without SMOTE, offer insightful information about the advantages and disadvantages of each model in managing the complexity of student performance data. The study emphasizes how important it is to adjust hyperparameters using ABC and ACO to enhance model performance. The ACO-DT model with SMOTE showed stability in measures including *Precision*, *Recall* and *F₁score*, and achieving greater accuracy. This demonstrates how well optimization strategies work to improve ML models’ ability to anticipate and manage imbalances in student performance data accurately. The results highlight how ML may help educational stakeholders pinpoint and address the variables affecting students’ achievement, eventually leading to better educational outcomes and interventions.

Implications of the study

The research’s strategies and outcomes significantly impact many educational practices and policy areas, particularly IT departments. The implications extend to future research paths, student support systems, administrative decision-making, and instructional methodologies.

- *Instructional Strategies:* Teachers may create individualized lesson plans that meet each student’s strengths and weaknesses and precisely forecast student achievement. This makes implementing tailored treatments that

raise student success and engagement possible. Early identification of pupils who are at risk of underperforming allows teachers to give remedial instruction and tailored support, enabling students to overcome obstacles in the classroom and achieve better results.

- *Administrative Decision-Making:* Predictive analytics is a tool administrators may use to allocate resources better, focusing on academic and support services where they are most needed. This guarantees the cost-effectiveness and efficacy of treatments. Understanding the elements that influence students' performance may help construct curricula, ensuring that classes are made to improve learning and tackle frequent problems.
- *Student Support Systems:* The study's prediction algorithms may be included in early warning systems to enable proactive interventions before problems worsen and real-time student performance monitoring. Predictive insights may be used by academic advisers and counselors to better assist students in making decisions regarding their study methods, course choices, and career pathways.
- *Educational Equity:* All student groups, especially minorities and those historically underrepresented in IT areas, may expect accurate projections thanks to the fair and inclusive predictions made using SMOTE when handling unbalanced data. Institutions may lower dropout rates and increase completion and retention rates among IT students by identifying at-risk individuals early on.
- *Technological Integration:* The potential for incorporating advanced analytics and optimization techniques into educational data systems is demonstrated by the successful use of ACO and ABC optimization approaches, opening the door to more complex data-driven educational decision-making. The benefits of predictive analytics may be extended outside of IT departments by adapting and scaling the current methodology across other educational contexts and disciplines.
- *Policy Implications:* Policymakers may use the results of this study to develop evidence-based policies that improve overall educational quality, optimize resource allocation, and strengthen student support services. Establishing standardized frameworks for predictive analytics in education may ensure consistency, dependability, and ethical concerns in using these technologies.

In addition to advancing educational data analytics, our work offers valuable tools and insights that have the potential to significantly enhance educational fairness, institutional effectiveness, and student outcomes. Implementing these approaches and insights may result in a more data-driven and student-centered approach to education, eventually promoting academic achievement and lifetime learning.

Conclusions, limitations, ethical and privacy considerations, and future work

Conclusions

This study illuminates the pivotal role of ML in forecasting the academic trajectories of IT students, presenting valuable insights for educational stakeholders and administrators. This study compares the effectiveness of ML classifiers for improved student academic prediction, utilizing hyperparameter tuning techniques and SMOTE to manage imbalanced datasets. Expanding upon the evaluation of three fundamental classifiers, DT, KNN, and XGB, it becomes evident that DT stands out prominently among the trio. Building on this revelation, we introduced two optimization techniques, ABC and ACO, aimed at augmenting the performance of the DT classifier. The ensuing results unequivocally illustrate the efficacy of both methods in bolstering classifier performance. Particularly striking is the exceptional performance exhibited by the ACO-DT model, boasting an *Accuracy* of 98.1%, an *F₁ score* of 96%, a *Precision* of 96.24%, a *Recall* of 96.19%, an ROC curve of 96%, and an *r²* of 84.75%. The finely tuned DT, complete with hyperparameters in the outcomes, demonstrates how crucial fine-tuning is to model optimization.

Limitations of the study

The study offers insightful information, but the dataset may be unique to a particular institution or demography, restricting the generalizability of the findings to other contexts or demographics. The study's characteristics could not have included all pertinent variables that affect academic achievement. The study did not investigate more sophisticated ML techniques, such as ensemble approaches or deep learning, that could provide higher predicted performance. Academic performance is dynamic and subject to change, so longitudinal studies are necessary to provide more comprehensive insights. Predictive modeling using student data presents ethical and privacy issues. It is essential to ensure student privacy and data are handled safely. The models run the danger of incorporating or maintaining biases, which might result in unfair or discriminating behaviors. Ensuring equity and justice in predictive analytics is a significant challenge.

Ethical and privacy considerations

Predictive modeling in educational contexts, mainly when dealing with sensitive student data, necessitates careful attention to ethical and privacy concerns. This study adheres to strict ethical standards to ensure students' privacy is upheld and their data is handled securely. Below are the key considerations taken into account,

- *Data Privacy:* The data used in this study was anonymized to prevent the identification of individual students. All data collection and usage complied with applicable data protection regulations, ensuring student information was protected throughout the research process. Secure storage methods were employed to safeguard the integrity and confidentiality of the data.
- *Informed Consent:* All participants were provided with clear information about the study's purpose and how their data would be used. Informed consent was obtained before data collection, ensuring that participants were fully aware of their rights and the scope of the research.
- *Mitigating Bias:* Predictive models in machine learning can unintentionally perpetuate biases, especially when trained on historical data that may reflect existing inequalities. To address this, the study carefully selected

features and applied techniques such as SMOTE to handle class imbalances and reduce the risk of biased predictions. Regular audits of model performance were conducted to ensure the results were fair and equitable.

- **Transparency and Interpretability:** The models used in this study were committed to enhancing their interpretability. Ensuring transparency in how models make predictions is crucial for fostering trust among stakeholders, including educators, administrators, and students. Future work should continue to explore ways to make machine learning models more interpretable and explainable.
- **Fairness in Predictive Analytics:** It is essential to ensure that predictive models do not reinforce existing educational inequalities. Careful attention was paid to the potential for unfair discrimination based on factors such as gender, socioeconomic status, or other demographic variables. The goal was to create models that are as equitable and inclusive as possible to benefit all students equally.

Addressing these ethical considerations will remain a priority in future studies, ensuring that machine learning applications in education are implemented responsibly and with respect for all individuals involved.

Future directions

The research on using ML to predict the academic achievement of IT students should concentrate on enlarging feature sets to include a wider variety of characteristics, such as socioeconomic determinants, mental health indicators, and engagement measures. Dynamic data-collecting techniques should record real-time changes in student behavior and performance to improve forecast accuracy and timeliness. Investigating cutting-edge ML strategies like ensemble methods and deep learning models may provide fresh perspectives on intricate patterns in educational data, enhancing prediction accuracy and resilience across student demographics and institutional settings. The ability to monitor and evaluate student performance over lengthy periods is made possible by longitudinal studies, which also provide insights into the factors that affect academic achievement over time and the efficacy of treatments. Building confidence and optimizing the valuable influence of predictive analytics on student outcomes and institutional decision-making will also require resolving ethical issues, improving model interpretability, and guaranteeing scalable deployment in educational contexts.

Data availability

The necessary data used in the manuscript are already present in the manuscript.

Code availability

The algorithm is freely available and can be downloaded from public GitHub repository: <https://github.com/KAJAL-CODER-BOT/StudentPerformancePrediction.git>.

Received: 22 July 2024; Accepted: 17 March 2025

Published online: 24 March 2025

References

1. Alhazmi, E. & Sheneamer, A. Early predicting of students performance in higher education. *IEEE Access* **11**, 27579–27589. <https://doi.org/10.1109/ACCESS.2023.3250702> (2023).
2. Albahli, S. Efficient hyperparameter tuning for predicting student performance with Bayesian optimization. *Multimed. Tools Appl.* **83**(17), 52711–52735. <https://doi.org/10.1007/s11042-023-17525-w> (2024).
3. Mochammad, A., Chowdhury Mohammad Masum, R., Kazuyoshi, H. & Keisuke, M. Movement optimization for a cyborg cockroach in a bounded space incorporating machine learning. *Cyborg Bionic Syst.* **4**, 0012. <https://doi.org/10.34133/cbsystems.0012> (2023).
4. Zhou, L., Sun, Q., Ding, S., Han, S. & Wang, A. A machine-learning-based method for ship propulsion power prediction in ice. *J. Mar. Sci. Eng.* **11**(7), 1381. <https://doi.org/10.3390/jmse11071381> (2023).
5. Gan, Y., Huang, H., Wu, X. & Meng, M. What doesn't kill us makes us stronger: insights from neuroscience studies and molecular genetics. *Curr. Opin. Behav. Sci.* **59**, 101431. <https://doi.org/10.1016/j.cobeha.2024.101431> (2024).
6. Hao, S. et al. Group membership modulates the hold-up problem: An event-related potentials and oscillations study. *Soc. Cogn. Aff. Neurosci.* <https://doi.org/10.1093/scan/nsad071> (2023).
7. Bingcheng, W. et al. A neural coordination strategy for attachment and detachment of a climbing robot inspired by gecko locomotion. *Cyborg Bionic Syst.* **4**, 0008. <https://doi.org/10.34133/cbsystems.0008> (2023).
8. Hao, S. et al. Group identity modulates bidding behavior in repeated lottery contest: Neural signatures from event-related potentials and electroencephalography oscillations. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2023.1184601> (2023).
9. Du, Y. et al. Neurometabolite levels in the brains of patients with autism spectrum disorders: A meta-analysis of proton magnetic resonance spectroscopy studies (N = 1501). *Mol. Psychiatry* **28**(7), 3092–3103. <https://doi.org/10.1038/s41380-023-02079-y> (2023).
10. Parhizkar, A., Tejeddin, G. & Khatibi, T. Student performance prediction using datamining classification algorithms: Evaluating generalizability of models from geographical aspect. *Educ. Inf. Technol.* **28**(11), 14167–14185. <https://doi.org/10.1007/s10639-022-11560-0> (2023).
11. Chen, S. & Y. Ding, A machine learning approach to predicting academic performance in Pennsylvania's schools. *Soc. Sci.* <https://doi.org/10.3390/socsci12030118> (2023).
12. Ouyang, F., Wu, M., Zheng, L., Zhang, L. & Jiao, P. Integration of artificial intelligence performance prediction and learning analytics to improve student learning in online engineering course. *Int. J. Educ. Technol. Higher Educ.* **20**(1), 4. <https://doi.org/10.1186/s41239-022-00372-4> (2023).
13. Wang, L. Metaheuristic optimization based machine-learning approach for analysis of academic students' strategies. *J. Intell. Fuzzy Syst.* **6**, 66. <https://doi.org/10.3233/JIFS-234021> (2024).
14. Xu, X., Lin, Z., Li, X., Shang, C. & Shen, Q. Multi-objective robust optimisation model for MDVRPLS in refined oil distribution. *Int. J. Prod. Res.* **60**(22), 6772–6792 (2022).
15. Ghorbani, R. & Ghousi, R. Comparing different resampling methods in predicting students' performance using machine learning techniques. *IEEE Access* **8**, 67899–67911. <https://doi.org/10.1109/ACCESS.2020.2986809> (2020).
16. El-kenawy, E.-S.M. et al. Greylag goose optimization: Nature-inspired optimization algorithm. *Expert Syst. Appl.* **238**, 122147. <https://doi.org/10.1016/j.eswa.2023.122147> (2024).
17. Zhang, Y. Application of data mining based on improved ant colony algorithm in college students' employment and entrepreneurship education. *Soft Comput.* <https://doi.org/10.1007/s00500-023-08701-z> (2023).

18. Sinha, A., Panchamia, J., Sachan, A. & Guru, S. Choice of electives among management students in India: A conjoint analysis. *Int. J. Manag. Educ.* **19**(3), 100557. <https://doi.org/10.1016/j.ijme.2021.100557> (2021).
19. Najieha, D., Alkawaz, M. H. & Johar, M. G. M. Students' performances analyzer using c4.5 algorithm and digital signature in validating report. *Int. J. Med. Toxicol. Legal Med.* **21**(3–4), 49–51. <https://doi.org/10.5958/0974-4614.2018.00027.X> (2018).
20. Gunasinghe, A., Hamid, J. A., Khatibi, A. & Azam, S. M. F. The adequacy of UTAUT-3 in interpreting academicians' adoption to e-Learning in higher education environments. *Interact. Technol. Smart Educ.* **17**(1), 86–106. <https://doi.org/10.1108/ITSE-05-2019-0020> (2020).
21. Fang, F. A study on the application of data mining techniques in the management of sustainable education for employment. *Data Sci. J.* <https://doi.org/10.5334/dsj-2023-023> (2023).
22. Cohausz, L., Tschalzev, A., Bartelt, C. & Stuckenschmidt, H. Investigating the importance of demographic features for EDM-predictions. *Educ. Data Min.* **6**, 66 (2023).
23. Verger, M., Lallé, S., Bouchet, F. & Luengo, V. Is Your Model "MADD"? A Novel Metric to Evaluate Algorithmic Fairness for Predictive Student Models (2023).
24. Bellaj, M., Bendahmane, A., Boudra, S. & Lamarti Sefian, M. Educational data mining: employing machine learning techniques and hyperparameter optimization to improve students' academic performance. *Int. J. Online Biomed. Eng.* **20**, 55–74. <https://doi.org/10.3991/ijoe.v20i03.46287> (2024).
25. Al-Alawi, L., Al Shaqsi, J., Tarhini, A. & Al-Busaidi, A. S. Using machine learning to predict factors affecting academic performance: The case of college students on academic probation. *Educ. Inf. Technol.* **28**(10), 12407–12432. <https://doi.org/10.1007/s10639-023-11700-0> (2023).
26. Nie, J. & Ahmadi Dehrashid, H. Evaluation of student failure in higher education by an innovative strategy of fuzzy system combined optimization algorithms and AI. *Heliyon* **10**(7), e29182. <https://doi.org/10.1016/j.heliyon.2024.e29182> (2024).
27. Teodorović, D. & Dell'Orco, M. Bee colony optimization—A cooperative learning approach to complex transportation problems. *Adv. OR AI Methods Transp.* **66**, 51–60 (2005).
28. Karaboga, N. & M. Çetinkaya, A novel and efficient algorithm for adaptive filtering: Artificial bee colony algorithm. *Turk. J. Electr. Eng. Comput. Sci.* **19**, 175–190. <https://doi.org/10.3906/elk-0912-344> (2011).
29. Brajevic, I., Tuba, M. & Subotic, M. Performance of the improved artificial bee colony algorithm on standard engineering constrained problems. *Int. J. Math. Comput. Simul.* **5**, 66 (2011).
30. Ye, A. Z. et al. High-dimensional feature selection based on improved binary ant colony optimization combined with hybrid rice optimization algorithm. *Int. J. Intell. Syst.* **1**, 1444938. <https://doi.org/10.1155/2023/1444938> (2023).
31. Xu, H. & Kim, M. Combination prediction method of students' performance based on ant colony algorithm. *PLoS ONE* **19**(3), e0300010. <https://doi.org/10.1371/journal.pone.0300010> (2024).
32. Helmi, R. A. A., Elghanuni, R. H. & Abdullah, M. I. Effect the graph metric to detect anomalies and non-anomalies on Facebook using machine learning models. In *2021 IEEE 12th Control and System Graduate Research Colloquium, ICSGRC 2021—Proceedings* 7–12. <https://doi.org/10.1109/ICSGRC53186.2021.9515227> (2021).
33. Dissanayake, K. & Johar, M. G. M. 2021 Comparative study on heart disease prediction using feature selection techniques on classification algorithms. *Appl. Comput. Intell. Soft Comput.* **2021**, Art no. 5581806. <https://doi.org/10.1155/2021/5581806>.
34. Ishaque, M., Johar, M. G. M., Khatibi, A. & Yamin, M. A novel hybrid technique using fuzzy logic, neural networks and genetic algorithm for intrusion detection system. *Meas. Sens.* **30**, Art no. 100933. <https://doi.org/10.1016/j.measen.2023.100933> (2023).
35. Li, Z. & Tham, J. Network accounting information security based on classification and regression tree algorithm. In *Proceedings of SPIE—The International Society for Optical Engineering* vol. 12941. <https://doi.org/10.1117/12.3011569> (2023). <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85180127175&doi=10.1117%2f12.3011569&partnerID=40&md5=5239fd945432eb58ca71b76394d306fc>
36. Mustafa, M. J., Chin, J. W., Nungsari, M. & Morris, K. J. Do proactive students benefit more from university support for entrepreneurship when it comes to choosing entrepreneurship as a career choice? An examination of Ghanaian and Malaysian students. *Int. J. Manag. Educ.* **21**(3), 100868. <https://doi.org/10.1016/j.ijme.2023.100868> (2023).
37. Dissanayake, K. & Johar, M. G. M. Two-level boosting classifiers ensemble based on feature selection for heart disease prediction. *Indones. J. Electrical Eng. Comput. Sci.* **32**(1), 381–391. <https://doi.org/10.11591/ijeecs.v32.i1.pp381-391> (2023).
38. Dehghani, H., Oghabian, M. A., Batouli, S. A. H., Kheradmand, J. A. & Khatibi, A. Research paper: Effect of physiological noise on thoracolumbar spinal cord functional magnetic resonance imaging in 3T magnetic field. *Basic Clin. Neurosci.* **11**(6), 737–752. <https://doi.org/10.32598/BCN.11.6.1395.1> (2020).
39. Pathirana, S., Asirvatham, D. & Johar, M. G. M. A multi-agent system for improving electroencephalographic data classification accuracy. In *IEEE Region 10 Humanitarian Technology Conference, R10-HTC* vol. 2018. <https://doi.org/10.1109/R10-HTC.2018.8629860> (2018).
40. Rozaini, M. A. N., Sariman, S., Abd Rashed, A. & Shari, N. The association of screen time with physical activity and weight status of autistic children in Kuala Lumpur, Malaysia. *Nutr. Food Sci.* **53**(3), 549–563. <https://doi.org/10.1108/NFS-02-2022-0038> (2023).
41. Febriani, I., Ekawati, R., Supriadi, U. & Abdullah, M. I. Fisher–Yates shuffle algorithm for randomization math exam on computer based-test. *AIP Conf. Proc.* <https://doi.org/10.1063/5.0042534> (2021).
42. Khatibi, A., Sharpe, L., Jafari, H., Gholami, S. & Dehghani, M. Interpretation biases in chronic pain patients: An incidental learning task. *Eur. J. Pain UK* **19**(8), 1139–1147. <https://doi.org/10.1002/ejp.637> (2015).
43. Lashayo, D. M., Alkawaz, M. H. & Johar, M. G. M. Model for measuring multiple factors in E-learning systems' adoption in Malaysia's universities: An experience from management and science university (MSU). *J. Theor. Appl. Inf. Technol.* **96**(18), 6180–6190 (2018).
44. Sen, B., Bhowmik, A., Prakash, C. & Ammarullah, M. I. Prediction of specific cutting energy consumption in eco-benign lubricating environment for biomedical industry applications: Exploring efficacy of GEP, ANN, and RSM models. *AIP Adv.* **14**(8), 085216. <https://doi.org/10.1063/5.0217508> (2024).

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through a Large Research Project under Grant Number RGP2/541/46.

Institutional Review Board Statement

This study was conducted in accordance with relevant guidelines and regulations. Lovely Professional University reviewed and approved the experimental protocol under approval number [Sr. No. 82; LPU/10.2/SC/2024/029/240923/0013; Code No. CRT206, dated 23 September 2024]. All procedures were carried out to ensure compliance with ethical standards.

Author contributions

Kajal Mahawar: Data curation, Investigation, Writing—original draft. Punam Rattan: Conceptualization, Supervision, Writing—review & editing. Ammar Jalameh: Formal analysis, Software, Writing—review & editing.

Mohd Shukri Ab Yajid: Software, Visualization, Writing—review & editing. Omar Abdeljaber: Methodology, Project administration, Writing—review & editing. Raman Kumar: Supervision, Validation, Writing—review & editing. Ayodele Lasisi: Funding acquisition, Resources, Writing—review & editing. Muhammad Imam Ammarullah: Project administration, Visualization, Writing—review & editing.

Funding

This research was funded by the Deanship of Research and Graduate Studies at King Khalid University through a Large Research Project under Grant Number RGP2/541/46.

Competing interests

The authors declare no competing interests.

Consent for publication

The authors consent for the publication of this manuscript.

Additional information

Correspondence and requests for materials should be addressed to M.I.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025