



Wrapper Feature Selection Method for Predicting Student Dropout in Higher Education

 Anuradha Kumari Singh^{1*},  S. Karthikeyan²

^{1,2}Department of Computer Science, Banaras Hindu University, India

✉ anuradha@bhu.ac.in*



Article Information:

Received June 05, 2025

Revised July 09, 2025

Accepted July 22, 2025

Keywords:

Ant Colony Optimization;
Artificial Intelligence in
Education; Machine Learning;
Student Dropout; Wrapper
Technique

Abstract

Background of Study: Student dropout in higher education is influenced by a variety of factors including demographic, socioeconomic, macroeconomic, admission-related, and academic performance data. Accurately identifying students at risk of dropping out is a significant challenge within educational data mining (EDM), especially when working with large, complex datasets.

Aims and Scope of Paper: This study aims to identify an optimal subset of features that can improve the accuracy of student dropout prediction. The scope includes comparing the effectiveness of different machine learning algorithms combined with a heuristic-based feature selection method to find the best-performing model.

Methods: A Wrapper-based feature selection approach was employed using Ant Colony Optimization (ACO) as the search strategy. ACO was integrated with five classifiers—Random Forest (RF), Logistic Regression (LR), K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Neural Network (NN)—to select the most relevant feature subsets. The performance of each combination was evaluated and compared.

Result: The study found that ACO combined with Random Forest (ACO-RF) outperformed the other combinations in feature selection effectiveness. The selected features were then validated using various machine learning algorithms and a neural network. Among them, the neural network achieved the highest accuracy of 93%.

Conclusion: The proposed ACO-RF wrapper method is an effective feature selection strategy for predicting student dropout in higher education. The method enhances model performance, especially when used with neural networks, and offers a promising approach for early identification of at-risk students.

A. Introduction

Educational Data Mining (EDM) is an important approach to learning analytics, obtaining extensive applications in tasks such as knowledge tracing, recommendation systems, aiding data-driven decision-making, predicting performance and early dropout of students, and many more. Predicting students' dropout in higher education institutions is one of the most challenging problems in EDM. EDM applies machine learning, statistics, data mining, educational psychology, cognitive psychology, and related methodologies to examine educational data. According to (Aleem & Gore, 2020), EDM methodology can be classified into six groups: data extraction, prediction, association mining, structure discovery, model-based recognition, and hybrid approaches. AI and machine learning technologies have the potential to enhance the learning experience of students. The machine learning approaches have used in several studies to assess the student's performance (Aulck et al., 2016) (Li et al., 2018). In this work, undergraduate students'

demographic, socioeconomic, macroeconomic, data at admission, and academic performance from the first and second semesters have been used to predict student dropouts. We have also used machine learning algorithms to address this prediction problem.

1. Student Dropout Prediction

In higher education research, Tinto's paper in (Nicoletti, 2019) introducing a new method for modeling and analyzing student dropouts. Another study (Siri, 2015) examined 810 enrollees in a health care professions degree program at the University of Genoa during the 2008-09 academic year. Data collection included administrative records of student progress, statistical data from a custom survey, and information from telephone interviews with dropout students. They utilized an artificial neural network (ANN) for classification task.

In another study addressing university dropout, presented in (Kadar et al., 2018), the focus is on predicting student dropout using a distinct student representation. This approach utilizes data collection through webcams, eye trackers, and similar devices within a smart classroom environment. By analyzing emotions and detecting patterns among students in the classroom based on this data, the study aimed to forecast potential dropouts. Another paper performs an integral systematic literature review about university dropout prediction through data mining, with studies from 2006-2018 (Alban & Mauricio, 2019). In the research conducted by (Del Bonifro et al., 2020), their tool assessed the likelihood of a student discontinuing an academic program. This tool's application is flexible, and suitable for use during the application process or within the first-year course credits. The study explored classification methods including Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and Random Forest (RF).

(Wan Yaacob et al., 2020) explored data mining techniques to predict dropout rates of Computer Science undergraduate students at University Technology MARA after three years of enrollment. The study compared various classifiers, including LR, RF, KNN, and Neural network, to determine the most effective model. Their analysis revealed that the LR model outperformed others in predicting dropout among students and identifying potential subject-related causes. There is another paper (Gardner & Brooks, 2018) in which the authors tried to find the best model for predicting student dropout rates in MOOCs using the Friedman and Nemenyi Two-Stage Produce. There is a lot of research being conducted to determine the optimal model and factors for predicting student dropout rates, either in MOOCs or in higher educational institutions.

To build a dropout prediction model, we faced the challenge of identifying the proper features that influence early student dropout from the course. To overcome this challenge we explored different feature selection methods. Feature selection (FS) is a process of selecting relevance subset of features from the original set. According to (Villa-Blanco et al., 2023), classified feature subsets into four kinds depending on their relevance and redundancy: 1) Noisy and irrelevant; 2) Redundant and weakly relevant; 3) Weakly relevant and non-redundant; and 4) Strongly relevant. Features that do not contribute to predicting accuracy are referred to as irrelevant features. FS methods are classified into three types based on their interaction with the learning model: Filter, Wrapper, and Embedded methods. The Filter method selects features using statistical measures, operating independently of the learning algorithm and demanding less computational time. Statistical measures such as information gain (Süpürtülü et al., 2025), chi-square test, Fisher score, correlation coefficient, and variance threshold are employed to assess the significance of features. The wrapper method's efficacy relies on the classifier employed, as it chooses the most suitable feature subset based on classifier performance. While Wrapper methods produce greater computational costs compared to Filter methods due to iterative learning and cross-validation, however they yield higher accuracy. The Embedded method, on the other hand, utilizes ensemble learning and hybrid learning techniques to perform FS.

2. Feature Selection

Feature Selection (FS) is the process of identifying a subset of features from the original feature space. Many methods exist for feature selection, such as Sequential Backward Selection (SBS) and Sequential Forward Selection (SFS). In the SFS method, the process starts with the smallest number of features and then incrementally adds features throughout the selection process. In contrast, the SBS method starts with all features and iteratively removes the least significant ones. After completing the process, the subset of features that persists in both method is deemed the optimal subset. The three types of FS methods: Filter method, Wrapper method, and Embedded method. Figure 1 represents an overview of different FS methods. In our study, we employed the wrapper method.

There have been many studies which utilizing evolutionary algorithms to reduce the features. In another work (Antony Gnana Singh et al., 2016), the authors used genetic algorithm (GA) for dimensionality reduction, to enhance accuracy in the classification process. Another study (Mukhlif et al., 2024), introduced an ant colony optimization (ACO) techniques for FS. It indicated that ACO exhibits reduce computational complexity compared to stochastic algorithms like GA. There is another study (Zahedi et al., 2022), which introduced an automatized bee colony based algorithm for FS aimed at addressing a classification problem. According to paper (Xiao et al., 2021), authors proposed the RnkHEU model, which combined ranking-based forward and heuristic search technique shown potential for enhancing the accuracy of predicting student's performance. Additionally, in paper (Turabieh et al., 2021), a modified version of the Harris Hawks optimization (HHO) algorithm was proposed, performing as a feature selection method to identify the most valuable features for the student performance prediction problem.

Most studies on student dropout prediction use filter methods. For example, one study (Cheng et al., 2022) applied genetic algorithms for FS in dropout prediction. Another compared correlation-based and entropy-based FS methods (Setiadi et al., 2024). (Febro, 2019) evaluated various filter methods (e.g., correlation analysis, chi-square, gain ratio) to identify retention factors. (Nuanmeesri et al., 2022) used filter methods and multilayer perceptron neural networks to forecast dropouts during the COVID-19 pandemic. We didn't find any studies that apply wrapper methods for student dropout prediction. One exception is (Youssef et al., 2019), who used a wrapper approach for predicting dropouts in MOOCs. Table 1 depicts some recent literature on feature selection techniques used for student dropout prediction.

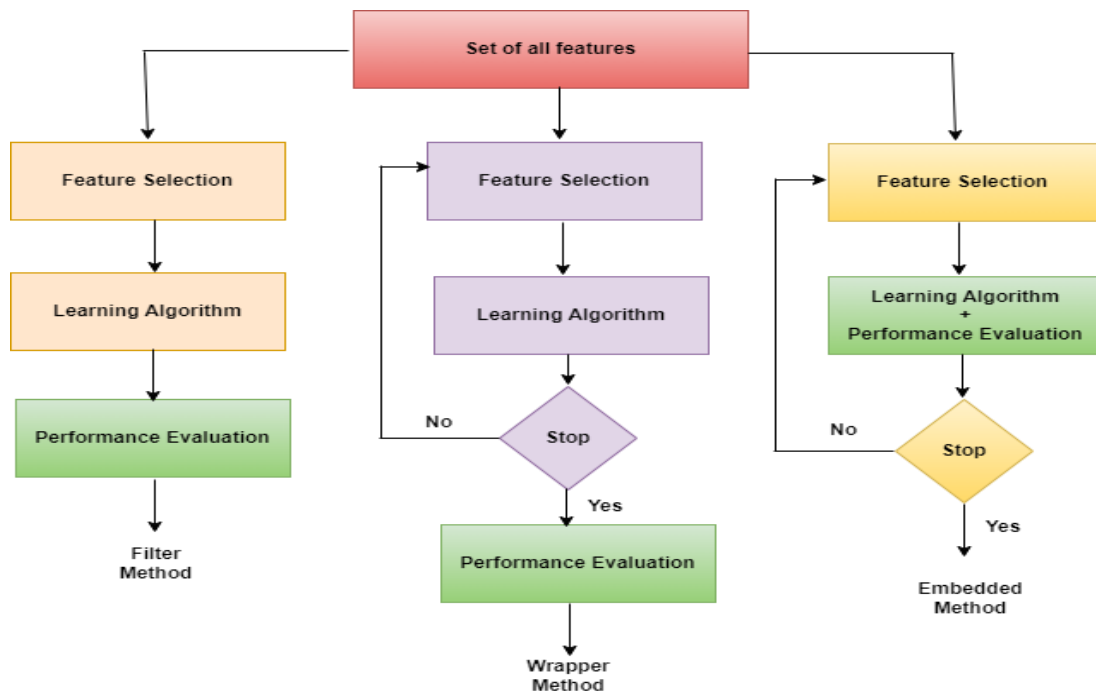


Figure 1. Overview of Filter, Wrapper, and Embedded methods

Table 1. Related literature for feature selection techniques used for student dropout prediction

Papers	Feature Selection Techniques
[23]	Genetic Algorithm
[24]	Correlation-based feature selection and entropy-based feature selection
[25]	Filter method (correlation feature selection, Chi-Square analysis, and gain ratio)
[26]	Filter method (correlation feature selection, Chi-Square analysis, and gain ratio)
[27]	Wrapper method
[28]	Ensemble method
[29]	Ensemble feature selection
[30]	Filter methods and classification algorithm
[31]	Feature X integrates filter-based and wrapper-based methods

3. Ant Colony Optimization

Ant Colony Optimization (ACO) is inspired by the foraging behavior of ants, which relies on indirect communication through chemical pheromone trails. This communication helps ants discover the shortest paths between their nest and food sources.

In ACO, optimization is achieved by continuously updating pheromone trails and moving ants throughout the search space based on simple mathematical formulas involving transition probabilities and the total pheromone in a given area. During each iteration, ACO generates global ants and evaluates their fitness. In this study, the fitness score is calculated using the classifier's accuracy as shown in equation (1). The pheromone levels and edges in weaker regions are updated accordingly. If the fitness improves, local ants are directed toward better regions; otherwise, a new random search direction is chosen. The pheromone levels are then adjusted, including evaporation to reduce old pheromones.

$$\text{fitness} = \text{accuracy}(\text{subset}) \quad (1)$$

Continuous ACO incorporates both local and global search mechanisms. Local ants are guided toward the most promising regions based on the transition probability (as shown in equation (2)) of region K, aiming to find the optimal solution.

$$P_K(t) = \frac{t_K(t)}{\sum_{j=1}^N t_j(t)} \quad (2)$$

where $t_K(t)$ total pheromone at region K and n is the number of global ants.

Pheromone is updated using the following equation (3),

$$t_i(t+1) = (1-r) * t_i(t), \quad (3)$$

where r is the pheromone evaporation rate.

The probability of solution of the region for local ants is proportional to the pheromone trail. The process flow for ACO is shown in Figure 2.

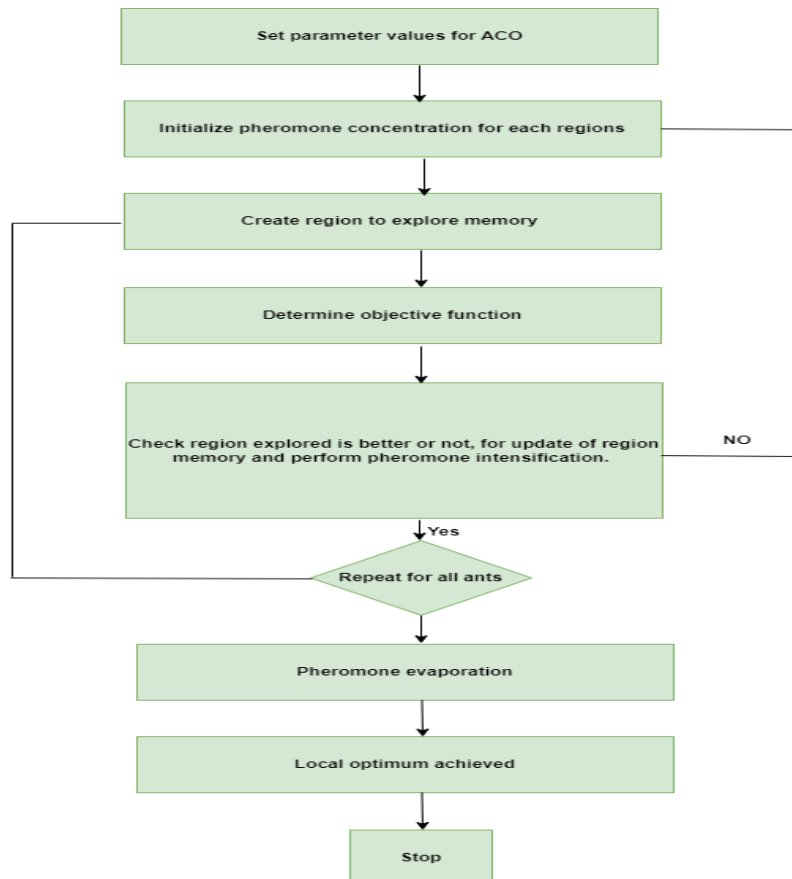


Figure 2. Process flow for ACO

Motivation and Justification: While Filter methods are widely used due to their simplicity, they may overlook complex interactions among features that are critical for dropout prediction. Similarly, Embedded methods, though efficient, often require tight coupling with specific algorithms. Therefore, this study adopts the Wrapper approach, which, despite its higher computational cost, allows feature selection tailored to the learning algorithm's performance, leading to better model outcomes. To address the limitations of deterministic wrapper methods, we employ a heuristic search strategy Ant Colony Optimization (ACO) due to its proven ability to explore large search spaces effectively and avoid local optima. Despite ACO's known utility in feature selection tasks, its application within student dropout prediction—especially using a wrapper framework remains underexplored. Our this work fills that gap.

Study Objectives: Our goal is to develop a model that identifies the most informative and non-redundant features for predicting student dropout. We applied the Wrapper method for FS using ACO as the search strategy. Given the class imbalance in our dataset, we used the SMOTE-Tomek links method to balance the data. Subsequently, we tested ACO with five classifiers Random Forest (ACO-RF), Logistic Regression (ACO-LR), K-Nearest Neighbors (ACO-KNN), Support Vector Machine (ACO-SVM), and Neural Network (ACO-NN) to evaluate the quality of selected features. The Random Forest classifier yielded the most optimal feature subset and was selected for further analysis. We propose the following research questions (RQ):

RQ1: *Will it be possible to obtain optimum features set to predict dropout student using ACO-RF?*

RQ2: *Will ACO-RF helps to increase the accuracy of the models?*

RQ3: *Will Neural networks perform better than machine learning algorithms on selected features?*

Addressing these questions is crucial within the context of Educational Data Mining because they directly impact the reliability, interpretability, and applicability of dropout prediction systems. Identifying the most relevant features (RQ1) supports the development of simpler and more interpretable models, which are essential for institutional stakeholders. Demonstrating that ACO-RF improves model performance (RQ2) validates the use of metaheuristic search strategies in educational domains. Finally, comparing neural networks and traditional models (RQ3) helps determine the appropriate computational complexity for deployment in real-world educational settings. By answering these (RQs), the study advances EDM practices and provides actionable insights for early intervention policies in higher education.

The rest of the article is structured as follows: Section B details the proposed methodology, Section C presents results and discussion, and Section D concludes the study and outlines future directions.

B. Methodology

The steps of the proposed methodology are detailed below and illustrated in Figure 3:

- a. *Preprocessing of the Dataset:* This step involves handling missing values, encoding categorical variables, and scaling numerical features to prepare the data for further analysis.
- b. *Balancing the Imbalanced Dataset using SMOTE-Tomek Links:* SMOTE (Synthetic Minority Over-sampling Technique) addresses class imbalance by generating synthetic examples of the minority class. Tomek links identify pairs of instances from different classes that are close to each other, and removing these pairs can improve the decision boundary. Combining SMOTE with Tomek links helps balance the dataset effectively.
- c. *Splitting the Dataset into Train and Test Sets:* The dataset is divided into training and testing sets. The training set is used to train the model, while the test set is used to evaluate its performance.

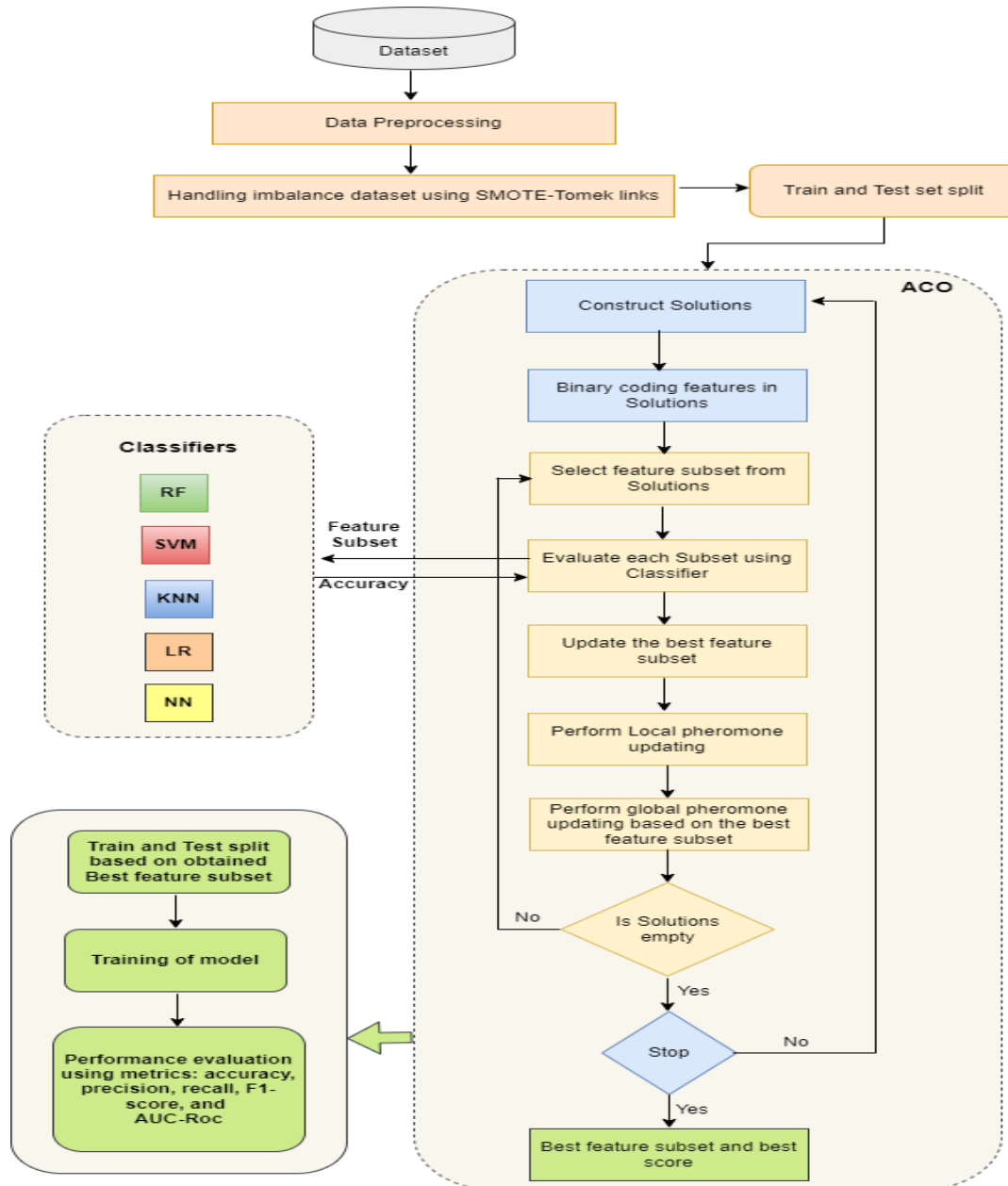


Figure 3. Overview of proposed methodology.

d. *Applying ACO Techniques for Feature Selection:*

1. **Initialize the Population:** Start with a set of candidate solutions (feature subsets).
2. **Construct the Solution:** Build solutions incrementally using heuristics or stochastic methods.
3. **Binary Encoding of Features in the Solution:** Represent each feature subset as a binary string. Binary encoding was chosen to represent feature subsets because it offers a simple, compact, and efficient representation of the feature selection problem. In this format, each feature is represented by a binary value—1 if the feature is selected, and 0 if it is not. This encoding is highly compatible with metaheuristic algorithms like Ant Colony Optimization (ACO), which explore solution spaces through combinatorial search. Binary encoding allows the algorithm to evaluate and update feature combinations in a structured way, making it easier to track inclusion/exclusion patterns, apply pheromone updates, and compute fitness scores. Moreover, this representation reduces the complexity of the search space and enhances convergence efficiency while maintaining flexibility across different classifier types.
4. **Select Feature Subset Using a Classifier:** Evaluate the quality of each feature subset using classifiers such as Random Forest, SVM, Logistic Regression, Neural Network, or KNN.

5. Update the Best Feature Subset: Keep track of the best-performing feature subset found so far.
 6. Perform Local Pheromone Updating: Update pheromone levels based on the quality of solutions found within a local neighborhood.
 7. Perform Global Pheromone Updating Based on the Best Feature Subset: Update pheromone levels globally based on the performance of the best solution identified.
 8. Return the Best Feature Subset and Best Fitness Score: Once the algorithm converges or reaches a stopping criterion, return the best feature subset found and its corresponding fitness score.
- e. *Performance Validation of the Best Feature Subset Using Five Learning Algorithms*: Train and evaluate models using the best feature subset obtained from the ACO technique. This step employs five different learning algorithms for comparison.
- f. *Performance Evaluation of Models Using Metrics*: Evaluate each model's performance using standard classification metrics such as accuracy, precision, recall, F1-score, and AUC-ROC (Area Under the Receiver Operating Characteristic Curve). These metrics provide insights into the models' classification accuracy and their ability to correctly identify positive and negative instances.

1. Proposed Algorithm

Algorithm 1: ACO solutions construction algorithm

Input : Initialize num_ants=10, alpha=1.0, beta=2.0, num_feature=X.shape[1], pheromone=0.1

Output: solutions

```

1: Initialize empty list solutions
2: For each ant in range(num_ants):
3:   Initialize empty list features
4:   For each feature index in range(num_features):
5:     Calculate selection probabilities probs using pheromone levels and heuristic information
6:     Normalize probs so that they sum to 1
7:     Randomly select a feature index based on the probabilities probs
8:     Append the selected feature index to features
9:   Encode the features list into a solution format
10:  Append the encode solution to solutions
11: Return solutions

```

The Algorithm 1 illustrates the solution construction process using the ACO technique. The parameter num_ants determines how many solutions are constructed per iteration. The parameters alpha and beta control the relative importance of pheromone levels and heuristic information, respectively. Pheromone levels guide the feature selection process, while num_features represents the total number of features in the dataset. For this experiment, we empirically chose the values for the parameters of the ACO optimizer such as num_ants=10, alpha=1.0, beta=2.0, and pheromone=0.1.

The proposed ACO-RF algorithm is detailed in Algorithm 2. The workflow begins with a main loop that continues until the termination criterion is met (a maximum of 100 iterations). Effective termination criteria help minimize computational complexity in identifying an optimal feature subset and prevent overfitting. The choice of stopping criteria is influenced by decisions made in earlier stages. Common stopping criteria include: (i) a predefined number of features, (ii) a predefined number of iterations, (iii) a percentage of improvement over successive iteration steps, and (iv) criteria based on the evaluation function. In our proposed algorithm, we used a predefined number of iterations.

Algorithm 2: Proposed ACO-RF feature selection algorithm

Input : Initialize best solution = None, Best score = $-\infty$, iteration = 0

Output: selected features, fitness score

```

1: While termination criterion (iteration):
2:   Solutions = construct solution()
3:   For each solution in Solutions:
4:     selected features=indices of selected features in solution
5:     score= evaluate solution RandomForest(training data with selected features,
        training labels, test data with selected features, test labels)
6:     if score > best score:
7:       update best score = score
8:       update best solution=selected features
9:   Perform local update on selected features

```

- 10: Perform global update on best solution
11. Increment iteration by 1
- 12: Return the best solution and best score.

Within each iteration, the first step involves constructing the solutions using Algorithm 1. In the second step, each solution in solutions is evaluated. The selected features from each solution are identified, and the Random Forest model is trained and tested using these features. If the current score surpasses the best score, both best score and best solution are updated. A local pheromone update is then performed based on the selected features. The third step involves a global pheromone update using the best solution. Once the loop concludes, the algorithm returns the optimal feature subset along with its fitness score.

2. Dataset

We have used a public dataset (Realinho et al., 2022) of the Polytechnic Institute of Portalegre. The same dataset that we used in our previous study (Singh & Karthikeyan, 2024), the dataset size is 743 KB and it contains 4424 records with 37 attributes. The dataset includes macroeconomic data, socioeconomic data, demographic data, and admission data and consists of semester first and semester second internal and external exam data from the institution. The original dataset has 3 targets class dropout, graduate, and enrolled, but for our work, we only used 2 classes i.e. dropout and graduate. To refine the classification task, we excluded students labeled as "enrolled" and focused solely on the two outcome-based categories: dropout and graduate. This decision was based on the fact that the enrolled students had not yet completed their academic journey at the time of data collection. Including them would introduce ambiguity, as their final status—whether they would eventually graduate or drop out—was unknown. Retaining only instances with definitive outcomes enables the model to learn more distinct patterns and relationships, thereby improving the reliability and interpretability of dropout prediction results. Hence after removing enrolled records, we left with 3630 records. Table 2 provides a comprehensive breakdown of the dataset's attributes, categorized into distinct classes: demographic characteristics, socioeconomic factors, macroeconomic indicators, data at the time of student enrollment, and academic data of the first and second semesters.

Table 2. Attributes used grouped by class of attribute

Class of Attribute	Attribute	Type
Demographic data	Marital status	Numeric/discrete
	Nationality	Numeric/discrete
	Age at enrollment	Numeric/discrete
	Displaced	Numeric/binary
	Marital status	Numeric/discrete
	Gender	Numeric/binary
	International	Numeric/binary
Socioeconomic data	Father's qualification	Numeric/discrete
	Mother's qualification	Numeric/discrete
	Mother's occupation	Numeric/discrete
	Father's occupation	Numeric/discrete
	Educational special needs	Numeric/binary
	Debtor	Numeric/binary
	Tuition fees up to date	Numeric/binary
	Scholarship holder	Numeric/binary
Macroeconomic data	Unemployment data	Numeric/continuous
	Inflation	Numeric/continuous
	GDP	Numeric/continuous
Admission data at enrollment	Application mode	Numeric/discrete
	Application order	Numeric/ordinal
	Course	Numeric/discrete
	Daytime/evening attendance	Numeric/binary
	Previous qualification	Numeric/discrete
Academic data at the end of 1st semester	Curricular units 1st semester (credited)	Numeric/discrete
	Curricular units 1st semester (enrolled)	Numeric/ordinal
	Curricular units 1st semester (evaluations)	Numeric/discrete

Class of Attribute	Attribute				Type
Academic data at the end of 2nd semester	Curricular (approved)	units	1st	semester	Numeric/binary
	Curricular (grade)	units	1st	semester	Numeric/continuous
	Curricular (without valuations)	units	1st	semester	Numeric/discrete
	Curricular (credited)	units	2nd	semester	Numeric/discrete
	Curricular (enrolled)	units	2nd	semester	Numeric/ordinal
	Curricular (evaluations)	units	2nd	semester	Numeric/discrete
	Curricular (approved)	units	2nd	semester	Numeric/binary
	Curricular (grade)	units	2nd	semester	Numeric/continuous
	Curricular (without valuations)	units	2nd	semester	Numeric/discrete
Target	Target				Categorical

3. Data Preprocessing, Balancing, and Train and Test split

In this phase, data cleaning and preprocessing were performed. We handled missing data using imputation and removal methods, transformed the data by encoding categorical variables, scaled the features, created new features, and split the data into training and testing sets. We used preprocessed data from our previous study, where the dataset included 3,630 records with two classes: dropout, and graduate.

Figure 4(a) illustrates the distribution of high-dimensional student data using the t-distributed Stochastic Neighbor Embedding (t-SNE) technique. Each point represents a student instance, where the blue dots (class 0) indicate students who did not drop out, and the orange dots (class 1) represent students who eventually dropped out. It is the distribution of student records before handling the class imbalance issue, with a clear dominance of the majority class (graduates) over the minority class (dropouts). This imbalance can lead to biased predictive models that favor the majority class, reducing the ability to correctly identify at-risk students. The significant overlap between the two classes in the plot also highlights the presence of noisy, irrelevant, or redundant features, which can obscure the true decision boundary. These observations emphasize the necessity of applying both class balancing techniques—such as SMOTE-Tomek links—and feature selection methods. Specifically, the ACO-based wrapper approach was employed to identify a compact, informative subset of features that improves model performance by enhancing class separation and reducing overfitting.



Figure 4. t-SNE Visualization of Student Dropout Dataset (with/without balancing)

We applied SMOTE-Tomek links to balance the imbalanced dataset. Figure 4(b), illustrates the t-SNE plot of balanced dataset. As in our dataset, the number of dropout students is fewer than the number of graduated

students. After applying the SMOTE-Tomek links we get 4186 records from 3630 records. The dataset was sorted by target features and then split into training (80%) and testing (20%) sets, with shuffling applied. After the split, we applied standardization to the training (X_train) data. Standardization, also known as feature scaling, is a crucial preprocessing step in machine learning and statistics. It involves transforming the values of different features in the dataset to a common scale or distribution.

4. Experimental setup

The experiments were conducted using Python programming language (version 3.7), the scikit-learn framework (version 1.3.2.) for data preprocessing tasks, and the tensorflow which provides access to the implementation of neural network, training, and testing run on Google Colab environment with T4 GPU. The details of the parameter we set for our neural network are shown in the Table 3. We optimized the hyperparameters of machine learning algorithms with different optimizers such as GridSearchCV and RandomSearchCV as shown in the Table 4.

Table 3. Parameters Set Up for the Neural Network Model.

Parameter	Value
Number of layers	4
Number of neurons in input unit	128
Number of neurons in dense layer 2	128
Number of neurons in dense layer 3	32
Number of neurons in dense layer 4	16
Batch size	16
Learning rate	0.01
Optimizer	Adam
Epochs	200
Activation function	Sigmoid
Computation mode	T4 GPU

Table 4. Parameters Set Up for the Neural Network Model.

Models	Parameters	Hyperparameter Optimizer
LR	C: 0.1, Penalty: l1, Solver: liblinear	GridSearchCV()
RF	Bootstrap: False, Criterion: entropy, Max depth: None, Max features: 6, Min samples leaf: 24, Min samples split: 16	RandomSearchCV()
DT	Criterion: gini, Max depth: 5, Max leaf nodes: 5, Min impurity decrease: 0.8, Splitter: random	GridSearchCV()
SVM	C: 1, Kernel: linear, Gamma: scale	GridSearchCV()
KNN	n_neighbors: 3	-
LR	C: 0.1, Penalty: l1, Solver: liblinear	GridSearchCV()
RF	Bootstrap: False, Criterion: entropy, Max depth: None, Max features: 6, Min samples leaf: 24, Min samples split: 16	RandomSearchCV()

5. Evaluation Metrics

The following metrics have been used to evaluate the models: Recall (REC), or true positive rate (TPR), represents the proportion of students who dropped out and were correctly predicted to do so. The false positive rate (FPR) measures the proportion of students who graduated but were incorrectly predicted to drop out. Specificity (SPEC), also known as the true negative rate (TNR), indicates the proportion of students who did not drop out and were accurately predicted to graduate.

The ROC (Receiver Operating Characteristic) curve plots the true positive rate (TPR) against the false positive rate (FPR) as the threshold for classifying events as positive or negative is varied. The AUC (Area Under the Curve) represents the area under the ROC curve, providing a single scalar value that summarizes the overall performance of a binary classification mode

The confusion metric parameters are used to examine the performance of our prediction models: Precision P, Recall R, and F1-score F. Their equations are given below (4), (5), (6):

$$P = \frac{TP}{TP+FP}, \quad (4)$$

$$R = \frac{TN}{TP+FN}, \quad (5)$$

$$F = \frac{2*(R*P)}{R+P}, \quad (6)$$

where TP (True Positive) counts non-successful students who are correctly classified as dropouts, TN (True Negative) counts successful students who are correctly classified as successful, FP (False Positive) counts successful students who are misclassified as dropouts (type II error), and FN (False negative) count non-successful students who are misclassified as successful (type I error).

Accuracy (7) is simply defined as a ratio of correctly predicted student at-risk retention status to the total number of students.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

C. Results and Discussion

1. Results

The study aimed to identify students who were at risk of dropping out their studies on time. Our dataset included three classes: enrolled, dropped out, and graduated. We focused solely on the dropout and graduated labels, removing the enrolled records. The preprocessed dataset was separated into two groups. 80% of the datasets were utilized for training, while 20% were for testing. The dataset had 2904 training samples and 858 testing samples, as indicated by the sample count.

ACO parameters are set to maximum-iteration= 100; maximum ants count = 10; pheromone evaporation rate = 0.1; alpha (pheromone importance) = 1.0; beta (Heuristic information importance)= 2.0, with an average run of 20 times. The experiment is performed using cross-validation (CV) to yield more robust results. CV (10 fold) splits the input into training data and test data independent of each other. Although CV may increase the computational time, it reduces the chance of overfitting and provides a more reliable model. In this study, we used parallel CV, and the final accuracy would be on average the accuracy for each of the folds.

In this study, we compared the four machine learning models and neural network performance with the feature subset selected by ACO-RF. The Tables 5 and 6 show all feature subsets selected by ACO-RF and ACO-LR respectively in 20 trials. Tables 7, 8, and 9 show all feature subsets selected by ACO-KNN, ACO-SVM, and ACO-NN respectively in the first 10 trials. The results indicate that ACO-RF achieves the highest fitness score of 0.9012. Therefore, we propose using ACO-RF for feature selection in predicting student dropout.

Table 5. Feature Selection Using ACO-RF

Experiment No.	Selected Features	No. of Features	Fitness Score
1	[0, 1, 2, 3, 7, 12, 18, 30]	8	0.8788
2	[0, 1, 2, 3, 7, 12, 17, 30]	8	0.8856
3	[0, 1, 2, 3, 4, 6, 24, 30]	8	0.8909
4	[0, 1, 2, 3, 7, 9, 13, 18, 31]	9	0.8474
5	[0, 1, 2, 3, 9, 10, 12, 14, 30]	9	0.8889
6	[0, 1, 2, 3, 7, 8, 11, 24, 30]	9	0.8961
7	[0, 1, 2, 3, 8, 9, 12, 22]	8	0.7458
8	[0, 1, 3, 6, 12, 16, 28]	7	0.7791
9	[0, 1, 2, 3, 5, 8, 11, 14, 19, 30, 34]	11	0.8856
10	[0, 1, 2, 3, 5, 6, 8, 9, 11, 13, 18, 34]	12	0.7712
11	[0, 1, 2, 3, 4, 5, 6, 7, 11, 18, 24]	11	0.8689
12	[0, 1, 2, 3, 6, 7, 25, 35]	8	0.8131
13	[0, 1, 2, 3, 4, 5, 30, 33]	8	0.8865
14	[0, 1, 2, 3, 4, 5, 10, 16, 17, 24, 26]	11	0.8934
15	[0, 1, 2, 3, 4, 8, 10, 11, 12, 20, 30]	11	0.9012
16	[0, 1, 2, 7, 9, 12, 30]	7	0.8641

Experiment No.	Selected Features	No. of Features	Fitness Score
17	[0, 1, 2, 3, 4, 5, 6, 11, 31]	9	0.8479
18	[0, 1, 2, 3, 6, 9, 25, 29]	8	0.8330
19	[0, 1, 2, 3, 4, 5, 24]	7	0.8566
20	[0, 1, 2, 3, 4, 6, 8, 16, 24]	9	0.8909

Table 6. Feature Selection Using ACO-LR.

Experiment No.	Selected Features	No. of Features	Fitness Score
1	[0, 1, 2, 4, 5, 6, 8, 12, 30]	9	0.8832
2	[0, 1, 2, 3, 6, 11, 16, 21, 23]	9	0.7118
3	[0, 1, 2, 7, 12, 25, 27, 34]	8	0.7536
4	[0, 1, 2, 4, 17, 30]	6	0.8766
5	[0, 1, 2, 4, 5, 10, 18, 24]	8	0.8361
6	[0, 1, 2, 3, 9, 12, 14, 16]	8	0.7166
7	[0, 1, 2, 3, 6, 14, 16, 18]	8	0.7378
8	[0, 1, 2, 3, 4, 26, 30]	7	0.8725
9	[0, 1, 2, 3, 4, 6, 31]	7	0.8124
10	[0, 1, 2, 4, 7, 16, 30]	7	0.7106
11	[0, 1, 2, 3, 4, 5, 6, 7, 21, 30]	10	0.8868
12	[0, 1, 2, 3, 5, 30]	6	0.8769
13	[0, 1, 2, 3, 4]	5	0.6323
14	[0, 1, 2, 4, 5, 9, 14, 23, 30]	9	0.8755
15	[0, 1, 2, 30]	4	0.8737
16	[0, 1, 2, 4, 5, 10, 30]	7	0.8716
17	[0, 1, 2, 18]	4	0.6908
18	[0, 1, 2, 4, 6, 8, 13, 16, 25]	10	0.8169
19	[0, 1, 2, 3, 4, 9, 25]	7	0.7656
20	[0, 1, 2, 3, 9, 22, 25, 30]	8	0.8862

Table 7. Feature Selection Using ACO-KNN.

Experiment No.	Selected Features	No. of Features	Fitness Score
1	[0, 1, 2, 3, 4, 21, 30]	7	0.8605
2	[0, 1, 2, 5, 31]	5	0.7916
3	[0, 1, 2, 3, 4, 10, 18]	7	0.6816
4	[0, 1, 2, 4, 5, 11, 14, 30]	8	0.8635
5	[0, 1, 3, 24]	4	0.8456
6	[0, 1, 2, 3, 5, 9, 24, 29, 31]	9	0.8325
7	[0, 1, 2, 3, 4, 6, 16, 22]	8	0.7306
8	[0, 1, 2, 3, 4, 6, 30]	7	0.8602
9	[0, 1, 2, 3, 4, 11, 14, 25, 34]	9	0.7545
10	[0, 1, 2, 3, 4, 6, 13, 15, 18, 25]	10	0.7727

Table 8. Feature Selection Using ACO-SVM.

Experiment No.	Selected Features	No. of Features	Fitness Score
1	[0, 1, 2, 3, 4, 5, 6, 20, 30]	9	0.8749
2	[0, 1, 2, 3, 4, 10, 23, 24]	8	0.8528
3	[0, 1, 2, 3, 4, 24]	6	0.8367
4	[0, 1, 2, 3, 4, 10, 11, 13, 14, 20, 25]	11	0.7682
5	[0, 1, 2, 3, 4, 6, 10, 30]	8	0.8695
6	[0, 1, 2, 3, 5, 6, 26, 31]	8	0.7975
7	[0, 1, 2, 3, 24]	5	0.8309
8	[0, 1, 2, 3, 4, 5, 7, 8, 15, 30]	10	0.8701
9	[0, 1, 2, 3, 5, 24]	6	0.8417
10	[0, 1, 2, 3, 4, 5, 6, 16, 21]	9	0.7506

Table 9. Feature Selection Using ACO-NN.

Experiment No.	Selected Features	No. of Features	Fitness Score
1	[0, 1, 2, 5, 30]	5	0.8829
2	[0, 1, 2, 3, 4, 6, 10, 11, 12, 25, 26, 28]	12	0.8495
3	[0, 1, 2, 3, 16]	5	0.7411
4	[0, 1, 2, 4, 5, 12, 31]	7	0.7984
5	[0, 1, 2, 4, 5, 12, 31]	7	0.7984
6	[0, 1, 2, 3, 24]	5	0.8394
7	[0, 1, 2, 3, 7, 12, 13, 30]	8	0.8831
8	[0, 1, 2, 3, 9, 28, 30]	7	0.8962
9	[0, 1, 2, 3, 4, 10, 11, 16, 17, 30]	10	0.7446

Table 10. Comparison of performance evaluation of models on different metrics with full features.

Models	Accuracy	Precision	Recall	F1-score
LR	0.9215	0.9441	0.8682	0.9045
RF	0.8953	0.9447	0.7789	0.8538
DT	0.9063	0.9286	0.8345	0.8790
KNN	0.8505	0.8874	0.7295	0.8007
NN	0.8981	0.8768	0.8514	0.8639

Table 11. Validation of feature subsets extracted by ACO-RF using different ML models, and a neural network.

Model	Selected Features	Accuracy	Precision	F1-score	Recall	AUC
KNN	[0,1,2,3,4,8,10,11,12,20,30]	0.8485	0.8784	0.8479	0.8194	0.85
	[0,1,2,3,5,8,11,14,19,30,34]	0.8628	0.8866	0.8596	0.8342	0.86
	[0,1,2,3,4,5,10,16,17,24,26]	0.8735	0.9021	0.8685	0.8374	0.87
	[0,1,2,3,4,6,8,16,24]	0.8699	0.9003	0.8688	0.8395	0.87
	[0,1,2,3,4,5,30,33]	0.8508	0.8628	0.8470	0.8317	0.85
	[0,1,2,3,7,8,11,24,30]	0.8592	0.8765	0.8622	0.8483	0.86
	[0,1,2,3,9,10,12,14,30]	0.8921	0.8475	0.8072	0.85	0.85
	[0,1,2,3,4,6,24,30]	0.8616	0.9111	0.8535	0.8029	0.86
RF	[0,1,2,3,7,12,17,30]	0.8640	0.8911	0.8659	0.8421	0.86
	[0,1,2,3,4,8,10,11,12,20,30]	0.8831	0.9239	0.8741	0.8293	0.88
	[0,1,2,3,5,8,11,14,19,30,34]	0.8795	0.9066	0.8767	0.8487	0.88
	[0,1,2,3,4,5,10,16,17,24,26]	0.8795	0.9144	0.8811	0.85	0.88
	[0,1,2,3,4,6,8,16,24]	0.8712	0.9127	0.8715	0.8338	0.87
	[0,1,2,3,4,5,30,33]	0.8915	0.9156	0.8841	0.8547	0.89
	[0,1,2,3,7,8,11,24,30]	0.8712	0.9084	0.8619	0.8200	0.87
	[0,1,2,3,9,10,12,14,30]	0.8843	0.9377	0.8771	0.8238	0.88
LR	[0,1,2,3,4,6,24,30]	0.8902	0.9280	0.8833	0.8426	0.89
	[0,1,2,3,7,12,17,30]	0.8891	0.9326	0.8816	0.8357	0.89
	[0,1,2,3,4,8,10,11,12,20,30]	0.8664	0.9146	0.8631	0.8172	0.87
	[0,1,2,3,5,8,11,14,19,30,34]	0.8533	0.8692	0.8513	0.8342	0.85
	[0,1,2,3,4,5,10,16,17,24,26]	0.8580	0.8997	0.8515	0.8081	0.86
	[0,1,2,3,4,6,8,16,24]	0.8687	0.9269	0.8659	0.8124	0.87
	[0,1,2,3,4,5,30,33]	0.8759	0.8988	0.875	0.8525	0.88
	[0,1,2,3,7,8,11,24,30]	0.8640	0.8735	0.8674	0.8614	0.86
DT	[0,1,2,3,9,10,12,14,30]	0.8807	0.8983	0.8786	0.8599	0.88
	[0,1,2,3,4,6,24,30]	0.8687	0.8843	0.8622	0.8411	0.87
	[0,1,2,3,7,12,17,30]	0.8831	0.8903	0.8816	0.8732	0.88
	[0,1,2,3,4,8,10,11,12,20,30]	0.8747	0.8922	0.8739	0.8565	0.87
	[0,1,2,3,5,8,11,14,19,30,34]	0.8675	0.8922	0.8677	0.8445	0.87

Model	Selected Features	Accuracy	Precision	F1-score	Recall	AUC
NN	[0,1,2,3,4,5,10,16,17,24,26]	0.8926	0.8781	0.8869	0.8959	0.89
	[0,1,2,3,4,6,8,16,24]	0.8807	0.8886	0.8778	0.8671	0.88
	[0,1,2,3,4,5,30,33]	0.8699	0.8921	0.8722	0.8532	0.87
	[0,1,2,3,7,8,11,24,30]	0.8854	0.8925	0.8884	0.8843	0.89
	[0,1,2,3,9,10,12,14,30]	0.8759	0.8875	0.8697	0.8526	0.88
	[0,1,2,3,4,6,24,30]	0.8831	0.8917	0.87841	0.8655	0.88
	[0,1,2,3,7,12,17,30]	0.8842	0.8936	0.8816	0.8699	0.88
	[0,1,2,3,4,8,10,11,12,20,30]	0.9317	0.9116	0.8420	0.7929	0.93
	[0,1,2,3,5,8,11,14,19,30,34]	0.9291	0.8295	0.8329	0.8605	0.93
	[0,1,2,3,4,5,10,16,17,24,26]	0.9212	0.8438	0.8392	0.8476	0.92
	[0,1,2,3,4,6,8,16,24]	0.9134	0.7609	0.7687	0.8412	0.92
	[0,1,2,3,4,5,30,33]	0.9325	0.8636	0.8481	0.8470	0.93
	[0,1,2,3,7,8,11,24,30]	0.9222	0.8015	0.7274	0.7587	0.92
	[0,1,2,3,9,10,12,14,30]	0.8916	0.8464	0.7721	0.7859	0.90
	[0,1,2,3,4,6,24,30]	0.9283	0.8511	0.7766	0.7802	0.93
	[0,1,2,3,7,12,17,30]	0.9307	0.8907	0.8327	0.8029	0.93

Tables 10 and 11 present a comparative analysis of model performance on the full feature set and on optimized subsets obtained through ACO-based feature selection, respectively. Notably, the neural network (NN) achieved the highest accuracy (93.25%) and AUC (0.93) after feature selection, surpassing its original accuracy of 89.81% on the full dataset. KNN showed the most significant improvement, with its F1-score rising from 0.8007 to 0.8685 and accuracy increasing by 2.3%. This indicates KNN's sensitivity to irrelevant features and its benefit from dimensionality reduction. In contrast, logistic regression (LR), which previously outperformed all models with 92.15% accuracy on the full dataset, experienced a decline to 88.31% post-selection, suggesting that LR benefits more from a complete feature space. Random forest (RF) and decision tree (DT) models showed minor changes in accuracy but consistent improvements in F1-score, highlighting their robustness to feature pruning. Overall, these results demonstrate that feature selection can significantly enhance model generalization—particularly for complex and non-parametric models such as NN and KNN.

Table 12. Selected Features by ACO-RF.

Sequence No.	Selected Features
1	Marital Status
2	Application mode
3	Application order
4	Course
8	Nationality
10	Father's qualification
11	Mother's occupation
12	Father's occupation
20	Age at enrollment
30	Curricular units 2nd semester (evaluations)

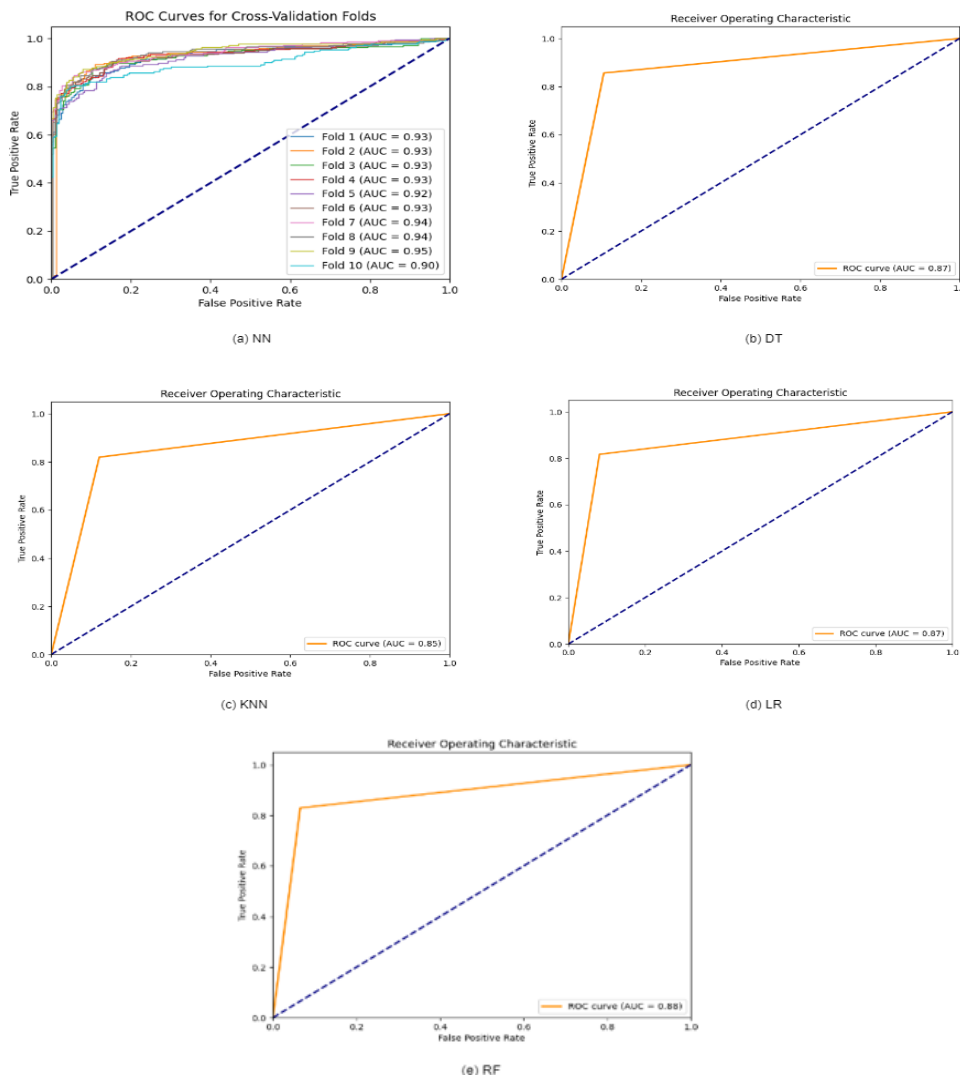


Figure 5. ROC curve of compared models using best subset [0,1,2,3,4,8,10,11,12,20,30] obtained by ACO-RF.

In Figure 5 represents ROC curves of all compared models are shown with a 10-fold CV. NN outperformed the other machine learning models. The Figure 6 depicts the models accuracy on the best subset selected [0,1,2,4,8,10,11,12,20,30] by ACO-RF.

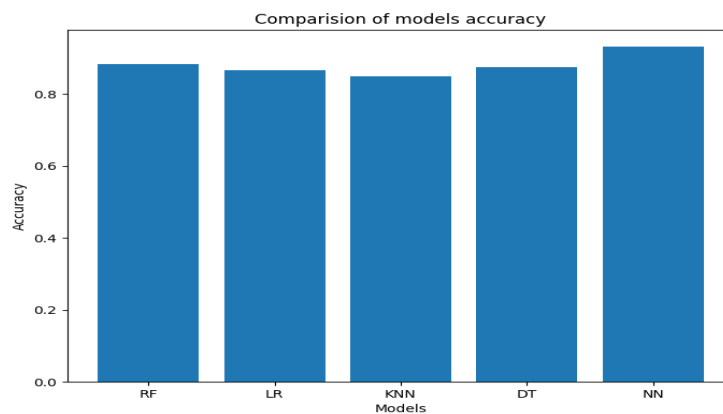


Figure 6. Models Accuracy on best subset selected [0,1,2,4,8,10,11,12,20,30] by ACO-RF.

The features selected by ACO-RF that affect the dropout rate of students in Higher Education as shown in Table 12 are marital status, application mode, application order, course, nationality, father's qualification, mother's occupation, father's occupation, age of enrollment, and curricular units second semester (evaluations). Most of the selected features subset included curricular units second semester (evaluations) feature which determined that the second-semester grade is one of the deciding features for students dropping out of Higher Education.

This study started with three research questions: first, *Will it be possible to obtain optimum features set to predict dropout student using ACO-RF?*; second, *Will ACO-RF helps to increase the accuracy of the models?*; and third, *Will Neural networks perform better than machine learning algorithms on selected features?* To address our research questions, yes, ACO-RF performed better with a maximum fitness score of 90%, and validating its selected feature with classifiers improved the classifier's accuracy. As the Figure 6 depicts the improvement in accuracy of the neural network model from 0.8981 to 0.9325. Among all the classifiers we used for our experiment, the neural network outperformed with an accuracy of 93%.

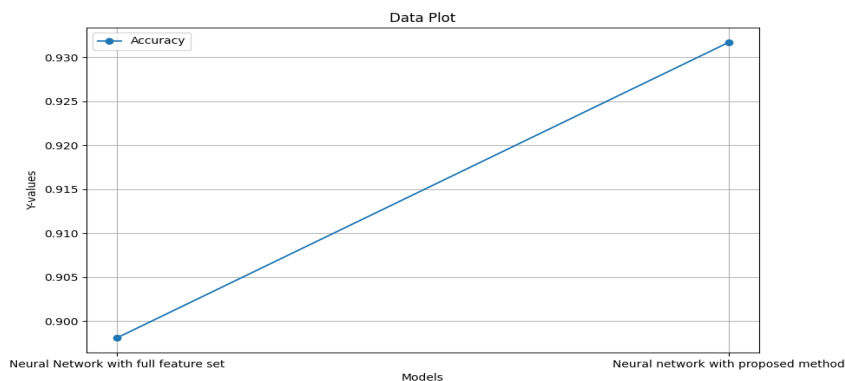


Figure 7. Represent the improvement in accuracy of the Neural network model using our proposed approach (ACO-RF).

2. Discussion

2.1 Implications

The findings reveal that applying ACO-RF-based feature selection significantly enhances the predictive performance of neural networks and KNN models. This suggests that educational institutions can implement models using compact and relevant feature subsets to more accurately identify at-risk students while reducing computational costs. Among the features selected, second-semester academic performance consistently emerged as a critical predictor, highlighting the importance of early academic intervention during the first year. Additional features such as parental occupation, application mode, and age at enrollment also demonstrated strong predictive value. These carry significant policy implications, as they offer concrete areas for targeted support. For instance, students with weak second-semester performance may benefit from remedial academic programs, while those from certain socioeconomic backgrounds could receive tailored counseling services. These insights enable institutions to design evidence-based policies that address student needs more effectively. Overall, the results reinforce the study's contributions—particularly in improving model interpretability and enhancing prediction performance using the ACO-RF approach.

2.2 Research Contribution

This study contributes to the field of educational data mining by proposing a novel integration of ACO with a wrapper-based feature selection approach tailored for student dropout prediction. The ACO-RF framework not only identifies relevant features but also enhances the performance of predictive models—especially neural networks. Furthermore, it offers an interpretable subset of features, which is crucial for policy design and intervention planning in higher education institutions. Our study makes several contributions:

1. Few studies have applied metaheuristic techniques for FS in student dropout prediction. We address this gap by proposing an ACO-RF approach.
2. We integrated ACO with various classifiers, compared their performance and selected the best one for feature selection.
3. We improved the predictive performance of models by using a reduced feature set.
4. Our approach identifies an optimal feature subset, providing insights that can help educational institutions formulate policies to support at-risk students.

2.3 Limitations

Our work has two main limitations. Firstly, the computational time increases as the number of ants and generations in the ACO-RF algorithm rises. This is because more ants and generations demand greater processing power and time to execute. However, increasing these parameters improves the likelihood of identifying the optimal set of features, ultimately enhancing the accuracy and effectiveness of the feature selection process. The second limitation is the unavailability of a dataset. We are actively addressing this issue by collecting our own dataset, which will support future research. To ensure the broad applicability and robustness of the proposed methodology, we aim to use diverse datasets sourced from multiple universities. This will help validate the effectiveness of our approach in various contexts.

2.4 Suggestions

To build on this research, future work could:

1. Explore other metaheuristic algorithms (e.g., Particle Swarm Optimization or Grey Wolf Optimizer) for feature selection.
2. Integrate temporal features such as progression patterns across semesters.
3. Apply the method across multi-institutional or international datasets for broader generalization.
4. Extend the study with SHAP or LIME to improve model interpretability and stakeholder trust.

Additionally, further qualitative analysis on the impact of each selected feature, supported by domain literature, would provide deeper insight into student behavior and the effectiveness of interventions.

D. Conclusion

The rising rate of student dropouts in universities signals underlying challenges in the education system. This study proposed an ACO-RF feature selection method that integrates Ant Colony Optimization (ACO) with Random Forest (RF) to identify the most influential factors contributing to student dropout. Using a dataset of undergraduate students, including demographic, socioeconomic, macroeconomic, admission, and academic performance information, our approach successfully selected the most relevant and non-redundant features. The ACO-RF method achieved a maximum fitness score of 90%, demonstrating its superior ability to enhance model accuracy compared to other classifier combinations. The selected features such as second-semester evaluations, parental occupation, application mode, and age at enrollment carry significant policy implications. Institutions can leverage these insights to design targeted interventions, such as early academic support for students struggling in their second semester or counseling services based on socioeconomic indicators. Our results strongly support the contributions claimed in the study, particularly regarding improved model interpretability and prediction performance. However, the method also has limitations, primarily the increase in computational cost with more ants and generations, and the restricted availability of diverse datasets. We are addressing this by actively collecting a more representative dataset from multiple universities. Future work will expand the scope by incorporating behavioral and interaction data—such as student engagement on learning platforms, attendance patterns, and course participation—and by benchmarking ACO-RF against other heuristic techniques like Particle Swarm Optimization (PSO) and Grey Wolf Optimizer (GWO). These enhancements aim to create more robust and actionable models for dropout prediction and educational policy planning.

E. Acknowledgment

We would like to express our deep appreciation to all entities and individuals who have made significant contributions to this research on Wrapper Feature Selection Method for Dropout Prediction in Higher Education. The support provided, whether in the form of enlightening academic guidance, provision of crucial datasets, access to computing facilities or invaluable moral support, has been the main foundation

for the smooth and successful analysis. All forms of participation were important determinants in the findings presented in this study.

F. Author Contribution Statement

A.K.S. formulated the research problem, conducted all the experiments, wrote the complete manuscript, validated the results, created the figures and tables, and performed the final review of the paper. S.K. supervised all the tasks, validated the results, and also reviewed the final manuscript.

References

- Alban, M., & Mauricio, D. (2019). Predicting University Dropout through Data Mining: A systematic Literature. *Indian Journal of Science and Technology*, 12(4), 1–12. <https://doi.org/10.17485/ijst/2019/v12i4/139729>
- Aleem, A., & Gore, M. M. (2020). Educational data mining methods: A survey. *Proceedings - 2020 IEEE 9th International Conference on Communication Systems and Network Technologies, CSNT 2020*, 182–188. <https://doi.org/10.1109/CSNT48778.2020.9115734>
- Antony Gnana Singh, D. A., Leavline, E. J., Priyanka, R., & Priya, P. P. (2016). Dimensionality reduction using genetic algorithm for improving accuracy in medical diagnosis. *International Journal of Intelligent Systems and Applications*, 8(1), 67–73. <https://doi.org/10.5815/ijisa.2016.01.08>
- Aulck, L., Velagapudi, N., Blumenstock, J., & West, J. (2016). Predicting Student Dropout in Higher Education. *ArXiv*, 1(4), 16–20. <https://doi.org/10.48550/arXiv.1606.06364>
- Cheng, Y., Pereira Nunes, B., & Manrique, R. (2022). Not Another Hardcoded Solution to the Student Dropout Prediction Problem: A Novel Approach Using Genetic Algorithms for Feature Selection. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-031-09680-8_23
- Del Bonifro, F., Gabbrielli, M., Lisanti, G., & Zingaro, S. P. (2020). Student dropout prediction. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12163 LNAI, 129–140. https://doi.org/10.1007/978-3-030-52237-7_11
- Febro, J. D. (2019). Utilizing feature selection in identifying predicting factors of student retention. *International Journal of Advanced Computer Science and Applications*, 10(9), 269–274. <https://doi.org/10.14569/ijacsa.2019.0100934>
- Gardner, J., & Brooks, C. (2018). Dropout model evaluation in MOOCs. *The Thirty-Ninth AAAI Conference on Artificial Intelligence*, 32(2), 7906–7912. <https://doi.org/10.1609/aaai.v32i1.11392>
- Kadar, M., Guevara, J. C., Sarraipa, J., & Gutiérrez Restrepo, E. (2018). An integrated approach for fighting dropout and enhancing students' satisfaction in higher education. *ACM International Conference Proceeding Series*, 240–247. <https://doi.org/10.1145/3218585.3218667>
- Li, H., Lynch, C. F., & Barnes, T. (2018). Early Prediction of Course Grades: Models and Feature Selection. *Journal Reference: The Proceedings of the 11th International Conference on Educational Data Mining (EDM 2018)*, 492–495. <https://doi.org/10.48550/arXiv.1812.00843>
- Mukhlif, Y. A., Ramaha, N. T. A., Hameed, A. A., Salman, M., Yon, D. K., Fitriyani, N. L., Syafrudin, M., & Lee, S. W. (2024). Ant Colony and Whale Optimization Algorithms Aided by Neural Networks for Optimum Skin Lesion Diagnosis: A Thorough Review. *Mathematics*, 12(7), 1–29. <https://doi.org/10.3390/math12071049>
- Nicoletti, M. do C. (2019). Revisiting the Tinto's Theoretical Dropout Model. *Higher Education Studies*, 9(3), 52. <https://doi.org/10.5539/hes.v9n3p52>
- Nuanmeesri, S., Poomhiran, L., Chopvitayakun, S., & Kadmateekarun, P. (2022). Improving Dropout Forecasting during the COVID-19 Pandemic through Feature Selection and Multilayer Perceptron Neural Network. *International Journal of Information and Education Technology*, 12(9), 851–857. <https://doi.org/10.18178/ijiet.2022.12.9.1693>
- Realinho, V., Machado, J., Baptista, L., & Martins, M. V. (2022). Predicting Student Dropout and Academic Success. *Data*, 7(11), 1–17. <https://doi.org/10.3390/data7110146>
- Setiadi, H., Larasati, I. P., Suryani, E., & Wardani, D. W. (2024). Comparing Correlation-Based Feature Selection and Symmetrical. *Jurnal RESTI*, 5(2), 542–554. <https://doi.org/10.29207/resti.v8i4.5911>
- Singh, A. K., & Karthikeyan, S. (2024). Heuristic Technique to Find Optimal Learning Rate of LSTM for Predicting Student Dropout Rate. *Communications in Computer and Information Science*, 2151 CCIS(May), 47–54. https://doi.org/10.1007/978-3-031-64312-5_6

- Siri, A. (2015). Predicting students' academic dropout using artificial neural networks. *Predicting Students' Academic Dropout Using Artificial Neural Networks*, 7(June), 1–159. <https://doi.org/10.14658/PUPJ-IJSE-2015-2-9>
- Süpürtülü, M., Hatipoğlu, A., & Yılmaz, E. (2025). An Analytical Benchmark of Feature Selection Techniques for Industrial Fault Classification Leveraging Time-Domain Features. *Applied Sciences (Switzerland)*, 15(3), 1–25. <https://doi.org/10.3390/app15031457>
- Turabieh, H., Azwari, S. Al, Rokaya, M., Alosaimi, W., Alharbi, A., Alhakami, W., & Alnfai, M. (2021). Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance. *Computing*, 103(7), 1417–1438. <https://doi.org/10.1007/s00607-020-00894-7>
- Villa-Blanco, C., Bielza, C., & Larrañaga, P. (2023). Feature subset selection for data and feature streams: a review. *Artificial Intelligence Review*, 56, 1011–1062. <https://doi.org/10.1007/s10462-023-10546-9>
- Wan Yaacob, W. F., Mohd Sobri, N., Nasir, S. A. M., Wan Yaacob, W. F., Norshahidi, N. D., & Wan Husin, W. Z. (2020). Predicting Student Drop-Out in Higher Institution Using Data Mining Techniques. *Journal of Physics: Conference Series*, 1496(1), 1–15. <https://doi.org/10.1088/1742-6596/1496/1/012005>
- Xiao, W., Ji, P., & Hu, J. (2021). RnkHEU: A Hybrid Feature Selection Method for Predicting Students' Performance. *Scientific Programming*, 1, 1–16. <https://doi.org/10.1155/2021/1670593>
- Youssef, M., Mohammed, S., Hamada, E. K., & Wafaa, B. F. (2019). A predictive approach based on efficient feature selection and learning algorithms' competition: Case of learners' dropout in MOOCs. *Education and Information Technologies*, 24(6), 3591–3618. <https://doi.org/10.1007/s10639-019-09934-y>
- Zahedi, L., Ghareh Mohammadi, F., & Amini, M. H. (2022). A2BCF: An Automated ABC-Based Feature Selection Algorithm for Classification Models in an Education Application. *Applied Sciences (Switzerland)*, 12(7), 1–16. <https://doi.org/10.3390/app12073553>

Copyright Holder

© Singh, A., & Karthikeyan, S.

First publication right:

JENTIK: Jurnal Pendidikan Teknologi Informasi dan Komunikasi

This article is licensed under:

