Artificial Intelligence Evolution
http://ojs.wiserpub.com/index.php/AIE/

Research Article

Prediction of the Computer Science Department's Educational Performance Through Machine Learning Model by Analyzing Students' Academic Statements

Md. Alamgir Hossain, Ishtiaq Ahammad*, Md. Kawsar Ahmed, Md Imtiaz Ahmed

Department of Computer Science and Engineering, Prime University, Dhaka, Bangladesh
Email: ahammadishtiaq27@gmail.com

Received: 25 February 2023; **Revised:** 16 April 2023; **Accepted:** 15 May 2023

Abstract: In this digital world, technology is changing every second. For this reason, tech-based educational departments should be aware of their strong and weak areas. By doing so, they can analyze their performance and emphasis the respected fields to prepare their students for the challenging job market and higher studies. In this research, a novel machine learning model is proposed to predict the performance of computer science department students by analyzing their academic statements. The academic areas of the departments are divided into fifteen fields, each of which is categorized as Excellent, Very Good, Good, Average, and Bad. One thousand students' academic statements were used as the dataset. The prediction outputs are labeled into multiple categories, with multiclass data in each. Different models are developed using multiclass-multioutput classification algorithms like Decision Tree (Tree), Extra Tree (Tree), Extra Trees (Ensemble), K-Neighbors (Neighbors), Radius Neighbors (Neighbors), and Random Forest (Ensemble) Classifier. From the developed models, the model that used "Random Forest Classifier (Ensemble)" provided the best results, and the accuracy was almost 94%. Finally, a comparison of our research with previous and existing different developed models is presented to show the novelty of this research.

Keywords: departmental performance analysis, performance prediction, strong areas of computer science, prediction with machine learning, departmental improvements

1. Introduction

Higher education institutions' primary goal is to provide students with an excellent educational experience [1]. A growing issue is the necessity to pinpoint students' areas of strength and predict their future professional field based on their strengths. The discovery of hidden links in educational data and the prediction of students' academic success are now both possible with the help of educational data mining [2]. Advanced machine-learning (ML) techniques along with big data management are used in this field of research to produce timely and useful information which can improve both students' as well as educational institutions' overall learning experiences [3]. Predictive applications for ML are typically found in fields like healthcare [4-5] and the stock market [6]. However, there are more and more opportunities to use ML algorithms to analyze educational data and enhance the learning experience [7]. Institutional technologies, e-learning tools, and online classes are all producing a ton of data. Educators might examine and comprehend student learning practices using this data. These data can then be used to identify a student's strong areas and predict their future career fields using ML approaches.

The primary problem that this research aims to solve is the identification of areas where a computer science department and its students may need improvement. By analyzing students' academic statements, we aim to predict their performance in different fields. By academic statement, we mean the academic results of the students. In our research, we have categorized the courses of the computer science department into fifteen fields. These fields might be selected as future professional places where the students might work. This is where our prediction is helpful. Using our prediction model, we can predict the future professional field of a student among those fifteen fields based on the students' academic statements. This might help to identify areas where students may need more guidance and where the department can focus its resources to improve its overall performance.

The benefits of solving this problem are multifold. Firstly, it can help the department to prepare its students for the challenging job market and higher studies. By identifying the areas where students are weak, the department can provide extra attention and resources to those fields to strengthen its students' skills. Secondly, it can enhance the department's reputation by ensuring that its graduates are well-prepared for the job market.

However, this problem is not easy to solve, and traditional methods may not be efficient or effective. The analysis of academic statements is a complex and time-consuming process that requires significant resources. Moreover, identifying patterns and trends in a vast amount of data is a challenging task. Therefore, we propose a machine learning model that can efficiently analyze the data and predict the performance of different fields in the department.

To implement our model, we use several machine learning algorithms, specifically the multiclass-multioutput classification algorithm. We divide the academic areas of the department into fifteen fields, each categorized as Excellent, Very Good, Good, Average, and Bad. We use a dataset of one thousand students' academic statements to develop and compare different models using the following algorithms: Decision Tree (Tree), Extra Tree (Tree), Extra Trees (Ensemble), K-Neighbors (Neighbors), Radius Neighbors (Neighbors), Random Forest (Ensemble) Classifier. Then, evaluation metrics are used for the predicted values to determine the effectiveness of the trained models. For this research, we consider four evaluation metrics, Accuracy, Precision, Recall, and F1 score. Based on these evaluation metrics, we found that the Random Forest (Ensemble) Classifier delivers the best performance among all implemented ML algorithms. The major contributions of this research are:

• The development of a novel machine learning model that analyzes students' academic statements and then predicts their performance, it contributes to the field of educational analytics and prediction.

• By identifying the areas where students are weak, the department can provide extra attention and resources to those fields to strengthen its students' skills which might help the students in the future challenging job market and higher studies.

• The prediction model's outcome can enhance the department's reputation by ensuring that its graduates are well-prepared for the job market.

• Deliver insights into factors that affect educational performance in computer science education. This could help develop targeted interventions or policies to improve educational outcomes.

• The proposed ML model is implemented with multiple ML algorithms, and their comparative evaluation metric values are presented. This helps in understanding and identifying the best performing algorithm.

• Improved prediction performance (with 94% accuracy) makes the suggested predictive ML model to be implemented in other institutions.

The paper is organized as follows. In the second section, we provide a detailed literature review of machine learning and its applications in education. In the third section, we present the methodology, which includes data collection, preprocessing, and model development. In the fourth section, we present the results of our analysis and a detailed discussion of our findings. Finally, we conclude our research, where we provide recommendations for future research and implications for practice.

2. Literature review

The use of machine learning in predicting academic performance has been an area of interest in recent years. The use of machine learning models to forecast academic performance based on a variety of variables, including demographic information, academic records, and student actions, has been the subject of numerous researches. However, the use of model-based machine learning to forecast the performance of a computer science department has received relatively little research attention.

El-Halees [8] used of academic data analysis to evaluate learning outcomes by collecting data from a course database. The study used various data analysis techniques, including association rules, classification, clustering, and outlier identification, to analyze the collected data. The study found that the four approaches used were effective in evaluating student performance based on the collected data.

In order to classify students as academically strong performers or poor performers, Ghassen Ben Brahim [9] generated an 86-dimensional feature space where only informative features were used. Using random-forest classifiers, the writer measured the effectiveness of the proposed methodology under three distinct experimental conditions and attained 97.4% efficiency.

Several machine-learning techniques were used by Nikola et al. [10] to evaluate the effectiveness of their suggested strategy. The authors approached the issue of exam prediction as a task requiring both categorization and regression. In the classification scenario, students were classified as “passing” or “failing”, and much like in the regression scenario, the expected exam grade was indeed the exam objectives result.

Sekeroglu et al. [11] employed a variety of machine learning approaches for the analysis of the Student Performance Dataset and the Student’s Academic Performance Dataset, where the former was used to predict the classification of the latter.

Yadav et al. [12] used decision trees to predict learning outcomes and manage student enrollment. Decision trees were found to provide easily understood classification rules and accurate forecasts, which could help reduce the dropout rate by identifying students who needed special attention. However, the study identified a research gap where the weaknesses in the predicted dropout students were not obvious enough to allow quick actions for repair.

Mesarić et al. [13] developed a model that uses data on students’ achievement in high school, programs after completing their first year of study, and teacher recommendations to classify students into two categories based on academic achievement at the end of their first academic year. The model was successful in identifying factors impacting student progress and accurately categorizing students based on academic achievement.

Rastrollo-Guerrero et al. [14] used a combination of random forest and SMOTE oversampling to analyze students’ academic performance and engagement with a virtual learning environment. The study found that the random forest classifier was successful in predicting academic performance, particularly toward the end of the course presentation when more accurate data were available.

The study utilized pre-processing with the Spearman correlation method and LR, SVM, RF, and NN algorithms to predict students’ learning outcomes. The study evaluated the models using a composite of AUC-ROC, ACC, and F1 score as evaluation metrics [15].

Table 1. Notations used in this paper

Notation	Description
DTC	Decision Tree Classifier (Tree)
ETC	Extra Tree Classifier (Tree)
ETCE	Extra Trees Classifier (Ensemble)
KNC	K-Neighbors Classifier (Neighbors)
RNC	Radius Neighbors Classifier (Neighbors)
RFCE	Random Forest Classifier (Ensemble)

There have been many investigations introduced in the past, and while they use the concept of machine learning to address the problem of predicting student achievement, there is still room for progress in the area of statistical analysis for classification using data, data conversion, and clarification models.

An explanation of the notations used in this research is given in Table 1.

3. Proposed methodology

3.1 *Workflow of our proposed model*

Figure 1 shows development workflow of the machine learning model. The data are initially gathered and transformed into comma-separated values. The actual results and grading system used for each course in the dataset are given in the Appendix section. Data that are partial, inaccurate, corrupted, or poorly formatted is processed. The complete academic statements of 1,000 students are included as a final record in the dataset. The next step is to generate labels or class columns from the entire academic result of a student. Thereby, the mean value of each category from fifteen is calculated and considered as the output column. Each value is grouped into the result category based on the range of the grade point. The performance category is divided into excellent, very good, good, weak, and bad sections. To perform the prediction analysis, Google Collaboratory, a web IDE for Python, was used in this research as an experimental environment. The grade point range and the performance category are represented in Table 2. A Label Encoder is applied to normalize each label for better performance.

The problem is a multiclass and multioutput problem because we must predict the values of multiple columns as outputs, each of which has multiple classes [16]. In this stage, the data are split into training and testing. 70% of data are used as training and 30% as testing.

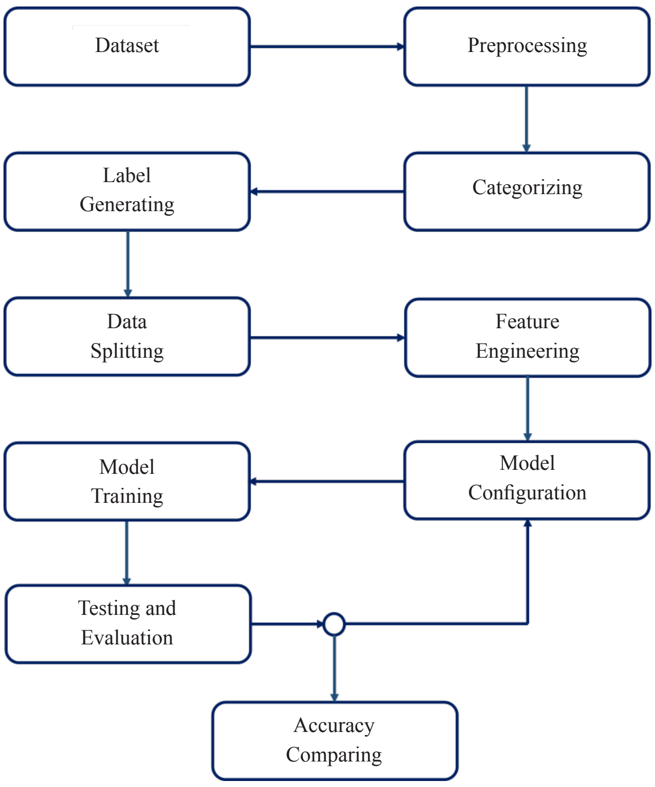


Figure 1. Pipeline for Developing Machine Learning Models

Table 2. Performance Category and the Range of Grade Points

Performance Category	Grade Points
Excellent	3.75 to 4.00
Very Good	3.50 to 3.74
Good	3.25 to 3.49
Average	3.00 to 3.24
Bad	< 3.00

Feature engineering was used to improve the model accuracy. It converts the unprocessed data into features that can be incorporated into this research's machine-learning methods to improve its accuracy. The dataset uses the four core feature engineering phases of feature creation, modification, extraction of features, and classification. The main component of a machine learning model used to solve a particular problem is the model configuration. We select a suitable machine learning technique to build the model to forecast our intended goals based on this difficulty. Since the problem is multiclass-multioutput classification, we applied the related different machine learning algorithms like the tree. Decision Tree Classifier (tree), Extra Tree Classifier (ensemble), Extra Trees Classifier (neighbors), K Neighbors Classifier (neighbors), Radius Neighbors Classifier (ensemble), Random Forest Classifier (ensemble). Now using the separated training and testing data, training, testing, and prediction are carried out. Finally, a comparison of all the evaluation metrics is carried out to evaluate each developed model.

3.2 *Dataset description*

The dataset used in this research contains academic records of 1,000 students from the Computer Science and Engineering Department. The dataset includes columns representing the result of each course taken by the students. From all the courses taught by the department, only those courses selected for our dataset have a direct impact on students' future professional market. The selected courses which acts as column names are: CSE-111, CSE-112, MAT-111, PHY-111, PHY-112, CEL-111, CSE-121, CSE-122, MAT-121, PHY-121, PHY-122, CEL-121, CSE-131, CSE-132, CSE-133, CSE-134, MAT-131, BDS-127, CSE-211, CSE-212, CSE-213, CSE-214, CSE-215, MAT-211, CSE-221, CSE-222, CSE-223, CSE-224, MAT-221, STA-221, CSE-231, CSE-233, CSE-234, CSE-235, CSE-236, IEC-231, CSE-311, CSE-312, CSE-313, CSE-314, CSE-315, BNA-311, CSE-321, CSE-322, CSE-323, CSE-324, CSE-325, CSE-327, CSE-331, CSE-332, CSE-333, CSE-334, CSE-335, CSE-336, CSE-337, CSE-338, CSE-411, CSE-412, CSE-413, CSE-415, CSE-416, CSE-417, CSE-483, CSE-489, CSE-491, and CSE-499.

The grading system used in the dataset ranges from 0.00 to 4.00, and letter grades are assigned based on a student's numerical grade. A+ is assigned to numerical grades of 80% and above, while F is assigned to numerical grades of less than 40%. The remaining letter grades (A, A-, B+, B, B-, C+, C, and D) are assigned to numerical grades that fall between these two extremes. The grading system which is used for our dataset of each course is presented in the Appendix section. Each row in the dataset represents a student's academic performance throughout their study in the department. Simply put, in our dataset, the columns represent the courses that were considered for our research, and each row represents the result of those courses for each student. After processing and refining the incomplete, inaccurate, corrupted, or poorly formatted data, a total of 1,000 student records are included in our final dataset. This dataset was used to develop a machine learning model to predict the performance of the department's students and identify areas that need improvement.

3.3 *Evaluation metrics*

Evaluation metrics are used for the predicted values to determine the effectiveness of the trained models. Different measurements independently reflect their assumptions on the difficulty. It is crucial to use a variety of measures to validate the results. In this instance, we have chosen to use the accuracy, precision, recall, and F1-score evaluation criteria to compare each estimator. Values for each metric are calculated using the predictions' confusion matrix. The ratio of the number of accurate predictions to the entire sample size is the accuracy. Equation 1 lays out the accuracy formula in detail. For values of accurately predicted event values, TP stands for true positives.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

False positives (FP) are values of wrongly predicted event values. For values of successfully anticipated no-event, TN stands for true negative. For values of no-event that were mistakenly anticipated, FN stands for false negative. A measurement called precision given in Equation 2 shows what proportion of all positive predictions is truly positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

The recall in Equation 3 is an assessment statistic that shows what proportion of all positive results was projected to be positive. It is equivalent to TPR (true positive rate).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

The harmonic mean between recall and precision is provided by the F1-score given in Equation 4. It is a statistical metric used to evaluate performance. Finally, compared to accuracy performance measurements, the confusion matrix, precision, recall, and F1 score present more useful insights into the prediction [17].

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

3.4 *Algorithms*

The working principles and a short summary of the algorithms used for this research are included below:

3.4.1 *Decision tree classifier*

A Decision Tree Classifier is a popular supervised learning algorithm used in classification problems. It works by partitioning the data into subsets based on the most significant attributes, using a tree-like model of decisions and their possible consequences. The algorithm iteratively splits the data into smaller subsets based on the attribute that provides the best split. This process continues until a stopping criterion is met, such as reaching a maximum tree depth or having no more attributes to split on. It can handle both categorical and numerical data, and it is capable of handling non-linear relationships between features. However, it is prone to overfitting, especially if the tree is too deep or the data is noisy. To mitigate this issue, pruning techniques such as cost-complexity pruning or minimum description length pruning can be used. Overall, the Decision Tree Classifier is a useful and widely used algorithm in machine learning, particularly for its simplicity and interpretability [18].

Information Gain is used to calculate the degree of homogeneity in the data set before and after the split. It is calculated using the following formula:

$$\text{Information Gain} = \text{Entropy(parent)} - [\text{Weighted Average}]\ \text{Entropy(children)} \quad (5)$$

Entropy is a measure of the degree of randomness or impurity in the dataset. It is calculated using the following formula:

$$\text{Entropy}(S) = -p_1\log_2(p_1) - p_2\log_2(p_2) - \ldots - p_n{*}\log_2(p_n) \quad (6)$$

Where p_1, p_2, ..., p_n are the proportion of examples in the data set that belong to each class.

Gini Impurity is another measure of impurity or degree of randomness in the dataset. It is calculated using the following formula:

$$\text{Gini}(S) = 1 - p_1^2 - p_2^2 - \ldots - p_n^2 \quad (7)$$

Where p_1, p_2, ..., p_n are the proportion of examples in the dataset that belong to each class. Pruning is a technique used to avoid overfitting in decision tree algorithms. It involves removing branches that do not improve the accuracy of the model on the test data [19].

3.4.2 *Extra Tree (Tree) classifier*

Extra Tree Classifier is an ensemble-based decision tree algorithm that randomly selects a subset of features at each node and creates a decision tree. The algorithm creates a forest of decision trees, where each tree is built on a random subset of data and features. The output of the Extra Tree Classifier is the class label that is most frequently predicted by the individual trees [20]. The algorithm works as follows:

I. Randomly select a subset of features at each node.

II. Split the node using the feature that provides the best split according to a random threshold.

III. Build a tree using the above two steps.

IV. Repeat the above steps to build multiple trees.

V. Aggregate the output of the trees to make a final prediction.

The Extra Tree Classifier equation is similar to the Decision Tree Classifier equation, but with an added step of randomization:

$Gain(T, F) = H(T) – H(T_F)$ (8)

Where T is the training set, F is the feature used to split the node, $H(T)$ is the entropy of the parent node, $H(T_F)$ is the entropy of the child nodes resulting from the split, and Gain is the information gained from the split.

3.4.3 *Extra Trees (Ensemble) classifier*

Extra Trees, also known as Extremely Randomized Trees, is an extension of the Random Forest algorithm. It is also an ensemble learning method based on decision trees, where a set of decision trees is built and combined to improve the overall performance of the model. In Extra Trees, each decision tree is built using a random subset of features and thresholds for splitting, making the algorithm less sensitive to noisy features and reducing overfitting. The final prediction of the Extra Trees model is the average of the predictions of all the decision trees [21].

The Extra Trees algorithm can be described mathematically as:

Given a dataset D, where D = $\{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$ where X_i is a vector of features and y_i is the corresponding class label, the algorithm can be summarized as follows:

I. Randomly select a subset of features and a threshold for splitting.

II. Split the data into two groups based on the selected feature and threshold.

III. Repeat steps 1 and 2 recursively until the stopping criterion is met (e.g., the maximum depth of the tree).

IV. Build multiple trees using bootstrapping to obtain different subsets of the original data.

V. Predict the class label for a new instance by taking the average of the predictions of all the trees.

The Extra Trees algorithm is a powerful and widely used classification algorithm, particularly in the fields of image and speech recognition, where it has demonstrated excellent performance.

3.4.4 *K-Neighbors (Neighbors) classifier*

K-Nearest Neighbors (K-NN) is a non-parametric and lazy learning algorithm used for classification and regression tasks. In the K-NN algorithm, the data points are classified based on the class of their K nearest neighbors. The value of K is set by the user [22-23]. The K-NN algorithm is simple to understand and implement. It does not make any assumption about the distribution of the data and can be used for both binary and multi-class classification tasks. However, the algorithm can be computationally expensive and may suffer from the curse of dimensionality. For classification, K-NN can be used as:

I. Calculate the distance between the test instance and all training instances

II. Select the K nearest instances based on distance

III. Determine the class of the test instance based on the majority class of its K nearest neighbors

3.4.5 *Radius Neighbors (Neighbors) classifier*

The Radius Neighbors Classifier is a machine-learning algorithm used for classification tasks. In this algorithm, each observation is classified based on the class of its neighbors within a fixed radius, rather than a fixed number of nearest neighbors as in the K-Nearest Neighbors algorithm. The radius is defined by the user, and any observation outside the radius is ignored.

The classification decision is based on the majority class of the neighbors within the radius. If there is a tie in the number of neighbors, then the class of the closest neighbor is chosen as the classification.

The equation for the Radius Neighbors Classifier is similar to that of the K-Neighbors Classifier, but instead of considering the k-nearest neighbors, it considers all neighbors within a fixed radius r. Here is the equation:

$$P(y|x) = \text{argmax } (1/|N\_r(x)|)\ \Sigma\_(x\_i, y\_i) \in N\_r(x)\ I(y\_i = y) \tag{9}$$

Where N_r(x) is the set of all neighbors within a radius r of observation x, and I(y_i = y) is the indicator function that takes the value 1 if the class of neighbor x_i is equal to y and 0 otherwise. The classifier predicts the class label of the observation x based on the majority class of its neighbors within the radius r [24].

3.4.6 *Random Forest (Ensemble) classifier*

Random Forest is a type of ensemble learning methods that uses multiple decision trees to make predictions. Each decision tree in the forest is built using a random subset of training data and a random subset of features. This randomness helps to reduce overfitting and improve the generalization ability of the model. During the prediction phase, each tree in the forest independently makes a prediction, and the final prediction is based on the majority vote of all the trees. During the training phase, multiple decision trees are built using different subsets of the data and features, and during the prediction phase, the output is the majority vote of all the decision trees [25]. Since in this research, the academic areas are divided into multiple categories, a multiclass-multioutput classification algorithm like the random forest classifier is used to make predictions. The results show that the random forest classifier provides the best accuracy.

4. Results and discussion

Based on the pipeline for developing machine learning models shown in Figure 1, each step is implemented in the Google Collaboratory using Python 3 programming language. The courses are divided into fifteen categories. Each category's performance is broken down into five sections. Excellent, Very Good, Good, Average, and Bad are the titles of the sections. The situation of each category is clearly visible from Figure 2 to Figure 16. Figure 2 shows the percentage of students in the Information System (IS) category. Only 14.10% of students are doing excellent in this section but 20.00% are weak and 27.00% are in bad. So, the department should emphasize this section. In Figure 3, the percentage of students in Network Design (ND) is shown. In this section, 12.30% of students are in excellent areas but 24.90% of students are in bad areas. The situations of this group are simply understandable. In Figure 4, the visualization of Theoretical Programming (TP) is given. In this group, a sizable 26.40% of students fall into the bad category.

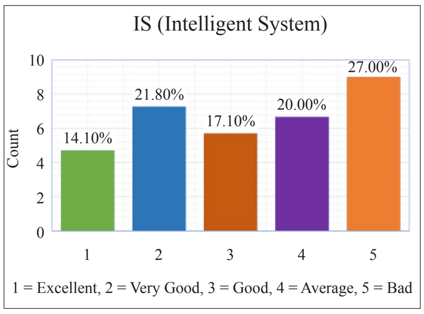


Figure 2. Percentage of Students in IS Category

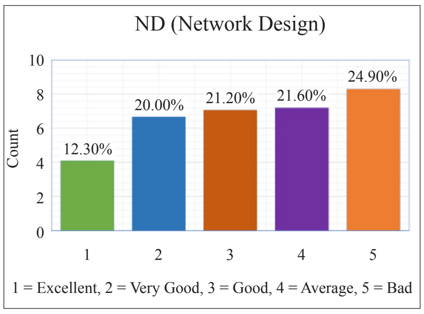


Figure 3. Percentage of Students in ND Category

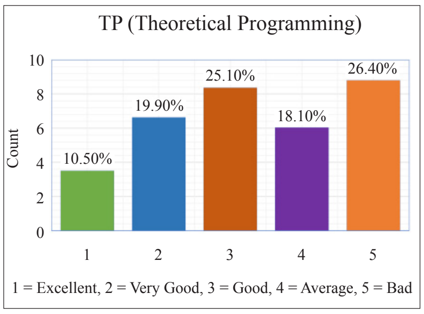


Figure 4. Percentage of Students in TP Category

The visualization of experimental programming (EP), logic building (LB), and software design (SD) is illustrated in Figures 5, 6, and 7. The represented visualizations make it simple to understand the department's performance in these groups.

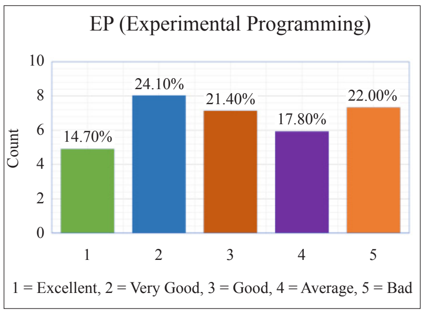


Figure 5. Percentage of Students in EP Category

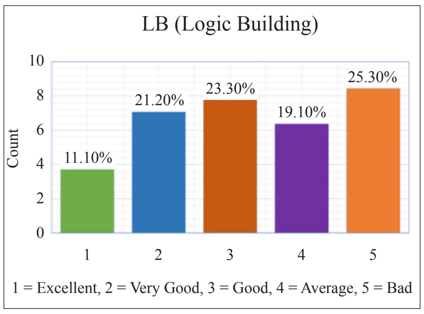


Figure 6. Percentage of Students in LB Category

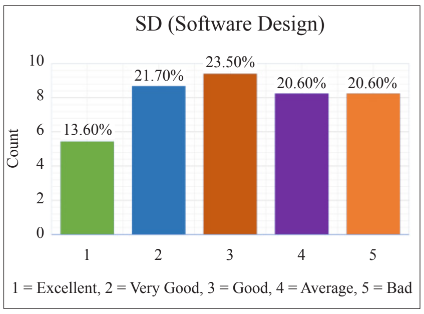


Figure 7. Percentage of Students in SD Category

The illustrations of Experimental Software Design (ESD), System and Hardware (SH), and Experimental Hardware (EH) are presented in Figures 8, 9, and 10. The graphical illustrations make it simple to determine the department's status in these groups.

The representations of vision and graphics (VG), numerical analysis (NA), and theoretical electrical electronics (TEE) are presented in Figures 11, 12, and 13. These sections' visual representations make it simple to comprehend the department's performance condition.

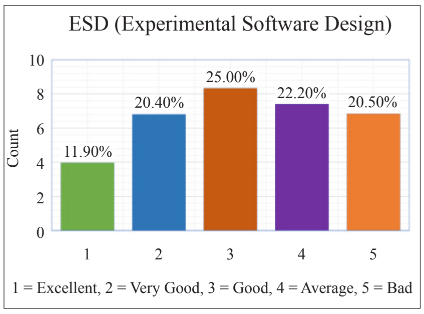


Figure 8. Percentage of Students in ESD Category

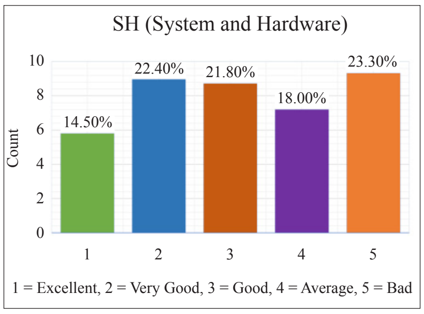


Figure 9. Percentage of Students in SH Category

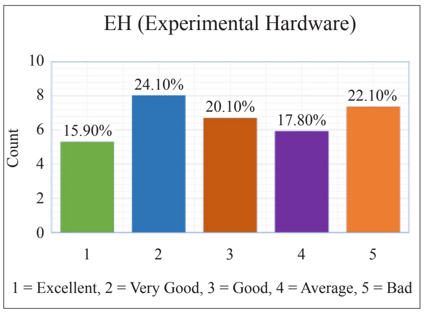


Figure 10. Percentage of Students in EH Category

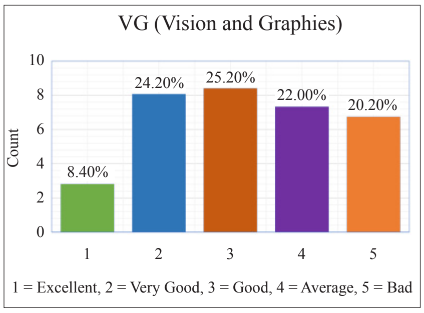


Figure 11. Percentage of Students in VG Category

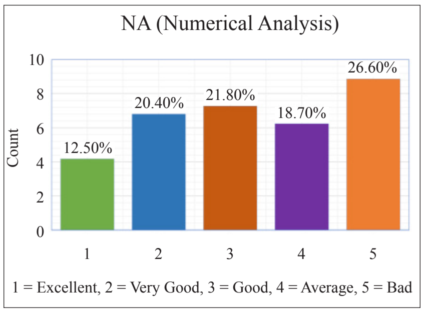


Figure 12. Percentage of Students in NA Category

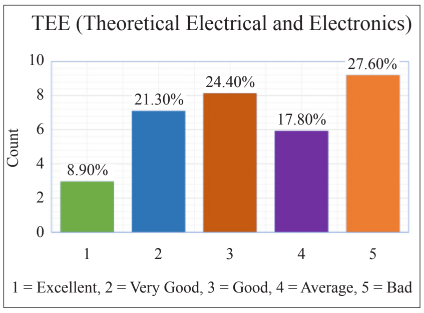


Figure 13. Percentage of Students in TEE Category

Figures 14, 15, and 16 demonstrate the depiction of Application Development (AD), Experimental Electrical and Electronics (EEE), and Experimental Vision and AI (EVI). The graphical infographics make it simple to understand how the department is still doing in each of these groups. In the AD division, 10.90% of students are in excellent areas, whereas more than 26% are in Bad areas. The percentage of students scoring Excellent or Very Good is higher in the EEE area. Additionally, 24.90% of students in the EVI category reside in a Bad region. Therefore, based on the aforementioned data, it is simple to identify the areas that need to be addressed to improve the department's performance.

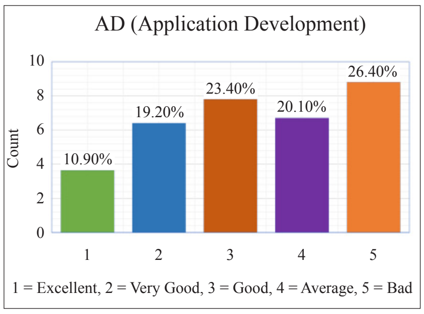


Figure 14. Percentage of Students in AD Category

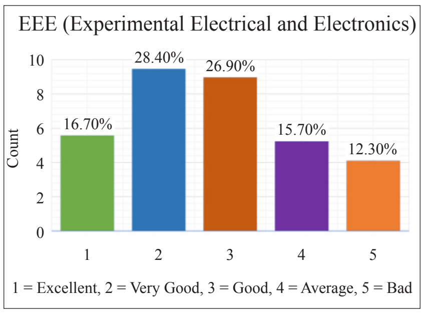


Figure 15. Percentage of Students in EEE Category

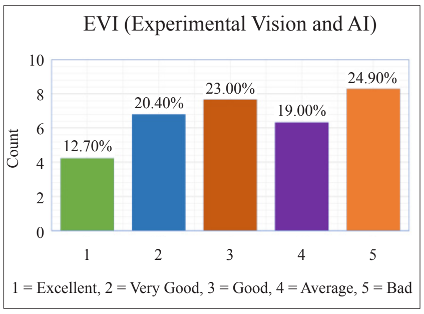


Figure 16. Percentage of Students in EVI Category

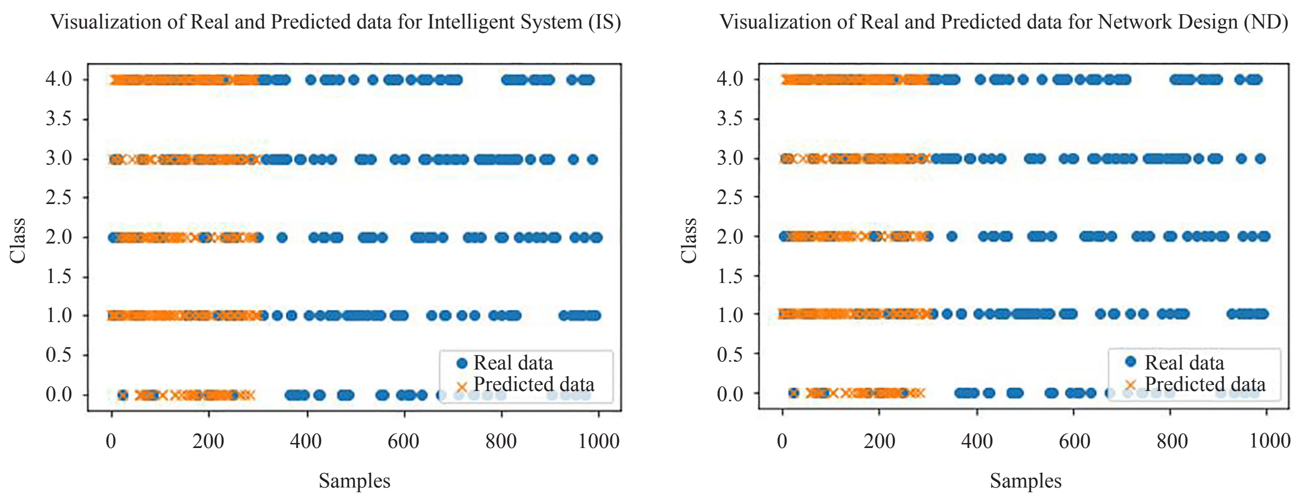


Figure 17. Visualization of the real and predicted data by Random Forest Classifier

Figure 17 shows the prediction results of a random forest classifier with real and prediction data on the used dataset. The prediction results are shown using a line chart with markers, where circles represent the real data and crosses represent the predicted data. The x-axis represents the samples, and the y-axis represents the class labels. In the figure, the predicted data points are generally close to the real data points. Overall, the prediction graph suggests that the random forest classifier is a promising method for predicting class labels in this dataset. From the fifteen categories, the first two categories Intelligent System (IS) and Network Design (ND) data are shown.

Table 3. Comparison of evaluation metrics of different used algorithms

Algorithm	Accuracy	Precision	Recall	F1-Score
DTC	0.916	0.948	0.948	0.948
ETC	0.922	0.951	0.951	0.951
ETCE	0.922	0.962	0.962	0.962
KNC	0.832	0.895	0.895	0.895
RNC	0.701	0.813	0.813	0.813
RFCE	0.940	0.963	0.963	0.963

Table 3 compares the evaluation metrics of several employed algorithms. The precision column in this table indicates how much one may rely on the classifier to accurately identify instances that belong to the positive class. A high accuracy rating indicates that there were few false positives and that the classifier's standards for categorizing something as positive are highly strict. Compared to other multiclass-multioutput machine learning algorithms utilized in the various established models, the Random Forest Classifier (Ensemble) has a greater level of precision. An algorithm with a high recall rate will often return the majority of pertinent results (whether or not ones that are irrelevant are likewise returned). The recall of the Random Forest Classifier (Ensemble) is also higher than other algorithms. The F1-core is an additional assessment metric. Such a maximum score of 1, the better the F1-score, which ranges from zero to one. F1-score for the developed models with multiclass-multioutput algorithms is also given in the table. F1-score of the Random Forest Classifier (Ensemble) is higher than other algorithms and it is almost 0.96 which is close to 1. It is evident from a comparison of the assessment metrics of several algorithms that the Random Forest Classifier (Ensemble) offers the highest level of accuracy for this research.

Figure 18 below compares model accuracy scores in terms of percentage. The model developed by the DTC algorithm accurately predicts the labels 91% of the time. The model created by ETC and ETCE has an accuracy rating of almost 92%, RNC is nearly 70%, RFCE is around 94%, and KNC is almost 83%. The RFCE has the highest accuracy score among the many models using multiclass-multioutput machine learning algorithms. Therefore, this model more accurately predicts the performance of the fifteen categories of these departments.

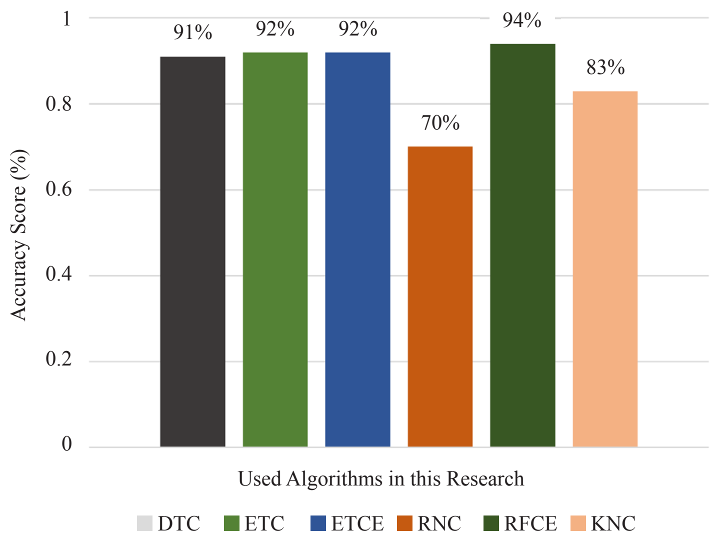
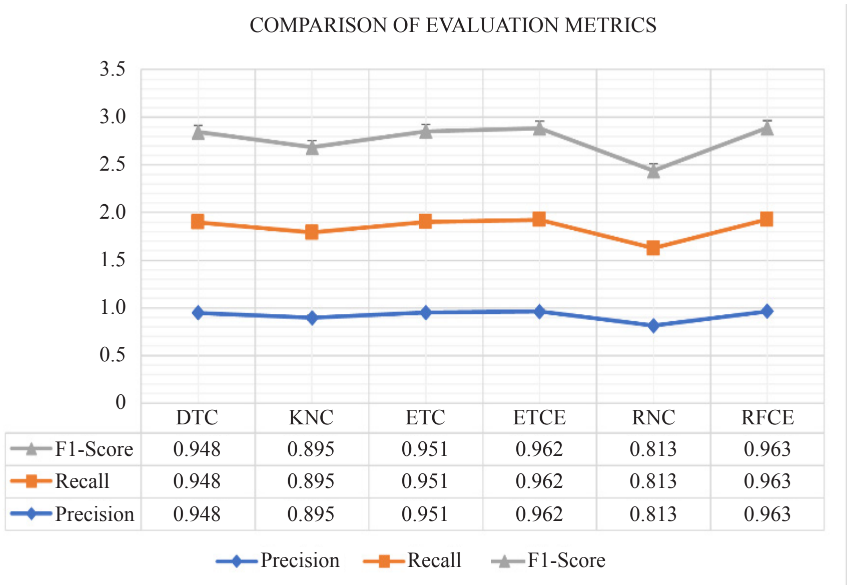


Figure 18. Model accuracy score in percentage (%)

Figure 19 below compares the Precision, Recall, and F1-Score evaluation metrics of the algorithms used in this research. In the X-axis, we have the utilized algorithms, which we have implemented in our experiment. In the Y-axis, we have the Precision, Recall, and F1-Scores for all of these implemented algorithms. From the figure, it's clear that the RFCE algorithm has the best score for all three metrics.

A comparison of the accuracy of the proposed model with the existing models is shown in Figure 20. In the existing machine learning model, various machine learning algorithms are used. Mesaric and Šebalj [13] proposed a model with the decision tree classifier and their accuracy was 70.54%. Hayder [26] performed a Comparative Study Between Classification Algorithms to predict student performance and proved that the model developed by support vector machine (SVM) provided an accuracy of 80.00%. Yağcı [2] designed a model with the random forest (RF) algorithm with an accuracy of 86.00%. After that, Sekeroglu et al. [11] proposed a model with Back Propagation (BP) and their accuracy was 87.78%. Su et al. [15] designed a model with Neural Network (NN) for the prediction of student performance with an accuracy of 81.30%. Karale et al. [27] proposed a machine learning and artificial intelligence-based model with the Random Forest (RF) algorithm to predict student performance with an accuracy of 80.29%. Sudais et al. [1] proposed another model for the prediction of student academic performance with the Naive Bayes (NB) algorithm with an accuracy of 63.70%. So, from the various existing model, the proposed model in this research with Random Forest Ensemble classifier accurately predict the student performance with an accuracy of 94.00%.



	DTC	KNC	ETC	ETCE	RNC	RFCE
F1-Score	0.948	0.895	0.951	0.962	0.813	0.963
Recall	0.948	0.895	0.951	0.962	0.813	0.963
Precision	0.948	0.895	0.951	0.962	0.813	0.963

Figure 19. Comparisons among the evaluation metrics of the used algorithms

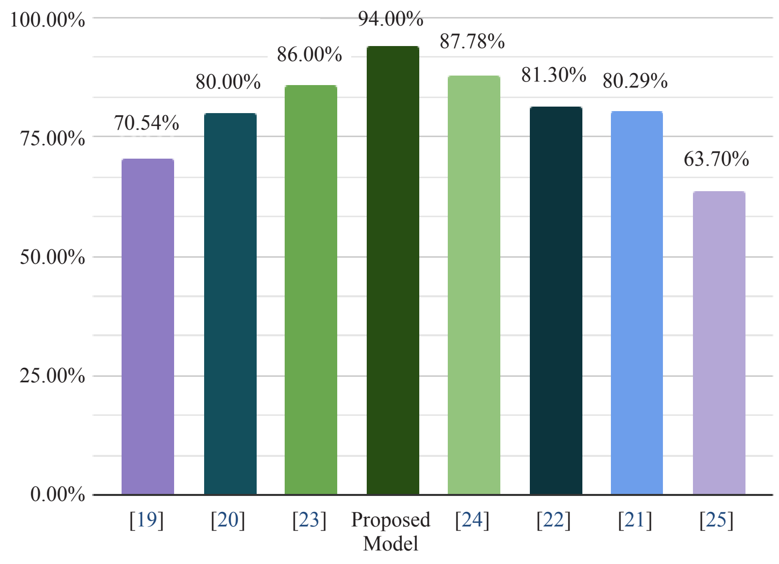


Figure 20. Comparison of the accuracy of the proposed model with the existing models

5. Conclusions

The performance of many university departments is improving because of technological improvements. It will be much simpler for them to make improvements if they are aware of the sector in which they need to improve. In this research, we have proposed and developed an efficient ML approach to address the strong and weak areas of computer science and engineering department students. Among multiple implemented ML algorithms, the Random Forest Classifier (Ensemble) model produces the highest performance in our research. This research will support the sustainability of this associated department in the global rankings. Additionally, by utilizing this model, other departments can increase their performance. Since this research can greatly improve any department's performance, it is important to keep in mind that other universities may have various performance standards, assessment standards, and grading systems. This model has to be updated in this situation. However, based on the model's accuracy, we may assume that it will be useful for both the research community and any other educational department.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Sudais M, Safwan M, Khalid MA, Ahmed S. *Students' academic performance prediction model using machine learning*. Research Square: Durham, NC, USA; 2022. Available from: https://doi.org/10.21203/rs.3.rs-1296035/v1 [Accessed 16th April 2023].

[2] Yağcı M. Educational data mining: Prediction of students' academic performance using machine learning algorithms. *Smart Learning Environments*. 2022; 9(1): 11.

[3] Dervenis C, Kyriatzis V, Stoufis S, Fitsilis P. Predicting students' performance using machine learning algorithms. *Proceedings of the 6th International Conference on Algorithms, Computing and Systems*. New York, NY, USA: ACM; 2022. p.1-7.

[4] Khan MAR, Akter J, Ahammad I, Ejaz S, Jaman Khan T. Dengue outbreaks prediction in Bangladesh perspective using distinct multilayer perceptron NN and decision tree. *Health Information Science & System*. 2022; 10(1): 32.

[5] Khan MAR, Afrin F, Prity FS, Ahammad I, Fatema S, Prosad R, et al. An effective approach for early liver disease prediction and sensitivity analysis. *Iran Journal of Computer Science*. 2023. Available from: https://doi.org/10.1007/s42044-023-00138-9.

[6] Khan AR, Uzzaman F, Ahammad I, Prosad R, Zayed-Us-salehin, Khan TZ, et al. Stock market prediction in Bangladesh perspective using artificial neural network. *International Journal of Advanced Technology and Engineering Exploration*. 2022; 9(95): 1397-1427.

[7] Poudyal S, Mohammadi-Aragh MJ, Ball JE. Prediction of student academic performance using a hybrid 2D CNN model. *Electronics*. 2022; 11(7): 1005.

[8] El-Halees AM. *Mining students data to analyze e-Learning behavior: A case study.* Computer Science, Education; 2009.

[9] Brahim GB. Predicting student performance from online engagement activities using novel statistical features. *Arabian Journal for Science and Engineering*. 2022; 47(8): 10225-10243.

[10] Tomasevic N, Gvozdenovic N, Vranes S. An overview and comparison of supervised data mining techniques for student exam performance prediction. *Computers & Education*. 2020; 143: 103676.

[11] Sekeroglu B, Dimililer K, Tuncal K. Student performance prediction and classification using machine learning algorithms. *Proceedings of the 2019 8th International Conference on Educational and Information Technology*. New York, NY, USA: ACM; 2019. p.7-11.

[12] Yadav SK, Bharadwaj B, Pal S. Mining education data to predict student's retention: A comparative study. *International Journal of Computer Science and Information Security*. 2012; 10(2): 113-117. Available from: https://doi.org/10.48550/arXiv.1203.2987.

[13] Mesarić J, Šebalj D. Decision trees for predicting the academic success of students. *Croatian Operational Research Review*. 2016; 7(2): 367-388.

[14] Rastrollo-Guerrero JL, Gómez-Pulido JA, Durán-Domínguez A. Analyzing and predicting students' performance by means of machine learning: A review. *Applied Sciences*. 2020; 10(3): 1042.

[15] Su Y-S, Lin YD, Liu TQ. Applying machine learning technologies to explore students' learning features and performance prediction. *Frontiers in Neuroscience*. 2022; 16: 1018005. Available from: https://doi.org/10.3389/fnins.2022.1018005.

[16] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*. 2011; 12: 2825-2830.

[17] Luque A, Carrasco A, Martín A, de las Heras A. The impact of class imbalance in classification performance metrics based on the binary confusion matrix. *Pattern Recognition*. 2019; 91: 216-231.

[18] Dai QY, Zhang CP, Wu H. Research of decision tree classification algorithm in data mining. *International Journal*

of Database Theory and Application. 2016; 9(5): 1-8.

[19] Jijo BT, Abdulazeez AM. Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*. 2021; 2(1): 20-28.

[20] Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Machine Learning*. 2006; 63(1): 3-42.

[21] Thankachan K. *What? When? How?: ExtraTrees Classifier*. Towards Data Science; 2022. Available from: https://towardsdatascience.com/what-when-how-extratrees-classifier-c939f905851c [Accessed 15th January 2023].

[22] Harrison O. *Machine Learning Basics with the K-Nearest Neighbors Algorithm*. Towards Data Science; 2018. Available from: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 [Accessed 25th February 2023].

[23] Sun J, Du W, Shi N. A survey of kNN algorithm. *Information Engineering and Applied Computing*. 2018; 1(1): 1-10. Available from: https://doi.org/10.18063/ieac.v1i1.770.

[24] Brownlee J. *Radius Neighbors Classifier Algorithm With Python*. Machine Learning Mastery; 2020. Available from: https://machinelearningmastery.com/radius-neighbors-classifier-algorithm-with-python/ [Accessed 25th February 2023].

[25] Shaik AB, Srinivasan S. A brief survey on random forest ensembles in classification model. In: Bhattacharyya S, Hassanien A, Gupta D, Khanna A, Pan I. (eds.) *International Conference on Innovative Computing and Communications. Lecture Notes in Networks and Systems, vol 56*. Springer, Singapore; 2018. p.253-260. Available from: https://doi.org/10.1007/978-981-13-2354-6_27.

[26] Hayder A. *Predicting student performance using machine learning: A comparative study between classification algorithms*. Mid Sweden University. Dissertation; 2022. Available from: http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-45356.

[27] Karale A, Narlawar A, Bhujba B, Bharit S. Student performance prediction using AI and ML. *International Journal for Research in Applies Science and Engineering Technology*. 2022; 10(6): 1644-1650.

Appendix A

Categorization of courses

1. Intelligent system (IS)
   a. Artificial Intelligence (CSE-415)
   b. Software Engineering (CSE-417)
   c. Machine Learning (CSE-483)
   d. Digital Image Processing (CSE-489)
   e. Data Mining (CSE-491)
2. Network Design (ND)
   a. Computer Fundamentals (CSE-111)
   b. Data Communication (CSE-327)
   c. Computer Networking (CSE-333)
   d. Computer Networking Laboratory (CSE-334)
   e. Cryptography and Network Security (CSE-413)
3. Theoretical Programming (TP)
   a. Structured Computer Programming with C (CSE-121)
   b. Object Oriented Programming with C++ (CSE-131)
   c. Data Structure (CSE-211)
   d. Object Oriented Programming II-Java (CSE-221)
   e. Design and Analysis of Algorithms (CSE-235)
4. Experimental Programming (EP)
   a. Structured computer Programming Lab (CSE-122)
   b. Object Oriented Programming I Lab (CSE-132)
   c. Object Oriented Programming II Lab (CSE-222)
   d. Data Structure Lab (CSE-212)
   e. Design and Analysis of Algorithms Lab (CSE-236)
5. Logic Building (LB)
   a. Data Structures (CSE-211)
   b. Discrete Mathematics (CSE-215)
   c. Digital Logic Design (CSE-223)
   d. Design and Analysis of Algorithms (CSE-235)
   e. Theory of Computation (CSE-315)
6. Software Design (SD)
   a. Engineering Drawing (IEC-231)
   b. Operating Systems (CSE-313)
   c. System Analysis and Design (CSE-325)
   d. Compiler Design (CSE-335)
   e. Software Engineering (CSE-417)
7. Experimental Software Design (ESD)
   a. Object Oriented Programming II Lab (CSE-222)
   b. Database Management System Lab (CSE-312)
   c. Operating Systems Lab (CSE-314)
   d. Web Engineering Lab (CSE-322)
   e. Compiler Design Lab (CSE-336)
8. System and Hardware (SH)
   a. Digital Logic Design (CSE-223)
   b. Computer Architecture and Organization (CSE-231)
   c. Microprocessors and Assembly Language (CSE-323)
   d. System Analysis and Design (CSE-325)

e. Digital System Design (CSE-331)

9. Experimental Hardware (EH)
   a. Computer Fundamentals lab (CSE-112)
   b. Digital Logic Design Lab (CSE-224)
   c. Microprocessors and Assembly Language Programming Lab (CSE-324)
   d. Digital System Design Lab (CSE-332)
   e. Computer Peripherals and Interfacing Lab (CSE-338)

10. Vision and Graphics (VG)
    a. Structured Computer Programming with C (CSE-121)
    b. System Analysis and Design (CSE-325)
    c. Computer Peripherals and Interfacing (CSE-337)
    d. Computer Graphics (CSE-411)
    e. Digital Image Processing (CSE-489)

11. Numerical Analysis (NA)
    a. Differential Equation and Numerical Analysis (MAT-121)
    b. Linear Algebra and Matrix Analysis (MAT-131)
    c. Coordinate Geometry and Vector Analysis (MAT-211)
    d. Complex Variable and Transform Mathematics (MAT-221)
    e. Probability and Statistics (STA-221)

12. Theoretical Electrical and Electronics (TEE)
    a. Circuit Analysis (CSE-133)
    b. Electronics (CSE-213)
    c. Digital Electronics and Pulse Technique (CSE-233)
    d. Basic Physics (PHY-111)
    e. Advanced Physics (PHY-121)

13. Application Development (AD)
    a. Object Oriented Programming with JAVA (CSE-221)
    b. Database Management System (CSE-311)
    c. Web Engineering (CSE-321)
    d. Software Engineering (CSE-417)
    e. Thesis/Project Work (CSE-499)

14. Experimental Electrical and Electronics (EEE)
    a. Circuit Analysis Lab (CSE-134)
    b. Electronics Lab (CSE-214)
    c. Digital Electronics and Pulse Technique Lab (CSE-234)
    d. Basic Physics Lab (PHY-112)
    e. Advanced Physics Lab (PHY-122)

15. Experimental Vision and AI (EVI)
    a. Structured computer Programming Lab (CSE-122)
    b. Compiler Design Lab (CSE-336)
    c. Computer Peripherals and Interfacing Lab (CSE-338)
    d. Computer Graphics Lab (CSE-412)
    e. Artificial Intelligence Lab (CSE-416)

Grading system used in the dataset

Table 4. Grading system used in this research's dataset

Numerical Grade	Letter Grade	Grade Point
80% and above	A+	4.00
75% to less than 80%	A	3.75
70% to less than 75%	A-	3.50
65% to less than 70%	B+	3.25
60% to less than 65%	B	3.00
55% to less than 60%	B-	2.75
50% to less than 55%	C+	2.50
45% to less than 50%	C	2.25
40% to less than 45%	D	2.00
Less than 40%	F	0.00