

A classification model for predicting course outcomes using ensemble methods

Emad Al-Momani¹, Ala'a Shatnawi², Mohammed Almomani², Ammar Almomani^{3,4},
Mohammad Alauthman⁵

¹Department of Industrial Engineering, Al Hussein Technical University, Amman, Jordan

²Department of Industrial Engineering, Jordan University of Science and Technology, Irbid, Jordan

³Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Irbid, Jordan

⁴Department of Computer Information Science, Higher Colleges of Technology, Sharjah, United Arab Emirates

⁵Department of Information Security, Faculty of Information Technology, University of Petra, Amman, Jordan

Article Info

Article history:

Received Apr 2, 2024

Revised Jul 25, 2024

Accepted Aug 14, 2024

Keywords:

Academic performance

Data mining

Ensemble methods

K-nearest neighbors

Machine learning

Synthetic minority over

sampling technique

Support vector machine

ABSTRACT

Educational data mining has sparked a lot of attention in latest years. Many machine learning methods have been suggested to discover hidden information from educational data. The extracted knowledge assists institutions in enhancing the effectiveness of teaching tactics and the quality of education. As a result, it improves students' performance and educational outputs overall. In this paper, a classification model was built to classify students' grades in a specific course into different categories (binary and multi-level classification tasks). The dataset contains features related to academic and non-academic information. The models were built using a variety of machine learning algorithms: decision tree (J48), support vector machine (SVM), and k-nearest neighbor (K-NN). Furthermore, ensemble methods (bagging, boosting, random subspace, and random forest) which combined multiple decision tree classifiers were implemented to improve the models' performance. The data set was modified under two stages: features selection method and data augmentation using a method called synthetic minority over sampling technique (SMOTE). Based on the results of the experiments, it is possible to predict the students' performance successfully by using machine learning algorithms and ensemble methods. Random subspace obtained the best accuracy at two-level classification task with modified data with 91.20%. At the three-level classification task, the best accuracy was obtained by random forest with 87.18%.

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



Corresponding Author:

Emad Al-Momani

Department of Industrial Engineering, Al Hussein Technical University

Amman, Jordan

Email: emad.almomani@htu.edu.jo

1. INTRODUCTION

Data mining techniques are widely used in the education field. This is attributed to the availability of educational data in universities about their students. Educational data mining (EDM) is an application of data mining techniques in the education field that integrates students' data with machine learning algorithms to extract unobserved information and patterns from education databases [1]–[4].

EDM incorporates various users groups, and these users apply the information discovered by EDM based on their claim vision and data mining (DM) objectives [5]. The discovered knowledge can assist the instructors to improve teaching methods [6], to identify students who are expected to fail [7], and students could use it to make a proper course scheduling [6], [8]. It also assists the administration to make appropriate

decisions to improve the education quality [9]. EDM can utilize a variety of DM techniques. Such as, classifications, which is the most widely used technique for predicting students' performance [10]. Several algorithms are under classification such as decision tree [11], support vector machine (SVM) [12], and k-nearest neighbor (K-NN).

The state-of-the-art in predicting student performance using machine learning focuses on leveraging academic and demographic features to forecast grades or identify at-risk students. Key recent contributions include using ensemble methods to predict final exam scores [13], employing educational data mining models with bagging to enhance accuracy [14], and combining filtering techniques like synthetic minority over sampling technique (SMOTE) with boosting to improve performance [15]. However, challenges remain in incorporating course difficulty and semester workload features, comparing a range of individual and ensemble machine learning (ML) models, and providing both binary and multi-class predictions. This study aims to address these gaps by i) introducing novel course difficulty and academic load features, ii) evaluating multiple individual and ensemble ML techniques on both original and augmented datasets, and iii) developing models for 2-class and 3-class prediction tasks. The main contributions are the novel features engineered, the extensive comparisons of ML approaches, and the multi-level, course-specific models developed. The following sections detail the methodology, present results comparing model performance, discuss implications of the findings, and summarize conclusions.

Students' performance varies in different courses and semesters. The performance could go up or down, which will affect the average of the students' marks. No system shows the students or the institutions how the performance situation is routing. A tracking system provides students and educational institutions with advanced knowledge of students' academic achievement in specific courses and their likelihood of success or failure. As a result, there is a need for a mechanism that allows students to assess their own performance. This will also enable institutions to take proactive measures to improve their performance.

This paper introduces classification models (binary and multi-level tasks) for future course marks through previous data such as preceding courses grades. The educational dataset is collected from Jordan University of Science and Technology (JUST). The semester difficulty feature (SDF) will be introduced for the first time to the best of our knowledge. Since the models were focused on predicting the students' performance in a single course, SDF will represent the total degree of difficulty imposed by other courses taken by the students in the same semester.

We will use three individual machine learning algorithms to build the classification models: decision tree, support vector machine, and k-nearest neighbor. Then we will apply ensemble methods which combine multiple decision tree classifiers to improve the models' performance. The ensemble methods are bagging, boosting, random subspace and random forest. Moreover, the data set will be modified using features selection method and oversampling technique. Furthermore, the performance of the selected methods will be evaluated and compared using different evaluation metrics, including accuracy, precision, recall, F-measure and area under receiver operating characteristic curve (AUC).

This paper is organized as follows: section 2 introduces the related works in the field of educational machine learning algorithms. Section 3 describes our research methodology. Section 4 discusses experimental evaluation and results, and section 5 presents our conclusions.

2. RELATED WORKS

In EDM, predicting students' performance is an important application. Several machine learning algorithms and ensemble methods are used to build a predictive model. Research by Pereira and Zambrano [16] utilized decision tree (DT) to study student dropouts across various students who are studying bachelor's level at Nariño University. 6,870 students' records were collected. 31 features that are related to social, economic, and academic factors were studied. The results of the study revealed that low grades, number of previously failed courses, department of studies, and distance from the university were the most influential features.

Jain [17] analyzed and classified students' grades into three categories: those who passed, those who failed, and those who dropped out, depending on various factors. Demographics and learning features were studied like age, gender, and learning elements like previous assessment grades are concerned. A publicly available dataset was used. Various machine learning algorithms such as artificial neural networks (ANN), random forests, decision trees, XGBoost, and support vector machines were used. Accuracy was considered as a measure of the models' performance. ANN achieved the best result with a 78.08 accuracy.

Recently, Ahmad *et al.* [13] presented a model that aims to forecast university students' achievement in final exams. They used various machine learning methods including support vector machine, logistic regression, naive Bayes, and gradient boosted trees to build their predictive model. The study compared the performance of these different algorithms in predicting student outcomes, providing insights into which methods were most effective for this educational data mining task. Additionally, their research highlighted

the potential of machine learning techniques in identifying at-risk students early, allowing for timely interventions to improve academic performance.

Also, Ragab *et al.* [14] developed an educational data mining (EDM) model. The model employed separate datasets to represent the student's interaction with the instructive model. They applied a variety of classifiers, such as logistic regression, naive Bayes tree, artificial neural network, support vector system, decision tree, and k-nearest neighbor. In addition, ensemble methods were implemented (boosting, random forest, and bagging). The models' results showed that the bagging method clearly improved with the DT model. It enhanced the accuracy of the individual decision tree method. The accuracy became 91.4% instead of 90.4%.

Ashraf *et al.* [15] applied individual machine learning algorithms such as J48, KNN, and Naive Bayes algorithms to classify students' grades into three intervals. After that, boosting ensemble methods and filtering techniques such as SMOTE were implemented. The best accuracy among individual machine learning was from naïve Bayes method with 95.5%. After applying SMOTE, the accuracy was enhanced to 97.15%. Applying Boosting also enhanced the accuracy for all classifiers.

Muchuchuti *et al.* [18] presented a comparative analysis of various classification algorithms that aims to aid in predicting and improving students' performance. A total of 124 students with 9 features were used. Five classifiers were implemented using WEKA software. The academic features during year 1 and year 2 were used to predict the final class grade of the student at the end of year 2. Different feature ranking methods were used to rank the features according to their effect on the output. The naive Bayesian algorithm was the most effective method.

In conclusion, various studies have been conducted to predict students' achievement using machine learning and ensemble techniques. To our knowledge, no studies have highlighted the semester difficulty feature (SDF) and whether it impacts academic success or failure. Furthermore, the extracted knowledge will allow students to evaluate their performance and institutions to take early action to help them improve their performance. Also, the model can be applied to various courses in the future.

3. METHOD

The applied methodology steps are visualized in Figure 1. The dataset was collected from the admission and registration department at Jordan University of Science and Technology. It contained records for 454 undergraduate industrial engineering students who graduated with a B.A. degree between 2018-2020. After data cleaning to remove 20 records with missing values, the final dataset consisted of 434 student records with 19 features. The experimental procedure involved: i) data preprocessing including cleaning, feature engineering, modification of dependent variables, and applying SMOTE oversampling, ii) implementing individual machine learning models (J48 decision tree, naïve Bayes, SVM) and ensemble methods (bagging, AdaBoost, random forest, and random subspace) in WEKA with 10-fold cross validation, and iii) evaluating models using accuracy, precision, recall, F-measure and ROC AUC metrics. Feature selection was performed using information gain to rank features by importance. SMOTE oversampling balanced the class distributions and increased total records to 1,091 for the 2-class dataset and 1,108 for the 3-class dataset. The individual and ensemble models were implemented with default hyperparameter settings in WEKA to allow fair comparisons. 10-fold cross-validation was selected as a robust evaluation approach. This procedure aimed to comprehensively and fairly assess multiple modelling approaches on original and augmented datasets for binary and multi-class prediction tasks.

3.1. Data collection and preprocessing

The data set for this study was obtained from Jordan University of Science and Technology (JUST), serving as the primary source of information for our analysis. The university's admission and registration department provided comprehensive data for 434 undergraduate students who had already graduated with a B.A. degree in industrial engineering over the past three years (2018-2020), covering first, second, and summer semesters. This rich dataset encompassed a wide range of academic and non-academic information, including students' grades in prerequisite courses, grade point averages (GPAs), demographic details, and other relevant factors that could potentially influence their performance in the target course. The collection of data from multiple academic years and semesters ensured a diverse and representative sample, allowing for a more robust analysis of factors affecting student performance in the operations research two course.

3.2. Description of the dataset

The dataset includes previous information about industrial engineering students that allow predicts students' performance in a specific course before taking it officially in the next semester. To predict the grade for the student in a future course (operations research two). At first, we looked at the student's grades in a

series of courses (preceding courses) that must be completed before enrolling in operations research two (OR2). The number of preceding courses in our case study is five. The course preceding course number five (OR1) is the primary prerequisite for the OR2 course. Other features were taken, such as those listed in Table 1. The dataset consists of 19 independent variables. The dependent variable is the student grade in the (OR2) course, as shown in Table 2. JUST has the following grading system: A- (3.75), A (4.0), A+ (4.2), B+ (3.5), B (3.25), B- (3.0), C+ (2.75), C (2.5), C- (2.25), D+ (2.0), D (1.75), D- (1.5), and F. (0.5).

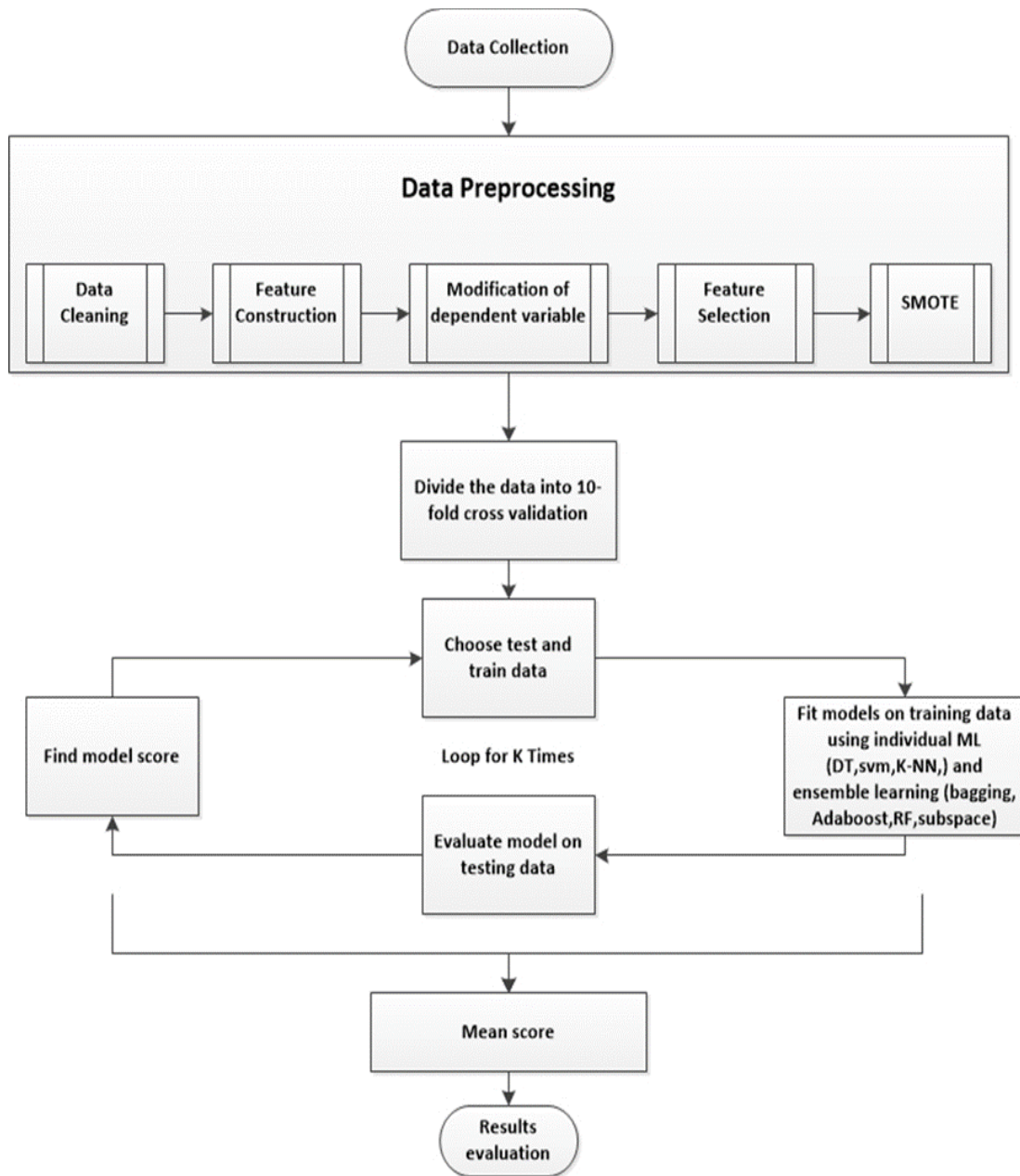


Figure 1. The classification model steps

3.3. Data preprocessing

Data preprocessing is a crucial step that precedes the application of specific machine learning techniques [19]. This process includes data cleaning, feature construction, dependent variable modification, feature selection, and data augmentation using techniques such as SMOTE [20], [21]. These steps are essential for preparing the raw data, optimizing the dataset for machine learning algorithms, and potentially improving model performance and prediction reliability for student academic outcomes.

Table 1. Student features (independent variables) description for the dataset

Sort	Feature	Description	Domain
1	Academic load	The total number of hours (courses load) that the student took in conjunction with the OR2 course over the full load semester	(Numeric: 0.5-1.166)
2	Living place	Living place of the students; how far is the place of residence from the university?	((Nominal: 1- Irbid (Distance ≤ 20 km), 2- Near cities ($20 < \text{Distance} \leq 60$ km), 3- Far cities (Distance > 60 km))
3	Semester difficulty	Total courses' difficulty	(Numeric: 10.92-70.04)
4	Funding source	Who is paying the students' tuition?	(Binary: private fund or grant fund)
5	Total income	Total family income (JD)	((Numeric: 1- (0-500) JD., 2- (500-1000) JD., 3- (equal or more than 1000 JD))
6	Gender	(Female, Male)	(Binary; female or male)
7	Number of trials in preceding course # 1	Did the student pass each of the preceding courses from the first trial or not?	(Binary: 1- Passed for the first time, 2- Does not pass for the first time)
8	Number of trials in preceding course # 2		
9	Number of trials in preceding course # 3		
10	Number of trials in preceding course # 4		
11	Number of trials in preceding course # 5		
12	High school final year grade (Tawjehi)	Student's GPA in high school	(Numeric: 1- ≤ 80 , 2- (80-89), 3- (90-100))
13	High School degree country	Did the student graduate from Jordanian-high school or not?	(Binary: Jordan, others)
14	Grades in preceding course # 1	The grades in the preceding courses that led to the target course (OR2)	(Nominal: A-, A, A+, B+, B, B-, C+, C, C-, D+, D, and D-)
15	Grades in preceding course # 2		
16	Grades in preceding course # 3		
17	Grades in preceding course # 4		
18	Grades in preceding course # 5		
19	Previous GPA	The grade point average that the student reached before taking the target course (OR2)	(Numeric: 1.82- 4.11)

Table 2. Dependent variable description for the dataset

Dependent variable	Description	Domain
Grade in target course	The grades in the target course (OR2)	(Nominal: A-, A, A+, B+, B, B-, C+, C, C-, D+, D, D-, and F)

3.4. Data cleaning

Preprocessing activities such as data cleaning are key activities and applied on the data set to eliminate the records with missing values [22]. The dataset contained 20 records with missing values, and as a result, these records were eliminated. After eliminating the missing values, the number of students' records decreased to 434 records.

3.5. Features construction

Feature construction creates new significant features from raw data [21]. In this study, we developed two novel features: the semester difficulty feature and the academic load feature. These constructed features aim to capture important aspects of a student's academic context, potentially improving our models' predictive power.

3.6. Semester difficulty feature

To show how difficult the courses are in the semester, we refer to the course schedule that includes the OR2 course (target course) in addition to other courses in the same semester, after sorting these courses in a table divided into three semesters for previous years as shown in Table 3. The total average will be calculated for the whole average values of $(\bar{x}_1, \bar{x}_2, \bar{x}_3)$. \bar{x}_1 refers to grades average in the first semester, \bar{x}_2 refers to grades average in the second semester and \bar{x}_3 refers to grades average in the third semester.

According to Figure 2, the higher the total average mark of each course is, the course is easy and the corresponding value in the course difficulty value will be low and vice versa. For example, if the total average mark of the course is 4, then the corresponding value in the course difficulty value is 1. Otherwise, if

the total average mark of the course is 0.5, then the corresponding value in the course difficulty line is 8. The Equation that proves these values is as shown in (1).

$$y = -2x + 9 \tag{1}$$

To have an expressive number of the value, we relate it to the load for each course by multiplying the value by the number of hours. As a result, this summation of the course's difficulty is being found to get the semester difficulty value, as shown in Table 4.

Table 3. An example for course difficulty computation

Course difficulty computation					
	Grades average in semester 1 (\bar{x}_1)	Grades average in semester 2 (\bar{x}_2)	Grades average in semester 3 (\bar{x}_3)	Total grades average	Course difficulty value
Course 1	2.2	2.58	2.9	2.56	3.88
Course 2	1.25	1.99	3.1	2.11	4.78
Course 3	1.2	2.9	2.8	2.3	4.40

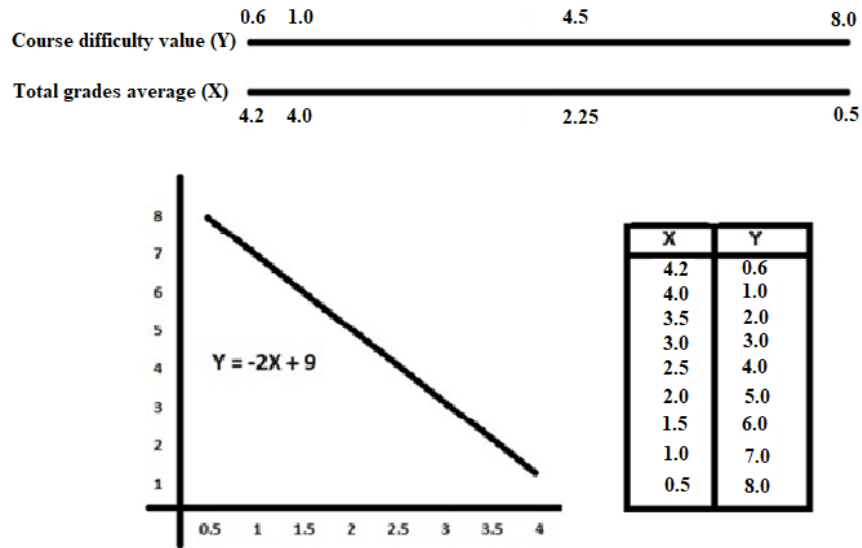


Figure 2. Course difficulty equation

Table 4. An example for semester difficulty feature computation

Semester difficulty computation										
Course 1 difficulty value (CD1)	Course hours (CH1)	Total course 1 difficulty value = CD1 * CH1	Course 2 difficulty value (CD2)	Course hours (CH2)	Total course 2 difficulty value = CD2 * CH2	Course 3 difficulty value (CD3)	Course hours (CH3)	Total course 3 difficulty value = CD3 * CH3	Semester difficulty	
Student 1	4.7	3	14.1	4.4	2	8.8	4.9	2	9.8	32.7
Student 2	4.4	3	13.2	4.5	1	4.5	4.8	3	14.4	32.1

3.7. Academic load feature

The academic load can be another factor to affect the students' academic performance in OR2 course. The total load number of hours in the semester that include OR2 course divided on the maximum allowed total hours will display the student's academic load. All of these details are shown in Table 5.

Table 5. An example for academic load feature computation

Student number	Semester type (full load in hours)	Semester load in that semester (hours)	Academic load (semester load/full load)
Student 1	Official semester (18.0)	12.0	0.7
Student 2	Summer semester (12.0)	6.0	0.5

3.8. Modification of dependent variables

To implement the classification task, we must convert the values of the dependent variable to nominal intervals. For the first data set, the dependent variable for the two-level classification task was converted to the binary variable, the value “fail” was converted to “1” and the value “pass” was converted to “2”. For the three-level classification task, the value “fail” was converted to “1”, the value “moderate” was converted to “2” and the value “good” was converted to “3”. Tables 6 and 7 show data set's two-level and three-level classification tasks.

Table 6. The two-level classification system for the first data set

Class	Description
1	Fail (F)
2	Pass (D-, D, D+, C-, C, C+, B-, B, B+, A-, A, A)

Table 7. The three-level classification system for the first data set

Class	Description
1	Fail (F)
2	Moderate (D-, D, D+, C-, C, C+)
3	Good (B-, B, B+, A-, A, A+)

3.9. Feature selection

This step aims to determine the important and appropriate groups of features which significantly affect the dependent variable. Irrelevant features do not have any significant effect on the dependent variable. The irrelevant features must be removed to enhance the model's performance [23]. The features were ranked using an information gain filter-based technique [24]. The ranking of the features will be shown from the most significant to the least important based on the value of information gained concerning dependent variables.

The features ranking for the dataset at two-level and three-level classification tasks are shown in Table 8. All the features that appear in the Table impact the dependent variable. Academic features such as (Previous GPA, preceding course grades, number of trials) and non-academic features such as (gender, living place and funding source and high school degree country) impact the dependent variable. The remaining features (semester difficulty, total income, high school GPA, and academic load) were excluded since they did not provide any information about the dependent variable (they did not have any rank).

Table 8. Features ranking for the dataset

First data set (two-level task)	Ranked	First data set (three-level task)	Ranked
Previous GPA	0.09929	Previous GPA	0.34605
preceding course 5 grades	0.07142	Preceding course 5 grades	0.3048
Preceding course 4 grades	0.06947	Preceding course 4 grades	0.28018
Preceding course 1 grades	0.06215	Preceding course 1 grades	0.16673
Preceding course 2 grades	0.04612	Preceding course 3 grades	0.12598
Trials in preceding course 5	0.03529	Preceding course 2 grades	0.12055
Trials in preceding course 3	0.03164	Trials in preceding course 4	0.05737
Preceding course 3 grades	0.0267	Trials in preceding course 3	0.05009
Trials in preceding course 4	0.02228	Trials in preceding course 5	0.04898
Gender	0.00865	Gender	0.03032
Living place	0.00551	Living place	0.02206
Trials in preceding course 1	0.00394	Trials in preceding course 1	0.02037
Funding source	0.00388	Funding source	0.00881
Trials in preceding course 2	0.00222	Trials in preceding course 2	0.00384
High School degree country	0.00026	High School degree country	0.00351

3.10. SMOTE technique

This step was done using the SMOTE [25]. SMOTE is an oversampling strategy that helps with the classes' imbalance issue. It is used to augment the data size and balance it [21]. For example, the number of students in the initial data at the two-level classification task was 67 for the first class, while the number of students in the second class was 367 as shown in Table 9. This big difference in the number of students among the different classes leads to good model's performance in the majority class, while it will be bad in the minority class. For this reason, we applied the SMOTE technique to augment the overall data size and

make the data balanced, therefore enhancing the models' performance at all classes. The data size was augmented and balanced for all classes at different tasks. As shown in Table 10.

Table 9. SMOTE at two-level for first data

Two-level task	Initial data		After SMOTE	
	Class 1 (fail)	Class 2 (pass)	Class 1 (fail)	Class 2 (pass)
Number of students	67	367	541	550

Table 10. SMOTE at three-level for first data

Three-level task	Initial Data			After SMOTE		
	Class1(fail)	Class2 (moderate)	Class3 (good)	Class1 (fail)	Class2 (moderate)	Class3 (good)
Number of students	67	185	182	374	370	364

3.11. Classification methods

The methodology starts with data collection, and this step follows with preprocessing steps to get the data ready for classification. Individual machine learning algorithms such as decision tree, k-nearest neighbor (K-NN) and support vector machine (SVM) were implemented. Decision tree (DT): consists of node, branch (link) and leaf. Each node represents a feature, link or branch that represents a rule between several feature choices, and each leaf represents an outcome. Nodes are used to classify groups of features and branches to classify their values [25]. SVM tries to draw a border between the points from different classes. The border is drawn so that the distance between the border and the different class points is maximizing and minimizing the classification error [26]. K-nearest neighbor (K-NN) classifier uses Euclidean distance function. After that, the majority sign of the k nearest points will be taken to classify the new point. The K is a hyperparameter for the k nearest neighbor algorithm [27].

Ensemble methods like bagging, random subspace, random forest, and boosting were implemented. Bagging is an independent ensemble method; each learner works separately, and their respective outputs are integrated through a voting procedure. The main steps of bagging techniques are: From the input training data set, random multiple data sets with replacement are produced, which are called bootstraps. Then, multiple classifiers from the same type are implemented and fitted on the bootstrap samples. Finally, the results of the individual classifiers through the voting procedure will be determined (aggregation process) [28]. Bagging helps in minimizing variation and avoiding overfitting problems [29]. The random subspace method (RSM) was presented by Ho [30]. It is known as feature bagging, in which random subsets of features are generated, while the number of training records remains the same. Random forest is the combination of bagging and random subspace algorithms [31]. Boosting (AdaBoost) This type improves weak learners into strong learners belonging to a group of algorithms known as "boosting". This approach trains a group of learners sequentially; then focuses on correcting the errors of the preceding learner through weight editing. This approach helps in reducing bias [32].

The K-fold cross validation method was used to split the data. This strategy divides the dataset into k equal-sized parts, (k-1) of them are used for training, while one is kept for testing. It is repeated for k times. In the first fold, the test data will be taken from the first part, in the second fold, the test data will be taken from the second part. It is repeated for k times until all the k folds are finished. Finally, the average of all testing subsets' accuracy for all the K folds is calculated [33]. Each selected method was trained and evaluated using the 10-folds cross validation method. decision tree (J48), KNN, SVM and Ensemble methods which combined multiple decision tree classifiers (bagging, AdaBoost, random subspace, and random forest) were implemented.

4. EXPERIMENT AND RESULTS

4.1. Environment

WEKA software was used to carry out the experiments. The WEKA program is widely used in data mining [34]. Furthermore, we divided the dataset into training and testing sets using a 10-fold cross validation method.

4.2. Evaluation metrics

Our experiments used five evaluation metrics to evaluate the classification performance for the different machine learning and ensemble methods: accuracy, precision, recall, F-measure, and AUC [35]–[37]. After finishing the 10-fold, the confusion matrix was generated and used to find the evaluation

metrics (accuracy, precision, recall, F-score and AUC). All the metrics were calculated using a confusion matrix [35]. Table 11 shows an example of the confusion matrix.

Table 11. Confusion matrix [38]

Actual Values	Predictive values	
	Positive (1)	Negative (0)
Positive (1)	TP	FN
Negative (0)	FP	TN

Accuracy is defined as the ratio of correctly classified cases over the total number of cases. Equation (2) shows the formula of accuracy [36]. Precision is defined as the ratio of correctly estimated positive cases over the total number of estimated positive cases. Equation (3) shows the formula of precision. Recall is the ratio of correctly estimated positive cases over the total number of actual positive cases. Equation (4) shows the formula of recall [37]. F-Measure is used to integrate the recall and precision values and make a balance between them. The formula of F-Measure is as shown in (5) [35].

$$Accuracy = (TP + TN) / (TP + FN + FP + TN) \quad (2)$$

$$Precision = TP / (TP + FP) \quad (3)$$

$$Recall = TP / (TP + FN) \quad (4)$$

$$F - Measure = 2 (Precision * Recall) / (Precision + Recall) \quad (5)$$

ROC curve plots the values of the true positive rate (y-axis) against the false positive rate (x-axis) at a various threshold value. After that, the intersection points between them are connected through a curve. Figure 3 depicts an example of the ROC curve. The area under the ROC curve represents the value of the AUC. Higher AUC indicates a better capability to correctly classify the different classes [37].

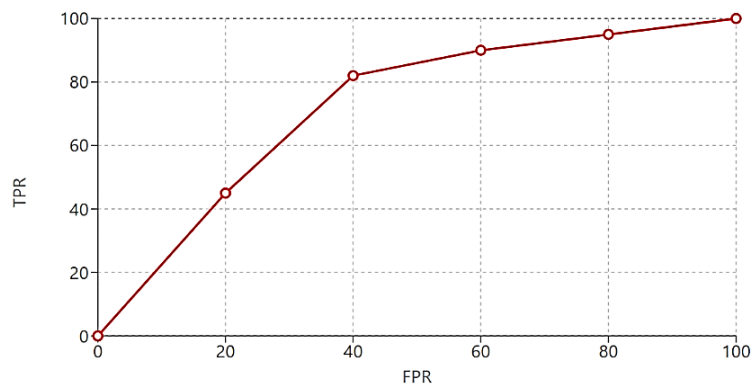


Figure 3. An example for ROC curve [38]

4.3. Evaluation results

This section will introduce and discuss algorithms results related to evaluation metrics like accuracy, weighted average of precision, recall, F-measure, and AUC for the data set. We applied each of the selected algorithms for the different classification task (two-level, three-level) in two stages: initial data (using all the 19 features and the initial number of students which was 434) and modified data (using the augmented number of students after applying SMOTE technique and the reduced number of features after applying the feature selection method). At two-level task with modified data, the number of students became 1,091 (back to Table 9), and the number of features became 15 (by removing all these features: semester difficulty, total income, high school GPA and academic load). At the three-level task with modified data, the number of students became 1,108 (back to Table 10). The number of features became 15 (by removing all these features: semester difficulty, total income, high school GPA, and academic load). The results for the data set

at the two-level task with initial data and modified data are shown in Table 12. The results at the three-level task with initial and modified data are shown in Table 13.

Table 12. Calculated metrics at the two-level classification task with initial and modified data

		DT (J48)	KNN	SVM	Bagging (with DT classifiers)	AdaBoost (with DT classifiers)	Random forest (with DT classifiers)	Random subspace (with DT classifiers)
Two levels for the data set (initial data)	Accuracy (100%)	82.95%	83.18%	80.65%	83.87%	80.65%	83.18%	84.56%
	Weighted Precision	0.713	0.763	0.755	0.746	0.776	0.756	Undefined
	Weighted Recall	0.829	0.832	0.806	0.839	0.806	0.832	0.846
	Weighted F-Measure	0.767	0.782	0.775	0.776	0.788	0.779	Undefined
	Weighted AUC	0.529	0.664	0.52	0.723	0.61	0.747	0.716
	Accuracy (100%)	87.26%	89.37%	89.55%	89.64%	90.19%	90.83%	91.20%
Two levels for the data set (modified data)	Weighted Precision	0.873	0.899	0.896	0.897	0.902	0.908	0.912
	Weighted Recall	0.873	0.894	0.896	0.896	0.902	0.908	0.912
	Weighted F-Measure	0.873	0.893	0.896	0.896	0.902	0.908	0.912
	Weighted AUC	0.902	0.954	0.896	0.956	0.946	0.973	0.971

Table 13. Calculated metrics at three-level classification task with initial and modified data

		DT (J48)	KNN	SVM	Bagging (with DT classifiers)	AdaBoost (with DT classifiers)	Random forest (with DT classifiers)	Random subspace (with DT classifiers)
Three levels for the dataset (initial data)	Accuracy (100%)	58.53%	54.84%	58.29%	63.59%	61.06%	62.67%	63.82%
	Weighted Precision	0.558	0.544	0.586	0.598	0.605	0.594	Undefined
	Weighted Recall	0.585	0.548	0.583	0.636	0.611	0.627	0.638
	Weighted F-Measure	0.561	0.545	0.582	0.606	0.604	0.59	undefined
	Weighted AUC	0.684	0.69	0.701	0.787	0.738	0.755	0.778
	Accuracy (100%)	77.62%	85.65%	77.17%	81.59%	87.09%	87.18%	84.12%
Three levels for the dataset (modified data)	Weighted precision	0.68	0.857	0.772	0.815	0.872	0.874	0.843
	Weighted recall	0.673	0.856	0.772	0.816	0.871	0.872	0.841
	Weighted F-Measure	0.676	0.855	0.771	0.816	0.871	0.873	0.842
	Weighted AUC	0.784	0.957	0.86	0.946	0.957	0.969	0.954

As shown in Table 12, the best result was generated by random forest with 83.18% accuracy at the two-level classification task from the initial data. The 83.18 accuracy means that 361 out of 434 students are correctly classified. They were distributed as follows: the number of correctly classified students in the “fail” class was 0 out of 67 and 360 out of 367 in the “pass” class. 73 out of 434 students are incorrectly classified. random subspace generated the best result from the modified data with 91.2% accuracy. The 91.2% accuracy means that 995 out of 1,091 students are correctly classified. They were distributed as follows: the number of correctly classified students in the “fail” class was 494 out of 541 and 501 out of 550 in the “pass” class. 96 out of 1,091 students are incorrectly classified. In general, using the modified data enhanced the overall models' performance. Especially in the “fail” class because the data size was augmented and became balanced for all classes.

From Table 13, at the three-level classification task with initial data, the best result was generated by Bagging with 63.59% accuracy. The 63.59 accuracy means that 276 out of 434 students are correctly classified. They were distributed as follows: the number of correctly classified students in the “fail” class was 5 out of 67, 134 out of 185 in the “moderate” class and 137 out of 182 in the “good” class. 158 out of 434 students are incorrectly classified. Random forest generated the best result of the modified data with 87.18% accuracy. The 87.18% accuracy means that 966 out of 1108 students are correctly classified. They were distributed as follows: the number of correctly classified students in the “fail” class was 331 out of 374, 316 out of 370 in the “moderate” class and 319 out of 364 in the “good” class. 142 out of 1108 students are incorrectly classified.

From Tables 12 and 13, we conclude that applying features selection methods and SMOTE technique (modified data) enhanced the performance for all methods (all the metrics values resulting from the modified data were higher than the initial data). Also, the ensemble methods (bagging, boosting, random forest, random subspace), which combine multiple decision tree classifiers, improved the individual decision tree (J48) performance. For example, the accuracy resulting from the decision tree with modified data at the two-level task was 87.26%. On the other hand, the accuracy resulting from the ensemble methods with modified data at the two-level task (Bagging, Boosting, random forest, random subspace) was respectively as follows: 89.64%, 90.19%, 90.83% and 91.20%.

The results demonstrate that ensemble methods, especially random subspace and random forest, outperform individual ML models for predicting student course outcomes when using an augmented dataset with SMOTE oversampling. The novel course difficulty and academic load features did not rank among the most important predictors, with preceding course grades and number of attempts having the greatest impact. Compared to prior studies using ensemble methods [13], this work achieves similar accuracy gains of 3-5% on the binary classification task but higher improvements of 9-10% on the 3-class task over baseline models. The use of SMOTE proved more beneficial than the filtering approach [14]. However, the best performing ensemble methods align with the findings [15]. The implications are that student performance can be accurately predicted using a range of academic features, with more granular grade categories providing additional insight beyond pass/fail classification. The methodology outlined here could be readily applied by other higher education institutions to proactively identify students needing support in specific courses. Future work could explore additional ensemble architectures, techniques for handling class imbalances, and experiments predicting outcomes in other courses

5. CONCLUSION

In this research, binary and multi-classification models for course outcomes were successfully created. The classification models can help students predict their performance in a specific course ahead of time (the dataset contains features that allow the prediction of course outcomes that will be taken in the next semester). Besides, it helps the teaching institutes to better understand the success chances of their students and provide the proper guidance accordingly. The model can help in performing classification tasks for any given data in the future. The data set was modified by removing irrelevant features and augmentation of the data size. To evaluate modified data, machine learning methods and ensemble methods were applied to the initial data and modified versions of the data separately. Using the modified data improved the models' performance at all classes. At the two-level task, the highest amount of improvement achieved by using modified data over initial data reached 9.55% (90.19-80.65) from the AdaBoost method. At the three-level task, the highest improvement achieved by using modified data over initial data reached 30.81% (85.65-54.84) from the K-NN method. In addition, the ensemble techniques, which combined multiple decision tree classifiers, improved the individual decision tree results (J48). For example, at two-level classification task with modified data, the best accuracy was obtained by random subspace with 91.20%, the accuracy achieved up to 3.94% (91.2-87.26) improvement over individual decision tree. At the three-level classification task, the best accuracy was obtained by random forest with 87.18%, the accuracy achieved up to 9.56% (87.18-77.62) improvement over individual decision tree results. All the tasks (two-level, three-level) built using ensemble methods with modified data resulted in good models' performance and can be used in any institution. The most significant features affecting the dependent variable (OR2 course grade) were preceding courses grades and the number of trials. In future, researchers could focus on using other ensemble methods such as the stacking method.




REFERENCES

- [1] R. Jindal and M. D. Borah, “A survey on educational data mining and research trends,” *International Journal of Database Management Systems*, vol. 5, no. 3, pp. 53–73, Jun. 2013, doi: 10.5121/ijdmms.2013.5304.




- [2] M. Al Luhaybi, "Explainable machine learning for educational data," Thesis, Brunel University London, 2021.
- [3] R. Nisbet, J. Elder, and G. Miner, "The data mining process," in *Handbook of Statistical Analysis and Data Mining Applications*, Elsevier, 2009, pp. 33–48.
- [4] C. Romero and S. Ventura, "Educational data mining: a survey from 1995 to 2005," *Expert Systems with Applications*, vol. 33, no. 1, pp. 135–146, Jul. 2007, doi: 10.1016/j.eswa.2006.04.005.
- [5] A. Peña-Ayala, "Educational data mining: a data mining-based analysis of recent works," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1432–1462, Mar. 2014, doi: 10.1016/j.eswa.2013.08.042.
- [6] C. Romero and S. Ventura, "Educational data mining: a review of the state of the art," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 6, pp. 601–618, Nov. 2010, doi: 10.1109/tsmcc.2010.2053532.
- [7] J. Simons, *A national study of student early alert models at four-year institutions of higher education*. 2011.
- [8] S. Ismail and S. Abdulla, "Design and implementation of an intelligent system to predict the student graduation AGPA," *Australian Educational Computing*, vol. 30, no. 2, 2015.
- [9] M. E. Zorrilla, E. Menasalvas, D. Marin, E. Mora, and J. Segovia, "Web usage mining project for improving web-based learning sites," in *Computer Aided Systems Theory – EUROCAST 2005*, Springer Berlin Heidelberg, 2005, pp. 205–210.
- [10] S. B. Aher and Lobo L.M.R.J., "Data mining in educational system using WEKA," in *Proceedings International Conference on Emerging Technology Trends (ICETT)*, 2011.
- [11] M. Quadri and D. Kalyankar, "Drop out feature of student data for academic performance using decision tree techniques," *Global Journal of Computer*, vol. 10, no. 2, pp. 2–5, 2010.
- [12] S. Umair and M. M. Sharif, "Predicting students grades using artificial neural networks and support vector machine," in *Encyclopedia of Information Science and Technology, Fourth Edition*, IGI Global, 2018, pp. 5169–5182.
- [13] D. M. Ahmed, A. M. Abdulazeez, D. Q. Zeebaree, and F. Y. H. Ahmed, "Predicting university's students performance based on machine learning techniques," Jun. 2021, doi: 10.1109/i2cacis52118.2021.9495862.
- [14] A. A. Alsulami, A. S. A. Al-Malaise Al-Ghamdi, and M. Ragab, "Enhancement of e-learning student's performance based on ensemble techniques," *Electronics*, vol. 12, no. 6, Mar. 2023, doi: 10.3390/electronics12061508.
- [15] M. Ashraf, M. Zaman, and M. Ahmed, "An intelligent prediction system for educational data mining based on ensemble and filtering approaches," *Procedia Computer Science*, vol. 167, pp. 1471–1483, 2020, doi: 10.1016/j.procs.2020.03.358.
- [16] R. Timaran Pereira and J. Caicedo Zambrano, "Application of decision trees for detection of student dropout profiles," 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 2017, pp. 528–531, doi: 10.1109/icmla.2017.0-107.
- [17] A. Jain, "Multi-class classification to track students' academic outcome," MSc Research Project, National College of Ireland, 2019.
- [18] S. Muchuchuti, L. Narasimhan, and F. Sidume, "Classification model for student performance amelioration," in *Advances in Information and Communication*, Springer International Publishing, 2019, pp. 742–755.
- [19] C. Romero, J. R. Romero, and S. Ventura, "A survey on pre-processing educational data," in *Educational Data Mining*, Springer International Publishing, 2013, pp. 29–64.
- [20] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised learning," *International Journal of Computer Science*, vol. 60, no. 1–2, pp. 111–117, 2011.
- [21] A. Saad Hussein, T. Li, C. W. Yohannese, and K. Bashir, "A-SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, 2019, doi: 10.2991/ijcis.d.191114.002.
- [22] K. Lakshminarayan, S. A. Harp, and T. Samad, "Imputation of missing data in industrial databases," *Applied Intelligence*, vol. 11, no. 3, pp. 259–275, 1999, doi: 10.1023/A:1008334909089.
- [23] A. G. Karegowda, A. S. Manjunath, and M. A. Jayaram, "Comparative study of attribute selection using gain ratio and correlation based feature selection," *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 271–277, 2010.
- [24] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *ICML 01 Proceedings of The Eighteenth International Conference on Machine Learning*, 1994, pp. 74–81.
- [25] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics*, vol. 18, no. 6, pp. 275–285, Jun. 2004, doi: 10.1002/cem.873.
- [26] S. B. Kotsiantis, "Use of machine learning techniques for educational proposes: a decision support system for forecasting students' grades," *Artificial Intelligence Review*, vol. 37, no. 4, pp. 331–344, May 2011, doi: 10.1007/s10462-011-9234-x.
- [27] L.-Y. Hu, M.-W. Huang, S.-W. Ke, and C.-F. Tsai, "The distance function effect on k-nearest neighbor classification for medical datasets," *SpringerPlus*, vol. 5, no. 1, Aug. 2016, doi: 10.1186/s40064-016-2941-7.
- [28] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.
- [29] S. B. Kotsiantis and P. E. Pintelas, "Combining bagging and boosting," *Computational Intelligence*, vol. 1, no. 4, pp. 324–333, 2004, doi: 10.1103/PhysRevD.77.085025.
- [30] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998, doi: 10.1109/34.709601.
- [31] R. Gondane and V. S. Devi, "Classification using rough random forest," in *Mining Intelligence and Knowledge Exploration*, Springer International Publishing, 2015, pp. 70–80.
- [32] A. Singh, N. Thakur, and A. Sharma, "A review of supervised machine learning algorithms," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 1310–1315.
- [33] E. A. Amrieh, T. Hamtini, and I. Aljarah, "Mining educational data to predict student's academic performance using ensemble methods," *International Journal of Database Theory and Application*, vol. 9, no. 8, pp. 119–136, Aug. 2016, doi: 10.14257/ijtda.2016.9.8.13.
- [34] R. Arora and S. Suman, "Comparative analysis of classification algorithms on different datasets using WEKA," *International Journal of Computer Applications*, vol. 54, no. 13, pp. 21–25, Sep. 2012, doi: 10.5120/8626-2492.
- [35] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
- [36] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *arXiv preprint arXiv:2010.16061*, Oct. 2020.
- [37] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315–1316, Sep. 2010, doi: 10.1097/jto.0b013e3181ec173d.
- [38] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jul. 2020, doi: 10.1016/j.aci.2018.08.003.

BIOGRAPHIES OF AUTHORS






Emad Al-Momani    is an assistant professor at Al Hussein Technical University, bringing over Emad Al-Momani 15 years of rich industrial experience to academia. He earned his Ph.D. in industrial engineering in 2011. His research interests include electronics manufacturing, ergonomics and machine learning. Dr. Al-Momani is committed to fostering a dynamic learning environment that prepares students for challenges in both academia and industry. He can be contacted at email: emad.almomani@htu.edu.jo.






Ala'a Shatnawi    is an industrial engineer who earned both her bachelor's and master's degrees in industrial engineering from the Jordan University of Science and Technology. Her research interests encompass artificial intelligence, machine learning, and the development of classification models for predicting course outcomes using ensemble methods. She has contributed significantly to these fields through her research and publications. Her work aims to improve predictive models and enhance the accuracy of course outcome predictions. He can be contacted at email: shatnawialaa10@gmail.com.






Mohammed A. Almomani    full professor of materials and industrial engineering at the Industrial Engineering Department of Jordan University of Science and Technology. Professor Almomani holds a Ph.D. degree in materials engineering from the University of Wisconsin Milwaukee, U.S.A in 2009. He got his master's degree in industrial engineering from the University of Jordan in 2004, and his bachelor's degree in mechanical engineering from Jordan University of Science and Technology in 2000. Prof. Almomani has research in lean manufacturing, machine learning, additive manufacturing, nanocomposite coatings, and advanced materials. He can be contacted at email: maalmomani7@just.edu.jo.



Ammar Almomani    is a renowned expert in internet infrastructure security and cybersecurity, specializing in AI-driven solutions. Holding a Ph.D. in computer science, he is recognized as one of the top 2% of global influential researchers by Stanford University and the Scopus Index (2023). With over two decades of academic experience, He has taught 42+ topics related to cybersecurity at five institutions. His research and leadership have been instrumental in managing AI and cybersecurity projects with international teams. Currently, he serves as a professor and head of the Research and Innovation Department at the School of Computing in Sharjah, UAE, contributing to course development and program accreditation efforts. He can be contacted at email: ammarnav6@bau.edu.jo.



Mohammad Alauthman    associate professor and chair of the Information Security Department at the Faculty of Information Technology, University of Petra in Amman, Jordan. His research interests lie in network security, intrusion detection systems, and the application of artificial intelligence techniques like machine learning and deep learning for botnet detection, DDoS detection, spam email filtering, IoT security, and network traffic classification. He obtained his Ph.D. in computer science with a focus on network security from Northumbria University in the UK in 2016. Alauthman has been awarded several research grants in the field of network security, supporting his work on advanced intrusion detection systems and the development of AI-driven security solutions. These grants have enabled him to conduct extensive research and collaborate with international experts. His funded projects have focused on enhancing the resilience of network infrastructures against emerging threats and improving the accuracy of threat detection mechanisms. He can be contacted at email: mohammad.alauthman@uop.edu.jo.