

Process-Based Multi-level Homogeneous Ensemble Predictive Model for Analysing Student's Academic Performance



Mukesh Kumar and Amar Jeet Singh

Abstract The aim of this study is to undertake an empirical inquiry and comparison of the effectiveness of various classifiers with ensemble classifiers in the prediction of student academic performance. A single classifier algorithm will be compared against the performance and efficiency of ensemble classifiers. Reducing student attrition is a major problem for educational institutions all over the world. The search for solutions to increase student retention and graduation rates continues for educators. This is only possible if at-risk students are identified and intervened with as soon as possible. However, the majority of regularly used prediction models are inefficient and inaccurate as a result of inherent classifier limitations and the inclusion of insignificant inputs in their calculations. The majority of data mining and machine learning researcher focused on developing an algorithm that can extract useful information from massive amounts of data after being processed by a computer. The most difficult problem in predictive modelling is identifying the most effective prediction algorithms that are also accurate enough to be useful. Therefore, a multi-level homogeneous ensemble predictive (MLHoEP) model is designed, which uses the different techniques of data mining like feature selection, ensemble learning techniques like boosting and bagging. Seven distinct machine learning algorithms were used on this model to predict and analyse the academic performance of the students. The performance of the classification algorithms in terms of prediction was evaluated using k -fold cross-validation. The study contributes to the body of knowledge by suggesting the development of homogeneous classifiers that may be used to accurately predict students' academic success. It also proposes the construction of homogeneous classifiers, which may be deployed for accurate student performance prediction, in order to provide a better explanation for the poor performance prediction. As a result of this research, it has been demonstrated that the technique of applying homogeneous ensemble approaches is incredibly efficient and accurate in terms of predicting student performance and assisting in identifying students, who are in danger of dropping out of school. The study compared the accuracy and efficiency of single classifiers to ensembles of classifiers in terms of performance. It was discovered in the

M. Kumar (✉) · A. Jeet Singh

Department of Computer Science, Himachal Pradesh University, Shimla, India

e-mail: mukesh.kumarphd2014@gmail.com

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

139

D. Gupta et al. (eds.), *International Conference on Innovative Computing and*

Communications, Lecture Notes in Networks and Systems 473,

https://doi.org/10.1007/978-981-19-2821-5_12

research that a homogeneous model with excellent accuracy and efficiency might be developed for anticipating student performance. These key problems have been successfully addressed by the findings of this research study: Which characteristics of students are the most effective predictors of academic performance? How accurate are approaches such as bagging and boosting ensembles for predicting student academic performance? The approach offered in this study will aid educational administrators and policymakers in designing new policies and curriculum-linked to student retention in higher education. This research can also aid in the identification of students who are at risk of dropping out of school early, providing for timely intervention and support. Prospective research will examine the creation and implementation of an automated prediction system known as the students' academic performance forecast framework, which will collect data from students via online submission and produce a prediction result for their academic performance.

Keywords Educational data mining · Ensemble learning · Multilayer perceptron · Random forest · Naïve Bayes · Correlation attribute evaluation · Information gain · Gain ratio

1 Introduction

Ensemble learning is frequently used to average the predictions of multiple classification models in order to obtain a more accurate forecast. This strategy is used to forecast the outcomes of insignificant classifier models drawn from a variety of input spaces. It is simply a technique for integrating the outputs of multiple models to obtain a more accurate result [1]. This is one of the simplest and most cost-effective strategies for enhancing the accuracy of your model's prediction outcomes. The majority of real-world applications employ some form of ensemble approach to improve the prediction model's performance. After the Netflix challenge, where all of the winning teams used ensembles of numerous inconsequential models to win, these ensemble learning techniques became increasingly prevalent. Netflix's primary goal with this challenge was to develop a new recommendation system that would allow users to suggest new films [15]. Additionally, these strategies are applied to deep learning. In data mining, certain classification algorithms, such as the random forests technique, are implemented using ensembles. The simplest technique for training a random forest algorithm is to train multiple decision trees on distinct subsets of the dataset using different feature subsets and then average the results [3].

Bagging and boosting are two further examples of assembly. Bagging is a technique that entails running numerous models on distinct sets of input samples and then averaging the results. Bagging is advantageous when the objective is to reduce variation while maintaining the same bias [6]. When used on an overfitted model with low bias and a large variation, bagging is beneficial. It is ineffective in cases where models exhibit a high degree of bias. Essentially, ensemble learning is group learning. Essentially, ensemble learning is a technique in which we train a large number of

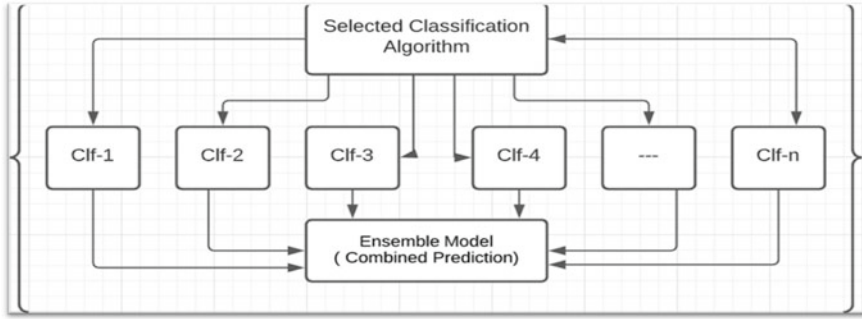


Fig. 1 Representation of ensemble learning methods

unimportant models and then integrate their predictions to arrive at a conclusion [2]. Combining the forecasts is a procedure that is determined by the models trained. If the models are homogeneous, that is, if all trained models utilise the same algorithm, such as decision trees, we can apply either bagging or boosting (Fig. 1).

These are the most often utilised ensemble learning approaches. If the trainers are diverse and a combination of multiple algorithms is utilised, such as decision trees, logistic regression, and so on, meta-learning can be applied. In this example, on top of all the predictions, you train another model that determines the final prediction [11]. Assume the learners generate class probabilities using a combination of 100 decision trees and logistic regression. You will end up with 100 values for each training instance; you can then train another model to predict the real outcome using these 100 values.

2 Literature Survey

During this study, we came across different research papers related to implementation of ensemble learning algorithms and see how these algorithms improve the prediction result of different classifier model. Different researcher groups of education data mining communities are working in different areas of education and its development [10]. Their research is focused on to find the effect of different student’s attributes on academic performance, predict the academic performance of the student’s, and predict the placement of the student [20]. In article [16], a survey of the literature is presented and certain theoretical methods are implemented in order to forecast student performance. For example, she discovered and compared the accuracy of Naive Bayes, Neural Network, and Decision Tree to predict students’ cumulative grade point average (CGPA),students’ demographics, high school, and study and social network attributes as the most critical factors in whether students pass or fail their studies [4]. The accuracy of naive Bayes is higher than that of neural networks and decision trees because it uses attributes that are more significant to forecast.

Educational data mining (EDM) is an interdisciplinary field that is concerned with the creation of methods to analyse a variety of unique data in the education area, with the goal of better understanding students' requirements and determining appropriate learning approaches [13]. In general, EDM is used to foresee difficulties in order to improve the quality of both student performance and the teaching–learning process [12], as well as the overall teaching–learning process. Due to the large amount of data in the educational dataset, it is concerned with how to adapt data mining methods and identify patterns, which are normally highly difficult problems to solve [14]. In order to identify datasets, data mining as a decision-making tool has been aided by a variety of approaches, including statistical models, mathematical methods, and machine learning algorithms [5]. Yet another piece of research, paper [18], examines numerous and relevant data mining approaches for classification in prediction, primarily for the purpose of determining the most important aspects of student performance forecasts. Using the random forest and J48 classification models, it is possible to forecast student achievement and to identify the most significant factors that influence it, such as study time spent, academic year attended, and parental education. In this paper [19], artificial neural networks, decision trees, and Bayesian networks were utilised to detect dropouts in order to investigate a large number of probable factors. Tan found two attribute variables as test inputs while doing empirical research on a dataset containing 3.59 million student records from an online training programme. These attribute variables were student characteristics and academic performance. As a result, the decision tree method was more exact in demonstrating that those variables are effectively used as key components in the prediction of student dropouts than before. As demonstrated in this work [9], Marquez presented a novel strategy for optimising the accuracy of predictive modelling, which he named modified interpretable classification rule mining, to improve predictive modelling accuracy. Marquez conducted an experiment in 419 schools to determine the elements that contribute to student dropouts. Six steps of evaluation were carried out, with a total of 670 students providing 60 different factors. As a result, modified classification rule mining is more accurate than JRip in terms of accuracy. Predictive modelling issues currently include the effectiveness and accuracy of various prediction models, which are mostly caused by insufficient variables in the basic classifier in most cases. In a related study [8], decision trees, naive Bayes, KNNs, and artificial neural networks were used to construct a predictive student dropout model and to adopt ensemble clustering based on students' demographic information, academic performance, and enrolment history, respectively. The accuracy of prediction models can be improved by using an experiment-verified ensemble approach to transform original data into a new form. Another similar study, as stated in [7], explored and investigated the ensemble technique, which was found to be effective in reducing errors and increasing the accuracy of student performance prediction. Below are some takeaway from this literature review:

Student's Attributes Which Effect the Academic Performance Prediction: There are so many student attributes, which effect the academic performance of any student from academic, family, institutional, social or personal attributes. Which attribute effect the student performance most is a matter of research for each and every researcher in the field of educational data mining. But it all depend upon the output you want from your predictive model. Some researcher wants to predict the student dropout status, some wants to predict the student placement, and some wants to predict the final grade of the student and many more. So, in the literature, there is no fixed attributes which we can say totally effect the overall performance of the student in school or any other institution during their study. But surely, we found that the categories of student attributes which overall play some role in predicting the academic performance of the students and these are academic attributes, family attributes, and institutional attributes.

Classification algorithms mostly used to predict academic performance student's: An important task for predicting academic performance of student's is to develop a superior classifier model by using classification algorithms. There are lots of families of classification algorithms which are built in the past by different researcher. At the time literature review, we came across such different algorithms which gave different types of accuracy on to the selected datasets for predicting academic performance of the students.

We want to say that in educational data mining, to improve the overall prediction accuracy of any classification algorithms, we have some ensemble learning techniques and these techniques are bagging, boosting, and random subspace. So, we proceed with our work by taking below-mentioned questions in our mind; first is how classification algorithms are valued for predicting academic performance of students and second is how classification algorithms performance is improved by using different ensemble learning techniques.

3 Materials and Methods

3.1 Data Description

This dataset pertains to student achievement in secondary education at two Portuguese educational institutions [17]. Among the information gathered from the students were student grades as well as demographic, social, and school-related attributes. The information was obtained through school reports and questionnaires. On the basis of performance in two independent subjects: mathematics (mat) and the Portuguese language (por), two datasets are offered. Cortez and Silva [7] used the two datasets to simulate classification and regression tasks that were either binary or five-level classification or regression tasks. One thing to keep in mind is that the target attribute G3 has a high association with the other two traits, G2 and G1. This is due to the fact that G3 is the final year grade (which is delivered at the end of the

Table 1 New class level assigned to the dataset

Range of initial class given in the dataset	New cluster number assigned to the class level
Grade between 16 and 20	Class-A
Grade between 14 and 15	Class-B
Grade between 12 and 13	Class-C
Grade between 10 and 11	Class-D
Grade less or equal to 9	Class-F

third period), whereas G1 and G2 correspond to the first and second period grades, respectively. Even though it is more difficult to anticipate G3 without first predicting G2 and then G1, such predictions are far more valuable. The desired output class initially has a range of 0–20, and there are 21 clusters. This is an unreasonable option for the classification task, as it makes classification extremely difficult, especially given the small number of instances available. In the given dataset, G1, G2, and G3 and the grade obtained by different students and for better result we find the final grade of the student by find the average of all grades and create a new attribute named as “total grade”. As a result, I have assigned a group of clusters to a few class levels denoted by the letters A, B, C, D, and F in Table 1.

3.2 Classification Algorithm Used

Classification is a data mining technique that classifies the elements in a dataset. The objective of classification is to accurately anticipate the target class for each occurrence of data. For instance, a classification model could be used to classify loan applicants into three categories based on their credit risk: low, medium, and high. Several classification techniques have been chosen for implementation, as follows:

Naïve Bayes: Naive Bayes is a model which is based on Bayes’ theorem and makes several fiercely independent assumptions. It forecasts the probability that a particular instance in a dataset belongs to a specific class. It is presumed that the prevalence of a feature in a class is unrelated to the presence of any other characteristic, i.e. that all features contribute independently in calculating the probability of data classification. This model is advantageous for very huge datasets and is simple to implement.

Random Forest: It is an ensemble method that combines various decision trees and a bagging technique. Bagging is the process of training each decision tree using a portion of the original dataset obtained through sampling and replacement. The final class is determined by conducting a majority vote on the outcome of all decision trees. It is an extremely efficient and effective technique when dealing with enormous datasets.

Decision Tree: The decision tree algorithm, also known as induction of decision trees, is a technique that is used in statistics, data mining, and machine learning to do predictive modelling and classification. It progresses from observations of an object's attributes to judgments about the item's desired value through the use of a decision tree.

Multilayer Perceptron: This is a sort of feedforward neural network that has multiple layers (ANN). Backpropagation is a supervised learning strategy that is used to train the algorithm. A MLP differs from a linear perceptron in that it has many layers and nonlinear activation, whereas a linear perceptron has only one layer. It has the capability of separating data that are not linearly separable, among other things.

Decision Table: Specific attributes are considered during the learning process of this classifier. This is accomplished by computing the table's cross-validation performance in various subsets of attributes and picking the subsets that performs the best. The cross-validation error is calculated by changing the class counts associated with each dataset entry, as the table structure remains constant, when instances are added or deleted. Typically, the feature space is searched using a best-first search method.

JRip: This class provides a learner for propositional rules, which can be used to automate the learning process. This approach was developed by William W. Cohen as an acceptable algorithm for the IREP. It employs a technique known as repeated incremental pruning in order to reduce error rates (RIPPER).

Logistic Regression: When there are many explanatory factors, logistic regression is used to compute the odds ratio. When there are multiple explanatory variables, logistic regression is used to calculate the odds ratio. The approach is quite similar to multiple linear regression. However, the response variable is a binomial distribution instead of a linear distribution. The outcome is defined as the effect of each variable on the odds ratio of the observed occurrence.

3.3 Ensemble Learning Method Used

When using ensemble learning, numerous data mining models are combined to create more efficient and effective learning algorithms, which ultimately improves the accuracy of any model's prediction output. This strategy combines numerous weak learners in order to increase the accuracy of our predictive models. In ensemble models, the decision tree is frequently chosen as the weak learner, and this is because of its simplicity. The core concept behind ensemble learning is that it involves training a large number of inconsequential models and then combining the predictions to get a conclusion. The strategy used to combine the predictions is determined by the models that were used in the training process. If the models are homogeneous, meaning that all of the trained models use the same algorithm, such as the decision tree, then

you can use either bagging or boosting to optimise the performance of the model. Gradient boosters, which are ensemble models, have grown increasingly popular.

Bagging Ensemble Learning: Bootstrap aggregation is the technical term for bagging. By producing some additional data for training from your original dataset, utilising combinations with repetitions to build multisets of the same size as your original data, it is possible to reduce variation in the outcome of your prediction. You will not improve the predictive accuracy of your model by increasing the size of your training set, but you will minimise the variance of your model, narrowing the forecast to the most likely outcome.

Boosting Ensemble Learning: It is a technique for creating a collection of predictive models that are used in conjunction with other techniques. Predictive models are taught sequentially using this technique, with early models fitting simple models to the data and then analysing the data for errors before learning more complex models. Remember that bagging requires each model to be run independently and then the outputs be aggregated at the end without giving any preference to any particular model.

3.4 Correlation Attribute Evaluator (CAE)

Methods for feature selection try to minimise the number of input variables to those that are deemed to be most beneficial in predicting the target variable. The purpose of feature selection is to exclude uninformative or redundant predictors from the model. Calculate the value of an attribute by calculating the correlation (Pearson's correlation coefficient) between it and the class. Nominal qualities are analysed value by value, with each value acting as an indicator. A weighted average is used to determine the overall correlation for a nominal property.

4 Proposed Multi-level Homogeneous Ensemble Predictive Model

In the below-mentioned Fig. 2, we demonstrate the working architecture of the proposed machine learning algorithms in conjunction with other important application algorithms of machine learning like feature selection (FS) and ensemble learning (EL) algorithms along with k -fold cross-validation as a testing method. At the start, first we need to select a dataset which is related to academic performance of the students with different features (independent and dependent). During the pre-processing phase, we need to remove all types of discrepancy be there in the dataset during data collection. Now, it is time to test our dataset in two different modes; first mode is to test our dataset with all the features present in it, and second

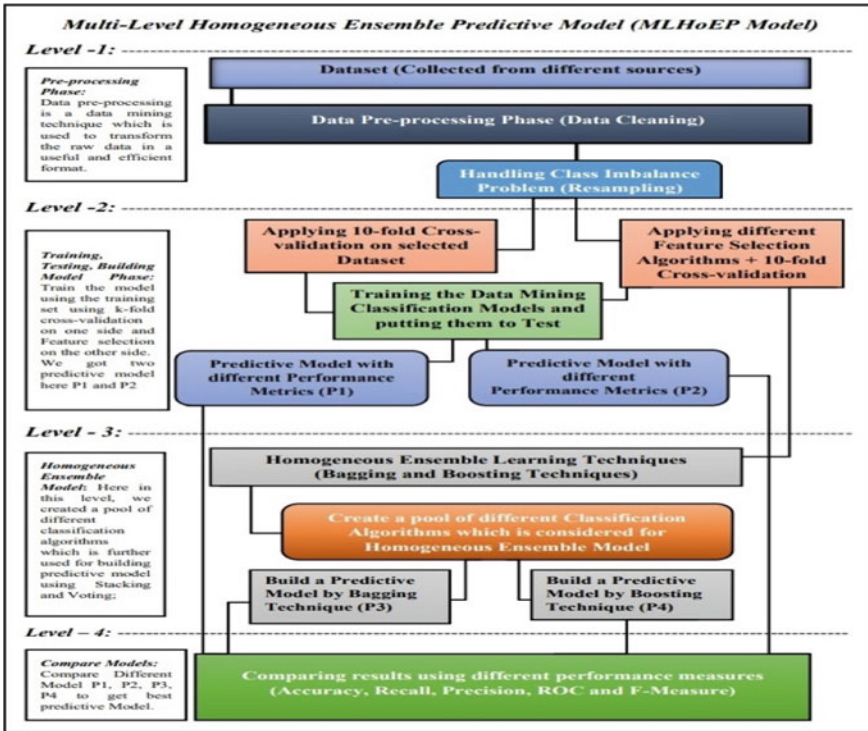


Fig. 2 Design of multi-level homogeneous ensemble predictive model

mode is to select some of the features with feature selection (FS) algorithm. Here, only correlation attribute evaluator (CAE) is used to implement FS and only top ten attribute are selected to find the accuracy of the classification algorithms. Now, move to next step where we need to select the testing mode along with the classification’s algorithms for the implementation. Now, it is time to select which ensemble learning algorithms need to be implemented to test the classifications algorithms.

A unique technique is called the multi-level homogeneous ensemble predictive model (MLHoEP model). As we saw throughout the literature review step, the majority of authors relied solely on data to arrive at the best outcome. However, in our MLHoEP model, we outlined a process that must be followed whenever homogeneous ensemble predictive modelling is used. In the MLHoEP model, we divide our predictive process into distinct levels, and each level will tackle its own set of problems. The following is a block diagram of the MLHoEP model:

Level-1: Prior to progressing to the next level, manage missing values (by mean and me–dian), outliers, and class imbalances in the dataset (Resampling Method).

Pseudo Code for level-1 in MLHoEP Implementation

Level-1: Data Pre-processing Phase

```

# Here, feature domain is {f1, f2, f3,...,fn}
# Handling Missing Value by mean()
1: Replace_Missing_Value_Mean(dataset)
2: return dataset ['f1', 'f2', 'f3', ..., 'fn']. replace ('O', mean())
# Handling Missing Value by median()
3: Replace_Missing_Value_Median(dataset)
4: return dataset ['f4', 'f5', ..., 'fn']. replace ('O', median())
5: Train_Test_Data_Split( diabetes)
# Handling imbalance problem by Oversampling
6: dataset minority oversampled(dataset)
7: retrun resample(1, replace = True, nsamples = majority class instance)
8: dataset = pd.concat([0, datasetminorityoversampled])
# Handling unbalance problem by Undersampling
9: dataset majority undersampled(dataset)
10: retrun resample (1, replace = True, n samples = minority class instance)
11: dataset = pd.concat([0, dataset_ majority undersampled])

```

Level-2: At this level, various classification methods are implemented and verified for accuracy (both with the complete dataset and with feature selection). Here, we have two clas=sifiers, P1 and P2.

Pseudo Code for level-2 in MLHoEP Implementation

Level-2: Training, Testing, Building Model Phase #Building Predictive Model (P1)

```

#Splitting dataset into Training and Testing dataset
12: Traing_Split, Testing_Split = split (dataset_feature_space,
dataset_class_level)
13: return TraingSplit, Testing Split
#Applying k-fold cross validation on selected dataset
14: CV = k_fold_cross_validation (n_splits=10. random_state = 1, shuffle =
True)
#Building Different Classifiers
15: Model-1: NBModel(Traing Split, Trainglabel, Testing Split)
16: Model-2: RFModel(Traing Split, Traing label, Testing Split)
17: Model-3: DTMndel(Traing Split, Traing label, Testing Split)
18: Model-4: MT PModel(Traing_Spl it. Traing_label, Testing_Split)
19: Model-5: DTModel(Traing Split, Traing label, Testing Split)
20: Model-6: JRipModel(Traing Split. Traing label, Testing Split)
21: Model-7: LRModel(Traing_Split, Traing label, Testing_Split)
#Building Predictive Model(P2)
#Applying different Feature Selection Algorithms
22: impattribute = model.CAE
23: for i, v in enumerate (imp_attribute):
24: Result v
25: Select top m feature according to your problem
#Applying k-fold cross validation on selected dataset

```

```

26: CV = k_fold_cross_validation (n_splits=10, random_state=1,
shuffle=True)
#Applying different Feature Selection Algorithms + k-fold cross validation
27: Model-1: NBModel (Traing Split, Traing label, Testing Split)
28: Model-2: RFModel (TraingSplit, Traing label, Testing Split)
29: Model-3: DTModel (Traing_Split, Traing_label, Testing_Split)
30: Model-4: MLPModel (Traing Split, Traing label, Testing Split)
31: Model-5: DTModel (Traing Split, Traing label, Testing Split)
32: Model-6: JRipModel (Traing Split, Traing label, Testing Split)
33: Model-7: (Traing_Split,Traing_label, Testing_Split)

```

Level 3: In this section, we develop a pool of diverse categorization methods that must be considered while constructing a homogenous ensemble model. Indeed, we are evaluating only those algorithms that have been selected for implementation at the level-2 level. We obtain P3 and P4 predictive models from this level.

Pseudo Code for level-3 in MLHoEP Implementation

Level-3: Homogeneous Ensemble Model

```

#Building Predictive Model (P3)
#Splitting dataset into Training and Testing dataset
34: Traing split, Testingsplit = split (dataset feature space, dataset class level)
35: return Traing split, Testing split
#Applying K-fold Cross-validation on selected Dataset
35: CV = k_fold_cross_validation (n_splits = 10, random_state = 1, shuffle
= True)
36: pool_of_classification_Model (Modell, Model2, Model7)
37: Compare accuracy of each model in the pool with the highest model
achieved
38: Ensemble_Model (TraingSplit, Trainglabel, TestingSplit)
39: ModellBagging.fit (Traing Split, Traininglabel)
40: Model2_Bagging.fit (Traing Split, Training label)
41: Model3_Bagging.fit (Traing_Split, Training label)
42: Model4_Bagging.fit (Traing_Split, Training_label)
43: Model5_Bagging.fit (Traing_Split, Training_label)
44: Model6_Bagging.fit (Traing Split, Training label)
45: Model7_Bagging.fit (Traing Split, Training label)
#Building Predictive Model (P4)
#Splitting dataset into Training and Testing dataset
46: Traing_split, Testing_split = split (dataset_feature_space.
dataset_class_level)
47: rehrn Traingsplit. Testingsplit
#Applying k-fold cross validation on selected dataset
48: CV = kfoldcrossvalidation (n_splits = 10, random_state = 1, shuffle =
True)
#Building Boosting Ensemble Model
49: pool_of_classification_Model (Modell, Model2, ..., Model7)

```

50: Compare accuracy of each model in the pool with the highest model achieved

51: EnsembleModel (TraingSplit, Traing label, TestingSplit)

52: Modell_Boosting.fit (Traing Split, Training label)

53: Model2_Boosting.fit (Traing_Split, Training label)

54: Model3_Boosting.fit (Traing_Split, Training_label)

55: Model4_Boosting.fit (Traing_Split, Training_label)

56: Model5_Boosting.fit (Traing Split, Training label)

57: Model6_Boosting.fit (Traing Split, Training label)

58: Model7_Boosting.fit (Traing_Split, Training label)

Level 4: Compare the predictive models (P1, P2, P3, P4) for better result on perform–mance metric.

All the necessary requirement are now set to implement the above-mentioned hybrid classification algorithms with the help of feature selection and feature selection algorithms. At the end, we need to compare all the implemented algorithms with each other to find the best on which gave use the maximum accuracy in prediction the result.

5 Implementation of the Proposed MLHoEP Model

Model Construction for the Standard Classifier: Numerous classification techniques were chosen and used to the dataset of student performance. We implement the following classification algorithms: naïve Bayes, random forest, J48, multilayer perceptron, decision table, JRip, and logistic regression. The table below summarises the implementation results of various categorisation algorithms using ten cross-validation (k -fold cross-validation) approaches. Our dataset is a balanced dataset with nearly equal distribution of data across five distinct classifications. According to Table 2, *decision tree* classification method had the greatest accuracy of 96.76% when

Table 2 Accuracy achieved by classification algorithm with all features of dataset

Classification algorithm	Accuracy (%)	MAE	Precision	Recall
Naïve Bayes + k -fold cross-validation	86.59	0.063	0.870	0.866
Random forest + k -fold cross-validation	92.14	0.118	0.924	0.921
Decision tree + k -fold cross-validation	96.76	0.017	0.968	0.968
Multilayer perceptron + k -fold cross-validation	84.59%	0.078	0.846	0.846
Decision table + k -fold cross-validation	90.13	0.146	0.909	0.901
JRip + k -fold cross-validation	96.14	0.024	0.962	0.961
Logistic regression + k -fold cross-validation	87.51	0.049	0.877	0.875

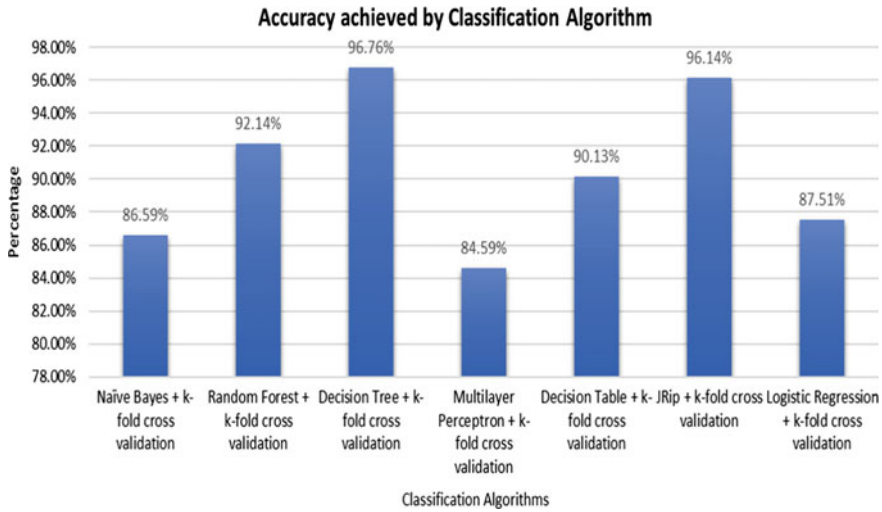


Fig.3 Graphical representation of accuracy level of classification algorithms

compared to other classification algorithms such as naive Bayes, random forest, decision table, multilayer perceptron, JRip, and logistic regression. As shown, the multilayer perceptron method achieves the lowest accuracy of 84.59%. Random forest and JRip algorithms also obtained an acceptable level of accuracy, at 92.14% and 96.14%, respectively. To implement these algorithms, all of the dataset’s attributes (up to 32) are considered. Other performance metrics such as mean absolute error (MAE), precision, recall, ROC area, and *F*-measure are also considered in this table. As our dataset contains no outliers, we will use accuracy as our primary parameter for evaluating our classifier’s effectiveness.

Classification algorithm accuracy is defined as the total number of correct predictions divided by the total number of predictions made by an algorithm for a given dataset. Figure 3 shows the graphical representation of the above-mentioned implementation of the classification algorithms with ten cross-validation (*k*-fold cross-validation) method. The graph clearly shows that decision tree classification algorithm performs exceptionally well as compared to other algorithms taken into consideration.

Implementation of Classification Algorithm after CAE feature selection: Classification, grouping, and regression algorithms all utilise a training dataset to establish weight factors that may be applied to previously unseen data for predictive purposes. Prior to executing a data mining technique, it is required to narrow down the training dataset to the most relevant attributes. Dimensionality reduction is the process of modifying a dataset in order to extract only the characteristics required for training. Due to its simplicity and computational efficiency, dimension reduction is critical since it minimises overfitting. Thus, dimensionality reduction is critical throughout the data pre-processing phase. A correlation-based feature selection method selects

Table 3 Accuracy achieved by classification algorithm with CAE

Classification algorithm used for implementing CAE	Accuracy (%)	MAE	Precision	Recall
Naïve Bayes + CAE + k -fold cross-validation	87.51	0.058	0.878	0.875
Random forest + CAE + k -fold cross-validation	95.83	0.061	0.959	0.958
Decision tree + CAE + k -fold cross-validation	96.91	0.0169	0.969	0.969
Multilayer perceptron + CAE + k -fold cross-validation	97.68	0.0172	0.977	0.977
Decision Table + CAE + k -fold cross-validation	90.13	0.146	0.909	0.901
JRip + CAE + k -fold cross-validation	95.53	0.0252	0.956	0.955
Logistic regression + CAE + k -fold cross-validation	97.84	0.0087	0.979	0.978

attributes based on the usefulness of individual features for predicting the class label, as well as the degree of connection between them. We avoid strongly linked and irrelevant features. The correlation attribute evaluator determines an attribute's value in a dataset by calculating the correlation between the attribute and the class attribute. Nominal qualities are assessed individually, with each value acting as a signal. A weighted average is used to generate an overall correlation for a nominal characteristic. We picked the top ten attributes with a threshold value larger than 1 using the aforementioned attribute evaluator CAE in conjunction with the ranker search strategy.

The following table summarises the results of the implementation of several classification algorithms using CAE and the test option as k -fold cross-validation approaches. As shown in Table 3, the combination (logistic regression + CAE + k -fold cross-validation) achieved the greatest accuracy of 97.84% when compared to other classification algorithms such as naive Bayes, random forest, decision tree, multilayer perceptron, decision table and JRip. As can be seen, the multilayer perceptron technique improves accuracy to 97.68%, which is significantly higher than the accuracy obtained without utilising the feature selection approach. The remainder of the algorithms is also accurate to an acceptable level. Only the top fifteen attributes of the dataset are considered when implementing these methods. Other performance metrics such as mean absolute error (MAE), precision, and recall value are also considered in this table. As our dataset contains no outliers, we will use accuracy as our primary parameter for evaluating our classifier's effectiveness.

Figure 2 is a graphical illustration of the implementation of the classification algorithms discussed previously using CAE and cross-validation (k -fold cross-validation) as testing methods. The graph clearly demonstrates that the logistic regression algorithm outperforms all other algorithms considered. However, as illustrated in Fig 1, practically all classification systems obtain a prediction accuracy of greater than 90% (Fig. 4).

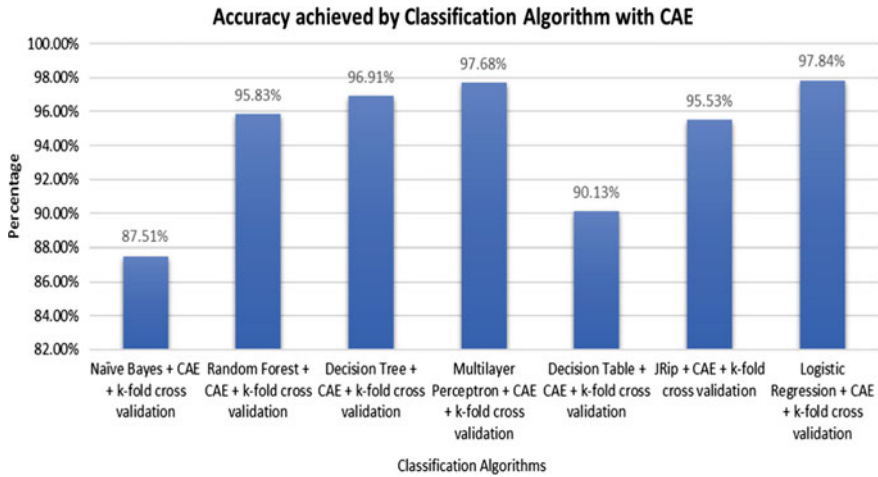


Fig. 4 Graphical representation of accuracy level of classification algorithms with CAE

Implementation of Bagging Ensemble after CAE feature selection: As part of this implementation, classification algorithms are applied to a dataset that has been reduced in features by employing the CAE feature selection technique in conjunction with the bagging ensemble and *k*-fold cross-validation selections, among other techniques. Table 3 shows that when compared to other classification algorithms taken into consideration, logistic regression and multilayer perceptron classification algorithms achieved the highest accuracy of up to 97.90%, as well as naive Bayes and random forest classification algorithms (also known as random forest and JRip classification algorithms). Using the multilayer perceptron technique, we can see that their prediction performance increased from 84.59% (without feature selection) to 97.90% (with feature selection). Using feature selection techniques, the performance prediction of the vast majority of algorithms improves significantly over time. A number of other performance metrics, including mean absolute error (MAE), precision, and recall, are taken into account in this table. We are just interested in accuracy in this example because the dataset does not contain any outliers; thus, we are only interested in accuracy when evaluating the performance of our classifier.

Figure 3 presents a graphical depiction of the data in Table 4, which is shown below the figure. When compared to other methods taken into consideration, the graph clearly demonstrates that logistic regression and the multilayer perceptron classification algorithm perform remarkably well. The accuracy of these two algorithms in terms of performance prediction is close to 97.90%, which is higher than the accuracy of the decision table method, which is also a rule-based classification system. Classification algorithms such as random forest, J48, decision table, and JRip attain accuracy levels of over 90% in several cases (Fig. 5).

Implementation of AdaBoostM1 Ensemble after CAE feature selection: As part of this particular portion of the implementation, classification algorithms are applied to

Table 4 Accuracy achieved by classification algorithm with CAE and bagging ensemble

Classification algorithm used for implementing bagging	Accuracy (%)	MAE	Precision	Recall
NB + CAE + bagging + <i>k</i> -fold cross-validation	89.67	0.0616	0.899	0.897
RF + CAE + bagging + <i>k</i> -fold cross-validation	94.60	0.0686	0.946	0.946
DT + CAE + bagging + <i>k</i> -fold cross-validation	95.99	0.0232	0.960	0.960
MLP + CAE + bagging + <i>k</i> -fold cross-validation	97.90	0.0297	0.968	0.968
DT (table) + CAE + Bagging + <i>k</i> -fold cross-validation	94.76	0.1501	0.950	0.948
JRip + CAE + bagging + <i>k</i> -fold cross-validation	96.45	0.0242	0.965	0.965
LR + CAE + bagging + <i>k</i> -fold cross-validation	97.90	0.0248	0.968	0.968

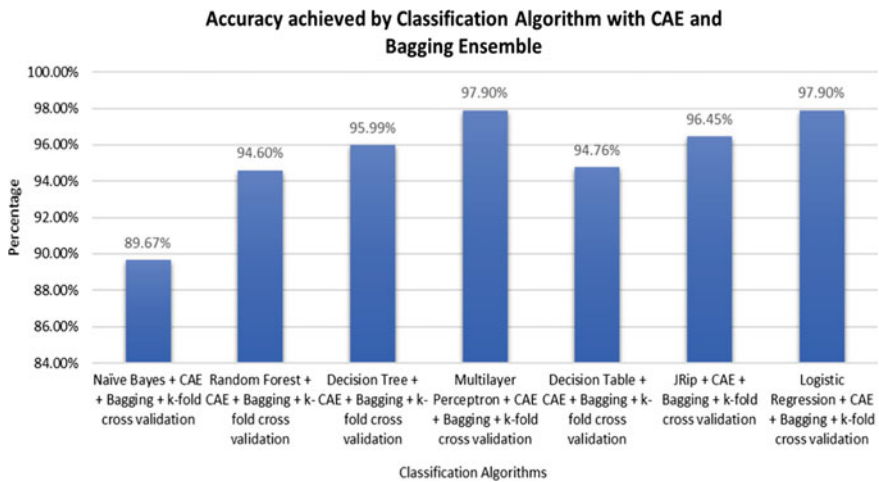


Fig. 5 Graphical representation of accuracy achieved by classification algorithm with CAE and bagging ensemble

a dataset that has been reduced in features by employing the CAE feature selection technique in conjunction with AdaBoostM1 ensemble learning with *k*-fold cross-validation, among other techniques. Table 4 shows that the decision table and logistic regression classification algorithms achieved the highest accuracy of up to 97.84% when compared to the other classification algorithms taken into consideration, which included naive Bayes, random forest, multilayer perceptron, decision tree, and JRip

Table 5 Accuracy achieved by classification algorithm with CAE and AdaBoostM1 ensemble

Classification algorithm used for implementing AdaBoostM1	Accuracy (%)	MAE	Precision	Recall
NB + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	91.83	0.0332	0.918	0.918
RF + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	95.83	0.0332	0.959	0.958
DT + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	97.38	0.0106	0.974	0.974
MLP + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	97.68	0.0131	0.977	0.977
DT(table) + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	97.84	0.010	0.979	0.978
JRip + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	97.07	0.012	0.971	0.971
Logistic regression + CAE + AdaBoostM1 + <i>k</i> -fold cross-validation	97.84	0.0087	0.979	0.978

classification algorithms. Using feature selection techniques, the performance prediction of the vast majority of algorithms improves significantly over time. A number of other performance metrics, including mean absolute error (MAE), precision, and recall, are taken into account in this table. One of the most interesting things about naive Bayes is that it achieves accuracy levels greater than 90%. We are just interested in accuracy in this example because the dataset does not contain any outliers; thus, we are only interested in accuracy when evaluating the performance of our classifier.

The graphical version of Table 5 is shown in the section Fig. 4. The graph clearly demonstrates that the decision table and logistic regression classification algorithms outperform all other algorithms taken into consideration when compared to one another. While the performance prediction accuracy of the decision table method is close to 97.84%, the accuracy of the JRip algorithm, which is also a rule-based classification system, is just slightly higher at 97.24%. Classification algorithms such as random forest, J48, decision table, and JRip attain accuracy levels of over 90% in several cases. However, the accuracy level of the naive Bayes algorithm has already risen to more than 90% (Fig. 6).

6 Analysis of All Predictive Classifier by MLHoEP Model

In this section, we will look at a comparative study of all of the algorithms that have been implemented. Using *k*-fold cross-validation as a testing option, we first examine the prediction accuracy of classification algorithms that use ensemble learning (bagging and AdaBoostM1) with and without ensemble learning. We can examine the following algorithms one by one in Table 6, which has a list of the algorithms taken into consideration:

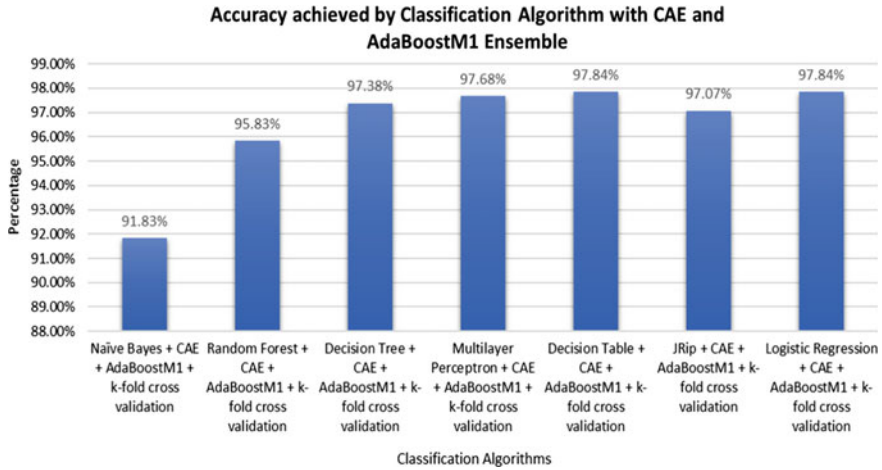


Fig. 6 Graphical representation of accuracy achieved by classification algorithm with CAE and AdaBoostM1 ensemble

Table 6 Accuracy achieved by all predictive classifiers by MLHoEP model

Classification algorithm	Accuracy (P1) (%)	Accuracy by CAE (P2) (%)	Accuracy by bagging ensemble (P3) (%)	Accuracy by AdaBoostM1 ensemble (P4) (%)
Naïve Bayes	86.59	87.51	89.67	91.83
Random forest	92.14	95.83	94.60	95.83
Decision tree	96.76	96.91%	95.99	97.38
Multilayer perceptron	84.59	97.68	97.90	97.68
Decision table	90.13	90.13	94.76	97.84
JRip	96.14	95.53	96.45	97.07
Logistic regression	87.51	97.84	97.90	97.84

Naïve Bayes: The ensemble learning methods used in our implementation included three different ensemble learning methods with k -fold cross-validation as the testing option method. Our observations revealed that the AdaBoostM1 ensemble performed exceptionally well on the supplied dataset, achieving the highest accuracy of 91.83%, which was significantly higher than that of another model like $P1$, $P2$, $P3$.

Random Forest: The ensemble learning methods used in our implementation included three different ensemble learning methods with k -fold cross-validation as the testing option method. Our observations revealed that the AdaBoostM1 ensemble

performed exceptionally well on the supplied dataset, achieving the highest accuracy of 95.83%, which was significantly higher than that of another model like *P1*, *P2*, *P3*.

Decision Tree: The ensemble learning methods used in our implementation included three different ensemble learning methods with *k*-fold cross-validation as the testing option method. Our observations revealed that the AdaBoostM1 ensemble performed exceptionally well on the supplied dataset, achieving the highest accuracy of 97.38%, which was significantly higher than that of another model like *P1*, *P2*, *P3*.

Multilayer Perceptron: The ensemble learning methods used in our implementation included three different ensemble learning methods with *k*-fold cross-validation as the testing option method. Our observations revealed that the bagging ensemble performed exceptionally well on the supplied dataset, achieving the highest accuracy of 97.90%, which was significantly higher than that of another model like *P1*, *P2*, *P4*.

Decision Table: The ensemble learning methods used in our implementation included three different ensemble learning methods with *k*-fold cross-validation as the testing option method. Our observations revealed that the AdaBoostM1 ensemble performed exceptionally well on the supplied dataset, achieving the highest accuracy of 97.84%, which was significantly higher than that of another model like *P1*, *P2*, *P3*.

JRip: The ensemble learning methods used in our implementation included three different ensemble learning methods with *k*-fold cross-validation as the testing option method. Our observations revealed that the AdaBoostM1 ensemble performed exceptionally well on the supplied dataset, achieving the highest accuracy of 97.07%, which was significantly higher than that of another model like *P1*, *P2*, *P3*.

Logistic Regression: The ensemble learning methods used in our implementation included three different ensemble learning methods with *k*-fold cross-validation as the testing option method. Our observations revealed that the bagging ensemble performed exceptionally well on the supplied dataset, achieving the highest accuracy of 97.90%, which was significantly higher than that of another model like *P1*, *P2*, *P4* (Fig. 7).

It is obvious from the preceding Fig. 5 that the AdaBoost1 ensemble approach did extraordinarily well in nearly all of the seven classification algorithms tested. It was discovered that classification algorithms such as naive Bayes and decision trees, as well as decision tables, random forests, and JRip, had higher accuracy than 97%. However, the bagging ensemble approach achieves the maximum accuracy in performance for multilayer perceptron and logistic regression, with a performance accuracy of up to 97.90%.

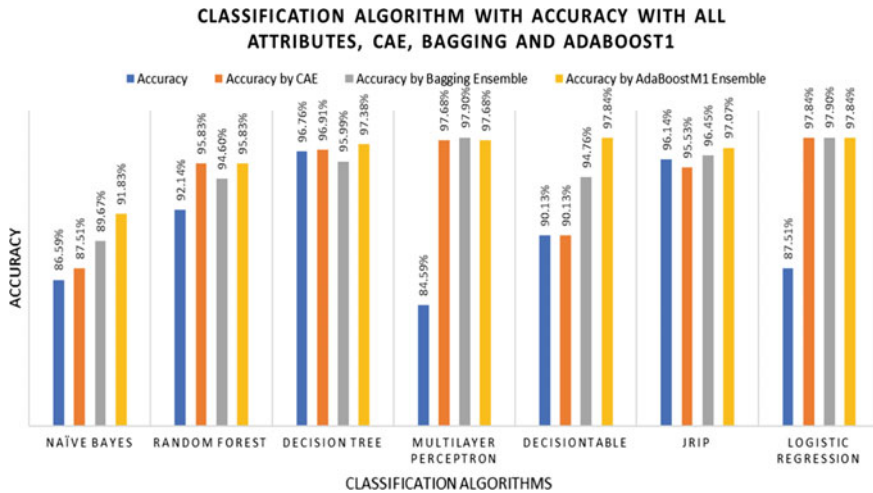


Fig. 7 Accuracy achieved by all predictive classifiers by MLHoEP model

7 Conclusion

When predicting the academic performance of students, many ensemble learning methods of data mining are taken into consideration. Feature selection methods such as bagging, boosting, and other ensemble learning methods are taken into consideration for implementation, as is the correlation attribute evaluator (CAE) as a feature selection algorithm. At the conclusion of this chapter, we can state that any classification algorithm that is implemented with the help of ensemble learning and the correlation attribute evaluator performs well when compared to algorithms that are implemented with ensemble learning but do not use the correlation attribute evaluator. It follows that ensemble learning, as well as feature selection, play an important role in improving the classification or prediction accuracy of the system.

Conflict of Interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Acknowledgements This work has been carried out in the Department of Computer Science, Himachal Pradesh University, Summer Hill Shimla-5, India. This research facility has not been funded from the university or any other funding agency.

References

1. Abubakaria MS, Arifin F, Hungilo GG (2021) Redicting students' academic performance in educational data mining based on deep learning using TensorFlow
2. Ashraf M, Zaman M, Ahmed M (2018) Using predictive modeling system and ensemble method to ameliorate classification accuracy in EDM. Asian J Comput Sci Technol:44–47

3. Ayyappan G, SivaKumar K (2018) A novel approach of ensemble models by using EDM. *Indian J Comput Sci Eng (IJCSSE)* 8:6
4. Bunkar K, Singh UK, Pandya B, Bunkar R (2012) Data mining: prediction for performance improvement of graduate students using classification. In: 2012 ninth international conference on wireless and optical communications networks (WOCN). IEEE, pp 1–5
5. Cano A, Zafra A, Ventura S (2013) An interpretable classification rule mining algorithm. *Inf Sci*:1–20
6. Christian TM, Ayub M (2014) Exploration of classification using NBTree for predicting students. In: International conference on data and software engineering (ICODSE), Nov, pp 1–6
7. Cortez P, Silva AM (2008) Using data mining to predict secondary school student performance
8. Gray G, McGuinness C, Owende P (2014) An application of classification models to predict learner progression in tertiary education. In: 2014 IEEE international advance computing conference (IACC), pp 549–554
9. Iam-On N, Boongoen T (2017) Improved student dropout prediction in Thai University using ensemble of mixed-type data clusterings. *Int J Mach Learn Cybern* 8:497–510
10. Jishan ST, Rashu RI, Haque N, Rahman RM (2015) Improving accuracy of students' final grade prediction model using optimal equal width binning and synthetic minority over-sampling technique. *Decis Anal* 2:1–25
11. Kalaivani S, Nalini S (2017) Analyzing student's academic performance based on data mining approach. *Int J Innov Res Comput Sci Technol (IJIRCST)*:2347–5552
12. Katare A, Dubey S (2017) A comparative study of classification algorithms in EDM using 2 level classification for predicting student's performance. *Int J Comput Appl* 9:35–40
13. Kumar M, Singh AJ, Handa D (2017) Literature survey on educational dropout prediction. *Int J Educ Manage Eng* 2:8
14. Márquez-Vera C, Cano A, Romero C, Noaman AY, Mousa Fardoun H, Ventura S (2016) Early dropout prediction using data mining: a case study with high school students. *Expert Syst* 1:107–124
15. Mishra T, Kumar D, Gupta S (2014) Mining students' data for performance prediction. In: Fourth international conference on advanced computing & communication technologies, February, pp 255–262
16. Natek S, Zwilling M (2014) Student data mining solution—knowledge management system related to higher education institutions. *Expert Syst Appl* 14:6400–6407
17. Osmanbegovic E, Natek S, Zwilling M (2012) Data mining approach for predicting student performance. *Econ Rev J Econ Bus* 10:3–12
18. Osmanbegović E, Suljić M, Agić H (2014) Determining dominant factor for students performance prediction by using data mining classification algorithms. *Tranzicija*, pp 147–158
19. Tan M, Shao P (2015) Prediction of student dropout in e-Learning program through the use of machine learning method. *Int J Emerg Technol Learn* 10
20. Valsan V, Mathai PP, Babu I (2021) Monitoring driver's drowsiness status at night based on computer vision. In: 2021 international conference on computing, communication, and intelligent systems (ICCCIS), pp 989–993