Selecting Optimal Subset of Features for Student Performance Model

Hany M. Harb[1], Malaka A. Moustafa[2]
[1]Computers and Systems Engineering Dept.,Faculty of Eng., Azhar University
Cairo, Egypt

[2]Faculty of Eng., Azhar University
Cairo, Egypt

Abstract

Educational data mining (EDM) is a new growing research area and the essence of data mining concepts are used in the educational field for the purpose of extracting useful information on the student behavior in the learning process. Classification methods like decision trees, rule mining, and Bayesian network, can be applied on the educational data for predicting the student behavior like performance in an examination. This prediction may help in student evaluation.
As the feature selection influences the predictive accuracy of any performance model, it is essential to study elaborately the effectiveness of student performance model in connection with feature selection techniques. The main objective of this work is to achieve high predictive performance by adopting various feature selection techniques to increase the predictive accuracy with least number of features. The outcomes show a reduction in computational time and constructional cost in both training and classification phases of the student performance model.
Keywords: *Educational data mining, Feature selection, Classification algorithm, ASSISTments Platform dataset.*

1. Introduction

Web-based education systems accumulate a great deal of information which may be valuable in analyzing student behavior and assisting teachers in the detection of possible errors, shortcomings and improvements of the educational process. However, due to the vast quantities of these accumulated data, it is very difficult to be managed manually and assisting tools may be needed by users to analyze the data. Recently, researchers have begun to investigate various data mining methods in order to improve e-learning systems [1]. The use of data mining is a promising area in the achievement of this objective. In the learning process, the data mining exploiting knowledge discovery in databases (KDD) may automatically extract implicit and interesting patterns from large data collections. Data mining techniques may be applied for the following objectives such as statistics, clustering, classification, outlier detection, association rule mining, sequential pattern mining, text mining, or subgroup discovery [1].
Mining in educational environment is called Educational Data Mining (EDM). EDM is a new growing research area and the essence of data mining concepts are used in the educational field for the purpose of extracting useful information on the student behavior in the learning process. One of the EDM key areas is the improvement of student models that would predict student characteristics or academic performances in schools, colleges and other educational institutions. Prediction of student performance with high accuracy is useful in many contexts in all educational institutions for distinguishing students who are likely to have low academic achievements. The end product of models would be beneficial to teachers, parents and educational planners [2].
Classification is the most commonly applied data mining technique, which employs a set of pre-classified examples to develop a model that can classify the population of records at large. This approach frequently employs decision tree or neural network-based classification algorithms. The data classification process involves learning and testing phases. In learning phase, the trained data are analyzed by the classification algorithm generating classification rules. In testing phase, data are used to calculate the accuracy of the classification rules. If the accuracy is acceptable, the rules can be applied to the new data tuples. The algorithm uses these pre-classified examples to determine the set of parameters required for proper discrimination. Then, these parameters are encodes into a model called a classifier [3].
It becomes an essential for any tutor to predict the student performance in an examination. If the student failure is predicated prior an examination, then extra efforts can be exerted to improve the student performance.
The main objective of this paper is to apply feature selection algorithms to reduce the number of features, remove irrelevant, redundant, or noisy data, thus reducing the size of the dataset. This feature reduction speeds up the data mining process and improves its performance parameters such as predictive accuracy (PA) and result comprehensibility.
In this work, we aim to filter out redundant information and thus significantly reduce the resources required to predict the student performance such as memory size and CPU time. We used ASSISTments Platform dataset (a web-based tutoring system developed at Worcester Polytechnic Institute) and used with 4th to 10th grade math students. The responses are all taken from skill building problem sets worked on by students in a suburban middle school in central Massachusetts during the 2009-2010 school year. The dataset consists of approximately one million student record which contains

19 features (attributes), three irrelevant features out of them are discarded. Table 1 show the features description [4].

Table 1: The 15 features of ASSISTments dataset

Number	Feature	Description
1	assignment_id	Two different assignments can have the same sequence id. Each assignment is specific to a single teacher/class.
2	user_id	student ID
3	assistment_id	ID of the ASSISTment (consists of one or more problems)
4	problem_id	ID of the particular problem the student answered
5	Original	1 = Main problem - 0 = Scaffolding problem
6	attempt_count	Number of student attempts on this problem
7	ms_first_response_time	The time in milliseconds for the student's first response
8	tutor_mode	tutor, test mode, pre-test, or post-test
9	answer_type	Multiple choice, Algebra-fill_in, open_response
10	sequence_id	ID of the collection of problems the student answered
11	student_class_id	The class ID
12	problem_set_type	Linear - Random – Mastery
13	list_skill_ids	IDs of the skills associated with the problem
14	teacher_id	ID of the teacher
15	school_id	ID of the school

2. RELATED WORK

Ramaswami and Bhaskaran [5] have constructed a predictive model called CHAID with 7-class response variable by using highly influencing predictive variables obtained through feature selection so as to evaluate the academic achievement of students at higher secondary schools in India. Data were collected from different schools of Tamilnada, 772 student records were used for CHAID prediction model construction. As a result, set of rules were extracted from the CHAID prediction model and the efficiency was evaluated. The accuracy of the present model was compared with other models and it has been found to be satisfactory.

Al-Radaideh *et al.* [6] proposed usage of data mining classification techniques to enhance the quality of the higher educational system by evaluating students' data that may affect the students' performance in courses. A classification model was built using the decision tree method. They used three different classification methods ID3, C4.5 and the NaïveBayes. The results indicated that the decision tree model had better prediction accuracy than the other models. As a result, a system was built to facilitate the usage of the generated rules to predict the final grades in the C++ undergraduate course.

Cesar *et al.* [7] proposed a recommendation system based on data mining techniques to help students to make decisions related to their academic track. The system supports students to better choose how many and which courses to enroll on. As a result, the authors developed a system that is capable of predicting the failure or success of a student in any course using a classifier obtained from the analysis of a set of historical data related to the academic field of other students who took the same course in the past.

Pathom *et al.* [8] proposed a classifier algorithm for building Course Registration Planning Model (CRPM) from historical dataset. The algorithm is selected by comparing the performance of four classifiers include Bayesian Network, C4.5, Decision Forest, and NBTree. The dataset were obtained from student enrollments including grade point average (GPA) and grades of undergraduate students. As a result, the NBTree was the best of the four classifiers. NaïveBayes classifier (NBTree) was used to generate the CRPM, which can be used to predict the student GPA and consider student course sequences for registration planning.

The most important point in all the studies is that predicting student final score based on online activities is the leading approach to examine the effectiveness of e-learning.

3. Student Performance Model

3.1 Feature selection

Feature selection is one of the important and frequently used techniques in data preprocessing for data mining. It reduces the number of features, removes irrelevant, redundant, or noisy data, and thus improves mining performance attributes such as response time, predictive accuracy and result comprehensibility. Feature selection algorithms designed with different evaluation criteria broadly fall into three categories: the filter model, the wrapper model and the hybrid model [9]. **The filter model** relies on general characteristics of the data to evaluate and select feature subsets without involving any mining algorithm. **The wrapper model** requires one predetermined mining algorithm and uses its performance as the evaluation criterion. It searches for features better suited to the mining algorithm aiming to improve mining performance, but it also tends to be more computationally expensive than the filter model. **The hybrid model** attempts to take advantage of the two models by exploiting their different evaluation criteria in different search stages. Table 2 shows the taxonomy of feature selection algorithms [9]. In this work, the wrapper model was used.

Table 2: Taxonomy of Feature Selection Algorithms [9]

Model Search		Algorithms
Filter	Univariate	X2, Euclidian distance, t-test, Information gain
	Multivariate	Correlation-based feature selection, Markov blanket filter, Fast correlation-based feature selection
Wrapper	Deterministic	Sequential forward selection , Sequential backward elimination, Plus L Minus R, Beam search
	Randomized	Simulated annealing, Randomized hill climbing, Genetic algorithms, Estimation of distribution algorithms
Hybrid		Decision trees, Weighted naïve Bayes, Feature selection using the weight vector of SVM

A typical feature selection process consists of four basic steps (shown in Fig. 1), namely, subset generation, subset evaluation, stopping criterion, and result validation [10].

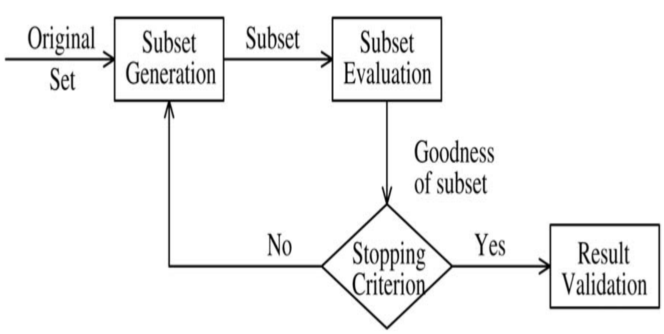

Figure1: Basic steps for feature selection process [10]

Subset generation is a search procedure that produces candidate feature subsets for evaluation based on a certain search strategy. Therefore, different search strategies have been explored such as complete, sequential, and random search. Each candidate subset is evaluated and compared with the previous best one according to a certain evaluation criterion. If the new subset turns out to be better, it replaces the previous best subset.

Subset evaluation each newly generated subset needs to be evaluated by an evaluation criterion. An evaluation criterion can be broadly categorized into two groups (based on their dependency on mining algorithms that will finally be applied on the selected feature subset). The two general groups are: *dependent and independent* criteria. An independent criterion (used in filter model) tries to evaluate the goodness of the feature subset by exploiting the intrinsic characteristics of the training data without involving any mining algorithm. While, the dependent criterion (used in the wrapper model) usually gives superior performance as it finds features better suited to the predetermined mining algorithm, but it also tends to be more computationally expensive, and may not be suitable for other mining algorithms [10]. As features are selected by the classifier that later on uses these selected features in predicting the class labels of unseen instances, accuracy is normally high, but it is computationally rather costly to estimate accuracy for every feature subset. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied. In this work, the dependent criterion was used.

Stopping Criteria determines when the feature selection process should stop. Some frequently used stopping criteria are: a) the search completes, b) some given bound is reached (minimum number of features or maximum number of iterations), c) subsequent addition/ deletion of any feature (a better subset and a sufficiently good subset is selected) [10].

Result Validation: a straightforward way for result validation is to directly measure the result using prior knowledge about the data. If we know the relevant features beforehand as in the case of synthetic data, we can compare this known set of features with the selected features. In real-world applications, however, we usually do not have such prior knowledge. Hence, we have to rely on some indirect methods by monitoring the change of mining performance with the change of features [10].

3.2 Dataset used for Experiments

The dataset used in this paper came from the ASSISTments Platform, (a web-based tutoring system developed at Worcester Polytechnic Institute [4]) and used with 4th to 10th grade math students. The responses are all taken from skill building problem sets worked on by students in a suburban middle school in central Massachusetts during the 2009-2010 school year. It contains 1 million students (rows) and it was made available for the 2011 Knowledge Discovery in Educational Data workshop. Each row in the dataset corresponds to a student answer which contains 19 columns: answer correctness, response time, problem information and several other metadata. The student's answer correctness can only be 1 for correct or 0 for incorrect and the response time is the time in milliseconds the student spent on the first attempt. In ASSISTments, there are "Main" and "Scaffolding" types of problems. "Scaffolding" problems are work steps of "Main" problems; students answer "Scaffolding" problems when they answered "Main" problems incorrectly, or they may choose to see the work steps, in which case the answer of "Main" problem will be marked as incorrect. There are two primary problem set types: "LinearSection" and "MasterySection" (where problems are given in a random order and students finish the problem set when they have answered three correct in a row). Some rows of the dataset are filtered out which are considered as unrealistic or gaming response time data and in the end of this process, 737586 of students (rows) are left in the dataset [11]

3.3 Methodology

In this section three basic steps are applied on the chosen ASSISTments dataset to test various algorithms.

I. Evaluating the environment (environment setup, data preprocessing, and choosing the data mining software).
II. Selecting a comprehensive set of the most popular and widely used classifier algorithms which represent a wide category of classifiers: Trees, Rules, Lazy, Bayes, General and Meta classifiers.
III. Implementing the four basic steps of feature selection.
    a. Subset generation: The six classifiers are used to rank all the features of the data set, then the highest three ranked features are selected as the subset to start with.
    b. Subset evaluation: Each classifier is applied to the generated subset (the highest three ranked features) and then features are successively added (wrapper model).
    c. Stopping criterion: The evaluation process continues till the best performance is detected for each classifier. In this work, the second method of stopping criterion was used.
    d. Result validation: The performance of the selected classifiers from the six categories is compared.

Environment setup

The experiments have been run on a system with a 2.66 GHz Intel Core (i5-560M) processor and 4 GB of RAM running Microsoft Windows 7 Home Premium (64-bit). The latest Windows version of Weka 3.6.2 has been used. Weka [12] is an open source machine learning package which is a collection of machine learning algorithms for data mining tasks. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes. This empirical study, however, only deals with a subset of classifier algorithms and uses the default options presented by Weka for each classifier. All the machine learning techniques that are used in this paper are implemented in Weka so that they will be easily and fairly compared to each other.

In data preprocessing, three irrelevant, redundant features out of 19 features of the dataset are discarded and some instances are filtered out which are considered as unrealistic or gaming response time data. The dataset used in our experiments contains 22000 students (rows) with 16 attributes which consists of 15 features among with a label that represents the instance class (either true or false). 66% of the dataset are used in the learning phase and the other part was used in the testing phase.

Classifier Algorithms Selection

In the experiment, six different classifiers are selected as our base classifiers: J48, IBK, Kmeans Clustering, NaiveBayes Updatable, ONER, and VFI Classifiers. Each classifier belongs to a different family of classifiers implemented in Weka where J48 related to Decision Trees, IBK belong to Lazy classifiers, ONER to Rules, NaiveBayes Updateable to Bayes classifiers, Simple Kmeans clustering to Meta classifiers, and VFI to general classifiers. Since they are from different classifiers family, they may yield different models that eventually will classify differently on some inputs. Also each one of these classifiers has the highest performance than other classifiers in its family [12].

VFI (Voting Frequency Intervals)

The VFI algorithm is a classification algorithm based on the voting frequency intervals. In VFI, each training instance is represented as a vector of features along with a label that represents the instance class. Feature intervals are then constructed for each feature. An interval represents a set of values for a given feature where the same subset of class values is observed. Thus, two adjacent intervals represent different classes [14].

Simple K means Clustering

The classic clustering technique is called k-means. First, the number of clusters must be specified. Then, k points are chosen at random as cluster centers. All instances are assigned to their closest cluster center according to the ordinary Euclidean distance metric. Next the centroid of the instances in each cluster is calculated. These centroids are taken to be new center values for their respective clusters. Finally, the whole process is repeated with the new cluster centers. Iteration continues until the same points are assigned to each cluster in consecutive rounds, at which stage the cluster centers have stabilized and will remain the same forever [15].

IBK (Instance-Based- K-nearest neighbor)

It is a simple instance-based learner K-nearest neighbor classifier. Instance-based learning schemes create a model by simply storing the dataset. A new data item is classified by comparing it with these 'memorized' data items, using a distance metric. The new item is assigned the category of the closest original data item (its 'nearest neighbor'). Alternatively, the majority class of the k nearest data items may be selected, or for numeric attributes the distance-weighted average of the k closest items may be assigned. IBk is an implementation of the k-nearest-neighbors classifier. The number of nearest neighbors (k) can be set manually, or determined automatically using cross-validation [16].

Naïve Bayes Updateable

This is the updateable version of NaïveBayes. NaiveBayes implements a NaïveBayesian classifier, which produces probabilistic rules—that is, when presented with a new data item, the NaiveBayes model indicates the probability that this item belongs to each of the possible class categories. The Bayesian classifier is 'naïve' in the sense that attributes are treated as though they are completely independent, and as if each attribute contributes equally to the model. If extraneous attributes are included in the dataset, then those attributes will skew the model. Despite its simplicity, NaiveBayes, like OneR, can give surprisingly good results on many real world datasets [16].

J48 (C4.5 Decision Tree Revision 8)

J48 is an implementation of C4.5 release 8, a standard algorithm that is widely used for practical machine learning. This implementation produces decision tree models. This algorithm works by forming pruned partial decision trees (built using C4.5's heuristics), and immediately converting them into a corresponding rule. C4.5 algorithm is the most popular tree classifier [16].

ONER (One-rules)

One-rules, are rules that classify an object on the basis of a single attribute (i.e. they are 1-level decision trees). ONER are a very simple classification rules performing well on most commonly used datasets. They use the minimum-error attribute for prediction, discretizing numeric attributes. They ranks attributes according to an error rate (on the training set). They treat all numerically-valued attributes as continuous and use a straightforward method to divide the range of values into several disjoint intervals [17].

4. Implementation and Results Analysis

4.1 Evaluation of algorithms on ASSISTments dataset

In this section the four basic steps of feature selection are applied. Firstly, the six classifiers were implemented in order to rank the 15 features of the ASSISTments dataset. Simulation results are given in Table 3.

Table 3: Lists of the Ranked Features.

Classifier	Ranked Features
VFI	4,2,1,3,13,7,6,9,5,15,11,14,8,12,10
IBK	4,2,1,3,5,6,7,8,9,10,11,12,13,14,15
NaiveBayes Updateable	4,2,1,5,6,9,3,12,8,7,15,13,14,11,10
ONER	4,1,2,3,5,6,7,8,9,10,11,12,13,14,15
J48	9,7,8,4,14,15,12,13,3,5,6,11,1,2,10
Kmeans Clustering	2,8,3,4,1,10,11,14,13,15,9,12,6,5,7

From the lists in Table 3, VFI and IBK classifiers give the same result for the first 4 features and differ in the rest. While the ONER classifier differs in the order of the first 3 features than IBK classifier but gives the same result in the rest from 4 to 15. While the other two classifiers (J48 and Kmeans Clustering) give different results.

Now, the first three features of each list are used as the generated subset to be evaluated by applying each classifier of the six selected classifiers, and then the features are successively added till the last ranked one. Then we notice the best performance for each classifier (which occurs if the prediction accuracy (PA) for the given subset of features is greater than the other subset PAs).

To compare the performance of the classifiers, we record the prediction accuracy (PA = Total correctly classified instances/Total instances) and the learning time to build the model (in seconds) of each algorithm. These parameters will be the most important criteria for the classifier to be considered and accordingly give the best subset of features.

4.2 Discussion and Experimental Results

Tables from 4 to 9 (in Appendix A) shows the evaluation criteria for the six classifiers applied to the ranked features by ONER (Table 4), IBK (Table 5), VFI (Table 6) J48 (Table 7), Simple Kmeans clustering (Table 8), and NaiveBayes Updateable (Table 9).

Tables 4-9 show that all classifiers give the best performance with only 3-7 features at most. While, Table 8 shows that Simple kmeans clustering is a good choice to rank the features, since it gives high prediction accuracy compared to the other classifiers (four classifiers have PA higher than PA achieved by the same classifiers with almost the same number of features but different in the ranking). From Table 4-9 we conclude the following:

• For NaiveBayes Updateable: the first 6 features give a best performance than the full 15 features. The first 6 subset gives best **PA (87.41%)** than the other subsets with Simple Kmeans clustering ranked classifier (table 8).

• For Simple Kmeans clustering: the first 3, 4 features subset gives the best PA than the other subset. It gives the less **PA (67.14%)** with the 3, 4 subset feature ranked by J48 (table 7).

• For J48 the PA is almost equal for all feature subset except for the feature ranked by J48 (table 7) which gives better PA (for the first 12 features) than the other ranked classifier.
• For ONER: PA is almost the same for all feature subset and for all the various feature ranked classifier applied except for Kmeans clustering classifier (table 8) which give best **PA(86.94%)** than the other ranked classifier.
• For IBK: the first 3- 5 subset features give better PA than PA obtained by the full 15 features, except for (table 8, feature ranked by Kmeans clustering) which gives high **PA(81.49%)** for the first three subset features.
• For VFI: it gives the highest **PA (87.48%)** with the Kmeans clustering (table 8) with the first 7 subset features.
Results also show that certain algorithms demonstrate superior detection performance compared to others. The prediction accuracy of **VFI** has the highest PA of **87.48%** and **NaiveBayes** has the PA **=87.41%** and **ONER** has the PA = **86.94%,** so we can say that **VFI, NaiveBayes , ONER** give the best accuracy for the subset features 6,7 ranked by Kmeans clustering. Also the three classifiers (**VFI, NaiveBayes , ONER)** take smallest time than the others. Figures 2 and Figures 3 show the variation of Prediction Accuracy and Time to build the model (the learning time of the Classifiers schemes) with the features ranked by Kmeans classifier which gives the best Prediction Accuracy with four classifiers from the six.

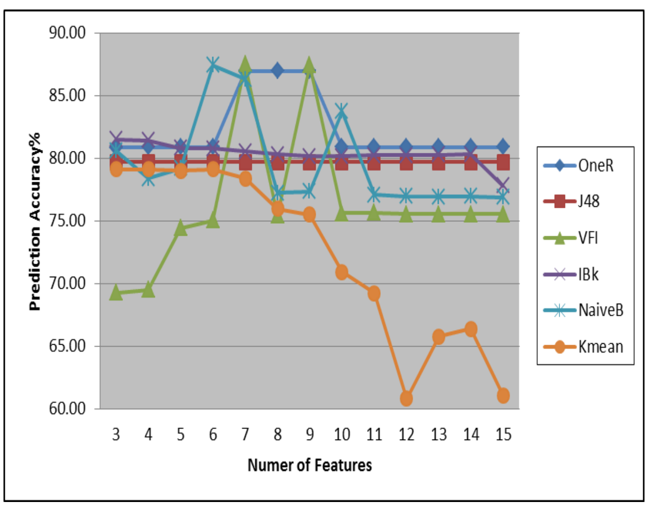

Figure 2: Prediction accuracy for features ranked by Kmeans classifiers

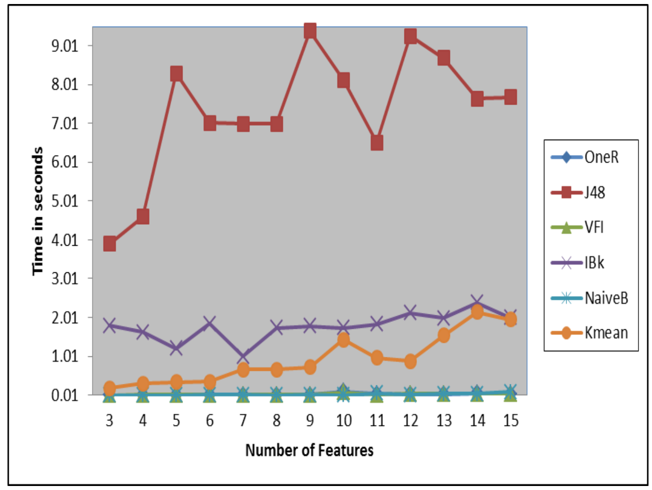

Figure 3: Time to build the model for features ranked by Kmeans classifier

Figure 3 shows that J48 classifier takes the largest time than the other five classifiers, and the three classifiers ONER, VFI and Naïve Bays take small time during building the model with all the subset features. Finally, results show that only the
first 7 features (2,8,3,4,1,10,11) user_id, tutor_mode, assistment_id,problem_id,assignment_id,sequence_id,student_class_id ranked by Kmeans are the optimal subset of features.

5. Conclusion

Six classifiers from different categories have been used to rank the 15 features of ASSISTments dataset, then they have been applied on the ranked features to get the optimal subset of features. Weka as an open source machine learning software has been used to conduct an extensive comparison of the six classifier algorithms performance. The experimental results show that all the classifiers give the best performance with only 3-7 features at most which results in reduction in data size and in processing time. The reduction of features (7 features at most) results in up to 80% reduction of the input dataset size without sacrificing of performance which means efficient utilization of the computing resources such as memory and CPU time.
The performance of the classification methods has been evaluated based on the basis of their predictive accuracy. The results indicate that the **Simple Kmeans clustering-meta** outperforms in ranking than the other classifiers, while **the NaiveBayes Updateable and ONER- Rules** outperform in both ranking and prediction than the other categories.

Appendix A

Evaluation Criteria and Performance Comparison for the Various Classifier Algorithms; PA is the Prediction Accuracy; Time is the time to build the model (In Seconds); #F is the number of features.

Table 4: Evaluation Criteria for the Various Classifiers applied to features ranked using ONER classifiers

#F	ONER		J48		VFI		IBk		NaiveUpdate		KMeans	
	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time
3	80.87	0.11	79.67	9.53	74.88	0.06	81.17	2.25	80.60	0.05	79.49	0.45
4	80.87	0.02	79.67	13.70	74.88	0.16	81.29	1.26	80.65	0.08	76.27	1.14
5	80.87	0.22	79.67	7.79	74.90	0.02	81.33	1.08	80.52	0.03	76.38	0.47
6	80.87	0.16	79.67	16.52	74.96	0.25	81.19	1.83	80.61	0.09	73.29	0.92
7	80.87	0.05	79.67	8.97	74.72	0.01	80.63	1.30	78.76	0.03	73.29	1.05
8	80.87	0.05	79.67	10.42	74.40	0.02	80.69	1.86	79.25	0.05	68.72	0.61
9	80.87	0.06	79.67	8.10	74.27	0.03	80.78	1.77	79.61	0.03	69.86	0.94
10	80.87	0.06	79.67	11.84	74.31	0.03	79.38	1.96	79.12	0.06	66.61	0.87
11	80.87	0.06	79.67	15.20	74.43	0.27	78.93	1.90	78.63	0.06	65.37	1.34
12	80.87	0.06	79.67	7.78	74.78	0.02	78.93	1.90	78.63	0.06	64.80	1.41
13	80.87	0.08	79.67	10.06	74.94	0.03	78.00	2.45	77.87	0.08	61.19	1.51
14	80.87	0.06	79.67	9.58	75.25	0.03	77.73	1.93	77.08	0.06	61.07	1.31
15	80.87	0.06	79.67	10.64	75.53	0.03	77.78	2.00	76.88	0.06	61.05	1.90
MAX	**80.87**		**79.67**		**75.53**		**81.33**		**80.65**		**79.49**	

Table 5: Evaluation Criteria for the Various Classifiers applied to features ranked using IBK classifier

#F	ONER		J48		VFI		IBk		NaiveUpdate		KMeans	
	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time
3	80.87	0.11	79.67	9.53	79.67	0.06	81.17	2.25	80.60	0.05	79.49	0.45
4	80.87	0.01	79.67	9.90	74.59	0.02	80.59	3.26	78.68	0.01	79.49	0.33
5	80.87	0.20	79.67	9.98	74.59	0.16	80.71	1.68	78.75	0.27	76.27	1.19
6	80.87	0.03	79.67	9.36	74.62	0.02	80.74	1.67	78.59	0.03	76.38	0.62
7	80.87	0.03	79.67	11.34	74.66	0.03	78.98	1.19	78.48	0.03	62.37	0.56
8	80.87	0.03	79.67	8.07	74.46	0.03	79.31	1.66	78.86	0.05	65.96	0.66
9	80.87	0.27	79.67	7.77	74.48	0.03	79.38	1.29	79.06	0.06	67.04	0.62
10	80.87	0.31	79.67	5.60	75.12	0.30	79.63	1.32	78.45	0.06	66.94	0.70
11	80.87	0.06	79.67	13.98	75.36	0.03	79.24	1.88	78.04	0.11	66.08	1.23
12	80.87	0.05	79.67	9.55	75.33	0.02	79.24	2.18	78.02	0.05	66.29	1.56
13	80.87	0.05	79.67	8.74	75.62	0.02	79.32	1.89	77.62	0.05	65.50	1.44
14	80.87	0.06	79.67	10.44	75.62	0.03	77.88	2.72	77.24	0.06	63.52	1.51
15	80.87	0.06	79.67	8.80	75.53	0.03	77.78	1.87	76.88	0.05	61.05	1.23
MAX	**80.87**		**79.67**		**79.67**		**81.17**		**80.60**		**79.49**	

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 1, September 2012
ISSN (Online): 1694-0814
www.IJCSI.org

Table 6: Evaluation Criteria for the Various Classifiers applied to features ranked using VFI classifier

#F	ONER		J48		VFI		IBk		NaiveUpdate		KMeans	
	PA%	Time	PA %	Time	PA	Time	PA %	Time	PA %	Time	PA %	Time
3	75.27	0.01	71.50	0.22	63.25	0.02	71.16	1.02	64.69	0.03	71.50	0.10
4	80.87	0.01	79.67	17.49	74.59	0.02	80.59	3.26	78.68	0.01	79.49	0.33
5	80.87	0.03	79.67	8.21	75.49	0.02	79.17	1.14	79.49	0.05	59.06	0.47
6	80.87	0.22	79.67	9.49	75.56	0.03	79.22	1.36	79.69	0.03	59.94	0.59
7	80.87	0.03	79.67	12.26	75.20	0.02	78.96	2.02	78.43	0.03	59.91	0.69
8	80.87	0.03	79.67	8.32	75.20	0.02	79.05	2.28	78.49	0.08	59.89	0.72
9	80.87	0.03	79.67	10.80	75.21	0.02	79.08	1.97	78.42	0.05	59.79	0.72
10	80.87	0.05	79.67	9.47	75.17	0.03	78.68	2.24	78.09	0.06	57.66	0.83
11	80.87	0.03	79.67	9.08	75.39	0.02	77.91	1.92	77.61	0.05	57.56	1.25
12	80.87	0.23	79.67	13.12	75.60	0.06	77.67	1.47	76.63	0.17	56.96	1.78
13	75.33	0.06	79.67	8.78	75.33	0.06	77.77	1.82	77.11	0.09	59.09	1.22
14	80.87	0.05	79.67	10.67	75.25	0.03	77.73	1.26	77.08	0.06	61.07	1.44
15	80.87	0.25	79.67	15.76	75.53	0.17	77.78	1.43	76.88	0.05	61.05	1.95
MAX	**80.87**		**79.67**		**75.60**		**80.59**		**79.69**		**79.49**	

Table 7: Evaluation Criteria for the Various Classifiers applied to features ranked using J48 classifier

#F	ONER		J48		VFI		IBk		NaiveUpdate		KMeans	
	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time
3	79.04	0.02	80.61	0.14	78.53	0.01	76.80	2.25	80.33	0.02	67.14	0.21
4	80.87	0.02	81.19	3.52	65.23	0.01	79.86	1.70	81.12	0.03	67.14	0.32
5	80.87	0.02	81.22	3.65	66.08	0.03	77.46	2.30	80.62	0.02	64.56	0.38
6	80.87	0.02	81.22	5.19	67.51	0.01	75.92	1.98	78.61	0.03	60.39	0.46
7	80.87	0.04	81.21	3.71	69.30	0.03	76.22	2.17	78.52	0.06	62.65	0.50
8	80.87	0.02	81.20	4.06	71.35	0.03	77.17	1.90	78.67	0.02	60.52	0.61
9	80.87	0.04	81.20	4.14	67.80	0.02	78.30	2.84	77.76	0.04	60.57	1.15
10	80.87	0.03	81.20	5.34	67.79	0.02	78.38	1.12	77.83	0.06	60.55	0.77
11	80.87	0.12	81.20	4.43	67.86	0.13	78.39	1.98	77.55	0.19	61.09	1.35
12	80.87	0.05	81.29	4.44	69.33	0.04	77.47	1.97	76.61	0.06	60.96	1.16
13	80.87	0.05	79.67	5.95	73.17	0.03	77.50	1.97	76.92	0.06	61.07	1.35
14	80.87	0.07	79.67	5.89	75.25	0.06	77.73	1.30	77.08	0.06	61.07	1.34
15	80.87	0.09	79.67	7.14	75.53	0.06	75.53	0.06	76.88	0.08	61.05	2.19
MAX	**80.87**		**81.29**		**78.53**		**79.86**		**81.12**		**67.14**	

Table 8: Evaluation Criteria for the Various Classifiers applied to features ranked using Kmeans classifier

#F	ONER		J48		VFI		IBk		NaiveUpdate		KMeans	
	PA%	Tim	PA%	Tim	PA%	Tim	PA%	Tim	PA%	Tim	PA%	Tim
3	80.83	0.01	79.67	3.90	69.26	0.01	81.49	1.80	80.57	0.01	79.12	0.19
4	80.87	0.01	79.67	4.60	69.50	0.02	81.40	1.63	78.37	0.01	79.12	0.31
5	80.87	0.01	79.67	8.30	74.42	0.02	80.80	1.21	79.18	0.01	79.01	0.34
6	80.87	0.02	79.67	7.02	75.02	0.02	80.76	1.85	87.41	0.02	79.09	0.36
7	86.94	0.02	79.67	6.99	87.48	0.02	80.54	1.01	86.29	0.02	78.37	0.67
8	86.94	0.02	79.67	6.99	75.39	0.02	80.28	1.74	77.22	0.01	75.94	0.67
9	86.94	0.02	79.67	9.39	87.40	0.02	80.16	1.79	77.35	0.02	75.50	0.73
10	80.87	0.11	79.67	8.13	75.59	0.08	80.16	1.74	83.72	0.01	70.92	1.44
11	80.87	0.04	79.67	6.51	75.65	0.02	80.22	1.84	77.07	0.06	69.20	0.97
12	80.87	0.03	79.67	9.26	75.51	0.04	80.22	2.13	76.95	0.02	60.79	0.89
13	80.87	0.03	79.67	8.69	75.51	0.04	80.24	1.99	76.92	0.04	65.75	1.55
14	80.87	0.06	79.67	7.65	75.51	0.05	80.31	2.39	76.94	0.05	66.36	2.15
15	80.87	0.05	79.67	7.68	75.53	0.05	77.78	2.00	76.88	0.09	61.05	1.96
MA	**86.94**		**79.67**		**87.48**		**81.49**		**87.41**		**79.12**	

Table 9: Evaluation Criteria for the Various Classifiers applied to features ranked using NaiveUpdate classifier

#F	ONER		J48		VFI		IBk		NaiveUpdate		KMeans	
	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time	PA%	Time
3	80.87	0.11	79.67	9.53	74.88	0.06	81.17	2.25	80.60	0.05	79.49	0.45
4	80.87	0.02	79.67	13.70	74.88	0.16	81.29	1.26	80.65	0.08	76.27	1.14
5	80.87	0.22	79.67	7.79	74.90	0.02	81.33	1.08	80.52	0.03	76.38	0.47
6	80.87	0.16	79.67	16.52	74.96	0.25	81.19	1.83	80.61	0.09	73.29	0.92
7	80.87	0.05	79.67	8.97	74.72	0.01	80.63	1.30	78.76	0.03	73.29	1.05
8	80.87	0.05	79.67	10.42	74.40	0.02	80.69	1.86	79.25	0.05	68.72	0.61
9	80.87	0.06	79.67	8.10	74.27	0.03	80.78	1.77	79.61	0.03	69.86	0.94
10	80.87	0.06	79.67	11.84	74.31	0.03	79.38	1.96	79.12	0.06	66.61	0.87
11	80.87	0.33	79.67	15.20	74.43	0.27	78.93	1.90	78.63	0.06	65.37	1.34
12	80.87	0.06	79.67	7.78	74.78	0.02	78.93	1.90	78.63	0.06	64.80	1.41
13	80.87	0.08	79.67	10.06	74.94	0.03	78.00	2.45	77.87	0.08	61.19	1.51
14	80.87	0.06	79.67	9.58	75.25	0.03	77.73	1.93	77.08	0.06	61.07	1.31
15	80.87	0.06	79.67	10.64	75.53	0.03	77.78	2.00	76.88	0.06	61.05	1.90
MAX	**80.87**		**79.67**		**75.53**		**81.33**		**80.65**		**79.49**	

References

[1] Mofreh A. Hogo, "Evaluation of E-Learners Behaviour using Different Fuzzy Clustering Models: A Comparative Study", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 7, No. 2, 2010.

[2] Ramaswami M. and Bhaskaran R., "A Study on Feature Selection Techniques in Educational Data Mining", Journal of Computing, Vol. 1, No. 1, 2009.

[3] Baradwaj B., Saurabh Pal "Mining Educational Data to Analyze Students' Performance", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 2, No. 6, 2011.

[4] http://teacherwiki.assistment.org/wiki/Assistments_2009-2010_Full_Dataset

[5] Ramaswami M., and Bhaskaran R., "CHAID Based Performance Prediction Model in Educational Data Mining", IJCSI International Journal of Computer Science Issues, Vol. 7, No.1, 2010.

[6] Al-Radaideh Q., Al-Shawakfa E., and AI-Najjar M., "Mining Student Data using Decision Trees", In Proceedings of the International Arab Conference on Information Technology (ACIT'2006), Yarmouk University, Jordan, 2006.

[7] Cesar V., Javier B., liela S., and Alvaro O., "Recommendation in Higher Education Using Data Mining Techniques",In Proceedings of the Educational Data Mining Conference, 2009.

[8]Pathom P., Anongnart S., and Prasong P., "Comparisons of Classifier Algorithms: Bayesian Network, C4.5, Decision Forest and NBTree for Course Registration Planning Model of Undergraduate Students", Sripatum University Chonburi Campus, Office of Computer Service, Chonburi Thailand, IEEE, 2008.

[9] Ladha L. and Deepa T., "Feature Selection Methods and Algorithms", International Journal on Computer Science and Engineering (IJCSE), 2011.

[10] Huan Liu, and Lei Yu, "Toward Integrating Feature Selection Algorithms for Classification and Clustering", IEEE Transactions On Knowledge and Data Engineering, Vol. 17, No. 4, 2005.

[11] Xiaolu X., Zachary A. Pardos and Neil T.,"An Analysis of Response Time Data for Improving Student Performance Prediction", USA, 2011.

[12] Weka – Data Mining Machine Learning Software, http://www.cs.waikato.ac.nz/ml/

[13] Hany M. Harb, Afaf A. Zaghrot, Mohamed A. Gomaa, Abeer S. Desuky "Selecting Optimal Subset of Features for Intrusion Detection Systems", Advances in Computational Sciences and Technology, Vol. 4 No. 0973-6107, India, 2011.

[14] Sabnani S. V., "Computer security: A machine learning approach", Master's thesis, Royal Holloway, University of London, 2007.

[15] Witten I.H., Frank Eibe, "Data Mining: Practical Machine Learning Tools and Techniques", Second Edition, Morgan Kaufmann Publishers, San Francisco, 2005.

[16] Cunningham S. and Holmes G., "Developing innovative applications in agriculture using data mining", Southeast Asia Regional Computer Confederation Conference (SEARCC'99), 1999.

[17] Robert C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets", Computer Science Department, University of Ottawa, Vol.11 No. 1, Canada, 1993.