ADABOOST ENSEMBLE WITH SIMPLE GENETIC ALGORITHM FOR STUDENT PREDICTION MODEL

AhmedSharaf ElDen[1] Malaka A. Moustafa[2]Hany M. Harb[3]AbdelH.Emara[4]

[1] Information Systems Dept., Faculty of Computer and Info.Sys.,Helwan University
profase2000@yahoo.com
[2] Al-Alson Higher Institute of Tourism, Hotels and Computer, Nasr City, Cairo
Samar2698@hotmail.com
[3,4] Computers and Systems Engineering Dept., Faculty of Eng., Azhar University
harbhany@yahoo.com abdemara@yahoo.com

ABSTRACT

Predicting the student performance is a great concern to the higher education managements.This prediction helps to identify and to improve students' performance.Several factors may improve this performance.In the present study, we employ the data mining processes, particularly classification, to enhance the quality of the higher educational system. Recently, a new direction is used for the improvement of the classification accuracy by combining classifiers.In thispaper, we design and evaluate a fastlearning algorithm using AdaBoost ensemble with a simple genetic algorithmcalled "Ada-GA" where the genetic algorithm is demonstrated to successfully improve the accuracy of the combined classifier performance. The Ada-GA algorithm proved to be of considerable usefulness in identifying the students at risk early, especially in very large classes. This early prediction allows the instructor to provide appropriate advising to those students. The Ada/GA algorithm is implemented and tested on ASSISTments dataset, the results showed that this algorithm hassuccessfully improved the detection accuracy as well as it reduces the complexity of computation.

KEYWORDS

Data mining, AdaBoost, Genetic Algorithm, Feature Selection, Predictive Model,ASSISTments Platform dataset.

1. INTRODUCTION

Predictingthe student performance is an important issue in e-learning environments. Student academic performance is based upon diverse factors such as personal, social, psychological issues and other environmental variables. Data Mining techniquesare a promising tool to attain these objectives. Data mining techniques are used to discover hidden patterns and relationships on a large amount of data which may be helpful in decision making.Classification is one of the most useful predictive data mining techniques used with e-learning,it maps data into predefined groups of classes,and it is often referred to as supervised learning (because the classes are determined before examining the data). Predictive models aim to predict the unknown values of variables of interest given known values of other variables.

The prediction of student performance with high accuracy is beneficial for identifying the students with low academic achievements. It is required that the identified students can be assisted more by the teacher so that their performance is improved in future [1].

DOI : 10.5121/ijcsit.2013.5207

This work aims to build a prediction model by analyzing the factors that affect the performance of the students using the “ASSISTments Platform dataset”.It is a web-based tutoring system developed at Worcester Polytechnic Institute [2]. In previous work [3], feature selection techniques were applied to that dataset to select the most relevant featurestoreduce the size of the dataset. The experimental results show that all the classifiers give the best performance with 3-7 features (out of sixteen) [3]. The proposed prediction model “Ada-GA” is implemented using the AdaBoost algorithm, with simple genetic algorithm, and it is found that the proposed “Ada-GA” classification model reduces the complexity of computation, while maintaining high detection accuracy compared to others. The prediction model would predict if the student will answer a certain problem true or false.

2. RELATED WORK

Al-Radaideh, et al.(2006)[4],studied the performance of a decision tree model by using three different classification methods (ID3, C4.5, and the NaïveBayes) to predict the final grade of students who studied the C++ course in Yarmouk University, Jordan in the year 2005. They found that the prediction of the decision tree model was the best one than the other models used.

Kotsiantis et al. (2003)[5]compared six classification algorithms to predict student drop-outs. The number of instances in the dataset was 350 containing numeric and categorical data. They showed thatNaïveBayes and neural networks were the best performed algorithms.

Wilhelmiina and Vinni (2006)[6],compared five classification methods for predicting the course outcomes by using very small datasets (125 rows and 88 rows). For numerical data, multiple linear regression and vector machine classifiers were used, while for categorical data, three variations of NaïveBayes classifier were used. They conclude thatNaïve Bayes classifier was the best classification method.

Dekker et al. (2009)[7], presented a case study to predict student drop-out demonstrating the effectiveness of several classification techniques and the cost-sensitive learning approach on several datasets over 500 instances with numerical and nominal attributes. They found that the use of simple classifiers (J48, CART) give useful results compared to other algorithms such as Bayes Net or JRip.

Kalles and Pierrakeas(2004)[8],studied the performance of different machine learning techniques:(decision trees, neural networks, Naive Bayes, instance-based learning, logistic regression and support vector machines). In addition, they compared themwith genetic algorithm basedon induction of decision trees. They analyze the students’ academic performance, as measured by the student’s homework assignments, and derived short rules that explain and predict success/ failure in the final exams.

Tuomas T.and Hannu T. (2010) [9],tack the problem of predicting student performance in anonline courses to identify students who have a high risk of failing by usingk-nearest neighbor method (KNN),theyfound that the KNN can predict student performance accurately, and even early at the first lessons. Furthermore,they conclude that early tests on skills can be strong predictors for final scores and for other skill-based courses. The results of their experiments are useful for teachers so they can quickly focus their attention to the students who need help.

3. THE PROPOSED PREDICTIONMODEL: ADA-GA

The objectives of this paper are framed so as to assist the low academic achievement by constructing a prediction model called “Ada-GA”, usingthe AdaBoost algorithmandgenetic algorithm, and then validating the developed model with the ASSISTments Platform dataset.The next subsections will describe the AdaBoost algorithm, the genetic algorithm and a brief description of the proposed algorithm “Ada-GA”.

The researcher in this field decided to use the DM techniques (classification tree models) because of some advantages they may have over traditional statistical models.

Typically, DM differs from traditional statistics on two issues: First, they can handle a large number of predictor variables, far morethan the traditional statistics. Secondly, the DM techniques are non-parametric and can capture nonlinear relationships and complex interactionsbetween predictors and dependent variable[10].

3.1 Boosting and AdaBoost

Boosting is a general method for improving the accuracy of any given learning algorithm. This is a widely used and powerful prediction technique that sequentially constructs an ensemble of weak classifiers. A weak classifier is a very simple model that has just slightly better accuracy than a random classifier, which has 50% accuracy on the training data set. The set of weak classifiers is built iteratively from the training data over hundreds or thousands of iterations. At each iteration or round, the examples in the training data are reweighted according to how well they are classified (larger weights given to misclassified examples). Weights are computed for the weak classifiers based on their classification accuracy. The weighted predictions from the weak classifiers are combined using voting to compute a final prediction of the outcome [11].

AdaBoost: AdaBoost is the most common boosting algorithm for binary classification which was proposed by Freund and Schapire [12]. It takes as input a training set “**S”** of “**m”** examples ($S =\{(x_1,y_1),...,(x_m ,y_m)\}$), where each instance(examples) x_i is a vector of attribute values that belongs to a domain or instance space *X*, and each label y_i is the class label associated with x_i that belongs to a finite label space $Y=\{-1, +1\}$for binary classification problems. Figure1,illustratea generalized version of the AdaBoost algorithm for binary classification problems.

```
Given:sequence of m instances $S=\{(x_1,y_1),\ldots, (x_m,y_m)\}$where $x_i$□$X$with labels $y_i$□$Y=\{-1, +1\}$, weaklearning algorithmWeakLearn, T(number of iterations)
InitializeD_1(i)=1/m        for all i=1,…,m
For t =1 to T
1-CallWeakLearnusing distribution$D_t$
2- Get a weak classifier (hypothesis)$h_t: X \rightarrow \{-1, +1\}$
3- Calculate the error$\varepsilon_t= \sum_{i=1}^{m} D_t(i)[h_t (x_i) \neq y_i]$ of $h_t$
If $\varepsilon_t>1/2$then        set T=t-1 and abort loop.
    4- Set  $\alpha_t=\frac{1}{2}\ln\frac{(1-\varepsilon_t)}{\varepsilon_t}$
    5- Update$D_{t+1}(i)= \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$($Z_t$ is a normalization factor)
Output the final classifier$H(x)=\mathrm{sign}(\sum_{t=1}^{T} \alpha_t h_t (x))$
```

Figure 1: A generalized version of the AdaBoost Algorithm

Adaboost weights the training samples with the probability distribution D_t(x) (weight function over the training examples) in each iteration t. The learning algorithm (**WeakLearn**) is then applied to produce a classifier h_t with error rate ε_t onthe training examples ($\varepsilon_t$was used toadjust the probability distribution D_t(x)). The effect of the change in weights is to place moreweight on training examples that were misclassified by h_t and less weighton examples that were correctly classified in the last stage. In subsequentiterations, therefore, Adaboost tend to construct progressively more difficultlearning problems. This process continues for T rounds, and, at last the final classifier, *H*, is constructed by a weightedvote of the individual weak classifiers h_1,h_2,...,h_T . Each classifier is weighted according to its accuracy of the distribution $D_t$that it was trained on [12]. The weak classifier is the core of an AdaBoost algorithm, in this work, classification and regression tree (CART) algorithm, proposed by Breiman et al. [13], was used as **WeakLearn** toAdaBoost algorithm.

3.2 Genetic Algorithm Overview

Genetic algorithm (GA)is an evolutionary based stochastic optimizationalgorithm with a global search potential proposed by Holland (1973) [14]. GA is among the most successful class of algorithms under EAs(Evolutionary Algorithms) which are inspired by the evolutionary ideas of natural selection. They follow the principles of Charles Darwin Theory of survival of the fittest. However, because of its outstanding performance with optimization, GA has been regarded as a function optimizer.

The algorithm begins by initializing a population of solution (chromosome) andcomprises representation of the problem usually in the form of a bit vector. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated, using some measures of fitness (using an appropriate fitness function suitable for the problem). To create the next generation, new chromosomes, called offspring, are formed by either merging two chromosomes from current generation using a crossover operator or modifying a chromosome using a mutation operator. A new generation is formed by selecting; fitter chromosomes have higher probabilities of being selected. After several generations, the algorithms converge to the best chromosome, which hopefully represents the optimum or a suboptimal solution to the problem. Thethree principal genetic operators in GA involve selection, crossover, and mutation[15].Figure2 show the outline of the GA algorithm.

1. **[Start]** generate a random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** evaluate the fitness *f (x)* of each chromosome *x* in the population
3. **[New population]** creates a new population by repeating the following steps until the new population is complete
    - a) **[Selection]** selects two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    - b) **[Crossover]** with a crossover probability, cross over the parents to form a new offspring (children). If no crossover was performed, the offspring is an exact copy ofparents.
    - c) **[Mutation]** with a mutation probability, mutate new offspring at each locus (position in the chromosome).
    - d) **[Accepting]** place new offspring in a new population
4. **[Replace]** use new generated population for a further run of the algorithm
5. **[Test]** if the end condition is satisfied, **stop**, and return the best solution in current population
6. **[Loop]** go to step **2**

Figure 2: Outline of the Basic Genetic Algorithm

3.3 Overview of the Proposed Model: Ada-GA

Freund and Schapire [12] concluded that the AdaBoost algorithm is less vulnerable to overfitting problem compared to most learning algorithms, because boosting is sensitive to noisy dataand outliers. Thus mislabeled cases or outliers may cause the overfitting problem, for the new classifier focuses more on those observations that have incorrectly classified, thus produce a large number of weak classifier to achieve better performance [11].

In this study, we develop a new boosting algorithm called **“Ada-GA”** whichoptimizes the **number of weak classifiers** and their**weights** using a genetic algorithm, in order to improve the performance of boosting. The genetic algorithm can control the effects of outliers by the choice of an appropriate fitness function that limits the number of weak classifiers and thereafter improve the predictive accuracy. The final model has an interpretability advantage over other boosting algorithms due to the reduced model complexity, the“**Ada–GA**”procedure is summarized in Figure3.

```
Input:a set S of m instances: S={(x_1,y_1) ,…, (x_m,y_m)}  where x_i□X with
labels y_i□Y={−1, +1}, P (population size), G (maximum number of
generations), T(initial number of weak learners).
Initialize: a randomly generated population of m solutions (consists of a set
of Tweak learners withtheir weightsproduced by AdaBoost)
Evolve: fork = 1, 2, …, G.
1-Generate a population of bit strings b
2- Evaluate the fitness of the solutions: $f(b)=w_1 * (1-L/T)+w_2 * (1/E_b^s)$
 where b isthe evolved best individual, $w_1$, $w_2$ are fitness weights ,$E_b^s$is
the validation error and $L=\sum_{i=1}^{T} b_i$
  3-Use vector b to update the weak classifiers (T)and their weights.
4-Produce new generation of solutions using the genetic operations
(selection, mutation and crossover).
   5- If end condition, stop, and return the best solution; else loop
Output:final hypothesis (with optimized classifiers and their weights).
```

Figure3: The Proposed Procedure of Ada-GA

The structure of “**Ada-GA”** is detailed in Figure4 which consists of three following phases:

1) **Pre-processing and featureextraction phase**: in this phase, randomly two separated training and testing datasets are selected from the ASSISTments dataset, the symbolic features are converted into numerical ones, and the mostsuitable features are selected. In our previous work we use six classifiers from different categories to rank the 15 features of ASSISTments dataset, then the same classifiers haven used on the ranked features to acquire the optimal subset of features. Weka have been used (open source machine learning software) to bring out an extensive performance comparison among the six classifier algorithms. The experimental results show that all classifiers give the best performance with only 3-7 features at most which mean that the input dataset has 80% -53.3% reductions in size. TheASSISTments dataset showed in Table 1 and the used dataset (after feature selection) showed in Table 2.

2) **AdaBoost training and testing phase**: this phase consists of two sub-phases including training sub-phase and testing sub-phase;

a. Training sub-phase: In this phase AdaBoost is trained using the trainingset. It iterates T rounds of AdaBoost training, produces numbers of weak classifiers h_t and ensembles weights α_t is yielded by learning to constitute the final strong classifiers. The size of the training sets is shown in Table 2.

b. Testing sub-phase: the performance of the system is measured with the testing set to choose the best hypothesis. The size of the testing sets is shown in Table 2.

3) **Post optimization procedure phase:**this phaseis composed of three parts: (i) initialization with *AdaBoost*(ii) fitness function, and (iii) genetic evolution.

i. *Initialization with AdaBoost*: initialize all individuals in the initial population with some proportion of the final classifierproduced by AdaBoost after a certain number of iterations (consists of a set of Tweak learners with their weights).

ii. *Fitness function:*it measures the goodness of a solutionwhich consists of a set of weak learners and their weights. Higher fitness solutions will influence the next generation more than lower fitness solutions. The following fitness formula is used to measure the fitness of the solutions.

$\boldsymbol{f(b)=w_1* (1-L/T)+w_2* (1/E_b^s)}$Where

$E_b^s$the validation error for a validation set S,

Lthe numbers of the selected weak classifiers($L=\sum_{i=1}^{T} b_i$),
Tthe total number of weak classifiers,
w_1 and $w_2$the fitness weights

iii.*Genetic evolution:*the evaluation of the fitness of new solutions is performed for each generation, and the evolution process is continued until some condition is satisfied. The solution in the final generation with the largest fitness value is chosen as the final solution. Each solution in the evolving population is a vector b composed of bit string (b_1, b_2, …, b_T) denoting the weak classifiers which constitute the strong classifier. The b_i=1 indicates the appearance of the i^{th} weak classifier in the ensemble while b_i=0 indicates its absence.

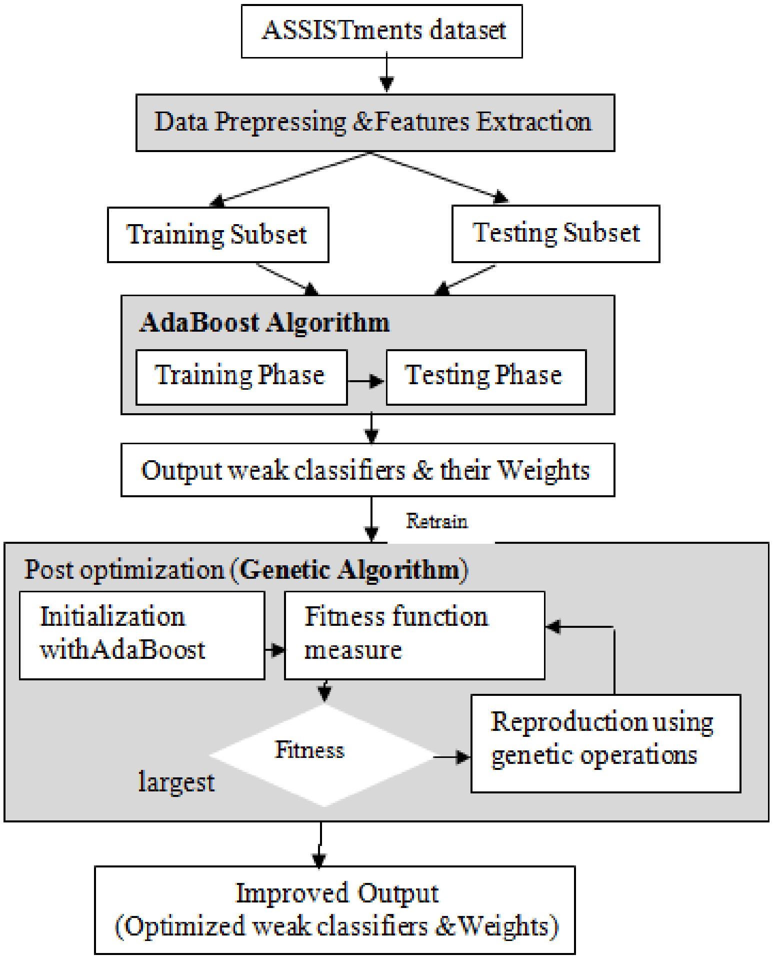
ASSISTments dataset
Data Prepressing &Features Extraction
Training Subset
Testing Subset
AdaBoost Algorithm
Training Phase
Testing Phase
Output weak classifiers & their Weights
Retrain
Post optimization (Genetic Algorithm)
Initialization withAdaBoost
Fitness function measure
Reproduction using genetic operations
Fitness
largest
Improved Output
(Optimized weak classifiers &Weights)

Figure4: The Structure of the Proposed Model Ada-GA

4. Experimental Results

The experiments have been run using Matlab 9, on a system with a 2.66GHZ Intel Core (i5-560M) processor and 4 GB of RAM running Microsoft Windows 7 Home Premium (64-bit). In the experiments, the dataset used are from the ASSISTments Platform, (a web-based tutoring system developed at Worcester Polytechnic Institute) and have been collected from 4th to 10th grade math students and it is available for the 2011 Knowledge Discovery in Educational Data workshop.It consists of approximately one million students record which contains 19 features (attributes), three irrelevant features out of them are discarded. Table 1 shows the features description [2].

Table 1: The 16 features of ASSISTments dataset

Number	Features	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	The ID of the ASSISTment (consists of one or more problems)
4	Problem_id	The 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	The 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	TheIDs of the skills associated with the problem
14	Teacher_id	The ID of the teacher
15	School_id	The ID of the school
16	Correct	Often used as target for prediction (1=correct,0=incorrect)

In our previous work [3] we have applied different feature selection techniques to get the optimal subset of features. Two subsets(of the ASSISTments dataset)called Dataset-One and Dataset-Two are used in this work (described in Table 2).

Table 2: Description of the Dataset Used

	Training Set Instances	Testing Set Instances	Selected Features	Numbers of Attributes
Dataset-One	10,841	3,098	4,6,8,16	4
Dataset-Two	387,439	110,697	4,6,8,16	4

The AdaBoost algorithm and the weak learner employed by our algorithm are implemented by the GML AdaBoost Matlab Toolbox developed by Alexander Vezhnevets [16]. Theparameters used for evolution are crossover rate=1 and mutation rate=0.003. Max iteration, the population size and the number of generations are given in Table 3.

Table3: The Parameter Settingsused by Ada-GA

Max-Iteration	**Population size**	**Number of generations**
5	100	10
10	100	10
15	100	15
20	100	15
25	150	20
30	150	25
35	150	20
40	200	25
45	250	30
50	350	35
55	350	35

In our proposed algorithm, the GA terminates if there is no better individual within the next 35 or 120 generations (Table 4 and Table 5 respectively) or if the validation error is equal to zero. The CART algorithm was used as the AdaBoost weak learner with “two” tree split in Table 4 and Table 6 and “three” tree split in Table 5 and Table 7.

Two experiments were done, one used the dataset-One and the other used the dataset-Twoshown in Table 2, the results of the first experiment are shown in Table 4 and Table 5 and the results of the second are shown in Table 6 and Table 7.

Experiment one:As illustrated in Table 4, the numbers of weak classifiers of the proposed classifier trained by standard AdaBoost reduced by about 62.8% due to using the genetic algorithm optimization while the accuracy of the proposed algorithm is increased by 0.28%, as shown in Figure5.

Table 4: Comparison of AdaBoostClassifier (two tree split) and the Proposed Ada-GA Classifier Using Dataset-One

Iteration	**AdaBoost Classifiers**	**Ada-GA Classifiers (Proposed)**	**Pop.-Size**	**Num.of Generation**	**AdaBoost Accuracy**	**Ada-GA Accuracy (Proposed)**
5	11	4	100	10	81.2137	81.2137
10	26	7	100	10	81.7624	**81.7947**
15	41	13	100	15	**81.9238**	81.8270
20	56	17	100	15	81.6656	**81.7301**
25	71	26	150	20	81.9884	**82.2466**
30	86	22	150	25	81.6010	**82.2466**
35	101	30	150	20	81.6656	**82.0529**
40	116	48	200	25	81.6010	**82.1498**
45	131	54	250	30	82.3112	**82.6985**
50	146	58	350	35	82.1498	82.1498
55	161	73	350	35	82.1498	**82.2466**
Average	**86**	**32**			**81.8485**	**82.0771**

As shown in Table 4, the accuracy of the AdaBoost algorithm is the same, (for 50,55iterations), so the experiment was done again with tree split=3 (for the CART algorithm)and the results aredetailed in Table 5. As shown in Table 5, the AdaBoost accuracy and weak classifiersare increased as max-iteration increased, whileAda-GA classifiers are reduced and Ada-GAaccuracy isslightly increased.

Table 5: Comparison of AdaBoost Classifiers(three tree split) and the ProposedAda-GA Classifiers Using Dataset-One

Iteration	**AdaBoost Classifiers**	**Ada-GA classifiers (Proposed)**	**Pop.-Size**	**Num.of Generation**	**AdaBoost Accuracy**	**Ada-GA Accuracy (Proposed)**
60	234	119	100	20	82.3112	82.4726
70	274	143	100	20	82.3112	82.5048
80	314	150	150	30	82.2466	82.4403
100	394	195	200	50	82.4080	**82.3757**
120	474	218	200	70	**82.4403**	**82.3757**
140	554	251	250	90	82.3112	**82.3757**
160	634	287	300	90	**82.4080**	**82.2466**
180	714	354	300	100	**82.2143**	**82.1498**
200	794	365	350	120	**82.2143**	**82.1498**
Average	**487**	**231**			**82.3183**	**82.3434**

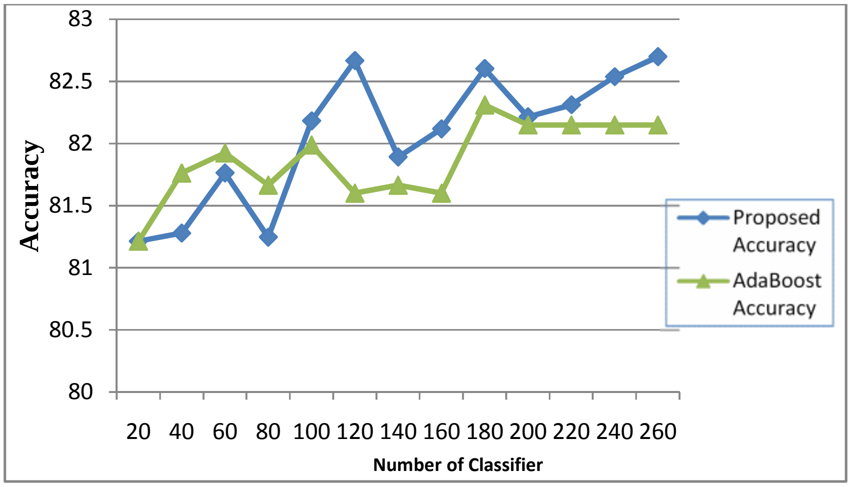

Figure5: AdaBoost Accuracy and Ada-GA Accuracy

Experiment two:as shown in Table 6 and Table 7, the accuracy of the AdaBoost algorithm and Ada-GA are decreased (using Dataset-Two) than the accuracy shown in Table 4 and Table 5(using Dataset-One). As illustrated in Table 6, the numbers of weak classifiers of AdaBoost algorithm are reduced by about 58.2% due to using the genetic algorithm optimization and the accuracy of Ada-GA algorithm is slightly increased than theaccuracy of the AdaBoost algorithm.

Table 6: Comparison of AdaBoost Classifier (two tree split)
and the Proposed Ada-GA Classifier Using Dataset-Two

Max-iteration	**AdaBoost Classifiers**	**ADa-GA Classifiers (Proposed)**	**Pop-Size**	**Num.of Generation**	**AdaBoost Accuracy**	**Ada-GA Accuracy (Proposed)**
20	60	12	100	20	77.9569	78.0771
50	150	44	250	35	77.9922	78.1322
70	210	58	250	50	78.1069	78.1322
100	300	96	300	60	78.1186	78.1692
150	450	157	300	60	78.1457	78.1909
200	600	270	350	80	78.1457	78.2072
260	780	331	200	80	78.1602	78.2785
300	900	418	250	80	78.1602	78.2325
350	1050	496	250	80	78.1602	78.2794
Average	**500**	**209**			**78.1051**	**78.1888**

Table 7, shows that when using “three” tree split with CART algorithm, the AdaBoost accuracy is increased than when using “two” tree split and therefore the Ada-GA accuracy is slightly increased. Also, the numbers of weak classifiers of AdaBoost algorithm are reduced by about 55.5% when using the proposed algorithm Ada-GA.

Table 7: Comparison of AdaBoost Classifier (three tree split)
and the Proposed Ada-GA Classifier Using Dataset-Two

Max-iteration	**AdaBoost Classifiers**	**ADa-GA classifiers (Proposed)**	**Pop-Size**	**Num.of Generation**	**AdaBoost Accuracy**	**Ada-GA Accuracy (Proposed)**
50	200	32	200	70	78.0798	78.1909
100	400	139	200	50	78.1051	78.1710
150	600	257	200	80	78.1367	78.1855
200	800	346	300	80	78.1060	78.1945
250	1000	456	300	100	78.2523	78.2361
300	1200	543	400	80	78.2343	78.3038
350	1400	617	300	100	78.3065	78.2081
400	1600	774	300	80	78.3653	78.3635
450	1800	837	300	100	78.3824	78.2605
Average	**1000**	**445**			**78.2187**	**78.2349**

Finally, a comparison of the prediction accuracy of the proposed algorithm (from Table 4) and other classification algorithms [3] (using ASSISTments dataset) is shown in Table 8. We can see from Table 8, that the proposed algorithm outperforms the others.

Table 8: Prediction Accuracy Comparison of Some Classification Algorithms and Proposed Algorithm

Classifiers	Prediction Accuracy(%)
ONER	80.87
J48	79.67
VFI	75.53
IBk	81.33
NaiveUpdate	80.65
KMeans	79.49
AdaBoost	81.85
Proposed model	**82.07**

5. CONCLUSIONS

Within this paper we have studied the problem of predicting the student performance using a web-based tutoring system(ASSISTments Platform dataset). With reliable predictions of the performances of the students, the teachers can focus their efforts on those students who are likely failingin the final test, and thus helping them before difficulties become overwhelming.

In addition, we have applied data mining classification techniques to improve the prediction results of the student academic performances. Also, we have introduced a new boosting method called “Ada-GA”, with higher accuracy and efficiency.In this correspondence, we apply the AdaBoost ensemble algorithm with Genetic algorithm.The experimental results showed that using genetic algorithm with boosting is another alternative boosting technique that produces better solutions (fewer weak classifiers and a slight increase of the classification accuracy) than the classical AdaBoost produces.

Finally, we compared the proposed algorithm with some classification algorithms as shown in Table 8, and the result showed that the proposed algorithm outperforms the others.

REFERENCES

[1] Bhardwaj K. andPal S., (2011)"Data Mining: A prediction for performance improvement using classification", (IJCSIS) International Journal of Computer Science and Information Security,Vol. 9, No. 4,pages 136-140.

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

[3] Hany M. and Malaka A., (2012)"Selecting Optimal Subset of Features for Student Performance Model",(IJCSI) International Journal of Computer Science Issues, Vol. 9, No. 1,pages253-262.

[4] AI-Radaideh Q.,AI-Shawakfa E.,and AI-Najjar M., (2006) “Mining student data using decision trees”, (ACIT’2006) International Arab Conference on Information Technology, Yarmouk University, Jordan.

[5] Kotsiantis S., Pierrakeas C., and Pintelas P., (2003)“Preventing studentdropout in distance learning using machine learning techniques”. InKES,Springer-Verlag Berlin Heidelberg,pages 267-274.

[6] Wilhelmiina H. and Vinni M., (2006)“Comparison of machinelearning methods for intelligent tutoring systems”,Intelligent Tutoring Systems, Vol. 4053, of Lecture Notes in Computer Science, pages 525-534.

[7] Dekker G., Pechenizkiy M., and Vleeshouwers J., (2009) "Predicting students drop out: a case study", Proceedings of the 2nd International Conference on Educational Data Mining, pages 41-50.

[8] Kalles D. and Pierrakeas C., (2004) "Analyzing student performance in distance learning with geneticalgorithms and decision trees", Laboratoryof Educational Material and Educational MethodologyHellenic Open University, Patras, Greece.

[9] Tanner T. and Toivonen H., (2010) "Predicting and preventing student failure – using the k-nearest neighbour method to predict student performance in an online course environment",International Journal of Learning Technology(IJLT) ,Vol. 5, pages 356-377.

[10] Zlatko J. Kovačić, (2010) "Early Prediction of Student Success: Mining Students Enrolment Data", Proceedings of Informing Science & IT Education Conference (InSITE).

[11] Dong-Yop Oh, (2012) " GA-BOOST: A Genetic Algorithm for Robust Boosting",A PhD thesis presented to the University of Alabama,Tuscaloosa, Alabama.

[12] Freund Y.andSchapire R.,(1996) "Experiments with a new boosting algorithm," in the proceedings of the13th International Conference on Machine Learning,Italy, pages 148 -156.

[13] Breiman L., Friedman J., Olshen R., and Stone C., (1984) "Classification and Regression Trees", Chapman and Hall, New York, USA.

[14] HollandJ.H., (1973) "Genetic algorithms and the optimal allocation of trials", SIAM Journal on Computing, Vol. 2, No. 2, pages 88-105.

[15] Binitha S., and Sathya S., (2012) "A Survey of Bio inspired Optimization Algorithms", (IJSCE) International Journal of Soft Computing and Engineering , Vol. 2, Issue-2.

[16] VezhnevetsA., (2006) , GML AdaBoost Matlab Toolbox [http://research. graphicon.ru/], Graphics and Media Laboratory, Computer Science Department, Moscow State University, Moscow, Russian Federation.