JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 243-257 (2021)
DOI: 10.6688/JISE.202101_37(1).0015

Building Student Course Performance Prediction Model Based on Deep Learning

JONG-YIH KUO[1], HAO-TING CHUNG[1], PING-FENG WANG[2] AND BAIYING LEI[3]
[1]*Department of Computer Science and Information Engineering*
National Taipei University of Technology
Taipei, 106 Taiwan
E-mail: jykuo@ntut.edu.tw; colin19940702@gmail.com
[2]*Institute for Information Industry*
Taipei, 106 Taiwan
E-mail: pfwang@iii.org.tw
[3]*School of Biomedical Engineering*
Health Science Center
Shenzhen University
Shenzhen, 518060 P.R. China
E-mail: leiby@szu.edu.cn

The deferral of graduation rate in Taiwan's universities is estimated 16%, which will affect the scheduling of school resources. Therefore, if we can expect to take notice of students' academic performance and provide guidance to students who cannot pass the threshold as expected, the waste of school resources can effectively be reduced. In this research, the recent years' student data and course results are used as training data to construct student performance prediction models. The *K*-Means algorithm was used to classify all courses from the freshman to the senior. The related courses will be grouped in the same cluster, which will more likely to find similar features and improve the accuracy of the prediction. Then, this study constructs independent neural networks for each course according to the different academic year. Each model will be pre-trained by using De-noising Auto-encoder. After pre-training, the corresponding structure and weights are taken as the initial value of the neural network model. Each neural network is treated as a base predictor. All predictors will be integrated into an Ensemble predictor according to different years' weights to predict the current student's course performance. As the students finish the course at the end of each semester, the prediction model will continue track and update to enhance model accuracy through online learning.

Keywords: deep learning, neural network, de-noising auto-encoder, ensemble learning, prediction model

1. INTRODUCTION

According to the statistics of the Chief Officer of the Executive Yuan [1], The deferral of graduation rate in Taiwan is about 16% in the 2016 academic year. When the number of students who postpone graduation increases it will affect the scheduling of school resources, including computer equipment, classroom space, and curriculum configuration. From the social cost side, students may be dissociated from society and even went astray if students dropped out of school when they do not plan for the future or have a professional skill. Baas [2] noted that the costs of dropout are six times more than preventing students

Received December 10, 2018; revised March 2, 2019; accepted June 4, 2019.
Communicated by Hung-Yu Kao.

from dropping out the school. From the above mentioned, it is extremely important for duly counseling students who have high possibility of dropout. In order to let students to graduate as scheduled, it is necessary to review the student's academic performance. In this way, the teachers and teaching assistants can provide guidance to those students who cannot pass the threshold as expected.

However, predicting student performance within a degree program faces some challenges [16]. First, students can differ tremendously in terms of backgrounds, and the same course can be taken by students in different areas. Since predicting student performance in a particular course relies on the student past performance in other courses. The key challenge for training an effective predictor is how to handle heterogeneous student data due to different areas and interests [17]. Similarly, predictions of students' performance in courses are often based on in-course assessments which are designed to be the same for all students [10]. Second, students should take many courses but not all courses are equally informative for predicting students' future performance. Utilizing the student's past performance in all courses that he/she has completed not only increases complexity but also introduces noise in the prediction, thereby degrading the prediction performance. Third, the collection and cleanup of data is difficult. The number of students is limited, so as the training data set is limited. However, the grades and student data attributes is diverged to easily cause overfitting. This study should train with different number of layers and the number of neurons, to cluster all the courses, and to filter the effective attributes.

This research proposed a deep learning model that used the student academic big data in the school information system to build a prediction model. First, this study grouped all the courses data of the students into different clusters by *K*-Means method. After clustering phase, the training input data may be more useful to be the predicting data. The *K*-Means method was used to classify all the courses in the first to fourth grades so that those courses have the same attributes and can be grouped together in the same cluster, which helps the model to find similar features faster when training, to improve the accuracy of the analysis. If all the courses are included in the training data set, the feature vector will be it is very large and the complexity is increased, thus reducing the accuracy of prediction. Therefore, this study used a compulsory course for students to do training. Before training phase, the model applied De-noising autoencoder as pre-training method to learn more stable data features. During the training phase, this study applied dropout, cross-validation method to prevent overfitting, and to integrate all model of each school year according to different weights to predict students' score. The current student data is used to track and update the proposed models continuously. Our proposed system can continuously track and update the model and increase the robustness of the model. For the school administrator, predicting a student's performance in the course is an ongoing task. As the student completes the course every semester, the prediction model needs to be continuously tracked and updated.

The remainder of this paper is organized as follows. Section 2 presented related work and method for finding the solution. Section 3 presented an effective model to improve the prediction for the student course performance. Section 4 conducted two experiments including the comparison for the effectiveness of using a specific algorithm and comparison the other machine learning method. Section 5 summarized the conclusions and the contributions of this paper.

2. RELATED WORK

Hinton *et al.* [5] proposed an Autoencoder model that is divided into two parts: an encoder and a decoder. In order to make the data reconstructed by the encoder and decoder close to the original input data x, the model will use mean square error as the loss function to measure the error between the input data and the reconstructed data. In order to let the hidden layer learn more stable features, Vincent *et al.* [15] mentioned that some random noises can be added to the input layer of the network data, which is called De-noising Autoencoder (DAE). Let the decoder restore the data without adding noise, a model can learn and eliminate the noise by itself. This model can be regarded as the pre-training of the neural network.

Freund *et al.* [4] proposed AdaBoost iterative algorithm that a weak classifier is added to each round. The misclassified data of the previous predictor will have a higher probability to be used to train in the next predictor. At the beginning of the training, each data will have the same weight, representing the probability that it will be selected into the training set. If the data is correctly classified, its weight will decrease, and in the next training, the probability of the data being selected will decrease; on the contrary, if the data is not misclassified, its weight will increase. Breiman [3] proposed the Bagging algorithm that is given a training set of size n, each time select n' data and put it back into the training set, and finally generate m subsets of size n' as new training and then build m models in sequence. Finally, integrate the results of all models. The Bagging is applied to complicated and overfitting model, since Bagging averages each model, and the final result will approach the overall average performance, so the variance will decrease and the possibility reducing the overfitting. Kohavi [8] proposed a cross-validation approach that effectively improves the accuracy of the model. A K-fold cross-validation method first cuts the training data into K equal parts, and the first set is used as the test data for validation, and the other $K-1$ sets are used for training. The cross-validation repeats K times, and finally the average the accuracy of K times to obtain a result without deviation. Ng [12] mentioned that using the best training results in K cross-validation as a prediction model is not better than average K time's cross-validation model.

Tanuar *et al.* [13] integrated generalized linear model, deep learning and decision tree techniques to predict the student's final year GPA. The data used in their experiment are from the computer science subjects, 6 subjects, 1 laboratories results and the GPA on their graduation year. According their experimental results, the important factors can be extracted to help students prepared themselves earlier. The accuracy results of the three proposed predication approaches are just 66.6%, 67.6%, and 60.6%. Tsiakmaki *et al.* [14] studied the predicting university students' grades based on previous academic achievements. They carried out several experiments using eight courses modeled by some familiar mining methods, including linear regression, support vector machines, decision trees, M5 rules, and k-nearest neighbors. The evaluation metric used in their study for determining the efficiency of each regression method is the Mean Absolute Error but not the accuracy. Xu *et al.* [16] constructed a two-layer structure to use the students' materials for three years, including the students' high school GPA and SAT scores. Through the two-tier architecture model, the data-driven approach can predict the GPA scores of college graduation, and discover the correlation between courses, and to establish a series of vector combinations of student learning results, which can effectively reduce the complexity between courses. However, this method is not suitable for us because the students' high school grades do not fully implement the GPA method to evaluate the scores, so that the forecast results should have a large error.

Our previous work [9] is to build a deep network model using the architecture of the Stacked De-noise Autoencoder, using the average ranking of the department, the average grade of the semester, and the scores of the professional subjects to establish the model. The study grouped the student's data by the course. The predictors for each course of the semester are different through the overall learning and training methods. The overall predictor performance proposed in the study is better than not using the overall predictor, and it is more accurate to predict the results. Our proposed approach adopts the idea of using students' relevant attributes as training data, then further analyzed each subject score and made a more accurate judgment to let teachers clearly understand the learning effectiveness of students. The study [16] predicted the finally college GPA and the study [9] predicted the dropout rate of a student. However, neither of these studies can truly reflect the performance of students. Therefore, our proposed approach takes advantage of the structure or method of the two studies, respectively, to focus on building a better, more accurate model, and to predict the score of each subject of students.

3. THE PROPOSED APPROACH

In order to use the data efficiently, this study divided the data into training data, validation data and test data to train, fine-tune and evaluate the performance of the model. The process of model is shown in Fig. 1.

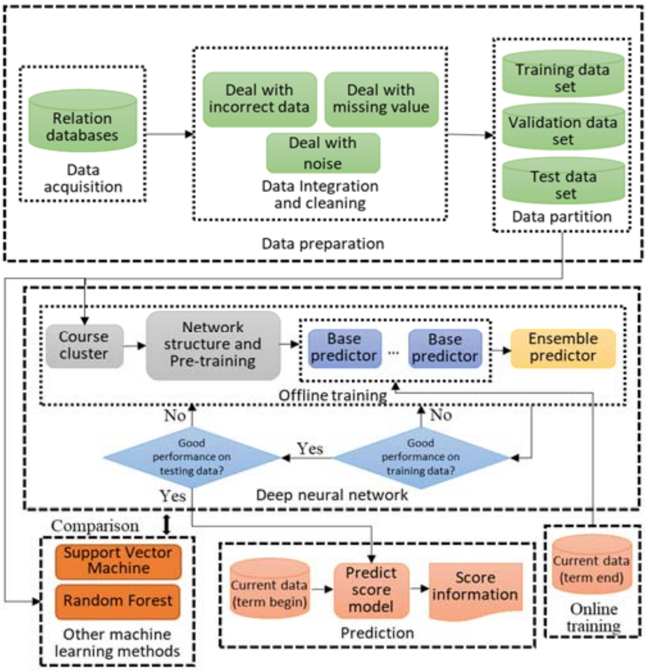

Fig. 1. The process of the model.

(A) Data Preparation

The quality of the training data affects the learning results of neural network and the accuracy of classification and prediction. In order to ensure the correctness of the model, this research performed data reprocessing and data transformation. This research used the course grades and related attributes of the Department of Electrical Engineering students in the 2010-2017 academic year as training data sets. The predictive target is those courses that are currently not taken by those students.

Table 1. Student course fields.

Required course name		Type	Values
Physics 1&2	Digital Logic	Numeric	[0, 100]
Physics Lab. 1&2	Programming and Lab		
Circuit Theory 1&2	Probability		
Engineering Mathematics 1&2	Microprocessor		
Electronics 1&2	Signals and Systems		
Electronics Lab. 1&2	Special Projects 1&2		

Table 2. Student related attributes.

Attributes	Values
Sex	Male, Female
Admission Type	{Joint University Entrance Exam, Recommendation and Screening-Based Admission, Admission, Star Plan, Transfer, Foreign Admission, Special Achievement and Screening-Based Admission, NA}
Birthplace	{North, Middle, South, East, Others}
Identity	{Normal Students, International Students, Others}
Father Education	{Senior, College, Junior, Bachelor, Master/ Ph.D., Others}
Mother Education	
Father Occupation	{Housewife, Laborer, Business, Service, Freelance, Officer, Others}
Mother Occupation	

The criteria of the course selection varied depending on the interest and learning field. Also, some courses are newly created and some have been closed. If all courses are included in the training data set, the feature vector will become very large. It is not only increase the complexity and also reduce the prediction accuracy. Therefore, this study focus on to predict the required subjects. The data field descriptions are shown in Tables 1 and 2, respectively.

Data Integration and Cleaning: The quality of the data and the selection of features greatly affect the analysis results of the model. There are three types of data preprocessing method. The value of training data must limit the range from 0 to 1 to normalize the weight of the neural network model in the training process. The formula is shown as follows:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}. \tag{1}$$

X_{max} and X_{min} are the maximum and minimum values of each attribute. By Eq. (1), the range of the attribute can be limited 0 to 1. There are two processing methods to deal with the missing values for the training data, one is to directly discard, and the other is to replace the average value of all the data in the column. For example, if a student data lacks too many course grades, this study directly deleted this data. If there is only one or two-course grade missing, the average score of that subject is taken. Non-numeric data must be encoded into numbers as the input training data. The training data can be separated into ordered and disordered. This study used one-hot encoding and Label Encoding to encode the disordered and ordered data. For example, the identity of the normal student is 100, the foreign student is 010, the others are 001. The parent's education level is ordered, according to the level of education, labeled in the range of 0 to 1.

The related attributes and course scores of students in the 2010-2016 academic year are used as training data, in which the *K*-Fold algorithm used for cross-validation. The 2017 academic year students are used as the testing data to evaluate the performance of the model. There are 22 required courses in the department of electrical engineering. It should be ineffective if all the completed courses are used to predict a new course because there is no correlation between many courses. For example, the correlation between calculus and physics may be small. However, the score of calculus 1 affecting calculus 2 may be more significant. The aim of this research is to find the course cluster of the related course so that the relevant courses were classified in the same cluster, as shown in Eq. (2). C^t represents the set of courses that have been completed up to the t semester, (j) indicates the set of courses in which the course j is located. The intersection vector of the two sets is regarded as the predictive input of the course j. The students and the corresponding course scores are organized into a matrix. The *K*-Means [6] is used to classify related courses to the same cluster.

$$\text{input of } j = C^t \cap k(j) \tag{2}$$

(B) Build a Deep Neural Network

A fully-connected four-layer deep neural network is used as the network architecture. The first hidden layer has 256 units, the second hidden layer has 128 units, and the output layer has one unit. The neural network of each course is considered as a predictor of the prediction model. Before training the neural network, we use DAE as the pre-training to initialize the value of the weight.

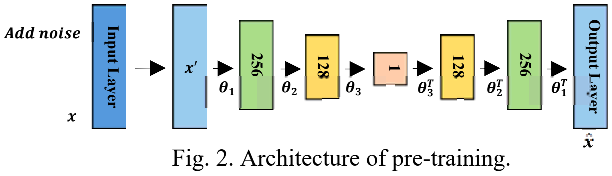

Fig. 2. Architecture of pre-training.

Pre-training: Because the traditional neural network usually uses 0 or a value close to 0 as the initial weight value. Such initial values tend to converge to a local minimum in the deep network of the multi-layer hidden layer. Using DAE as a pre-trained weight can reduce the probability of the neural network converge to a local minimum. The architecture of the pre-training model is shown in Fig. 2. The model is divided into two parts: encoder and decoder. Input $\boldsymbol{x}$ will first add some noise to prevent overfitting and increasing restore the ability of the model. $\boldsymbol{x}'$ reduces the dimension through neural network layer encoding, and then decodes the restored data into $\hat{x}$. The mean square error is used to calculate the loss between x and the decoded restored data $\hat{x}$ in each iteration, as Eq. (4) shown.

$$RMSE=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)},loss=RMSE+\alpha\times\frac{1}{2}\sum w_i \tag{4}$$

After the pre-training, the corresponding structure and weight θ are taken as the initial values of the neural network model. The pre-training algorithm is shown in Fig. 3. The initial value weight θ is set, and the iteration is 3,000 times. In each iteration, the training data will contain noise. The ratio c is 0.3. Adam method [7] is applied as the optimization method to reduce the loss to the lowest, after finish training, return the weight θ.

```
procedure De-noising Autoencoder($x$):
$\theta = \{W, b, b_{prime}\}$ which is the parameters
  for epochs in range(iteration):
  $x'$ = masking Noise($x$, $c$), $c$ is the corrupted level
  $h$ = ReLU($x' * W + b$), the encoder
  $\hat{x}$ = ReLU($h * W^T + b_{prime}$), the decoder
    loss = RMSE($x$, $\hat{x}$)
    Compute the gradients of the loss with respect to $\theta$
  end for
  return $\theta$
end procedure
```

Fig. 3. The pre-training algorithm of DAE.

Base Predictor: This study established a base predictor for each course in each academic year. Each base predictor is an independent neural network. The trained weight is used as the initial value of the predictor. As shown in Fig. 4, in each academic year, the study trained n base predictors h, n is the number of courses to be predicted. Finally, each base predictors h of the same course is combined into the ensemble predictor f.

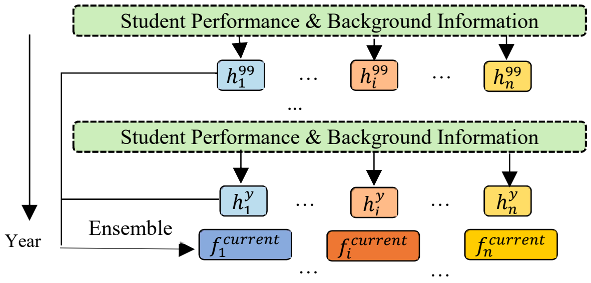

Fig. 4. Architecture of base predictor.

Ensemble Predictor: The ensemble predictor integrated the base predictor of all the same courses each year, as shown in Eq. (3). *Current* represents the current year, n is the course be predicted and $f_n^{Current}$ is the ensemble predictor of the course to be predicted. This study used the students' academic data for 99th to 106th school year as training data. "99th school year" is equal to A.C 2010. Therefore, the "totalYear" should be "Current-2010". The assignment of weights increased by the year as shown in Fig. 5. Because the closer year may be taught by the same teacher and there should be higher similarity in the scores.

$$f_n^{Current} = \sum_{y=99}^{Current-1} w_y \times h_n^y \tag{3}$$

```
$totalYear = Current - 99$
$index = totalYear \times (totalYear + 1) / 2$
for $y$ in $range(99, Current - 1)$:
    $w_y = (1/index) \times y$
```

Fig. 5. Algorithm of weight assignment.

Even though a particularly base predictor performs best during training, it may not have good predictive power because the scores for each semester will combine many different factors. If only use one specific base predictors, the result may be very unsatisfied. Therefore, integrating the results of the annual predictor can improve the generalization ability and robustness of the predictor.

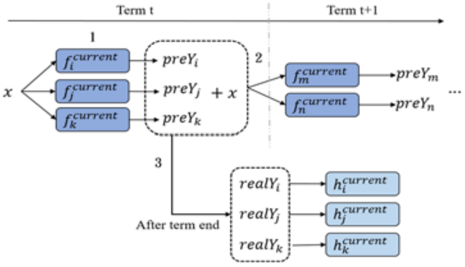

Fig. 6. prediction flow chart.

Combined neural network model h_i^y of each course in different years with different weights w_y to form an ensemble predictor, as shown in algorithms of Figs. 6 and 7.

```
$C^t$ is the course set that has been taken,
$K(i)$ is the course cluster set belongs to course $i$
for $i$ in $range(course)$:
    for $j$ in $range(year)$:
        Get the student training data of year $j$ and
        Select $C' \cap K(i)$ set as features
        $h_i^j$ ← Cross Validation
        Save each year's model $h_i^j$ in course $i$
    end for
    $f_n^{Current} = \sum w_y \times h_i^y$
end for
```

Fig. 7. Ensemble learning algorithm.

Prediction: The ensemble predictor is used to predict each course performance of current semester, and then use the predicted scores to predict future performance. By the end of the semester, this study further used the student's grades to update the model and establish a system that continuously tracks student grades and accurately predicts their future performance. Fig. 5 introduces the process of prediction: (1) Use the student's data x as input to fit into different ensemble predictors model $f^{Current}$ to predict the course score in the t-semester; (2) Use the predicted score $preY$ and x as new input to predict the score for the next semester, and so on, and get the score for each semester course iteratively; (3) After the end of the semester, use the actual score $realY$ to create a new base predictor model $h^{current}$. This research used CSV format to present the results of the prediction, showing the student class, id, name and the course score. The loss function is used to estimate the accuracy of the model's predicted value $f(x)$ and the true value y. The L2 regularization is used to measure the complexity of the model. Reduce unimportant weight values in the neural network.

After pre-training, every weight θ have been trained. The ReLU function [11] is used as the activation function of the network. The training of the model adopts K-fold cross-validation to divide the training data and validation data. The K value is 5 times. In order to ensure that the model is not over-trained, each cross-certification will cooperate with early stop. Although this method should reduce the accuracy of the model to the training data, it can effectively improve the versatility of the model. The root mean square error of the validation data is obtained by each epoch. If the error rate continues to decrease, save each updated model; if the error rate of the validation data does not improve more than 500 epochs, then the model stop training. After completing five cross-validations, this study averaged five models to obtain the final neural network model.

```
Initialization: $\alpha$ is the learning rate, $p$ is the dropout rate, iteration
Deep Neural Network ($x, y, x\_val, y\_val, \theta$):
  $x = [x^{(1)}, x^{(2)}, \ldots, x^{(m)}] \in R^{m*n}$ is the input matrix, where $m$ is the number of data
  $y = [y^{(1)}, y^{(2)}, \ldots, y^{(m)}] \in R^{m*1}$ is the real score
  $\theta = [\theta_1, \theta_2, \ldots, \theta_l]$, pre-training by De-noising Autoencoder, where $\theta = \{W_i, b_i\}$
  $O = [O_0, O_1, \ldots, O_l] \in R^{m*h_i}$ is the output of each layer, which $O_0 = x$.
   for $epoch$ in $range(iteration)$:
       $\theta$ = dropout($\theta, p$)
     for $i$ in $range(1, l-1)$:
       $O_i$ = ReLU($O_{i-1} * W_i + b_i$)
       $regularization$ = L2Regularization($W_i$)
     end for
     $O_l$ = Linear Regression($O_{l-1}, \theta$)
     $loss$ = RMSE($y, O_l$) + $regularization * \alpha$
     $y\_val\_predict$ = Linear Regression($x\_val, \theta$)
     $val\_loss$ = RMSE($y\_val, y\_val\_predict$)
     EarlyStop($val\_loss$)
      $g$ = Compute the gradients of the loss with respect to $\theta$
     For $\theta_i, g_i$ in $(\theta, g)$:
       $\theta_i = \theta_i - \alpha * g_i$
     end for
   end for
end procedure
```

Fig. 8. Deep neural network algorithm of model.

Fig. 8 is the deep neural network algorithm. The number of iterations is set to 5,000 times. Each iteration will do dropout with dropout ratio p0.2, the learning rate α is 0.01, and use L2 regularization for loss function. Use cross-validation, early stop, and other algorithms to make the model more stable. Finally, this study used Adam method to update the weight and then training for the next iteration.

4. EXPERIMENTAL RESULTS

The system environment is i7-6700 and 16G memory, the prediction model is implemented by Python and TensorFlow framework. This section is divided into two categories of experiments, one for comparing the effectiveness of using a specific algorithm, another for comparing the proposed model with other machine learning methods. The chart for each experiment is the average of ten experiments. Table 3 is the code number of each course to be predicted, and it is sorted according to the order of the courses specified in each semester from the first year to the fourth year.

Table 3. Class code.

Course Name	Code	Course Name	Code	Course Name	Code
Engineering Mathematics 1	A	Probability	B	Electronics 1	C
Electronics Lab. 1	D	Circuit Theory 1	E	Engineering Mathematics 2	F
Microprocessor	G	Electronics 2	H	Electronics Lab. 2	I
Circuit Theory 2	J	Signals and Systems	J	Special Projects 1	L
Special Projects 2	M				

(A) The Efficiency of DAE

This experiment compared the performance between using DAE or not, and uses the scores of the students as test data to evaluate the loss of the model.

In Fig. 9, it can be seen that the model without pre-trained has a very high loss at the initial stage, and the loss segment is jagged, which means that the training process cannot be stably reduced. Although the loss of training data is getting smaller and smaller at a late stage, the gap between the training data and the validation data is gradually widening, which means the model does not have the general ability to adapt to data. On the contrary, the error between the pre-trained model prediction result and the expected result is dropped very fast in the early stage, and after about 1500 iterations, the best local minimum can be quickly found to early stop the training.

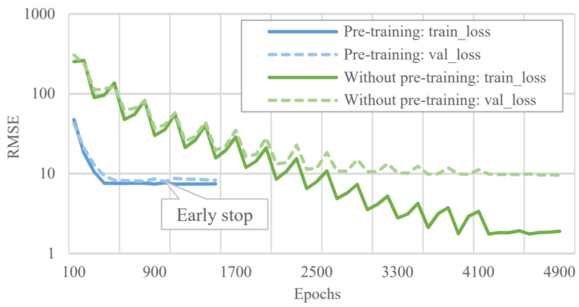

Fig. 9. Loss of pre-training and without pre-training.

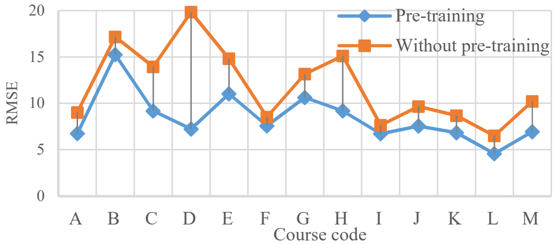

Fig. 10. Compare with each course.

This experiment used the student's grades as test data to compare the differences between the pre-trained and without pre-trained courses in each subject. As shown in Fig. 10, we can observe that each course performs better after pre-training.

(B) The Efficiency of Course Cluster

In order to understand whether the clustering of the course helps to improve the performance of the model, this experiment selected three course: Electronic Internship 1, Electronics 2, and Circuit 2, based on the pre-training model of the previous chapter. To explore the comparison of the loss of training scores between clustering and the non-clustering. As shown in Fig. 11, the degree of course loss of the clustering has been far better than that of the non-clustering at the beginning of the training. After 3000 iterations, all of the clustering courses perform better than the non-clustering courses.

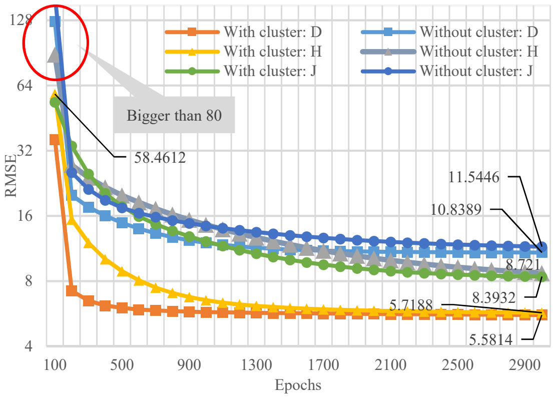

Fig. 11. Efficiency of course cluster.

(C) The Efficiency of Ensemble Learning

The purpose of this experiment is to understand whether the performance of the model combination in each school year has improved. Take the course of "signal and system" as an example.

Table 4. Loss of different year.

Training Model (Academic year)	2012	2013	2014	2015	2016	Ensemble Predictor
Loss	8.268	8.296	8.379	7.234	7.123	6.831

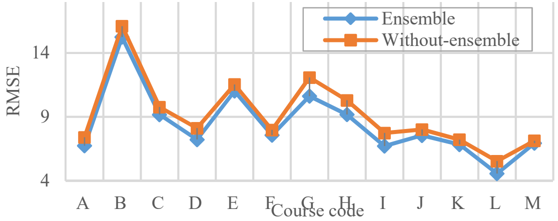

Fig. 12. Loss of ensemble and without ensemble.

The students who take this course in each school year will be used as training data to train the model. The loss of the model to the test data is shown in Table 4. The loss distribution is from 7.123 to 8.268, and the closer to the year of test data is, the better performance it is. Combine 2012 to 2016 academic years with different weights to build an ensemble predictor. Use the students who took the 2016 academic year course as test data obtained the loss of 6.831. Therefore, the ensemble predictor did improve the performance architecture or not. This experiment compared the test data using the ensemble predictor performance difference in each course, as shown in Fig. 12, because the network model has been pre-trained. The two models have performed very well, but our proposed model combines the results of different predictor, making the prediction more precise.

(D) The Efficiency of Ensemble Learning

This section compares the predictive power of our proposed model with Support Vector Regression, Linear Regression, and Random Forest, three model learning algorithms. Fig. 13 shows the loss of each course in different methods. The loss function of our model is the smallest and performs the best among the test data.

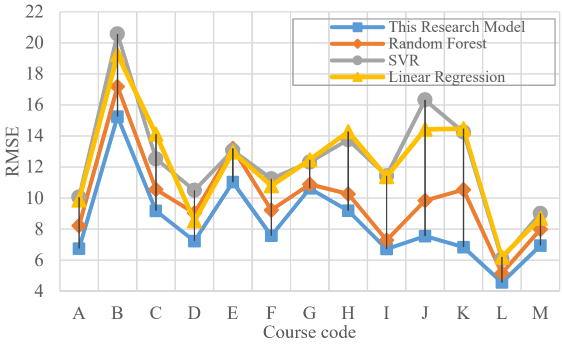

Fig. 13. Loss of different algorithms.

In order to examine the performance, this experiment establishes the confusion matrix, uses the score of the pass as a judgment of whether the model performs well. The condition of True Positive is the number of students who cannot pass the course in the real situation and the model correctly predicts the student didn't pass the course. The results are shown in Table 5. The F-measure score of our model is the highest, which means that we can effectively pick out the failed students.

Table 5. Efficiency comparsion of different algorithms.

	Our Model	Random Forest	SVR	Linear Regression
RMSE	7.625	9.222	11.241	10.79
Accuracy	89.62%	85.84%	78.3%	77.35%
Precision	76.47%	60.86%	43.47%	40.9%
Recall	65%	70%	50%	45%
F-measure	70.27%	65.11%	46.51%	42.8%

5. CONCLUSION

This study built a pre-trained ensemble learning neural network model that used the student's academic performance and attributes to construct the prediction model. The proposed approach continuously tracked the data and updated the model to provide teachers with predictive reports, that it allowed teachers to review student performance, and to reduce the possibility of deferral graduation of students. This study applied *K*-Means to cluster all courses, and thus the method can effectively find similar course features to improve prediction accuracy through experiments. The pre-trained models help the training process relatively stable and avoid into local optimization. The experiments also proved that the training data according to different years of the course can be effectively reduced one to two points of loss. Finally, the proposed model was compared with other machine learning approaches to show that it presented the highest predictive ability for measuring the student's performance.

ACKNOWLEDGMENT

This research was supported by Ministry of Science and Technology, Taiwan, program MOST 106-2221-E-027-018-MY2, and National Taipei University of Technology program NTUT-SZU-107-06.

REFERENCES

1. Directorate General of Budget, Accounting and Statistics, Executive Yuan, R.O.C. "Number of students who postpone graduation in colleges and universities in 2016 academic year," https://www.dgbas.gov.tw/public/Data/7417174752W75YTOV0.pdf.
2. A. Baas, "Promising strategies for at-risk youth," ERIC Digest No. 59, 1991.
3. L. Breiman, "Bagging predictors," *Machine Learning*, Vol. 24, 1996, pp. 123-140.
4. Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proceedings of the 13th International Conference on International Conference on Machine Learning*, 1996, pp. 148-156.
5. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313, 2006, pp. 504-507.
6. J. A. Hartigan and M. A. Wong, "Algorithm as 136: A K-Means clustering algorithm," *Journal of the Royal Statistical Society – Series C* (*Applied Statistics*), Vol. 28, 1979, pp. 100-108.

7. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv: 1412. 6980, 2014.
8. R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, Vol. 2, 1995, pp. 1137-1143.
9. J. Y. Kuo, C. W. Pan, and B. Lei, "Using stacked denoising autoencoder for the student dropout prediction," in *Proceedings of IEEE International Workshop on Mining and Application of Multimedia*, 2017, pp. 483-488.
10. Y. Meier, J. Xu, O. Atan, and M. van der Schaar, "Personalized grade prediction: A data mining approach," in *Proceedings of IEEE International Conference on Data Mining*, 2015, pp. 907-912.
11. V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines." in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 807-814.
12. A. Y. Ng, "Preventing 'overfitting' of cross-validation data," in *Proceedings of the 14th International Conference on Machine Learning*, 1997, pp. 245-253.
13. E. Tanuar, Y. Heryadi, Lukas, B. S. Abbas, and F. L. Gaol, "Using machine learning techniques to earlier predict student's performance," in *Proceedings of International Conference on Indonesian Association for Pattern Recognition*, 2018, pp. 85-89.
14. M. Tsiakmaki, G. Kostopoulos, G. Koutsonikos, C. Pierrakeas, S. Kotsiantis, and O. Ragos, "Predicting university students' grades based on previous academic achievements," in *Proceedings of the 9th International Conference on Information Intelligence Systems and Applications*, 2018, pp. 1-6.
15. J. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1096-1103.
16. J. Xu, K. H. Moon, and M. van der Schaar, "A machine learning approach for tracking and predicting student performance in degree programs," *IEEE Journal of Selected Topic Signal Process*, Vol. 11, 2017, pp. 742-753.
17. KDD Cup, "Educational data minding challenge," https://pslcdatashop.web.cmu.edu/KDDCup/, 2010.

Jong-Yih Kuo (郭忠義) received the Ph.D. degree in Computer Science and Information Engineering from National Central University in Taiwan in 2007. He is a Professor in the Computer Science and Information Engineering at Taipei University of Technology in Taiwan. His current research interests include software engineering and intelligent system.

Hao-Ting Chung (鍾皓廷) received the B.S. degree in Computer Science and Information Engineering from Taipei University of Technology, Taiwan. His research interests include software engineering, and machine learning.

Ping-Feng Wang (王秉豐) received the Ph.D. degree in Computer Science and Information Engineering from National Central University in Taiwan in 2015. He is a Deputy Director in the Institute for Information Industry. His current research interests include software engineering and service-oriented computing.

Baiying Lei (雷柏英) received the Ph.D. degree from Nanyang Technological University, Singapore, in 2013. She is currently an Associate Professor with the School of Biomedical Engineering, Shenzhen University, China. Her current research interests include medical image analysis, machine learning, digital watermarking, and signal processing.