
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 4, pp. 2122–2132, DOI 10.12694/scpe.v25i4.2156

A STUDY ON THE PREDICTION METHOD OF ENGLISH PERFORMANCE IN UNIVERSITIES BASED ON THE STACKING INTEGRATED MODEL

TONGSHENG SI*

Abstract. Students' performance in higher education reflects their overall quality in higher education. By predicting the performance, students with greater learning problems can be screened out early and given appropriate guidance. To predict students' performance in English, the knowledge information of courses, examination papers, and historical examination records are used to build a feature project of students' examinations. Meanwhile, the features strongly correlated with their performance are filtered out. Then the next step of performance prediction is carried out. The results showed that a neural network long and short-term memory performance prediction model incorporating an attention mechanism was more effective than other models in predicting English performance in higher education. Further experiments found that the model reduced the error by 1.04% on the MAE metric, 0.53% on the RMSE metric, and increased its value by 4.12% on the R2 metric. Adding the new feature dataset led to better forecasting by the Att-LSTM model in all metrics. This indicated that the enhanced dataset temporality could improve the effectiveness of the Att-LSTM model in predicting English grades in higher education. The stacked integrated prediction model, by integrating multiple strong regressors, can avoid poor prediction and excessive overall bias due to one regressor and increase the soundness and prediction precision of the mode.

Key words: Stacking, Achievement Prediction, Integrated Learning, Attention Mechanism, English in higher education

1. Introduction. With the fast growth of national education computerization, a large amount of data on education has been generated in the education management system[15]. Data mining has been a new power in these sectors [5]. However, the application of this technology in student performance prediction is still underdeveloped and still being explored [12]. Big data has made a huge difference in how people work, live and think, and its impact on the education sector cannot be ignored [9]. Big data is becoming a huge force for progress and change in the education system [2]. Education is undergoing a dramatic change due to the influence of big data. The education sector continuously generates and accumulates a large amount of data, which constitutes big data in agriculture [10]. Big data's importance in education and its huge value is gaining more attention. With the construction of information systems on campus, there is a hardware basis for collecting and processing the various data generated by students on campus [6]. Therefore, this study uses the Stacking integrated model to study English performance prediction in colleges. A multi-model overlay prediction model based on the integrated learning stacking method is proposed for college students' English grades. The algorithm takes the prediction results of XGBoost, LightGBM and Att-LSTM models as input and then uses multiple regressors to integrate the prediction results to obtain the students' grade prediction results.

2. Related Works. Under the influence of big data, the education sector is undergoing a dramatic change. It constantly generates and accumulates large amounts of data, which constitute a large amount of energy in agriculture. The importance of big data in education and its huge value is becoming increasingly visible. With the continuous construction of information systems on campuses, there is a hardware basis for collecting and processing the various data generated by students on campus. So that college English can be promoted and students can achieve better grades. Achievement prediction has become an important topic and has been thoroughly studied by many scientists. Mubarak et al. proposed a supermodel of deconvolutional networks and long-term and short-term memory, dubbed CONV-LSTM, to extract features automatically from the raw MOOC dataset and to forecast each student to drop or finish the course. The proposed model showed an

*School of Culture, Tourism and International Education, Henan Polytechnic Institute, Nanyang, 473000, China (tongshengst@outlook.com)

improved performance [11]. Wu L proposed a spatial recurrent mould grounded in deep attention, which learnt to focus on key target components and encoded those into space expression characterization. The method was experimentally demonstrated to outperform two typical fine-grained recognition tasks [13]. Thanh et al. proposed a profound study using classroom data conversion and factorial data over time to predict the pupils' grades. The experiment was built on 16 statistical collections relevant to various fields of study, collected from about 4 million samples from the student message board of a large-scale discipline network in Vietnam. The outcomes indicated that the presented approach offered excellent forecasting results, particularly when transforming the data and applied to practical cases [4].

Liu et al. proposed a new framework for student performance prediction using machine learning to capture features and fused attention mechanism-based recurrent neural networks. Experiments demonstrated the effectiveness and practicality of the method with an accuracy of 98% [7]. Baruah et al. presented a deeply neuro-fuzzy web based on the MapReduce framework for multiple universe optimization of grade competition-based student performance prediction. The values of the proposed method in terms of mean square error, root mean square error and mean absolute error were 0.3383, 0.5817, and 0.3915, respectively [3]. Abdollahi et al. presented a depth study-based multi-step algorithm to predict trip times using 5-fold cross-validation to detect the ability of the prediction mode. The proposed algorithm performed on a mean of 4 minutes outperforming a deep neo-network applied to the original eigenspace [1]. Meng et al. presented a deep feedforward forecasting model of neighbourhood neural networks to implement a suitable software system for employment forecasting and guidance of college and high school students, and the results showed some applicability [8]. Zhang et al. presented a method for predicting exploits using a multi-step N-gram characteristic selection and hierarchical integrated study. The results showed that the presented approach was validated on the server exploit buffer spill exploit and resource management exploit datasets with the lowest false positive and false negativity rates of 1.58% and 4.06%, respectively [16]. Xie et al. proposed a model for predicting grades through an attentive multilayer LSTM that combined students' demographic and clickstream datasets for integrated analysis. The results showed higher prediction accuracy and could provide timely interventions [14].

Currently, the research on student performance prediction is less concerned with the association between course knowledge data and performance. Its applicability and extensibility are narrow, which is not conducive to the practical application of relevant research findings. Meanwhile, existing studies have also paid little attention to the effect of time-series factors on predicting pupil achievement. Thus, this study proposes a neural network long- and short-term memory performance prediction mode incorporating an attention mechanism and a multi-model overlay of university students' English performance prediction mode. The algorithm takes the prediction results of XGBoost, LightGBM and Att-LSTM models as input, and then uses multiple regressors to integrate the prediction results. Finally, the prediction results of students' performance are obtained, and a new idea is proposed for the comprehensive research of English education in colleges.

3. A Stacking Integrated Multi-Model Overlay Based Approach for Predicting English Performance in Higher Education.

3.1. A neural network long and short-term memory performance prediction model incorporating attention mechanism. The first step for predicting English grades in higher education is pre-processing students' English grade data. The pre-processing allows richer information to be obtained on language test data and makes it easier to access questions. The data set records all knowledge points besides the student's English grades, with classification, sectioning, numbering and complexity information. Also, for each English test, the dataset records the points in the question paper, and the scores accounted for by those points. They are pre-processed through cleaning, feature engineering, and the generation of training datasets to investigate the intrinsic correlation between the data. These data provide a view of the overall distribution of student performance and also enable an understanding of whether there are any anomalies in the examination performance. Student performance is plotted according to a range of statistics per 10 points while being able to draw a box plot of student performance, as shown in Figure 3.1. Figure 3.1 shows that most students in this English course are within 60-100. Among them, the median score of students is more than 80, and a few students score under 60, indicating the good overall performance of college students. However, a small number of students have a score of 0 and might miss exams. The exam score of 0 is low, so this part of the anomaly is removed. In the construction of feature engineering, a computer program is usually used to record data, which is convenient,

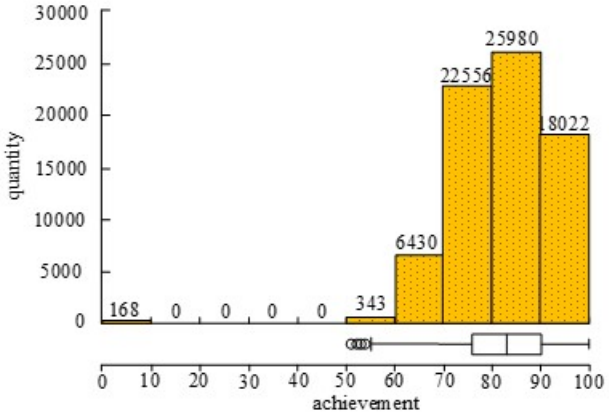

Fig. 3.1: Statistical Chart of Student Test Results

fast, and has a large amount of data. Attributes with strong correlations with scores are generally screened out, while those with weak correlations are excluded. Point-biserial correlation coefficients are used to analyse correlations between bivariate and continuous variables. The relation between student performance and gender is obtained by computing the point-biserial correlation coefficient, which is given in equation 3.1.

$$r_p b = \left(\overline{(Y_1)} - \overline{(Y_0)}\right) / S_Y \sqrt{((N_0 N_1)/(N(N-1)))} \tag{3.1}$$

In equation 3.1,Y_0 is the observed mean of the gender metric coded as 0. Y_1 is the mean of observations for gender indicators grouped under the code 1.N_0 is the approximate value of the observed values of males and females coded as 0. N_1 is the quantity of gender-specific views encoded as 0. N is the number of observations. S_y is the reference deviation of all metric measurements. If one starts with the complexity of the test paper, Pearson's coefficient of correlation is used to assess the intensity of the online relationship between two continuous entities. Pearson's coefficient is used to calculate the total complexity and its performance correlation, as shown in equation 3.2.

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - x_m)^2 \sum (y - y_m)^2}} \tag{3.2}$$

In equation 3.2, m_x represents the mean of the total complexity of x. m_y denotes the average of the outcomes. y indicates the outcome of the correlation. The paper's complexity greatly impacts students' final grades and could be used to forecast their grades. In turn, the different points of English knowledge can have a differential effect on students' achievements. The data size is effectively reduced by reducing the dimensionality of the original data. To perform a non-negative matrix decomposition for an already existing matrix$V \in R_+^{n*m}$, the matrix $W \in R_+^{n*r}$ needs to be found with the matrix $H \in R_+^{r*m}$ such that it satisfies equation 3.3.

$$V_{n*m} \approx W_{n*r} H_{r*m} \tag{3.3}$$

In equation 3.3, $\approx$ is used mainly because the solution used is only an approximation. $r < n, r < m$. In the general case of $(n+m)\,r < nm$, the relationship between a more complex feature set is constructed by feature engineering. Table 3.1 shows the specific feature descriptions. Table 3.1 divides the college English exam papers into three granularities for each knowledge point from coarse to fine, namely, class, subsection and knowledge point. The student's performance will change in each test. However, the long-term performance will stabilize within the same range. After multiple exams, students learn from their previous mistakes and can avoid the same mistakes, so their performance on exams will have some influence on subsequent exams.

Table 3.1: Description of college English achievement characteristics

Features	Format	Describe
Student Id	String	Student ID
Exam Id	Integer	Exam ID
Complexity	Integer	Complexity
ComplexityRate1-5	Float	Proportion of complex 1-5
Type1-10	Integer	Chapter classification 1-10
Subjects1-10	Integer	Subjects classification 1-10
Knowledge1-100	Integer	Knowledge points 1-100

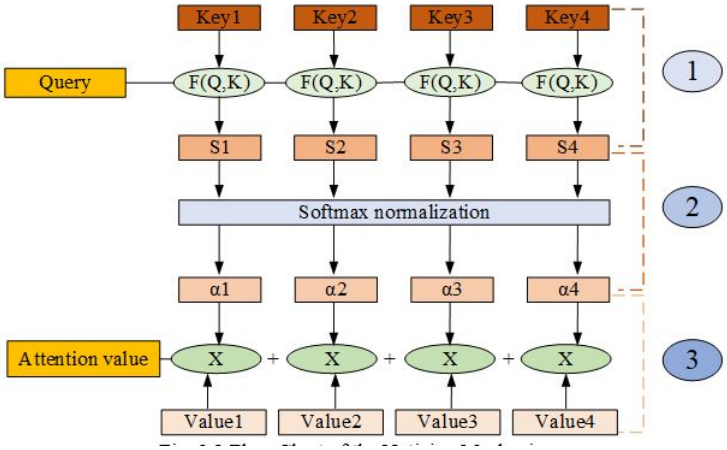

Fig. 3.2: Flow Chart of the Noticing Mechanisms

Therefore, the prediction of English grades in higher education can be seen as a time series prediction. In this study, student performance is predicted by the LSTM model, which incorporates an attention mechanism. The flow processing of the general attention mechanism is shown in Figure 3.2. In the general concentration mechanism, four key data values are available. The input is represented as a key-value pair (Q,K) of Q and K. The weights of the key-value pair are obtained by calculating the similarity between the Query and each Key. Then the values representing the importance are obtained by normalization. Finally, the attention values are obtained by summing the results for each value according to the importance values. The attention function can use addition or dot product. The attention function can be expressed as shown in equation 3.4.

$$Atteneion\left(Query, Source\right) = \sum_{i=1}^{L_x} Similarity\left(Query, Key_i\right) * \ Value_i \tag{3.4}$$

In equation (4), L_x is the total length of the input features. There are 3 main stages in the attention mechanism. The first step is to choose the appropriate method to find the correlation coefficient between the query and the key. Methods such as vector dot product method, vector similarity method and MLP neural network method can be used. The formula for the calculation method, see equation 3.5.

$$\begin{cases} F\left(Q, K_i\right) = Q^T K_i \\ F\left(Q, K_i\right) = \frac{Q^T K_i}{\|Q^T\| \|K_i\|} \\ F\left(Q, K_i\right) = MLP\left(Q^T, K_i\right) \end{cases} \tag{3.5}$$

In equation 3.5, $F\left(Q, K_i\right)$ is the correlation coefficient. Qis the Query. K_i is the different keyword Key. For K_i, the correlation coefficient $F\left(Q, K_i\right)$ can be expressed as S_i. The previous correlation coefficients are

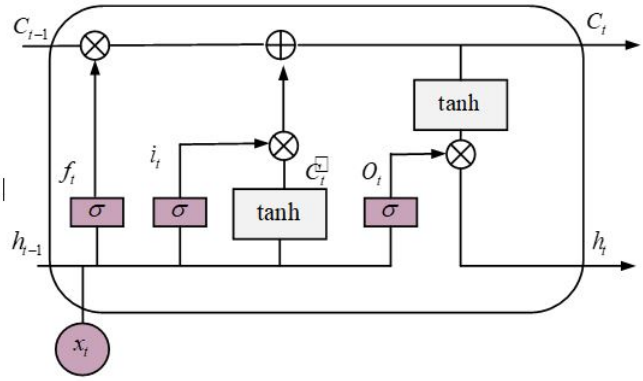

Fig. 3.3: Cell Architecture of LSTM

processed using Softmax on the second stage to obtain the weighting coefficients. The two advantages of this processing are that all keyword weights are processed as a probability distribution with a sum of 1, and that the importance of important keywords can be increased by its own function. The result of the Softmax formula is shown in equation 3.6.

$$a_i = Softmax\left\{(\right\} S_i) = Softmax\left\{(\right\} F\left(Q, K_i\right)) = \frac{e^{F(Q,K_i)}}{\sum_{j=1}^{L_x} e^{F(Q,K_i)}} \tag{3.6}$$

In equation 3.6, a_i is the weighting factor corresponding to K_i. In the third stage, the weighting coefficients a_i and V_i are calculated to obtain the final attention values, as shown in equation 3.7.

$$Attention\left(Q, K, V\right) = \sum_{i=1}^{L_x} a_i V_i \tag{3.7}$$

In equation 3.7, V_i denotes the value of the weight coefficient. LSTM is controlled by input, output, and forgetting gates, which enhance its long-term memory capability and effectively solve the gradient disappearance while filtering relatively unimportant information and reducing training time. The complete LSTM cell state is displayed in Figure 3.3.

From Figure 3.3, in the LSTM cell state, C_{t-1} is the cell preservation state after the previous processing. $h_{t-1}$is the retained state-of-play of the latent level after the last processing. σ is the input data where the sigmoid function is performed. $\otimes$ represents the need to perform a dot product operation on the data. *tanh* represents the activation function operation to be performed on the data. In the LSTM cell, the forgetting gate is represented by f_t, the entry gate is represented by i_t, and the out gate is represented by o_t. The input data of LSTM is represented by x_t. The hidden state input to the next cell after completing LSTM is represented by h_t. The cell state input to the next cell after completing LSTM is represented by C_t. LSTM is mainly used to obtain information about the input feature. The attention mechanism helps it quickly sift out the important messages from many features and focus on the most helpful information. Unimportant data are filtered out to increase the training efficiency of the mode. The $Att - LSTM$ market mode is designed by combining the attention mechanism. The model has good results for predicting English grades in colleges, and its architecture is shown in Figure 3.4.

Figure 3.4 shows that the college English grade prediction model of Att-LSTM contains a five-layer structure with an input layer, an LSTM, an attention mechanism, a fully connected layer and an output layer. The input layer is to import the student performance data into the model. Various data related to students and courses have been filtered and constructed through feature engineering. Feature vectors are constructed based on historical examination paper information, knowledge and performance information that can be identified.

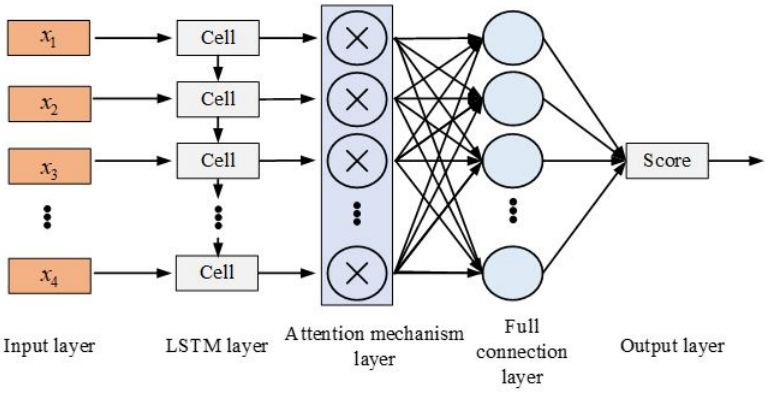

Fig. 3.4: Att LSTM Model Architecture

The exam count window is n. The feature window for consecutive exams is X_n. See equation (3.8) for details.

$$\begin{cases} X_n = (x_1, x_2, \ldots, x_n) \\ x_n = (r_1, r_2, \ldots, r_k) \end{cases} \tag{3.8}$$

In equation (3.8), x_n represents the nth test, where the data features of this test are represented by r_k. The LSTM layer is governed by three doors: the forgetting door, the input and the lower gate. The LSTM first filters the cell states by the activation function of the forgetting gate. A vector is calculated from the secrecy state h_{t-1} at the former time $t-1$ and the current input information x_t, and the vector takes values within 0-1, indicating the degree of data retention corresponding to the cell state. No retention at all is represented by 0 and full retention is represented by 1. The specific calculation is shown in equation (3.9).

$$\begin{cases} f_1 = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ \sigma(x) = \text{sigmoid}(x) = \frac{1}{1+e^{-x}} \end{cases} \tag{3.9}$$

In equation (3.9), f_1 is the forgetting gate. x_t is the input message for the present. W_f and U_f are the parameters of importance. b_f is the parameter for the correction offset. After the calculation of the forgetting gate is completed, the calculation of what new data should be entered for the cell state is then done. The data are updated by the hide status h_{t-1} from the previous time t-1 and the present input information x_t. Then the new candidate memory cell $\widetilde{C}_t$ is obtained using h_{t-1} and x_t via *tanh* function, calculated as shown in equation (3.10).

$$\begin{cases} i_t = \sigma\left(W_i x_t + U_t h_{t-1} + b_i\right) \\ \tilde{C}_t = \tanh\left(W_{\tilde{c}} x_t + U_{\tilde{c}} h_{t-1} + b_{\tilde{c}}\right) \\ \tanh(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \end{cases} \tag{3.10}$$

In equation (3.10), i_t is the input gate for the current operation t. The parameters of importance are W_i, U_i, $W_{\widetilde{C}}$, and $U_{\widetilde{C}}$. $b_{\widetilde{C}}$ is the correction offset. $\widetilde{C}_t$ is the state of the memory cell selected for the current operation. $tanh\left\{(\right\} x)$ is the hyperbolic tangent function. Once $\widetilde{C}_t$ is calculated, C_{t-1} is updated and the value after C_t is determined. However, after x_t passes through i_t, i_t will use $\sigma\left(x\right)$ to fuse the data from $\widetilde{C}_t$ into C_t. The calculation results are shown in equation 3.11.

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \widetilde{C}_t \tag{3.11}$$

In equation 3.11), C_{t-1} represents the memory cell state from the previous operation t-1. After calculating C_t, the information is sent to the next hidden layer of cells based on h_{t-1} and x_t. Then *tanh* is used to calculate

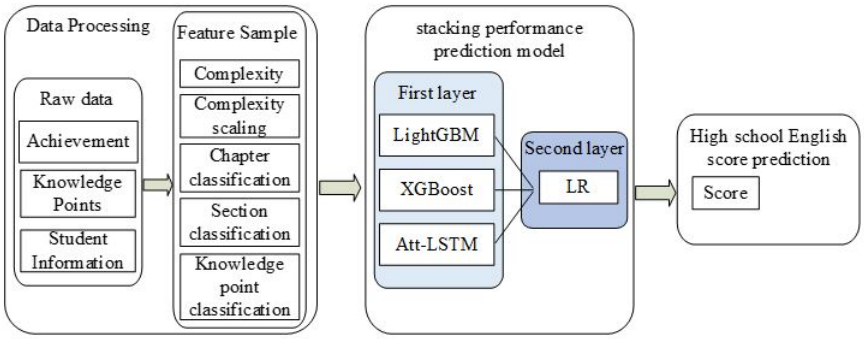

Fig. 3.5: Schematic Diagram of Student Achievement Prediction Model

a vector at o_t to obtain the final result. The calculation steps are shown in equation (3.12).

$$\begin{cases} \sigma_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \\ h_t = o_t \otimes \tanh\left(C_t\right) \end{cases} \tag{3.12}$$

In equation (3.12), o_t is the outgate. W_o and U_o are the significance parameters. b_o is the correction offset. x_t is the data source for the current operation t. h is the current hidden status. The input feature information is captured at the attention mechanism layer, and low-impact features are filtered out. The fully connected layer aggregates the results of the attention regime tier into the final prediction data.

3.2. Stacking-based multi-model overlay prediction model for English language performance in higher education. The stacking algorithm combines multiple prediction models and uses the prediction results of each model as a new training set for retraining, thus improving the prediction performance and making the predictions more stable. The construction of the Stacking model consists of two main layers, one consisting of three base regressors, LightGBM, XGBoost and Att-LSTM. The second is a meta-regressor using the LR algorithm, which uses the output data from the first layer as its input, and LR to make the final prediction of student performance. This constitutes a stacking prediction model with a two-layer structure, which is shown in Figure 3.5 below.

Figure 3.5 shows that in the first level of the training model, the whole data collection is organized first into a single set for training D_{train} and a test set at D_{test}. The three machine learning algorithms are then trained separately to obtain the three base regressors M_1, M_2, and M_3. To guard against modes that perform well on already existing known domains and behave badly on an unknown dataset, it is essential to ensure the generalization capability of the scale. In this study, a five-fold cross-validation method is applied to each underlying regressor during the definition. The set of training is partitioned into D^1_{train}, D^2_{train}, D^3_{train}, D^4_{train}, and D^5_{train} and each copy is taken as the validation set in turn for five times iterative coaching. The mode expression for each training is shown in equation 3.13

$$M^k_t = N_i\left(D_{train} - D^k_{train}\right) \tag{3.13}$$

In equation 3.13, M^k_i is the mode produced by the algorithm at the crisscross certification training. N_i is the $i-th$ algorithm. $i \in (1,3)$, $k \in (1,5)$. The trained base regressor M^k_i predicts the remaining validation set. The specific calculation is shown in equation 3.14.

$$\begin{cases} \hat{Y}^k_i = M^k_i\left(D^k_{\text{train}}\right) \\ \hat{Y}_i = \left(\hat{Y}^1_i, \hat{Y}^2_i, \hat{Y}^3_i, \hat{Y}^4_i, \hat{Y}^5_i\right) = \left(\hat{y}^1_i, \hat{y}^2_i, \hat{y}^3_i, \ldots, \hat{y}^n_i\right)^T \end{cases} \tag{3.14}$$

In 3.14, $\hat{Y}^k_i$ is the set of predictions from base regressors for the cross-validation training using M^k_i. $\hat{Y}_i$ is the predicted value of D^k_{train} for all validation sets for the $i-th$ base regressor after 5 cross-validations. During

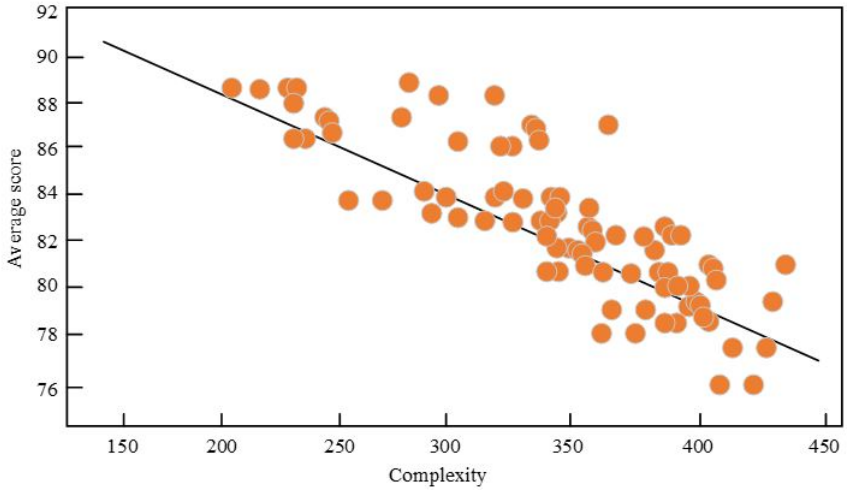

Fig. 4.1: Complexity of Papers and Mean Score Spread of the School Community

the cross-validation process, the base regressor M_i^k predictions at D_{test} are calculated. This average is then used as the test set for the next regressor. The detailed calculation is shown in equation 3.15.

$$\begin{cases} \hat{Z}_i^k = M_i^k\left(D_{\text{test}}\right) \\ \hat{Z}_i = \frac{1}{5}\sum_{k=1}^{5}\hat{Z}_i^k = \left(\hat{Z}_i^1, \hat{Z}_i^2, \hat{Z}_i^3, \ldots, \hat{Z}_i^m\right) \end{cases} \tag{3.15}$$

In 3.15, $\hat{Z}_i^k$ is the prediction result of the crisscross certification training of the $i-th$ base regressor used for the test set. $\hat{Z}_i$ is the mean of the predictions from the base regressor for the $i-th$ cross-validation. In the run of the linear regression model, the mediated results derived from the basis regressor in the first run are used in the stacking algorithm to construct a new training set S_{train} and test set S_{test}. The data formats for S_{train} and S_{test} are $\left(\left(\hat{Y}_{1i}, \hat{Y}_{2i}, \hat{Y}_{3i}\right), y_i\right)$ *and* $\left(\left(\hat{Z}_{1j}, \hat{Z}_{2j}, \hat{Z}_{3j},\right), y_j\right)$, and the prediction results are shown in equation '3.16.

$$\begin{cases} M_{\text{stacking}} = LR\left(S_{\text{train}}\right) \\ \text{PredSet} = M_{\text{stacking}}\left(S_{\text{test}}\right) \end{cases} \tag{3.16}$$

4. Experiments and Analysis of English Performance Prediction in Universities Based on Stacking Integration Model.

4.1. Experiments and analysis based on the Att-LSTM prediction model. In terms of feature-building choices, students' test scores were generally influenced by test difficulty. The harder the test question, the more challenging it is to obtain a high score. All knowledge points in English in higher education were categorized, and labelled, and their complexity determined. The distribution of the correlation between the complexity of a paper and the average score in a particular exam is shown in Figure 4.1 below.

Figure 4.1 showed that there were 5 levels of paper complexity, with the difficulty categorized into 1-5. The complexity of the paper was determined for a given examination. Overall, the complexity of the paper was negatively correlated with the mean score, the more difficult the paper, the lower the student's score. To verify the effectiveness of a neural network long and short-term memory performance prediction model incorporating an attention mechanism in predicting English performance in higher education, three benchmark models were used in the experiments for comparison. To further ensure the stability of the experimental results, a quintuple across the verification process was employed to split the database into 5 equal portions. One of them was used as the validation set, and the remaining four were used as the training set. The models were trained on the

Table 4.1: Experimental results

Models	Evaluation Indicators	1	2	3	4	5
RF	MAE	4.597	4.581	4.569	4.602	4.599
	RMSE	6.564	6.547	6.534	6.571	6.564
	R2	0.241	0.211	0.211	0.268	0.262
RNN	MAE	4.435	4.378	4.388	4.413	4.399
	RMSE	6.415	6.34	6.348	6.325	6.366
	R2	0.284	0.291	0.28	0.281	0.284
LSTM	MAE	4.269	4.295	4.284	4.292	4.291
	RMSE	6.23	6.266	6.261	6.262	6.251
	R2	0.244	0.239	0.245	0.233	0.247
Att-LSTM	MAE	4.213	4.219	4.208	4.217	4.204
	RMSE	6.136	6.175	6.164	6.178	6.146
	R2	0.244	0.236	0.248	0.238	0.248

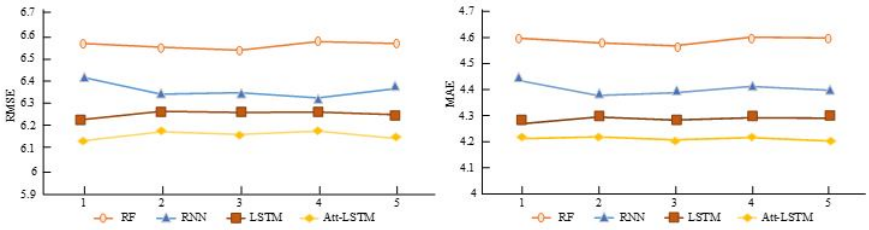

Fig. 4.2: Comparison of Forecast Results of Different Indicators

training set using different machine-learning algorithms. By iterating five times, the trained model was tested on the validation set for its effectiveness and predicted a result on the testing machine to obtain five test results. finally, the five results were averaged as the final model prediction results in Table 4.1.

In Table 4.1, the experiments provided statistics on the prediction findings of the different modes. The prediction results of the prediction models constructed by the four model algorithms tended to be stable over the five training sessions over multiple training sessions. Figure 4.2 compares the five prediction experiments for different students' performance prediction models at different metrics for a more intuitive view of the individual metrics.

In Figure 4.2, the performance metrics for each model were increasingly better from RF to Att-LSTM. The Att-LSTM model outperformed the other benchmark modes. Since the Att-LSTM model incorporated an attention mechanism, it could better capture the more valuable information when processing the data, and the prediction results would be better. However, to further ensure the stability of the experimental results evaluation, similar to the previous experimental setup, the constructed dataset was divided randomly into training and testing sets, and the ratio of the two data subsets was kept 8:2. A five-fold validation of crossover was used and the mean value was used as the final mode metric. The experimental outcomes are listed in Table 4.2.

From Table 4.2, the effectiveness of the Att-LSTM model in predicting English grades in higher education could be improved when the temporality of the dataset was enhanced. By comparing the different data results, the error was reduced by 1.04% on the MAE metric, 0.53% on the RMSE metric, and its value was improved by 4.12% on the R2 metric. Across the metrics, the addition of the new feature dataset resulted in better prediction results for the Att-LSTM mode. It was demonstrated that the Att-LSTM grade prediction model could achieve the task of student grade prediction on both datasets and had relatively good results.

4.2. Experiments on predicting English performance in higher education based on stacking integration model. To verify the effectiveness and accuracy of a stacking multi-mode overlay prediction

Table 4.2: Att-LSTM model prediction results

Models	Dataset	MAE	RMSE	R2
Att-LSTM	D1	4.201	6.142	0.214
	D2	4.112	6.121	0.221

Table 4.3: Att-LSTM model prediction results

Models	Evaluation Indicators	1	2	3	4	5
LightGBM	MAE	4.211	4.213	4.239	4.211	4.179
	RMSE	6.154	6.121	6.201	6.189	6.159
	R2	0.124	0.101	0.101	0.118	0.132
XGBoost	MAE	4.175	4.148	4.188	4.163	4.201
	RMSE	6.180	6.129	6.168	6.151	6.159
	R2	0.132	0.141	0.130	0.150	0.141
Att-LSTM	MAE	4.145	4.125	4.139	4.142	4.139
	RMSE	6.123	6.126	6.129	6.122	6.131
	R2	0.224	0.231	0.230	0.223	0.239
Stacking	MAE	4.131	4.121	4.121	4.122	4.114
	RMSE	6.114	6.125	6.101	6.101	6.046
	R2	0.230	0.231	0.229	0.232	0.238

model for college English grades, a comparison experiment was conducted using a Stacking integrated model with the base regressors LightGBM, XGBoost, and Att-LSTM. Five experiments were conducted on each of the four models in the training set, and the mean was obtained as the final test result, as indicated in Table 4.3.

In Table 4.3, the Stacking integrated model had improved in terms of evaluation metrics compared with the single performance prediction models, namely XGBoost, LightGBM, and Att-LSTM. To more intuitively verify the various performance of the Stacking integrated model and observe its stability, the experimental results of cross-validation of each model on different data sets were counted, and the change curves of MAE and RMSE of different prediction models under five experiments were plotted. This is shown in Figure 4.3. From Figure 4.3, the Att-LSTM model outperformed XGBoost and LightGBM in all indicators of the first three prediction models and achieved better experimental results by capturing the changes in students' historical performance. In the above figures (a) and (b), the XGBoost model had large fluctuations in the prediction of grades in the face of different data, while the Stacking integrated prediction model could avoid poor prediction and large overall deviation due to one regressor by integrating multiple strong regressors, which improved the robustness and prediction accuracy of the model. After determining the base regressors and meta-regressors, the stacking-integrated student performance prediction model was constructed. The experimental results showed that the Stacking integrated college English performance prediction model would have better prediction results than using only the base regressor.

5. Conclusion. The study addressed the problem of predicting English grades for students in higher education. Using a dataset of course knowledge points and historical English grades of college students, two prediction models were designed for grade prediction, thus enhancing the applicability of student grade prediction. Feature engineering was established based on students' test scores. Attributes with strong correlation to grades could be filtered out, while those with weak correlations were excluded. Attributes with test paper complexity were selected for the next grade prediction. The neural network long and short-term memory performance prediction model incorporating the attention mechanism is more effective than any other models. In further experiments, by comparing the results of different data, the error was reduced by 1.04% on the MAE metric, 0.53% on the RMSE metric, and its value was improved by 4.12% on the R2 metric. Across the metrics, the addition of the new feature dataset resulted in better prediction findings for the Att-LSTM model. This

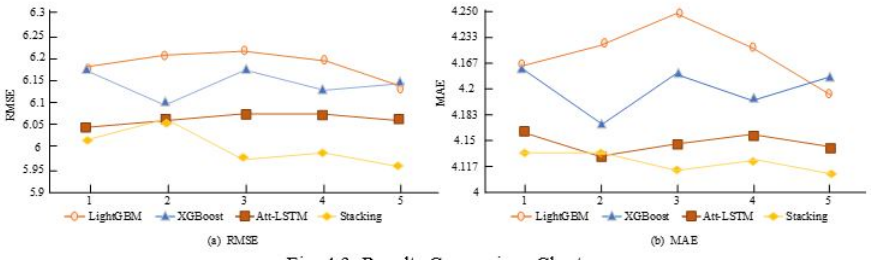

Fig. 4.3: Results Comparison Chart

indicated that enhanced dataset temporality could improve the efficiency of the Att-LSTM mode in predicting English grades in higher education. The stacked integrated prediction model, by integrating multiple strong regressors, could avoid poor prediction and excessive overall bias due to one regressor and improve its robustness and prediction accuracy.

REFERENCES

[1] M. Abdollahi, T. Khaleghi, and K. Yang, *An integrated feature learning approach using deep learning for travel time prediction*, Expert Systems with Applications, 139 (2020), p. 112864.
[2] M. S. Abubakaria, F. Arifin, and G. G. Hungilo, *Predicting students' academic performance in educational data mining based on deep learning using tensorflow*, Int. J. Educ. Manage. Eng.(IJEME), 10 (2020), pp. 27–33.
[3] A. J. Baruah and S. Baruah, *Data augmentation and deep neuro-fuzzy network for student performance prediction with mapreduce framework*, International Journal of Automation and Computing, 18 (2021), pp. 981–992.
[4] T. T. Dien, S. H. Luu, N. Thanh-Hai, and N. Thai-Nghe, *Deep learning with data transformation and factor analysis for student performance prediction*, International Journal of Advanced Computer Science and Applications, 11 (2020).
[5] W. Feng, J. Tang, and T. X. Liu, *Understanding dropouts in moocs*, in Proceedings of the AAAI conference on artificial intelligence, vol. 33, 2019, pp. 517–524.
[6] A. Khan, S. K. Ghosh, D. Ghosh, and S. Chattopadhyay, *Random wheel: An algorithm for early classification of student performance with confidence*, Engineering Applications of Artificial Intelligence, 102 (2021), p. 104270.
[7] D. Liu, Y. Zhang, J. Zhang, Q. Li, C. Zhang, and Y. Yin, *Multiple features fusion attention mechanism enhanced deep knowledge tracing for student performance prediction*, IEEE Access, 8 (2020), pp. 194894–194903.
[8] X. Meng, G. Ren, and W. Huang, *A quantitative enhancement mechanism of university students' employability and entrepreneurship based on deep learning in the context of the digital era*, Scientific Programming, 2021 (2021), pp. 1–12.
[9] M. Mohd, R. Jan, and M. Shah, *Text document summarization using word embedding*, Expert Systems with Applications, 143 (2020), p. 112958.
[10] V. Moscato, A. Picariello, and G. Sperlí, *A benchmark of machine learning approaches for credit score prediction*, Expert Systems with Applications, 165 (2021), p. 113986.
[11] A. A. Mubarak, H. Cao, and I. M. Hezam, *Deep analytic model for student dropout prediction in massive open online courses*, Computers & Electrical Engineering, 93 (2021), p. 107271.
[12] A. Polyzou and G. Karypis, *Feature extraction for next-term prediction of poor student performance*, IEEE Transactions on Learning Technologies, 12 (2019), pp. 237–248.
[13] L. Wu, Y. Wang, X. Li, and J. Gao, *Deep attention-based spatially recursive networks for fine-grained visual recognition*, IEEE transactions on cybernetics, 49 (2018), pp. 1791–1802.
[14] Y. Xie, *Student performance prediction via attention-based multi-layer long-short term memory*, Journal of Computer and Communications, 9 (2021), pp. 61–79.
[15] W. Xing and D. Du, *Dropout prediction in moocs: Using deep learning for personalized intervention*, Journal of Educational Computing Research, 57 (2019), pp. 547–570.
[16] B. Zhang, Y. Gao, J. Wu, N. Wang, Q. Wang, and J. Ren, *Approach to predict software vulnerability based on multiple-level n-gram feature extraction and heterogeneous ensemble learning*, International Journal of Software Engineering and Knowledge Engineering, 32 (2022), pp. 1559–1582.

Edited by: Achyut Shankar
Special issue on: Machine Learning for Smart Systems: Smart Building, Smart Campus, and Smart City
Received: Mar 20, 2023
Accepted: Apr 28, 2024