The Application of Artificial Neural Network, Long short-term Memory Network and Bidirectional Long short-term Memory Network in the Grades Prediction

HAIQI DONG

Research School of Computer Science, Australian National University
Email: u6872079@anu.edu.au

Abstract. The purpose of the study is to enhance the prediction accuracy of the final grades of students with various neural network models and techniques. The experiments are based on a small sample of data, and this study discusses the appropriate preprocessing strategy for the small dataset. Combined with control variables method, the study compares the effects of pruning, mini-batch gradient descent, random removal of neurons on the final accuracy of the Artificial Neural Network. The study implements the Long short-term Memory Network and Bidirectional Long short-term Memory Network as a further comparison. It shows the superiority of deep learning models compared to Artificial Neural Network in this problem. The study applies the classification accuracy and balanced accuracy as a comparison to measure the model performance comprehensively. By applying these techniques and models, the accuracy of the experiment in this study is significantly improved compared to the accuracy of the paper using the same data set. The report describes in detail how to improve accuracy step by step and puts forward some thoughts about the techniques and models.

Keywords: Artificial Neuron Network, pruning, mini-batch gradient descent, neuron dropout, Long short-term Memory Network, Bidirectional Long short-term Memory Network.

1 Introduction

There has existed the system to predict the final performance of the students in the educational institution [1]. The system can help teachers find potential risks of students failure, and pull them back when situations are getting too risky. The bad performance of homework and labs may cause more severe consequences. Thus, it is significant to intervene in the learning behaviours of students early based on the forecast [2]. The purpose of this study is based on this and aimed to enhance the grades prediction accuracy of students and make the educational institutions help students avoid failing in the courses.

The motivation for selecting dataset is to test the capability of various neural network models based on small data. The dataset used for this study is a marks recording of an undergraduate course from the University of New South Wales, Australia. The marks recording contains 40% part of all assessment marks of the course, which includes lab marks, homework, tutorial assessments and mid-term exam. The proportion of final exam is 60%, which is omitted. The final grades are given [3]. The objective of the model is to apply partial assessment marks of the students to predict and classify their final grades to four classes, which represent their grade levels.

This study compares the performances of The artificial neural network, Long short-term memory and Bidirectional Long short-term Memory Network on this dataset.

The artificial neural network (ANN) in the study has two hidden layers, and the model uses the latest modified version of Adam optimizer, the AdamW [4]. The ANN model adopts pruning technology to ensure the network small and efficient. In some situations, it is also helpful to increase the accuracy. The pruning neurons principles was proposed in "Network Reduction Techniques" [5]. The study also compares the ANN optimization effects of dropping out neurons randomly in each epoch with pruning specific neurons by specific rules. The study applies the mini-batch gradient descent approach in ANN to explore better accuracy, which also reflects the adaption effect of mini-batch training in a small data set.

This study also applied the Long short-term memory (LSTM) and Bidirectional Long short-term Memory Network (Bi-LSTM). LSTM is a particular Recurrent Neural Network (RNN). It aims to avoid the gradient disappearance and gradient explosion when training the long sequences [6]. LSTM can perform better than RNN on long sequence data [7]. In the LSTM internal structure, there are three main stages. The first stage is the forgotten stage; it ignores the unimportant information in the input value of the previous node, leaving only the critical information. The second stage is selectively memorizing the current input. And then combine the results of first and the second stages results. The final stage is through an output layer to decide the current states output value. Complete the entire sequence in this approach. When the current state has relationships with the former state and the future state, Bi-LSTM can be applied to this situation. Bi-LSTM is

composed of forwarding and backward LSTM, and these two networks both have a connection to the same output layer [8]. This study found the optimal models by setting the number of hidden neurons and LSTM layers.

The study still adopted the cross-validation method to ensure the reliability of the accuracy, which is consistent with the previous work on this problem [9]. The comparison found that the fitting effect of the new network used in this study far exceeds the past results. The final accuracy of ANN, LSTM and Bi-LSTM models are respectively 71.254%, 75.926% and 76.540%.

2 Method

2.1 Data analysis

The models aim to apply the neural networks to implement the prediction and classification of students final grades. The model only takes 40% of the results of the assessments from a total of 145 (The original data contains 153 patterns) students as input data. In the original dataset, there is a total of fifteen features and one label column. Extract the ten numeric features as the input of the models. Moreover, ignores the other nominal features. The final prediction grade levels contain four classes. See the Table below about the details [9]

Table 1. Correspondence table of final marks and grade level [9].

Grade level	Final marks
Distinction or above	Marks >= 75
Credit	65 <= Marks <= 74
Pass	50 <= Marks <= 64
Fail	Marks < 50

2.2 Data preprocessing

The size of the dataset using for the study is relatively small. Consequently, applying appropriate data preprocessing methods can helpfully improve the model performance. It is necessary to remove the outliers. The outliers in this study include the patterns only contain null values, and the pattern is missing the label value. After removing these outliers, the test accuracy increases and becomes stable. Next is vacant processing. The null values for the assessment marks represent the student did not submit the homework or absent in the examination. Thus, these null values should be set to zero. Then, another efficient approach is to reduce the number of data features. Based on observations, some feature values account for a small marks proportion, so mapping these feature values to the rest of the features is beneficial to explore the relationships among features, and make the ANN model more easily to learn. The principal component analysis is a good principle to implement the dimensionality reduction. It guarantees the integrity of the data to the greatest extent and improves the efficiency of the model [10]. Applying principal component analysis method on ANN and reduce the number of data features to seven can help the model enhance learning efficiency. While in LSTM and Bi-LSTM, the data features are still ten. Because these two models predict the results by time series, so ensuring the originality of the data is conducive to higher prediction accuracy.

The small dataset may cause the imbalanced data type problem, which may lead to insufficient model learning. When splitting sample data into training and test sets, it is likely to assign all or most of a certain class of data to the test set. This could result in the model not learning this specific type of data. Such a situation will significantly affect the test accuracy and reduce the credibility and dependability of the model. Therefore, assigning each type of data to the test set and training set is a solution to this problem. First, divide the data into four groups according to the level of the final marks. Then split each group of data into sub training and sub testing set. Finally, joint four sub training and sub testing set to the final training and testing set respectively. This method can ensure that each type of data can be in both the training set and test set, thus improve the adaptability of the model. Through the above data preprocessing methods, it can make small sample data perform better in the model.

2.3 Common Design Principles

2.3.1 Loss Function

The loss function in this study is a cross-entropy loss (1). In the formula, n represents the total samples, x is the current sample. The cross-entropy loss function can help to avoid gradient decay. When the loss is larger, the gradient will be larger as well (2), this can lead to improvements in training speed and also can avoid the underfitting problem.

$$L = -\frac{1}{n}\sum_x [\, y \ln a + (1-y) \ln (1-a)] \tag{1}$$

$$\frac{\partial L}{\partial w_i} = \frac{1}{n}\sum_x x_i \, (\sigma(z) - y) \tag{2}$$

2.3.2 Optimizer

This study uses AdamW optimizer. AdamW is the Adam optimizer with the decoupled weight decay. Experiments results demonstrated that the AdamW with the best parameter settings has a higher performance than SGD, SGDW and Adam with L2 regularization [4]. AdamW can avoid overfitting to improve the weaknesses of Adam [4]. In this study, it also helps to fixes the defect that the sigmoid function is likely to overfit in ANN. The weight decay regularization in AdamW can decrease the fluctuation range of batch losses, which makes the model unable to reach the lowest loss of training [4]. Therefore, the weight decay regularization can also help to avoid overfitting of the model.

2.3.3 Cross-Validation

The Cawley and Nicola [11] state that the cross-validation method can be used to find problems with overfitting and data bias. The average result of cross-validation is also a more accurate estimate of model performance [12]. In this study, the models apply the 10-fold cross-validation to assign training and testing datasets and get the average value of the ten runs results to evaluate the model performance.

2.4 Artificial Neural Network Model

2.4.1 Design Principle

The neural network in the study has two hidden layers. The first layer contains 78 neurons, and the second contains 26 neurons. The neurons amounts have the best performance in testing accuracy. Both of the hidden layers apply the sigmoid function as the activation functions. The sigmoid function as the basic logistic function is more easily to promote data fitting and also is commonly used in the feed-forward network. One obvious advantage is that sigmoid has the smooth gradient and results are normalized to range 0 to 1 [13]. For the output layer, it contains four neurons and applies the softmax activation function, which is useful for multiply label classification. The final optimal model with the parameters shown in the below Table.

Table 2. Artificial Neural Network with optimal parameter Setting

Parameter	Neurons of the first hidden layer	Neurons of the second hidden layer	Neurons of the Input layer	Neurons of the Output layer	Optimizer	Epochs	Learning Rate	Weight Decay	Mini-batch size
Value	78	26	7	4	AdamW	1500	0.006	0.45	12

2.4.2 Techniques for ANN model improvements

The model applies the mini-batch gradient descent to relatively increase the noise for every gradient and accelerate convergence.

The model prevents overfitting issues by applying random neurons dropout. In neural networks, the approach to avoid overfitting is to apply different neuron structures to the same set of data as much as possible and average each prediction results. The principle of dropout neurons is based on this approach. In training epochs, the input and output connections of some neurons would be ignored randomly. It is equivalent to that these neurons do not work in the current network structure. This is similar to fitting data with different network structures. This can make the model becomes more robust and reliable without overfitting problems [14].

Unlike dropout neurons randomly, the pruning technique is to delete the specified neurons according to the algorithm. The pruning neurons principle in this model is based on the theory in "Network Reduction Techniques" [5]. Firstly obtain the output vector of each hidden layer, every column represents one neuron. Next, calculating the vector angles among each pair of vectors. The angles which are smaller than a certain degree means these two neurons are similar enough because they have a similar output. When the angles are bigger than 180 minuses this certain degree, it means these two neurons are complementary, and their impacts on the network can offset. So when the two neurons in the same layer are sufficiently similar, delete one of them, and when they are complementary, delete all of them at the same time. The neurons in different hidden layers have not the similarity, so do not consider the neurons in different layers here. In the original paper, the angle is 15 degrees, because its network structure is quite small. However, the degree is only three in this study because the model is bigger and has a larger number of neurons. Three degrees is the most appropriate. Before removing the target neurons, it is necessary to assign the weights and bias of these models to their similar neurons. Because it can prevent the loss of information and also do not need to train the model after pruning. In this model, my implementation is to construct a new network which has a reduced number of neurons and the corresponding initial weights and bias. When a neuron is similar to multiple neurons, my principle of pruning is to keep as many neurons as possible and remove the neurons with the highest frequency of similarity. The testing accuracy evaluates the performance of pruning compared with the test accuracy before pruning. Pruning techniques should not reduce the accuracy too much, usually within 5%.

By comparison, the goal of pruning neurons is to make the network more efficient without losing too much accuracy. However, dropout neurons randomly in each epoch expect to avoid overfitting and improve the testing accuracy, which can make the model more robust. This study combines the two techniques to improve stability while maintaining the efficiency of the model.

2.5 Long Short-term Memory Model

LSTM is a particular Recurrent Neural Network (RNN). It aims to avoid the gradient disappearance and gradient explosion in the training process of long sequences [6]. LSTM can perform better than RNN on long sequence data [7]. This study apple the many-to-one Long Short-term Memory Model to predict the final student performance. Taking each pattern as a sequence and enter it into the LSTM model one by one and taking the last output as the final output result.

According to Figure 1 [15]. The LSTM structure, starting from the bottom, the current input is combined with the state of the past cell, and input to the forget gate, input gate, output gate and the cell itself. When deciding what information the cell state should forget, it can be done by the forget gate. It is a sigmoid function and the range between zero to one, one means to save all the information and zero means forget all the information. When deciding what the next new information the cell state should save is, there are two parts in the LSTM will work. The input gate with a sigmoid function and the block input with the tanh function. The input gate decides to update what information and block input creates a vector, which will be added to the next cell state. Firstly make the multiplication of previous state and f to forget the unnecessary information. Then, adding i*z, which is the candidate of the new state. When degerming what information should be the output, there are two parts. Firstly, through an output gate (sigmoid function) to determine which information of the cell state should be the output. Then, make the cell state conduct the tanh function and multiply with the output value of the output gate. Then we get the output value. Finally, the many-to-one LSTM model has a linear layer to map the last one output from LSTM to four categories. After that, the model applies the softmax function to identify the category of the grades for each student.

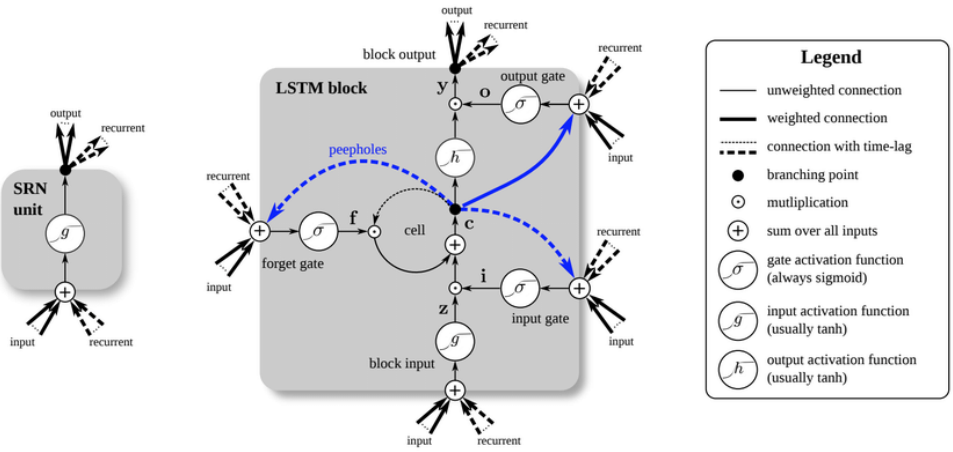

Figure 1. The LSTM structure [15]

The optimal Long Short-term Memory Model has one stacked layer and the thirteen output timesteps (hidden size), which predict the last hidden state.

Table 3. Long Short-term Memory Model with optimal parameter Setting

Parameter	Optimizer	Epochs	Learning Rate	Weight Decay	Number of Layers	Hidden size
Value	AdamW	120	0.004	0.4	1	13

2.6 Bidirectional Long Short-term Memory Network

The bidirectional Long short-term Memory network (Bi-LSTM) is the extension of the LSTM. Bi-LSTM model is to train two LSTM model on the input sequence, which is forward and reverse LSTMs, respectively. When the current state is not only related to the previous one but also related to the future state, Bi-LSTM model can provide the additional information and performs more efficiently than LSTM [16]. The comparison of LSTM and Bi-LSTM is shown in Figure 2 [17]. The many-to-one Bi-LSTM model in this study also has a linear layer to map the last one output to four categories. After that, the model applies the softmax function to identify the category of the grades for each student.

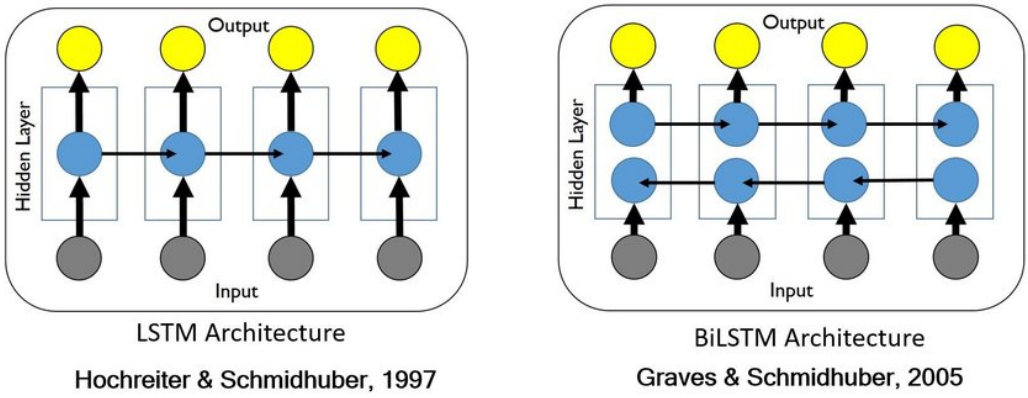

Figure 2. The comparison of LSTM and Bi-LSTM structures [17]

The optimal Bidirectional Long Short-term Memory Model has two stacked layer and the eight output timesteps (hidden size), which predict the last hidden state.

Table 4. The Bidirectional Long Short-term Memory Model with optimal parameters Setting

Parameter	Optimizer	Epochs	Learning Rate	Weight Decay	Number of Layers	Hidden size
Value	AdamW	120	0.006	0.4	2	8

3 Result and Discussion

The indicator of the model performance is averaged accuracy and averaged balanced accuracy. Balanced accuracy is a good indicator of the unbalanced multi-label classification problem [18]. The experiments adopted the 10-fold cross-validation method to ensure the reliability of the results. The experiments carried out in the form of variable control. The following tables will show the impact of different techniques on the results.

3.1 Dropout in Artificial Neural Network

The artificial neural network model prevents overfitting issues by applying random neurons dropout. The experiments set the dropout rate of the first hidden layer from 0.06 to 0.3 to find the optimal parameter value. The results below are average accuracy with 50 runs.

Table 5. Dropout rate - Experiment Results with 50 runs average accuracy

	Dropout rate	Ave Accuracy(%)	Ave Balanced Accuracy(%)
Run A	0.06	65.658	66.153
Run B	0.07	68.359	69.568
Run C	0.08	69.441	71.830
Run D	**0.10**	**71.254**	**72.176**
Run E	0.12	71.047	71.374
Run F	0.13	69.764	71.176
Run G	0.15	66.792	67.456
Run H	0.17	66.321	67.551
Run I	0.19	66.110	67.173
Run J	0.20	68.258	68.999
Run H	0.30	66.725	66.880

From Table 5, when the dropout rate is 0.1, the model performance is the best. The appropriate dropout rate can help the model weaken the influence of overfitting. Because the model randomly drops 10% of the neurons in each training epoch so that the model can have different structures with the remaining 90% neurons every time. Thus, the model can have a stronger ability in generalization.

3.2 Pruning Vector Degree in Artificial Neural Network

Pruning is based on the similarity between neurons in each layer. The pruning vector degree parameter is the similarity angle mentioned in 2.4.2. The dropout rate of the experiments is still set to 0.1. The experiments compare the model accuracy with different similarity angles and time spent of the model before and after pruning. The results below are average accuracy with 50 runs.

Table 6. Pruning vector degree - Experiment Results with 50 runs average accuracy

		Before pruning			After pruning		
	Pruning vector degree	Ave Accuracy(%)	Ave Balanced Accuracy(%)	Time(s) spent of prediction	Ave Accuracy(%)	Ave Balanced Accuracy(%)	Time(s) spent of prediction
Run A	15	67.608	67.608	0.00123	66.048	69.762	0.00097
Run B	13	65.262	66.858	0.00138	63.442	66.318	0.00107
Run C	11	66.128	67.201	0.00118	62.352	65.738	0.00092
Run D	9	66.260	67.956	0.00126	63.758	66.111	0.00093
Run E	7	65.885	66.995	0.00120	64.089	66.693	0.00098
Run F	5	67.513	67.924	0.00153	64.531	65.927	0.00124
Run G	**3**	**70.436**	**71.335**	**0.00143**	**69.327**	**70.431**	**0.00097**
Run H	0.5	66.254	67.668	0.00143	65.250	67.930	0.00114

From Table 6, the pruning vector degree in the above Table is to measure the similarity between neurons. As for the distinctiveness, the distinctiveness degree is 180 minus the pruning vector degree. In the original investigation, the degree of similarity is 15 [5]. Nevertheless, in this study, the model is over-parameterized, which has 78 neurons in the first hidden layer and 26 neurons in the second hidden layer. Thus, when the degree is 3, the performance of the model is the best, and the pruning effect is the best as well because the loss of accuracy is minimal (The accuracy before pruning is 70.436%, after pruning, it becomes 69.327%). When the degree is 3, the model after pruning remains the high accuracy and pruning can help to increase efficiency.

By horizontal comparison, the pruning of neurons will affect the final accuracy to a certain extent. Even if the initialized weights and bias are assigned to the model after pruning, the pruning without minor adjustments will still affect the model accuracy. However, the reduction of averaged accuracy is within an acceptable range, usually only 1% to 3%. At the same time, the model operation will be more efficient. When predicting a set of test data, the model that removes the neuron takes less time (Reduced from 0.00143 seconds to 0.00097 seconds in Run G), which is conducive to reducing CPU consumption.

3.3 Mini-batch Size in Artificial Neural Network

The mini-batch size is another factor which may affect the testing accuracy. The experiments set the mini-batch size form 10 to 20 to find the optimal parameter. The results below are average accuracy with 50 runs.

Table 7. Mini-batch - Experiment Results with 50 runs average accuracy

		Before pruning		After pruning	
	Mini-batch size	Ave Accuracy(%)	Ave Balanced Accuracy(%)	Ave Accuracy(%)	Ave Balanced Accuracy(%)
Run A	20	68.804	70.625	63.493	67.236
Run B	17	68.559	70.715	66.456	71.130
Run C	15	71.100	72.919	63.276	67.365
Run D	**12**	**71.104**	**72.080**	**69.332**	**70.705**
Run E	10	64.534	65.543	58.008	61.096

After applying the mini-batch gradient descent, the accuracy has a prominent improvement and maintains higher accuracy. When the batch size is 12, the model has the best performance.

3.4 Learning Rate in Long Short-term Memory Network

The optimal parameters of the LSTM model are:

Table 8. Optimal Parameters of LSTM

Parameter	Weight Decay	Number of Layers	Hidden size	Epochs
Value	0.4	1	13	120

The results below are average accuracy with 50 runs with different learning rate.

Table 9. Learning Rate - Experiment Results with 50 runs average accuracy

	Learning Rate	Ave Accuracy(%)	Ave Balanced Accuracy(%)
Run A	0.01	73.147	73.167
Run B	0.009	73.783	73.633
Run C	0.008	73.696	72.625
Run D	0.007	73.912	74.075
Run E	0.006	75.672	74.000
Run F	0.005	73.319	72.500
Run G	**0.004**	**75.926**	**74.491**
Run H	0.003	72.934	70.858

From Table 9, the highest accuracy with the learning rate equals 0.004 is 75.926%, and the balanced accuracy is 74.491%. Compared with the optimal result of artificial neural network (Accuracy is 71.254%, and balanced accuracy is 72.176%), the LSTM model has a higher accuracy, which increases by 4.672%. This demonstrates that the LSTM Network is more suitable for grades prediction problem, which the data has time-series significance.

3.5 Bidirectional Long Short-term Memory Network

The optimal parameters of Bi-LSTM model are:

Table 10. Optimal Parameters of Bi-LSTM

Parameter	Weight Decay	Number of Layers	Hidden size	Epochs
Value	0.4	2	8	120

The results below are average accuracy with 50 runs.

Table 11. Learning Rate - Experiment Results with 50 runs average accuracy

	Learning Rate	Ave Accuracy(%)	Ave Balanced Accuracy(%)
Run A	0.01	74.801	74.500
Run B	0.009	74.174	74.541
Run C	0.008	76.180	75.633
Run D	0.007	76.361	76.084
Run E	**0.006**	**76.540**	**76.680**
Run F	0.005	74.951	75.200
Run G	0.004	73.341	73.350
Run H	0.003	72.803	72.783

From table 11, the highest accuracy with the learning rate equals 0.006 is 76.540 %, and the balanced accuracy is 76.680%. The accuracy results of Bi-LSTM network have a further improvement than LSTM (The Accuracy increases by 0.614% and the Balanced Accuracy increases by 2.189%). The results prove that the Bi-LSTM network outperform LSTM network in grades prediction problem.

Compared with the experimental results in the original paper, results in this study have been improved (Shown in the below Table). The best result of averaged accuracy after running model fifty times is from Bidirectional Long Short-term Memory Model with accuracy 76.540%, which is larger than the previous result, 62.3% [9].

Table 12. Results comparison - with 50 runs average accuracy

Model	Ave Accuracy
ANN Model in original paper [9]	62.3%
ANN Model in this study	71.254%
Long Short-term Memory Model	75.926%
Bidirectional Long Short-term Memory Model	**76.540%**

4 Conclusion and Future Work

The experimental results have obtained a relatively satisfactory accuracy compared with the original paper. It has a certain role in helping to predict final student grades. However, because the sample size is too small, the accuracy of this model is not very high and stable.

Data preprocessing is dramatically significant for a small sample. It can decrease the influence of the unbalanced allocation of training and test set. It can also help the model distinguish the importance of each feature. Removing the outlier can help the stability of the model as well.

Pruning technique in ANN can help refine the model to a certain extent and can shorten the running time, But at the cost of losing accuracy. Dropout neurons randomly in ANN can help model reduce overfitting issues. The mini-batch approach can make the prediction results of ANN more stable. The deep learning models LSTM and Bi-LSTM have better performances than ANN. The model with the best performance in this study is Bi-LSTM. However, because the sample size of data is too small, it is necessary to verify the validity of the models on a larger dataset. In order to enhance the accuracy of prediction and assessment of student performance, the future work is to use the genetic algorithms to explore better solutions.

5 Reference

1. Fernandes, E., Holanda, M., Victorino, M., Borges, V., Carvalho, R., Van Erven, G.: Educational data mining: Predictive analysis of the academic performance of public school students in the capital of Brazil. Journal of Business Research, 94, 335--343 (2019)

2. Lara, J. A., Lizcano, D., Martínez, M. A., Pazos, J., & Riera, T.: A system for knowledge discovery in e-learning environments within the European Higher Education Area–Application to student data from Open University of Madrid, UDIMA. Computers & Education, 72, 23--36 (2014)

3. Gedeon, T. D., Turner, S.: Explaining student grades predicted by a neural network. In: Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan), Vol. 1, pp. 609--612. IEEE (1993)

4. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)

5. Gedeon, T. D., Harris, D.: Network reduction techniques. In: Proceedings International Conference on Neural Networks Methodologies and Applications, Vol. 1, pp. 119--126 (1991)

6. Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J.: A novel connectionist system for unconstrained handwriting recognition. IEEE transactions on pattern analysis and machine intelligence, 31(5), 855-868 (2008)

7. Gers, F. A., Schmidhuber, J., Cummins, F.: Learning to forget: Continual prediction with LSTM. (1999)

8. Graves, A., & Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM networks. In: Proceedings of 2005 IEEE International Joint Conference on Neural Networks 2005, Vol. 4, pp. 2047--2052. IEEE (2005)

9. CHOI, E. C. Y., GEDEON, T. D.: Comparison of Extracted Rules from Multiple Networks. In: Proceedings of ICNN'95-International Conference on Neural Networks, Vol. 4, pp. 1812--1815. IEEE (1995)

10. Jolliffe, I. T., Cadima, J.: Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202 (2016)

11. Cawley, G. C., Talbot, N. L. : On over-fitting in model selection and subsequent selection bias in performance evaluation. Journal of Machine Learning Research, 11(Jul), 2079-2107 (2010)

12. Seni, G., & Elder, J. F.: Ensemble methods in data mining: improving accuracy through combining predictions. Synthesis lectures on data mining and knowledge discovery, 2(1), pp.1-126 (2010)

13. Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S.: Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378 (2018)

14. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958 (2014)

15. Kwak, G. H. J., Hui, P.: DeepHealth: Deep Learning for Health Informatics. arXiv preprint arXiv:1909.00384 (2019)

16. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks, 18(5-6), 602-610 (2005)

17. Mohan, Arvind & Gaitonde, Datta.: A Deep Learning based Approach to Reduced Order Modeling for. Turbulent Flow Control using LSTM Neural Networks (2018)

18. García, V., Mollineda, R. A., & Sánchez, J. S.: Index of balanced accuracy: A performance measure for skewed class distributions. In Iberian conference on pattern recognition and image analysis, pp. 441-448. Springer, Berlin, Heidelberg (2009)