Problem Chosen	2023	Team Control Number
C	MCM/ICM Summary Sheet	2322318

Application of LSTM and WOA optimized Neural Network model in Wordle Game

Summary

Wordle, as a popular word puzzle, offers the statistical data of this game in a file。Based on these statistical data, including the number of reported results, the percentage of different tries, word attributes, etc., this paper establishes a mathematical model to explain the existing data, predict the unknown data, and discuss the relationship between various kinds of data.

For model I， first of all, we pre-process the given data, eliminate some unreliable data and encode the words. The LSTM model is established to simulate the change of the number of reported results with time, and predict it of a specific date. In addition, we also examined the relationship between different attributes of words such as word frequency, whether there are repeated letters, and the number of difficult patterns.

For model II, we take the corresponding relationship between each word and its percentage of different tries as the input test set. Then we establish a neural network model, and use whale algorithm to optimize it, aiming to achieve the prediction of the percentage of the number of tries corresponding to a specific word in the future.

For model III, by using the K-means clustering analysis method, the words in the given data set are classified according to the percentage of tries, then the difficulty coefficient is defined. In this way, the correspondence between the code of the word and the difficulty coefficient is constructed. With this as an input, the neural network model is established and is used to predict the difficulty coefficient of any given word. Spearman correlation test is used to examine the relationship between attributes of words and the difficulty coefficient of words. Finally, comprehensive qualitative and quantitative analysis is carried out for other statistical characteristics of the data set.

During the course of establishing the above model, we also compare the actual value with the predicted value of the model to calculate various error indicators. Moreover, we test the optimization effect, draw a graph for comparison, and constantly seek the best parameter setting. The final model is tested with the word EERIE, and the prediction results are given in our paper.

Keywords: LSTM neural network; WOA-BP neural network; Prediction; K-means cluster analysis; Spearman correlation test

Content

1 Introduction

1.1 Background

Wordle[1] is a puzzle game from The New York Times that is being released daily and is becoming widely known and rapidly gaining popularity, and is now an up-and-coming game in over 60 languages. In this game, players are asked to guess a five-letter word in six or fewer guesses, and are given feedback after each guess. Upon entering the game, players will see six rows of spaces, which equates to six chances, and an English keyboard below. After the player enters a valid English word and clicks "Enter", the color of the letters will change. Green means the letter is in the word and in the correct position; yellow means the letter is in the word but in the wrong place; gray means none of the letters are included in the word. When the player enters a word and all five letters are green, that means the player guessed the right answer!

Wordle also has a hard mode, which, on top of the easy mode, requires players to use the correct letters once they find them, i.e. tiles colored green or yellow, in the next guesses to make the game more difficult. Along with its growing popularity, there are many interesting phenomena and problems hidden in it, and there are many rules to be discovered behind the huge number of daily players and the riddles that people are eager to solve.

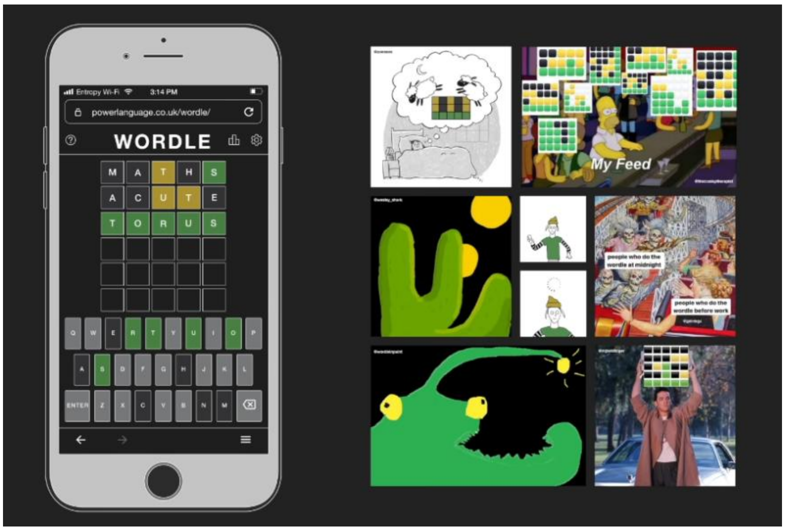


Figure1： Wordle has taken the world by storm

1.2 Restatement of the Problem

In this issue, we are given a copy of Wordle's daily user statistics. We will address the following questions based on the above data file.

• Build a model to explain the varying number of daily reports, use this model to predict about the March 1, 2023 reported outcome measures, and create a prediction interval. Consider the impact of word attributions on daily analysis of the percentage of scores in hard mode.

• Develop a model to predict the distribution of relevant percentages in the reported results at future dates. Accurately analyze and adjust the uncertainty of the model. Predict a percentage distribution of scores on the word EERIE on March 1, 2023 based on this model.

• Based on the attributions of the words, develop and summarize a word classification model, and from this determine the attributes of the words associated with each classification. Analyze the word EERIE and discuss the accuracy by classification model.

• List and describe other interesting features of the dataset based on the above model and other information already available.

• Provide a one- to two-page letter to the New York Times editor concluding the results of our analysis.

1.3 Our Approach

This topic requires us to develop a model to explain the variation of reported results, and give a prediction of the number of percentages of different tries according to a given word, as well as to classify a given word by difficulty. In the meantime, we should identify the attributes of a given word that are associated with its difficulty or the number of hard mode. The main aspects of our works are as follows：

• Build a LSTM model to simulate the change of the number of reported results with time, and predict it of a specific date.

• Establish a neural network model optimized by whale algorithm to predict the percentage of the percentage of tries corresponding to a specific word in the future.

• Use the K-means clustering analysis method to classify the words and define their difficulties. The neural network model is established through those data, and is used to predict the difficulty coefficient of any given word.

• Spearman correlation test is used to examine the relationship between different attributes of words.

2 General Assumptions and Model Overview

In order to simplify the problem, we make the following basic assumptions, each of which is properly justified.

• **Assumption 1**. The reported score percentages and the number of players scoring in the data are accurate and reliable; these data will not be significantly affected by uncertainty over a short period of time.

The model building and prediction in this paper need to be supported by correct data, because the model we build is based on the data provided in the subject, and only the high validity of the data can guarantee the high reliability of the model, and we need to ensure that the data will not fluctuate significantly due to unexpected events.

• **Assumption 2**. The words of the day are randomly generated.

Only if the words are randomly generated, our model is associated with the word attributes and has researchable value and predictability for the relationship between the number and percentage of scores and the word attributes.

• **Assumption 3**. The number of scoring players and the relationship between the percentage of scores and the attributes of the words are predictable.

Although there is no obvious linear relationship between the number and the percentage of scores and the attributes of the words, the prediction can be made in a short period of time after accumulating certain previous data and with appropriate mathematical methods.

First, in order to better adapt and predict the reported outcome quantity fluctuations, we first build a quantity prediction model based on LSTM neural network. After modifying the parameters of the model with appropriate data for training, the model was more successful in predicting the number of reports. By thinking about the observation of word characteristics, we identified several word attributes that may influence the number of people who choose the difficult model, and identified the attributes that really have an impact by Spearman's correlation test.

Secondly, we built a neural network model and optimized it with the whale algorithm, combined with word coding performed in the data pre-processing stage, trained it with existing data, and tested it to achieve a better prediction of the daily result distribution. Next, we measure and evaluate the difficulty of each word based on the average number of successful responses required, and use K-means clustering analysis to classify it into five difficulty levels, and use this as input to train a neural network model that can then evaluate and analyze the difficulty levels of other words. has a positive correlation with the difficulty level of the word. Finally, the hidden patterns within the given data are further explored to obtain more valuable conclusions, and the robustness and sensitivity of the model are tested and analyzed using statistical indicators to summarize the strengths and weaknesses of the model.

In summary, the whole modeling process can be shown as follows:

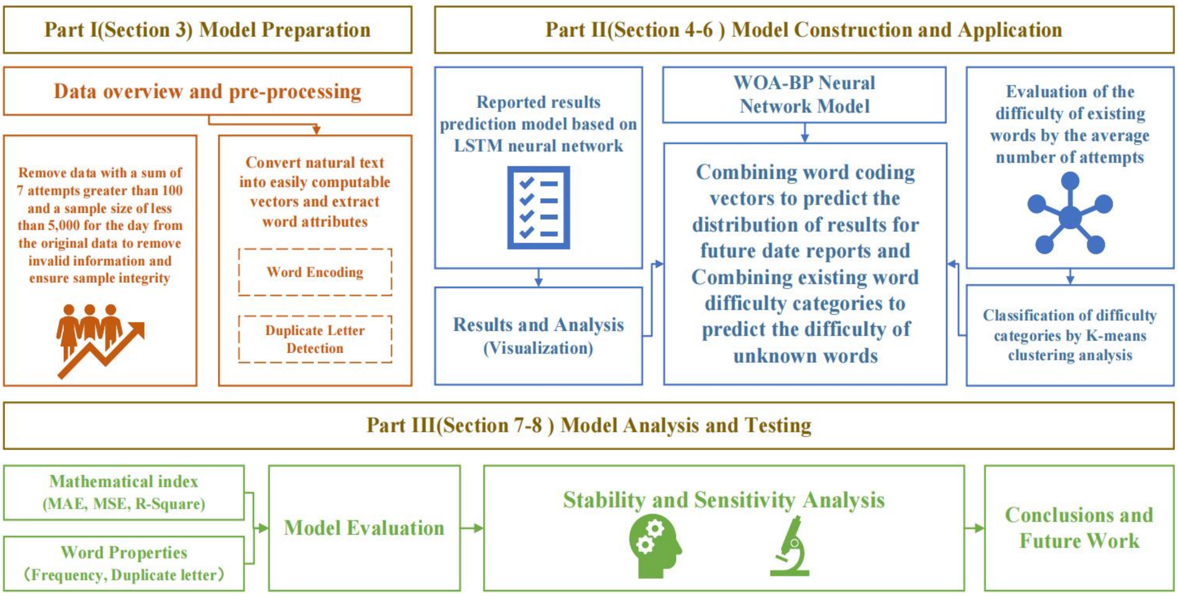


Figure2：Overall modeling idea flow chart

3 Model Preparation

3.1 Notations

Table 1 Notations

Symbol	Description
ρ_s	Spearman correlation coefficient
$input\_train$	input training data set vector
$output\_train$	output training data set vector
$input\_test$	input test set vector
$output\_test$	output test set vector
$inputnum$	the number of input neurons
$outputnum$	the number ofoutput neurons
$hiddennum$	hidden layer nodes of the neural network
$hiddennum\_best$	The best value of the number of hidden layer nodes
MSE	Minimum mean square error
dim	The number of independent variables in the whale algorithm
$fitnessci$	Self-compiled function for calculating fitness
$compute\_error$	Self-compiled function to calculate error
ρ	Correlation coefficient between variance vector and difficulty
ρ_2	Correlation coefficient between variance vector and word

where we define the main parameters while specific value of those parameters will be given later.

3.2 Data Pre-processing

The data given in the question are large and complex to apply and there are some anomalous data mixed in, so we need to pre-process the data appropriately.

The raw data was preprocessed using matlab software. The data from the excel table was read into matlab using the xlsread function, noting that numbers and letters were read into separate numeric and character matrices. When the sum of the percentages of the data in 7 attempts is greater than 100, or when the number of samples on a particular day is less than 5000, the corresponding data rows and character rows are deleted. A total of 280 sets of valid data were obtained after the operation was completed. This filtering criterion was chosen because it can screen out some invalid information while maintaining the integrity of the sample. If we strictly limit the sum of percentages to 100 and the number of samples to not less than 10,000, we will end up with only 180 sets of valid data, which is not conducive to the subsequent modeling analysis. Write the data processing results into a new excel sheet.

Encoding of words in the data. According to the alphabetic order a-z are mapped as numbers 1-26 respectively, then each word containing 5 letters corresponds to 5 decimal numbers, and the words are encoded with these numbers to reflect the characteristics of the words. A loop structure is used in matlab to traverse each row of data for coding. The coding results were added to an excel sheet. For example, word "manly" corresponds to the code [13 1 14 12 15].

Detects whether the words contain duplicate letters. Based on the coding in the previous step, simply compare the 5 numbers corresponding to each line of words to see if they are the same. If they are the same record 1, otherwise record 0. Use matlab to write the results in an excel sheet.

4 Model I：LSTM neural network prediction model

The extremely high popularity and propagation trend of wordle has led to data that is highly volatile, a very sophisticated nonlinear dynamic system. In this nonlinear dynamic system, it is tough to achieve satisfactory fitting and prediction results using conventional linear prediction methods. Prediction methods are difficult to achieve satisfactory fitting and prediction results. However, the combination of computer technology and artificial intelligence enables the modeling and prediction of nonlinear chaotic gray systems using new technical tools. Among them, artificial neural networks (ANNs) are very prominent in predicting and fitting nonlinear dynamic systems. It has been shown that ANNs can be applied in the areas of language recognition and automatic control, as well as for prediction and evaluation. Several studies have shown that its accuracy is considerably better than that of autoregressive models.

4.1 LSTM neural network

This topic limits us to knowing only the number of reported results as of that date. Moreover, we cannot predict the change in the number of reported results by dint of theory. Therefore, we consider a prediction model that uses the number of reported results for the past almost one year to forecast the number for a future date as one of the indicators for our decision.

If the randomness is substantial and irregular, conventional time series models (such as ARIMA, etc.) cannot achieve our expected results, so we transform our perspective and consider using recurrent neural networks to build models to help us solve the problem (RNN), which has achieved good performance results in areas such as natural language processing. However, due to the enormous amount of data for this problem, it is prone to difficulties such as the disappearance of data gradients. And the LSTM of long time line is a special kind of recurrent neural network (RNN) that can be used for time series prediction and solves the problem of gradient disappearance and gradient explosion in the training process of RNN.

The LSTM features valve nodes added to each layer outside the RNN structure. the LSTM has 3 types of valves: forgetting valves, input valves and output valves. These valves can be opened or closed and are used to determine if the memory state of the model network (the previous state of the network) has reached a threshold for the output of the layer to be added to the current computation of that layer. The valve node computes the memory state of the network as input; if the output reaches the threshold, the valve output is multiplied by the current layer's computation and used as input to the next layer; if the threshold is not reached, the output is forgotten. The weights of each layer, including the valve nodes, will be updated in the backpropagation training of each model.

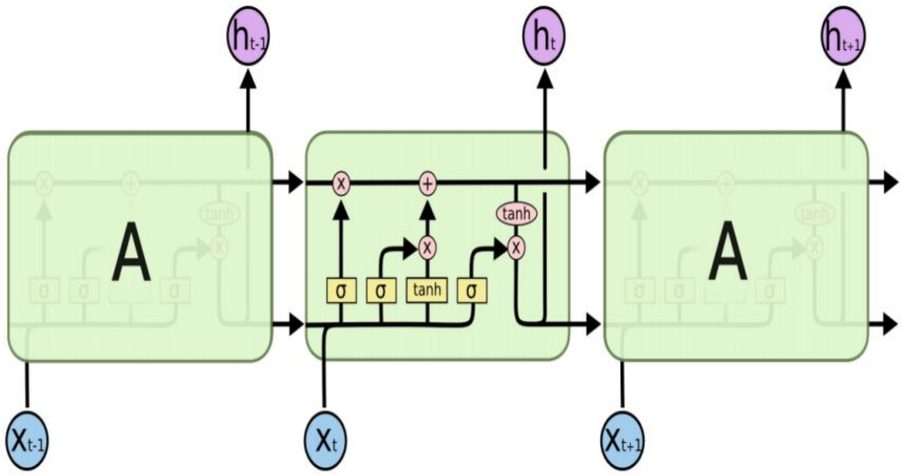


Figure3：The repetition module in the LSTM contains four interaction layers

4.2 Development of prediction model based on LSTM neural network

The LSTM neural network received sufficient data, and after the training was completed, we took the number of reported results for one year as input and delivered them to the LSTM neural network. The structure of the LSTM neural network is shown in the following figure.[2]

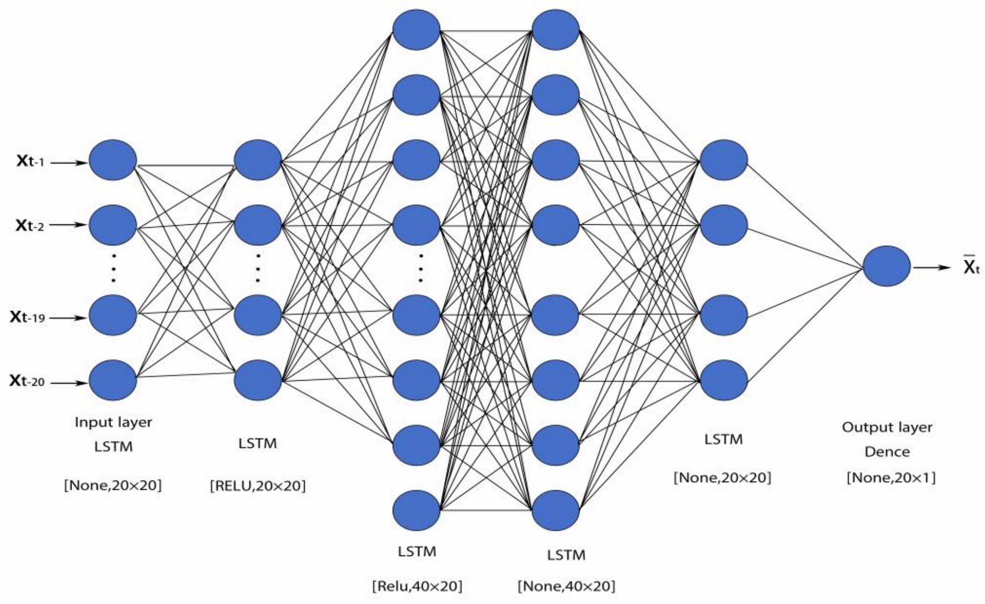


Figure4：Schematic diagram of LSTM neural network structure

The RELU nonlinear activation function is chosen for the first and second hidden layers, and no activation function is used for the remaining layers. Each layer is an LSTM layer, except for the output layer which is. The layer relationships of the model and the parameters to be estimated are shown in the following table.

Table2: LSTM model hierarchical relationships and parameters to be estimated

Layer	Output shape	Param
$lstm$	(None,20,20)	1760
$lstm_1$	(None,20,20)	3280
$lstm_2$	(None,20,40)	9760
$lstm_3$	(None,20,40)	12960
$lstm_4$	(None,20)	4880
$dense$	(None, 1)	21

The loss function uses mean squared error, and the optimization algorithm chooses the ADAM algorithm

The model is pre-trained with a divided pre-training dataset. Epoch is 100, $batchsize$ is 16.[3] After training, make the model predict the number of reported results and compare with the real value, the result is shown in the following figure.

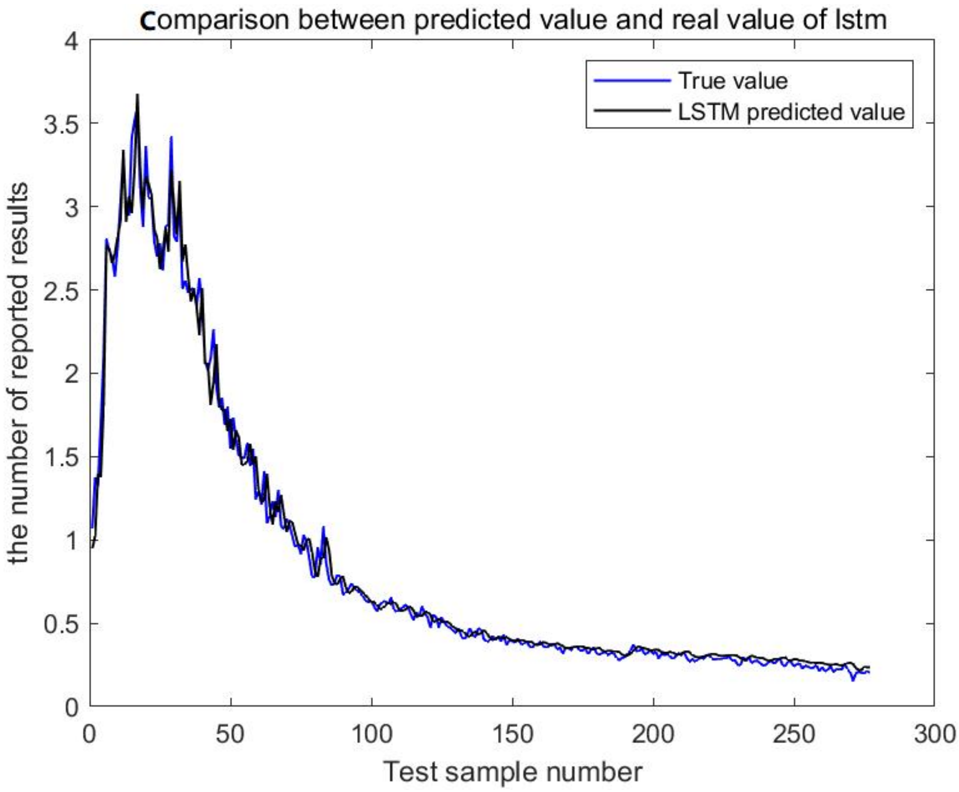


Figure5：Reported results prediction VS Real value

As you can see, the pre-trained model has the ability to predict future reported results. After pre-training, the LSTM model was used to predict the number of results reported on

2023/3/1 required by the question, and the predicted result was obtained as 16559.9±1474.3.

4.3 Related word properties

After further analysis to predict the number of reported results, we explored the words in this data file in depth to try to identify some of the word attributes that would affect the percentage of people with difficult patterns.

After observing the given word characteristics, we guessed that there are two main influencing factors: the presence of repeated letters in the word and the frequency of the word's use in daily life. For the attribute of whether there are repetitive letters, it has been marked in advance in the data preprocessing stage and can be directly analyzed by correlation; for the word frequency, we used the $WordFrequencyData$ function [4] that comes with mathematica, and the data used in this function is calculated from Google's e-book dataset, which has a large enough amount of data and authenticity. There is a guarantee of authenticity.

In the following, it is specifically analyzed by Spearman correlation analysis, and the process is expressed with appropriate statistical indicators, from which the strength of the correlation coefficient can be obtained. On this basis, the significant difference test, i.e. t-test, is done to test whether the two sets of data are significantly correlated, and the main formulas used in the process are the following:

$$\rho_{\mathrm{s}}=\frac{\sum_i\left(x_i-\bar{x}\right)\left(y_i-\bar{y}\right)}{\sqrt{\sum_i\left(x_i-\bar{x}\right)^2\left(y_i-\bar{y}\right)^2}}$$

Where ρ is the correlation coefficient.

The proportional relationship between these two attributes and the number of difficult patterns is shown in the figure below, where the shades of color represent the strength of the correlation, and the numbers on the figure are the magnitude of the correlation coefficient.

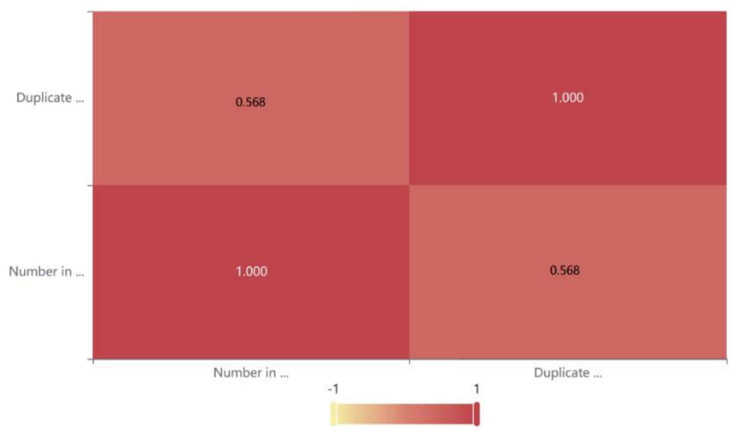


Figure6：Spearman correlation analysis of the percentage of people in the difficult mode with the repeated letter attribute

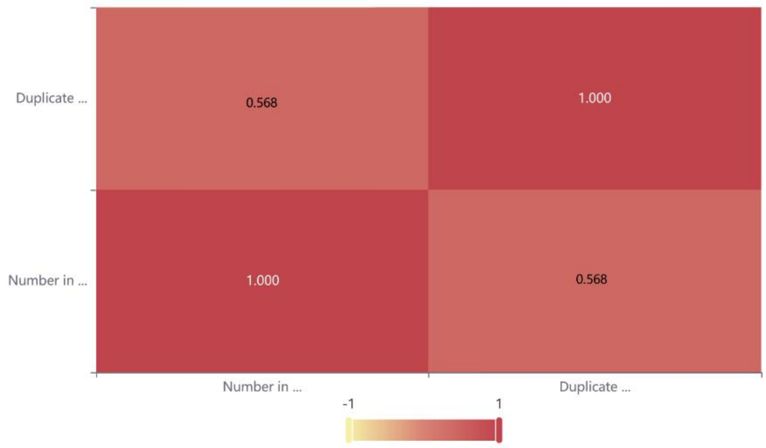


Figure7： Spearman correlation analysis of the percentage of the number of people in the difficulty mode and the word frequency attribute.

From the figure, we can see that the attribute of the presence of repeated letters is positively correlated with the proportion of the number of people in the difficult mode, and it is an extremely significant correlation, with a correlation coefficient of 0.568 in the case of such a large sample size, indicating that the two become strongly correlated. On the contrary, the correlation between word frequency and the proportion of people with difficulty mode is very light, with a correlation coefficient of 0.007, which leads to the conclusion that the two are not correlated.

5 Model II：WOA-BP neural network model

For the BP neural network training process, the initial weights and thresholds are generated by random numbers, which have an impact on the structure of the trained network, the whale optimization algorithm[5] is used to optimize the initial weights and thresholds of the BP neural network, so as to obtain a more stable WOA-BP neural network model. Its model building flow chart is shown below.

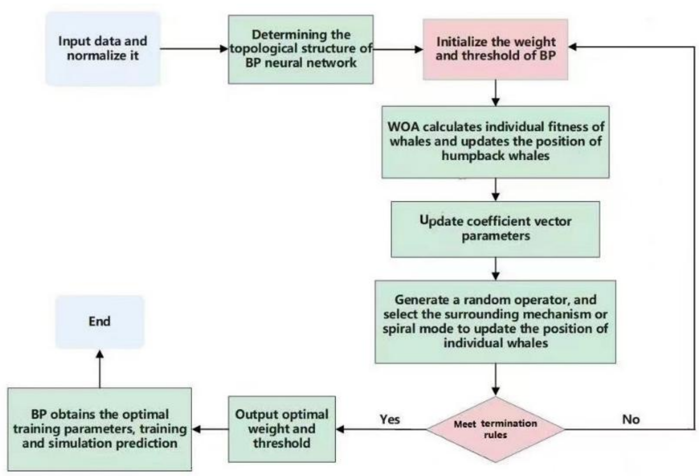


Figure8：Model building flow chart

5.1 WOA-BP neural network prediction model for percentage of attempts

5.1.1 Neural network modeling

The purpose of this model is to train with known data such that the prediction of the percentage of attempts is given for an arbitrary letter code input. The letter code is considered as the input to the model and the percentage of attempts is the desired output of the model.

The data is read from a preprocessed data table and the data set is randomly scrambled using the $randperm$ function. The number of training sets is set to 5/6 of the valid data, and the number of test sets is set to 1/6 of the valid data. $mapminmax$ function is used to normalize the data to obtain $input\_train, output\_train, input\_test, output\_test$ vectors.

According to the question, set the number of input neurons to $inputnum = 5$ and the number of output neurons to $outputnum = 7$, According to the empirical formula:

$$hiddennum = \sqrt{inputnum + outputnum} + a$$

a is generally taken as an integer between 1-10, it is known that $hiddennum$ should be taken between 5-13. Establish the initialized minimum squared error as $MSE = 1e + 5$, establish a loop to traverse each value of $hiddennum$, get $hiddennum = 11$, MSE minimum, record the $hiddennum$ as $hiddennum\_best$ at this time. substitute $hiddennum\_best$ to establish the BP neural network model, respectively, the number of trainers, learning rate and other network parameters are configured. The network parameters such as training times, learning rate, etc. are configured. The trained model is then simulated using the sim function. The inverse normalization of the prediction results is calculated to obtain the error indicator as follows:

Table3：BP Neural Network Prediction Error Indicator

Standard BP neural network				
Mean absolute error	1.0236	0.28489	0.34304	0.92976
Mean square error	10.8026	0.949715	1.04519	7.57045
Root of mean square error	3.2867	0.97453	1.0223	2.7514

5.1.2 Model optimization using the whale algorithm

The neural network model is optimized using the whale algorithm. Firstly, the WOA parameters are initialized and according to the formula:

$$dim = inputnum * hiddennum_{\mathrm{best}} + hiddennum_{\mathrm{best}} + hiddennum_{\mathrm{best}} * \mathrm{outputnum} + \mathrm{outputnum}$$

The number of independent variables dim is calculated, the upper and lower bounds of the independent variables are given according to the number of independent variables, and the position vector and leader score are initialized. The population is initialized using a loop structure.

In the cycle of optimization, it is necessary to establish the objective function and calculate the value of the objective function and update the leader position so that the parameters can be updated. The more optimal parameters are approached gradually in a continuous loop. We wrote the $fitnessci$ function to calculate the value of the objective function, also known as the fitness degree. The principle is to extract the weights from the neural network to the input layer connected with the hidden layer, the threshold of neurons in the hidden layer, to the weights of the hidden layer connected with the output layer, and the threshold of neurons in the output layer respectively to divide, so as to assign values to the network weights. After the network training, the individual whale fitness is calculated. The smaller the value of the fitness function, the more accurate the training and the better the prediction accuracy of the model. According to the calculation results, the humpback whale position is updated and the coefficient vector parameters are updated. Subsequently, the random operator is generated and the shrinkage envelope is selected to update the position of individual whales. When the parameters obtained from the cyclic process satisfy the termination rule, the cycle ends and the optimal parameter values are output. The optimization curve obtained according to the optimization process is plotted as shown in the figure below.

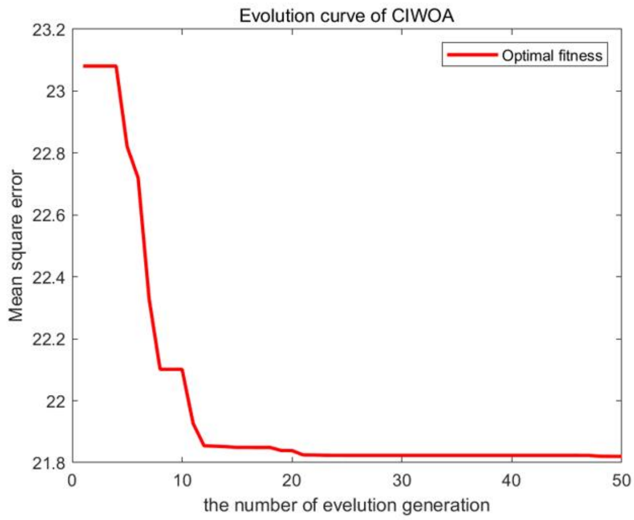


Figure9：Whale algorithm optimization curve

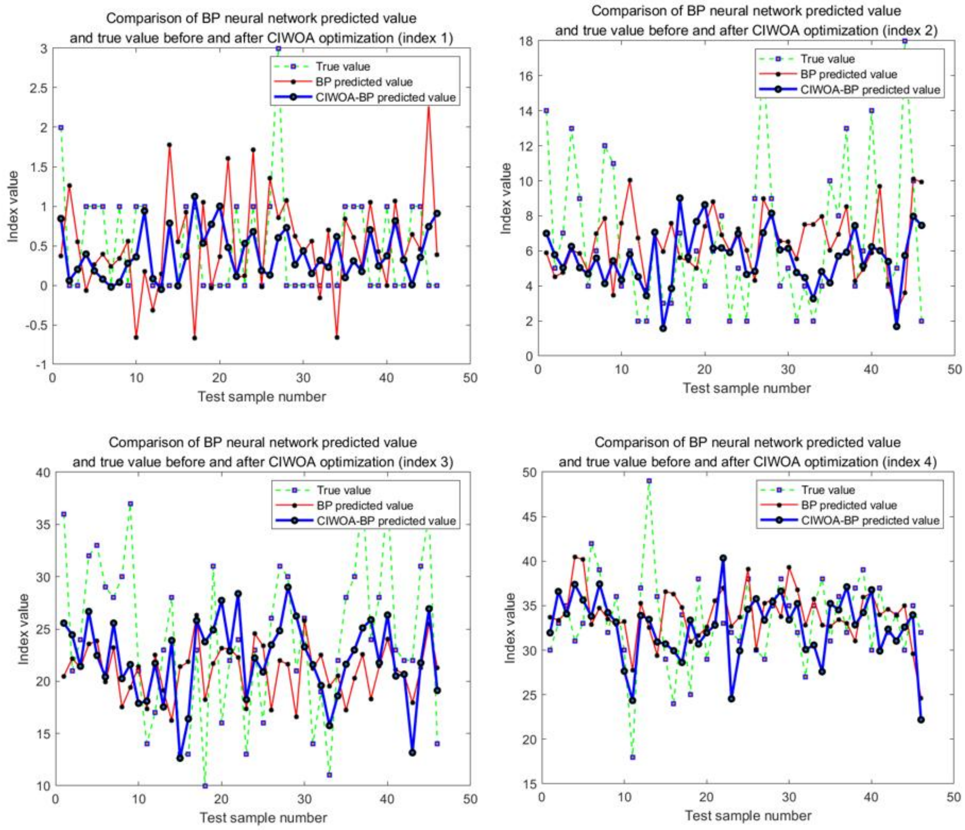
From the optimization curve, we can see that the MSE as a whole shows a decreasing trend and finally stabilizes at about 21.8, indicating that the whale optimization algorithm has played a certain effect.

The optimized neural network is trained, the optimized neural network is tested, the data is denormalized using the $mapminmax$ function and the error metric is calculated using the $compute\_error$ function. The resulting optimized error metrics were output and compared with the pre-optimized metrics, and it was found that all errors were significantly reduced.

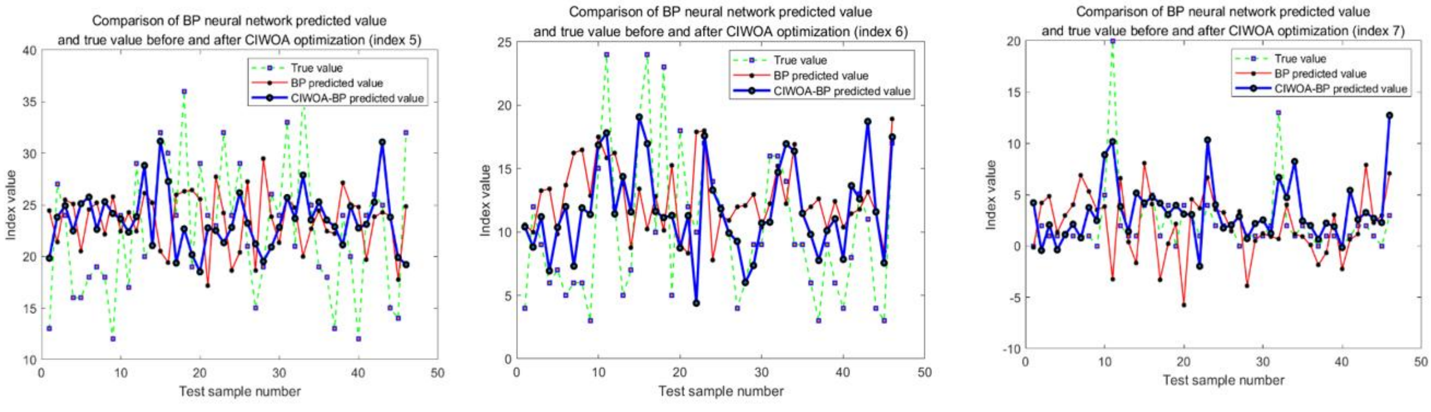
Table4：WOA-BP Neural Network Prediction Error Indicator

CIWOA optimized BP neural network				
Mean absolute error	0.82632	0.3621	0.21417	0.60572
Mean square error	5.84205	1.11832	0.395535	3.47683
Root of mean square error	2.417	1.0575	0.62892	1.8646

For the seven different attempted results of the test set data, three curves are plotted for the true value, the BP neural network predicted value, and the CIWOA-BP neural network predicted value, as shown in the following figure. As can be seen from the figures, compared with the BP predicted values, the CIWOAP-BP predicted values are more consistent with the true values in most data points, and the change trends are more similar to the true values.



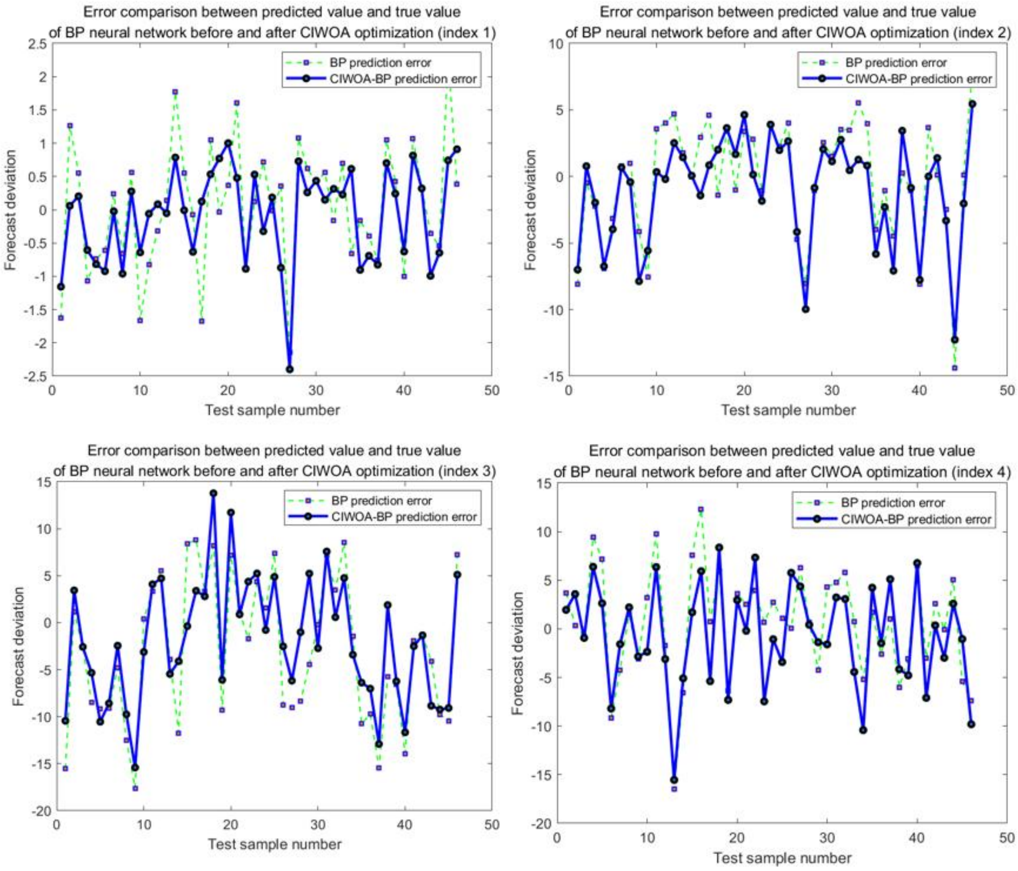
(a)



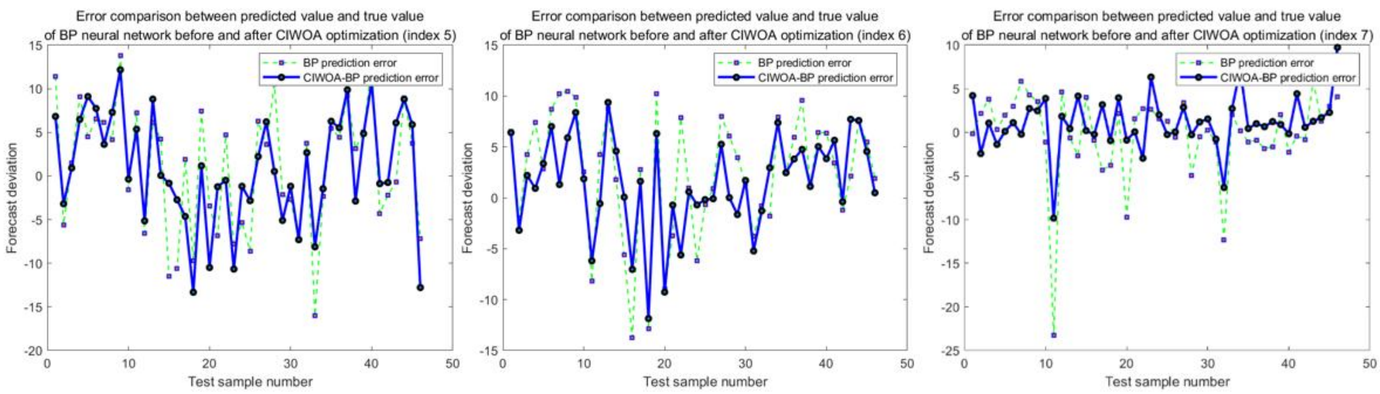
(b)

Figure10：(a)Comparison of test set output curves (1-4 tries);
(b)Comparison of test set output curves (5-7 tries)

For the seven attempts on the test set data, two curves, the difference between the predicted and true values of the BP neural network and the difference between the predicted and true values of the CIWOA-BP neural network, are plotted, as shown below. From the figures, it can be seen that the mean value of the prediction error of the CIWOA-BP curve is closer to 0 and its variance is smaller, so the prediction effect of CIWOA-BP is better than the direct prediction of BP.



(a)



(b)

Figure11：(a) Comparison of test set data difference curves (1-4 tries); (b) Comparison of test set data difference curves (5-7 tries)

5.1.3 EERIE prediction results

For the word $EERIE$, its encoding vector is $vector = [5\ 5\ 18\ 9\ 5]$, which is fed as data into the WOA-BP neural network model for testing, and the test results are normalized to obtain the final prediction percentages for the word for attempts 1-7 as shown in the table below.

Table5：Projected percentage of EERIE

Percentage of 1 to 7 tries of EERIE
2.9282
5.9659
23.8872
24.2293
16.0794
3.7902
23.1198

5.2 Difficulty classification prediction model based on WOA-BP neural network

5.2.1 Training data preparation

First, use the $xlsread$ function to read in the data that has already been processed in the data preprocessing step. In this problem, we need to solve the difficulty corresponding to each word, so the valid data extracted are the 5-digit codes and the number of attempts corresponding to each word for the words during the data pre-processing.

After calculating the difficulty index of each known word, the known words were classified into five different difficulty levels using the K-means cluster analysis method and used as input data for this model, and the specific classification results are shown in the figure below.

Figure12：Known word classification results

5.2.2 Model construction and optimization

The data set is randomly disrupted, the number of test set is set to 18, and the number of training set elements is set to 262. Use the $mapminmax$ function to normalize the above data. Since the number of letters in the input word is 5, the number of input nodes i.e.

$inputnum = 5$ and since the output is the difficulty factor of the word, the number of output nodes i.e. $outputnum = 1$.

Similar to the second question, some initial parameters of the neural network are set and configuration parameters of the BP neural network for constructing the best hidden layer nodes are configured. The model is built using the $newff$ function, the training is started using the $train$ function, and the trained model is used to make predictions using the sim function. The prediction results are denormalized and the $compute\_error$ function is written for error calculation to obtain the error parameter values of mean absolute error, mean square error, root of mean square error, etc. The running results are shown in table below:

Table6：BP Neural Network Prediction Error Indicator

Standard BP neural network	
Mean absolute error	0.62592
Mean square error	0.59638
Root of mean square error	0.77225

Next, the BP neural network is optimized using the whale algorithm. First, the WOA parameters are initialized, the upper and lower bounds of the independent variables are given according to the number of independent variables, and the position vector and leader score are initialized, and the population is initialized using the loop structure.

In the cycle of optimization, it is essential to establish the objective function and calculate the value of the objective function, and update the leader position so that the parameters can be updated. The better parameters are approximated gradually in a continuous cycle.

After the network training, the individual whale fitness is calculated. The smaller the value of the fitness function, the more accurate the training is and the better the prediction accuracy of the model. According to the calculation results, the humpback whale position is updated and the coefficient vector parameters are updated. Subsequently, the random operator is generated and the shrinkage envelope is selected to update the position of individual whales. When the parameters gained from the cyclic process satisfy the termination rule, the cycle ends and the optimal parameter values are output. The optimization curve obtained according to the optimization process is plotted as shown in figure below.

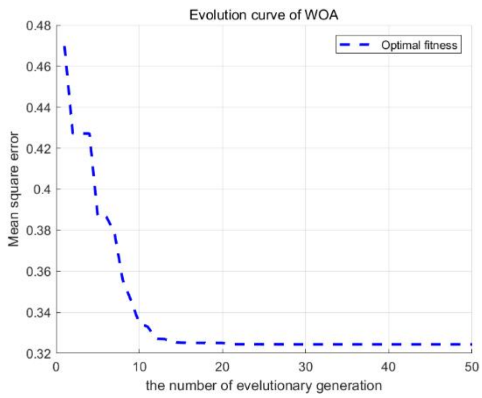


Figure13：Whale algorithm optimization curve

From the optimization curve, we can see that the MSE shows a decreasing trend and finally stabilizes at about 0.32. This indicates that the optimization algorithm has played a certain optimization effect.

5.2.3 Model Application

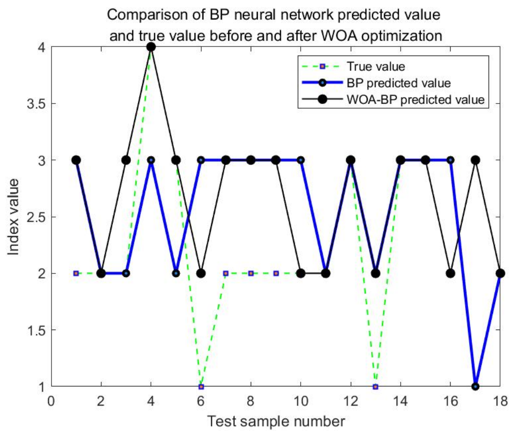
The optimized network is again trained and predicted using the train function, and the error is calculated for the prediction results and compared with the results when it is not optimized, and the data results are shown in the table below.

Table7：WOA optimized BP Neural Network Prediction Error Indicator

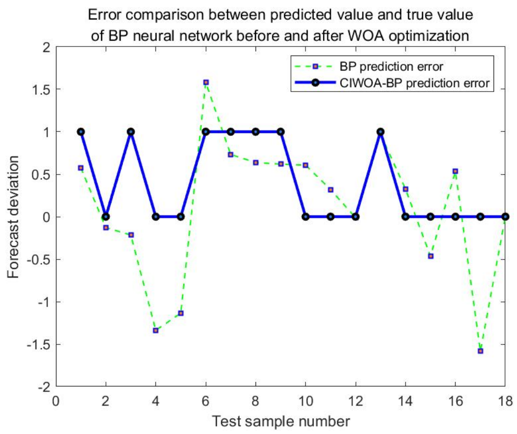
Standard BP neural network	
Mean absolute error	0.38889
Mean square error	0.38889
Root of mean square error	0.62361

It can be seen that the optimized model is evidently smaller than the pre-optimized model in all the three error metrics mentioned in the previous section, which indicates that the optimized model will be closer to the real results when making predictions.

For the data in the test set we plot the predicted data against the real data, and the specific results and errors are shown in the following figure.



(a)



(b)

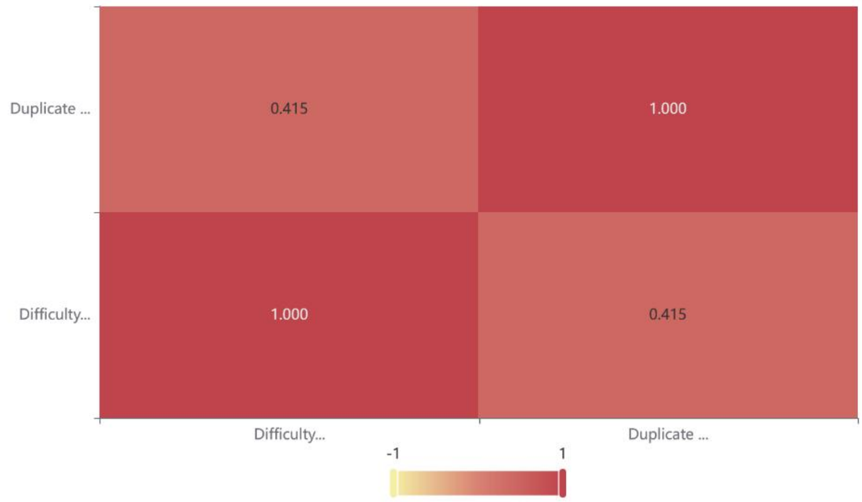
Figure14：(a) Difficulty classification result comparison curve;
(b) Difficulty classification result error curve.

5.2.4 EERIE prediction results

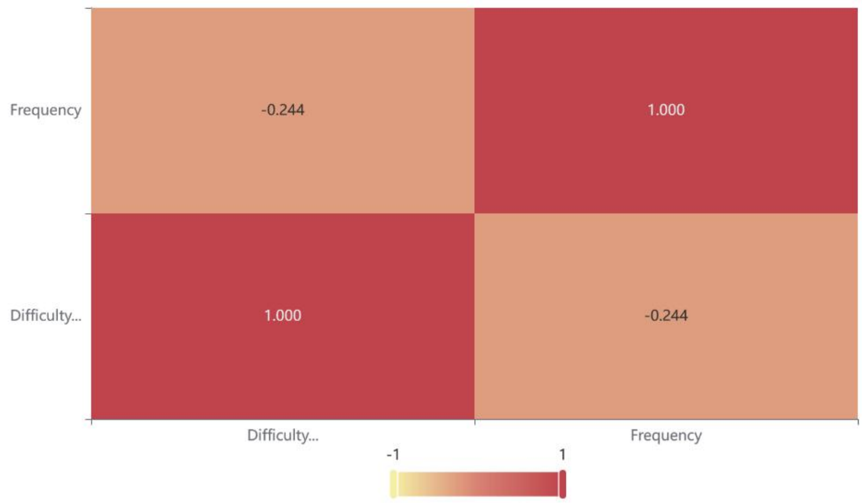
For EERIE, the model optimized with the whale algorithm was tested and the $classify\_difficulty\_after\_algothorim$ function (the assignment principle is similar to $classify\_difficulty\_by\_data$) was called to calculate its final score of 4. This indicates that the difficulty of the word guess is on the high side.

5.2.5 Related Word Properties

Similarly to subsection 4.2, here it is required to explore the word attributes associated with the difficulty level, again considering the two attributes of the presence of repeated letters in the word and the frequency of the word, and testing their relevance using Spearman's correlation analysis to obtain the results shown in the following figure:



(a)



(b)

Figure15：(a) Spearman correlation analysis of the percentage of difficulty score with the repeated letter attribute;(b) Spearman correlation analysis of difficulty score and the word frequency attribute.

Figure 15(a) shows the correlation between word repetition and word difficulty classification. The correlation coefficient of "0.415" indicates that the correlation is extremely intense and positive. Figure 15(b) shows the correlation relationship between word frequency and word difficulty classification. The correlation coefficient of "-0.244" indicates a weak correlation and a negative correlation between the two. This also indirectly confirms the conclusion drawn in section 4.2.

6 Results

We also analyzed and explored some other statistical characteristics of the data set. First, we analyzed the relationship between the percentage data of different tries and the difficulty coefficient of words. The analysis results are shown in figure below.

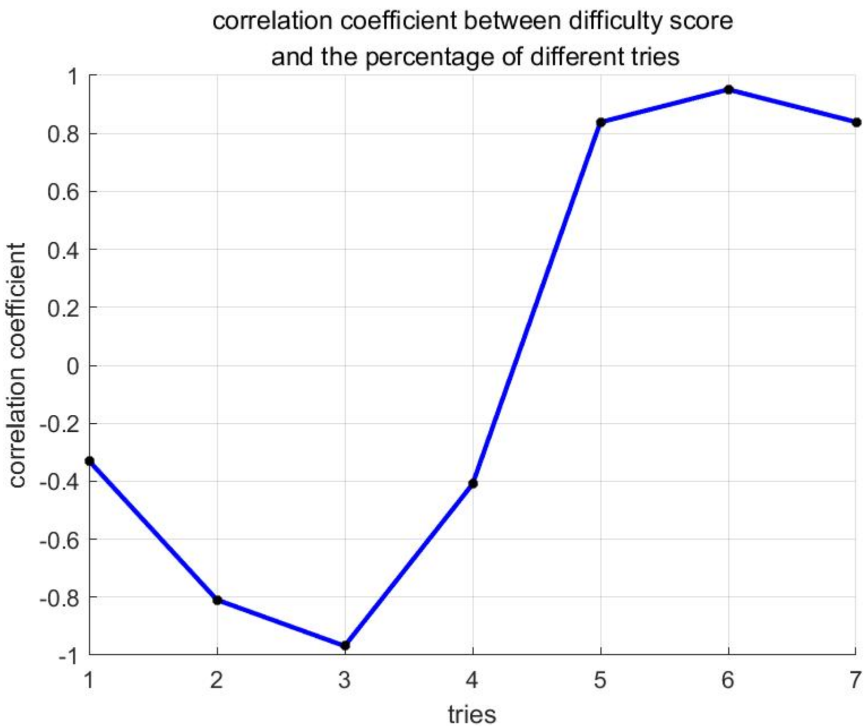


Figure16：Correlation coefficient between difficulty and number of attempts

It can be seen from the figure that the data of 2tries and 3tries are significantly negatively correlated with the difficulty, while the data of 5tries-7tries are significantly positively correlated with the difficulty. The more times you try, the more difficult it is, which is basically in line with our expectations. However, due to the small size and high randomness of the data in the data set, 1 try cannot observe significant negative correlation. Because of its middle position, the size of 4tries cannot reflect the difficulty as well.

Further, we calculated the correlation between the percentage of different tries, and the statistical results are shown in figure below：

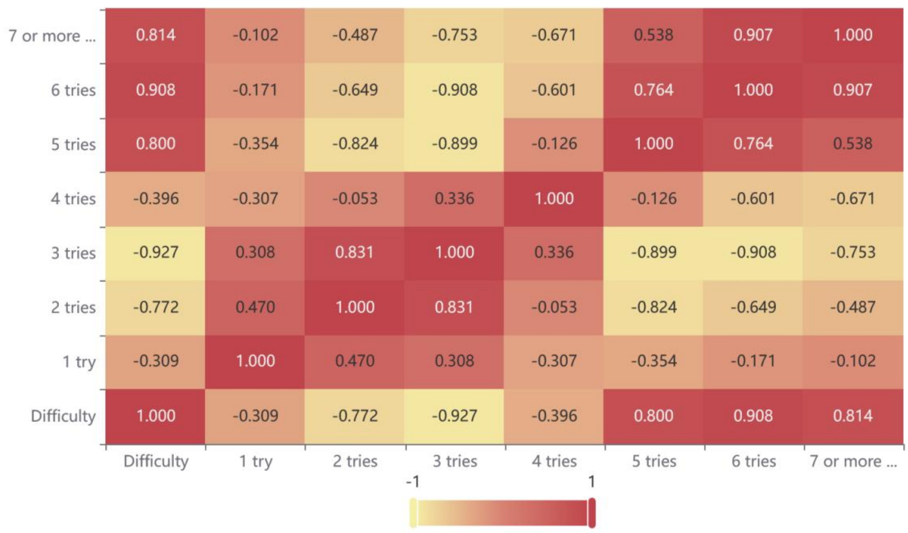


Figure17: Correlation statistics between different tries

It can be seen from the figure that two similar tries tend to show a good positive correlation, while the tries with a large difference in the number of tries tend to show a strong negative correlation. We can understand that the data of the proportion of similar tries often changes synchronously, while the data of the proportion of tries far away often changes in the opposite direction: for example, when the overall difficulty increases, the percentage of 5tries-7tries tends to increase at the same time, while the percentage of 2tries-3tries tends to decrease at the same time. Here, 4tries is an exception. It is often not closely related to the number of other tries.

For the percentage of the number of tries per word, we calculate its variance to get the vector var_{vector}. We calculate the correlation coefficient between it and the difficulty coefficient, and get $\rho = 0.5269$, we can see that there is a positive correlation between the two. The more difficult the word is, the greater the variance of the number of attempts will be. Calculate the correlation coefficient between the variance and whether there are repeated letters in the word to get $\rho_2 = 0.4151$, so it can be seen that when the letters in the word are repeated, the variance of the number of guesses will also increase.

7 Stability and Sensitivity Analysis

For the established model, test the robustness of the system: input commonly used words, rare words and wrong words respectively, such as horse, train, abide, appal, abcde.

It can be seen from the data that when the input test words are our commonly used words, the percentage of small tries is significantly higher than that of uncommon words or wrong words, and the percentage of 7tries is significantly lower. The percentage of uncommon words is more scattered and the percentage of 7tries significantly increases. The percentage of wrong words is concentrated in 6tries and 7tries. This shows that the system is very sensitive to wrong words or strange words, and the system is robust when disturbed by errors.

For the model in question 3, the test results are shown in table below.

Table9: Robustness test data for difficulty

Words	Difficulty
eeeee	4
zzaaa	5
apple	2

It can be seen from Table 9 that when the input test word is our common used word, the difficulty coefficient obtained is significantly lower, while when the input is a string of letters with repeated letters and does not form a word, the difficulty coefficient is significantly higher. This shows that the system is very sensitive to wrong words or repeated letters, and the system is robust when disturbed by errors.

8 Strengths and Weaknesses

8.1 Strengths

• With powerful memory function: LSTM is a recurrent neural network whose neuron's output will act on itself as the next input. This structure gives it a memory function that is well suited for solving time series prediction problems.

• Avoiding the gradient disappearance or explosion problem: Normal RNN networks have difficulty in preserving information in learning for a long time and may suffer from serious problems such as gradient disappearance or gradient explosion. LSTM uses special units that enable it to preserve inputs for a long time.

• Highly generalizable: In theory, this model can be used to monitor and predict the volume fluctuations of any game, and as long as we adjust and modify it according to the database updates of that game, we can get a model with wide applicability.

8.2 Weaknesses

• During the training process of the neural network model, some of its initial parameter values have randomness, which makes the results obtained from each prediction may have small fluctuations.

• Observation of the evolutionary curve shows that the mean square error still exists in the optimized model. This can also cause random perturbation errors on the prediction data of the model.

• Unable to handle too large data: Although LSTM can handle long sequences, it does not work well in the later stages when faced with large data.

9 Letter to the Puzzle Editor

Since you've asked as to do an analysis of the results in a file including the statistical data of Wordle, we're now writing to explain our results. To sum up, based on your statistical data, we established a mathematical model to explain the existing data, predict the unknown data, and discuss the relationship between various kinds of data.

First of all, we preprocess the given data, eliminate some unreliable data and encoded the words. In the meantime, we mark the words which have repeated letters. The LSTM model is established to simulate the change of the number of reported results with time, and predict it of a specific date. We draw the curve of LSTM predicted value and true value in the figure below.

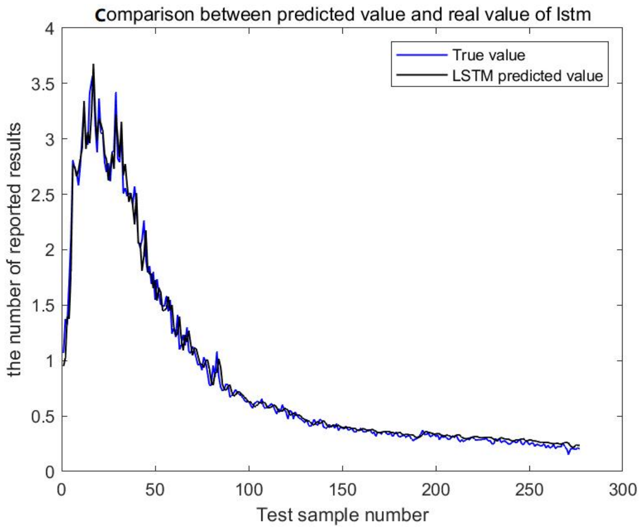


Figure1: Reported results prediction VS Real value

We take the corresponding relationship between each word and its percentage of different tries as the input test set. Then we establish a neural network model, and use whale algorithm to optimize it, aiming to achieve the prediction of the percentage of the number of tries corresponding to a specific word in the future. The figure below shows the predicted value of our model before and after being optimized.

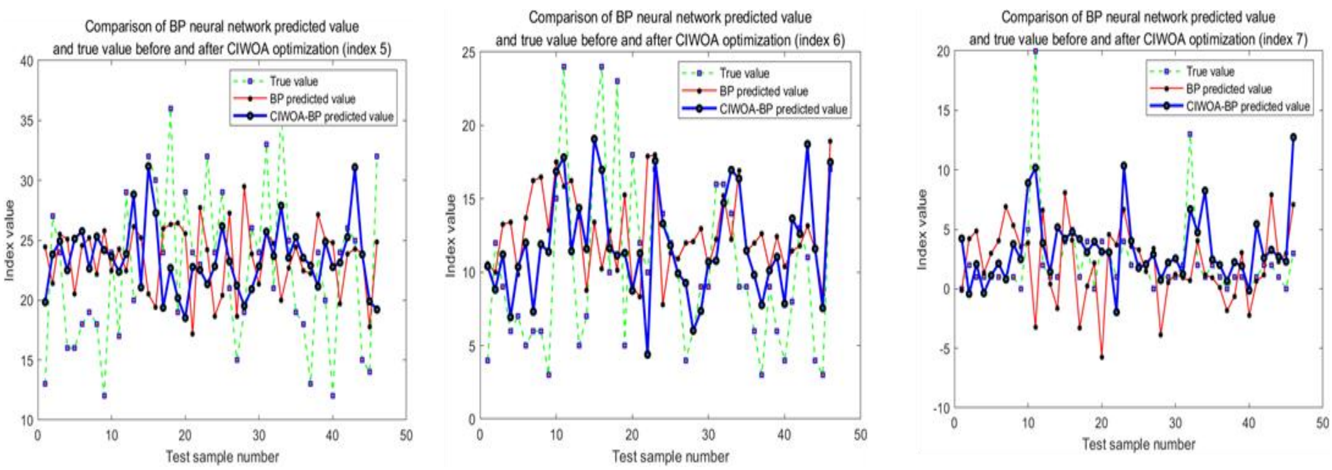


Figure2: Comparison of test set output curves (5-7 tries)

By using the K-means clustering analysis method, the words in the given data set are classified according to the percentage of tries, then the difficulty coefficient is defined, the classification results are shown in the chart below. In this way, the correspondence between the code of the word and the difficulty coefficient is constructed. With this as an input, the neural network model is established and is used to predict the difficulty coefficient of any given word.

Figure3：Known word classification results

During the course of establishing the above model, we also compare the actual value with the predicted value of the model to calculate various error indicators. Moreover, we test the optimization effect, draw a graph for comparison, and constantly seek the best parameter setting. The final model is tested with the word eerie, and the prediction results are given in our paper.

In addition, we also examined the relationship between different attributes of words such as word frequency, whether there are repeated letters, and the number of difficult patterns and the difficulty coefficient of words. Quantitative conclusions are given through Spearman correlation test, which is shown in the figure below. Finally, comprehensive qualitative and quantitative analysis is carried out for other statistical characteristics of the data set.

Figure4： Correlation statistics between different tries

References

[1] https://wordleonline.co
[2] Hochreiter S , Schmidhuber J . Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.
[3] https://github.com/jaungiers/LSTM-Neural-Network-for-Time-Series-Prediction
[4] https://reference.wolfram.com/language/ref/WordFrequencyData.html
[5] Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm.Advances in engineering software,95, 51-67.