Hindawi
International Journal of Computer Games Technology
Volume 2022, Article ID 1828293, 11 pages
https://doi.org/10.1155/2022/1828293

Research Article

Automatic Quality Assessment of Speech-Driven Synthesized Gestures

Zhiyuan He

The University of Edinburgh, UK

Correspondence should be addressed to Zhiyuan He; zedyuanhe@hotmail.com

Received 7 January 2022; Accepted 11 February 2022; Published 16 March 2022

Academic Editor: Peican Zhu

The automatic synthesis of realistic gestures has the ability to change the fields of animation, avatars, and communication agents. Although speech-driven synthetic gesture generation methods have been proposed and optimized, the evaluation system of synthetic gestures is still lacking. The current evaluation method still needs manual participation, but it is inefficient in the industry of synthetic gestures and has the interference of human factors. So we need a model that can construct an automatic and objective quantitative quality assessment of the synthesized gesture video. We noticed that recurrent neural networks (RNN) have advantages in modeling advanced spatiotemporal feature sequences, which are very suitable for use in the processing of synthetic gesture video data. Therefore, to build an automatic quality assessment system, we propose in our work a model based on Bi-LSTM and make a little adjustment on the attention mechanism in it. Also, the evaluation method is proposed and experiments are designed to prove that the improved model of the algorithm can complete the quantitative evaluation of synthetic gestures. At the same time, in terms of performance, the model has an improvement of about 20% compared to before the algorithm adjustment.

1. Introduction

Speech-driven synthetic gestures play an important role in the modern film industry and 3D virtual games and have a decisive influence on the construction and expression of virtual characters. Studies have shown that a gesture-synchronized animation can effectively attract people's attention [1]. Due to the large demand in the fields of animation, avatars, and communicative agents, a variety of methods for speech synthesized gestures have been proposed, such as the style-controllable speech-driven gesture synthesis [2] and speech driven HMM-based body animation synthesis [3]. These technologies have achieved quickly synthesizing speech-driven gestures, but relevant systems for evaluating the quality of synthesized gestures have not been proposed. In fact, most gesture evaluations are based on gesture recognition systems [2, 4, 5]. These systems only pay attention to the feature similarity of the gestures in evaluation. For example, a letter Y with a deformed bottom may be misjudged as a "victory" gesture, then get a high score, which is not accepted. Evaluation based on several other indicators such as human-likeness, appropriateness, and style-control conditions can avoid misjudgment [2], cause these indicators are more closer to the human's acceptance to the synthesized gestures. But the currently known synthetic gesture evaluations based on these indicators are done manually, this is undoubtedly interfered with huge subjective factors, such as personal preference or observer attention, which makes this evaluation method neither accurate nor objective. At the same time, manually watching synthetic gesture videos and collecting evaluation results will consume a lot of human resources and time. Such low efficiency is unacceptable. Thus, the quality assessment of speech synthesized gestures requires an automatic and objective assessment method. In this paper, we aim to introduce a system based on neural networks to learn data features from synthetic gestures and generate a quantitative score to quantify the synthesis quality of speech synthesized gestures. Here, we will only pay attention to the motion characteristics of some joints of the synthesized gesture data, because they have a larger proportion change of vector coordinates with time, so they are more representative of the

overall motion characteristics of the gesture data. The system is trained on the speech & text synthesized Mo-Cap data set of KTH Royal Institute of Technology and the subjective judgment data set of GENEA challenge [2, 6]: body movement data, speech, text, and a set of scores for different synthesized videos given by 125 participants. We use the synthesized gestures as the input of our system, and the output is a quantitative score of the synthesized quality.

We treat synthesized gesture evaluation as a nonlinear regression task and plan to implement a network based on the combination of long short-term memory (LSTM) and linear layers. The LSTM network is selected to extract features from the motion data and encode them into high-dimensional vectors. The linear layer processes the output from the LSTM and then gradually generates prediction scores. Based on the baseline model, we will upgrade the model by adding network layers and improving calculation algorithms. Through comparative evaluation, we hope to explore the performance of the model in evaluating synthetic gestures and select the best evaluation model.

In addition, we will also study the impact of using different regularization and data enhancement methods on model prediction accuracy. After completing the parameter tuning and model optimization, the final model performance will be evaluated. The next section will introduce the related work and basic knowledge; the third section will introduce the experimental methods, including data processing, system structure, and evaluation method. Section 4 will concentrate on the experiments on the baseline and improved models, as well as model optimization and evaluation. Section 5 will summarize the work and make some comments on future work.

The experimental codes of the projects introduced in this article are uploaded to OneDrive: https://1drv.ms/u/s!Ahud_qK07whbkxbxlqwXVeZtL7Q5?e=eyhxhz.

In summary, the main contributions of this paper are as follows:

(i) Propose new evaluation indicator to assess the system

(ii) Propose a single LSTM layer model as baseline system to evaluate the synthesized gestures and improve the model structurally by adding a attention layer

(iii) A novel and simple attention enhancement algorithm is proposed, and the evaluation results of the models are compared

2. Related Works

2.1. Current Assessment. Evaluation on indicators such as human-likeness, appropriateness, and style-control conditions mentioned above is proposed by Simon Alexanderson [2], which is based on the extensive subjective assessment of the MoGlow-based system baseline [2]. Here, the human-likeness, appropriateness, and style-control conditions were evaluated in the different user studies with the full-body synthesis, which will enable a meaningful cross-comparison.

All perceptual studies of the quality assessment were carried out using online experiments in the Figure Eight crowdworker platform, only the most accurate contributors are allowed to provide subjective evaluation data, and the country/region origin is set to English-speaking countries (United States, Canada, United Kingdom, Ireland, Australia, and New Zealand) in order to minimize the subjective interference caused by language errors. In all experiments, the evaluator was asked to watch and listen to the 18-second video clips of the speech synthesis gesture generated by 8 different gesture synthesis systems and to evaluate them on a 5-point scale according to the given scoring standard [2]. The evaluation criteria include human body similarity, appropriateness, style control human body similarity, and body posture similarity [2]. The evaluation results are comprehensively considered, in the calculation, and different criteria are given different weights, and the final score is calculated to find the best synthesis system. This score is the corresponding quantitative score of synthetic gestures after manual evaluation.

2.2. Long Short-Term Memory Model. Long short-term memory (LSTM) is a special kind of RNN that adds network storage to solve the problem that RNN is difficult to learn and store information for a long time [7–9]. LSTM is an excellent variant of RNN that inherits most of the characteristics of the RNN model and at the same time solves the gradient backpropagation process due to the gradual reduction, vanishing gradient problem that arises. It is especially useful for many tasks that require "long-term memory". Thus, it is very useful in dealing with the time series motion data of our project.

As mentioned earlier, we treat the project as a regression task, and that the output is a single-dimensional quantitative score (0 to 100). Therefore, we need to build a model that can process the input which is multidimensional time-series data and output a corresponding single-dimensional scalar. Due to the need to perform feature learning on time series data, LSTM is chosen to encode the data into high-dimensional vectors [10]. However, the output of the LSTM network is not a single scalar, but a multidimensional vector, and the dimension is determined by the number of LSTM hidden layers. Therefore, we consider adding a linear layer to the output of the LSTM network to convert the vector into a single scalar. The overall architecture of the baseline model can be shown in Figure 1(a).

2.3. Attention. The attention algorithm is very similar to the logic of human viewing pictures. When we look at a picture, we do not see all the content of the picture but focus on the focus of the picture [10–13]. In the calculation of long sequence data, the attention algorithm focuses limited attention on key information, thereby saving resources and quickly obtaining the most effective information. Even if the text is relatively long, the key points can be grasped from the middle without losing important information. Compared with CNN and RNN, the model complexity is smaller and the parameters are also fewer. Therefore, the requirement for computing power is even smaller [10–13]. In a

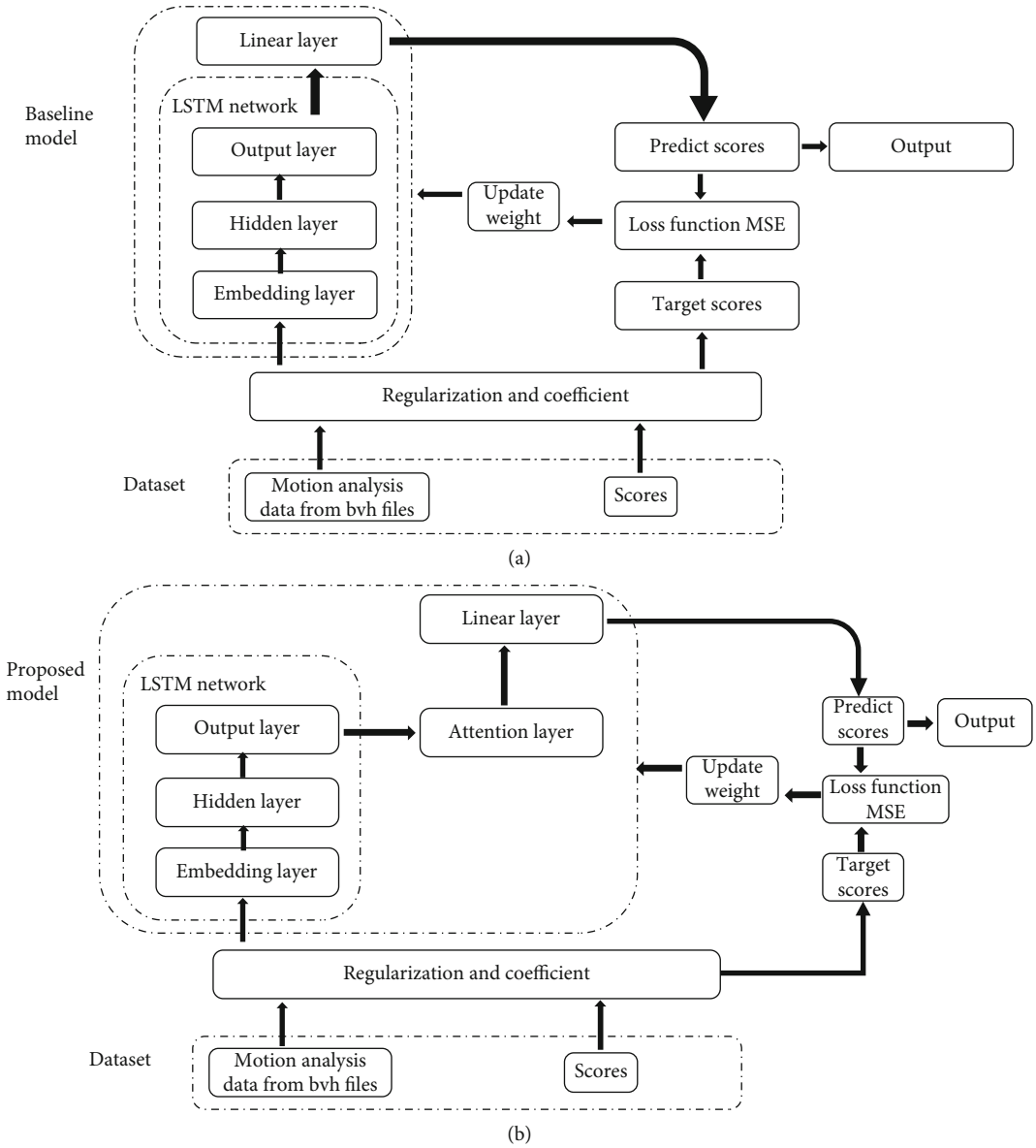

Figure 1: (a) The architecture of the baseline model. (b) The architecture of the proposed model.

typical encoder to decoder model, the attention algorithm is actually a series of attention distribution coefficients, that is, a series of weight parameters. Therefore, it is very useful when assigning weights to key motion data in a project.

3. Method

3.1. Data Preprocessing

3.1.1. Data Overview. The system will be established based on MoCap data set of KTH Royal Institute of Technology and the subjective judgment data set of GENEA challenge [2, 6] which is used as data and labels, respectively. The MoCap data was obtained by analyzing the generated synthetic gesture video, which are motions of full body and head, generated from 8 different gesture synthesis systems. The sampling rates are respectively 60 Hz for grand truth and 20 Hz for synthetic gestures. Labels contain 10000 scores given by 125 participants for 80 video clips of synthesize gestures based on two criteria: appropriateness and anthropomorphism, corresponding video segments, and participant information, etc. These two types of data will be recorded as the objective data and subjective data which require extra works of preprocessing and dataset making. The specific data processing and analysis will be discussed in this chapter.

3.1.2. Objective Data. The objective data are the files generated after human movement is captured of the synthesized gesture video in the form of .bvh. This is a typed dataset that contains the motion feature data of 57 human motion joints (from head to toe), that is, the three-dimensional position and the three-dimensional rotation angle in different frames. All data types are three-dimensional floating point coordinates [14, 15]. Use the built-in function to extract the data

contained in the file. Observing the position and rotation coordinates, it can be found that the coordinates of some joints have not changed over time (or changed very slightly). So, first use the filtering method to extract the features. We calculate the standard deviation of the movement for all joints over time. We found that the maximum change in coordinates of all data whose average movement standard deviation is less than 0.01 over time is less than 10%. By setting the std threshold as 0.01, that is, deleting all joints whose motion standard deviation is less than 0.01, 174 joints containing effective information can be obtained. These joint data are exactly all the joints of the upper body of the human body in the synthesized gesture video.

After cutting out the "unimportant" joints, the data needs to be divided according to time. This is because the subjective score is not to score the complete synthetic gesture video but to score the video segment. Therefore, it is necessary to divide the objective data according to the timing segmentation of the video in the GENEA data [6], that is, to divide the data according to the frames, so that the 10 synthetic gesture data files generated by each of the eight systems will be divide into 320 segments, which will be used as the input sample.

3.1.3. Subjective Data. The subjective data contains the information of the participants, scored video segments, and the corresponding score, including the interaction log. The information needed in our system is the score corresponding to the video segments. To analyze the data, we introduce the matlab-based subjective data reading method [6] to gather the dataset of scores as the label of our input sample. Due to the lack of text and speech materials for synthesizing gestures, we only consider the evaluation scores for human-likeness of every synthesized gestures.

The figure that plots the human-likeness scores to every gesture synthesis systems ais shown in Figure 2. It can be observed that the score for each system is distributed in an interval, the distribution of the scores is all larger than 0.995 in calculation, and most scores are distributed near the average score. Thus, we just consider excluding the outliers and use the remaining scores as labels. After the same step of dividing the subjective data according to the timing segmentation of the video in the GENEA data [6], we got a total of 8375 scores on 320 different files, each with a number of scores ranging from 17 to 34.

While the scores are used as the label for model training, it cannot be in the form of an array. Thus, a special value is required to replace the overall scores. In order to verify whether the mean value can be used to represent the overall scoring situation, we continue to analyze the distribution of scores. The figure that plots the standard deviation of scores given to all video segments is shown in figure 3. It can be seen from the figure that the variance of the scores given by the participants to any video segment is around 20 (±5), and the mean value is 21.398, which means that the degree of dispersion of the scores set is relatively high. Therefore, any single score or median value in the score set of any video segment cannot represent the overall score set given by the participants. In order to obtain a single quantitative score to represent the synthesis quality of video segments, we need to clarify the data distribution to find a suitable alternative scalar. We perform the Kolmogorov-Smirnov (KS) test on the set of scores corresponding to all video segments [6] to check which distribution that the data conforms to. Here, we set the distribution model as the normal distribution and calculate the mean and standard deviation of each score set. By setting the P value threshold of 0.05 ($P > 0.05$ corresponds to the normal distribution), the KS test is performed [15, 16]. The test result shows that, except for the score set of the second segment of the ninth gesture video file given by the grand truth, the rest of the score sets conform to the normal distribution. Therefore, the average

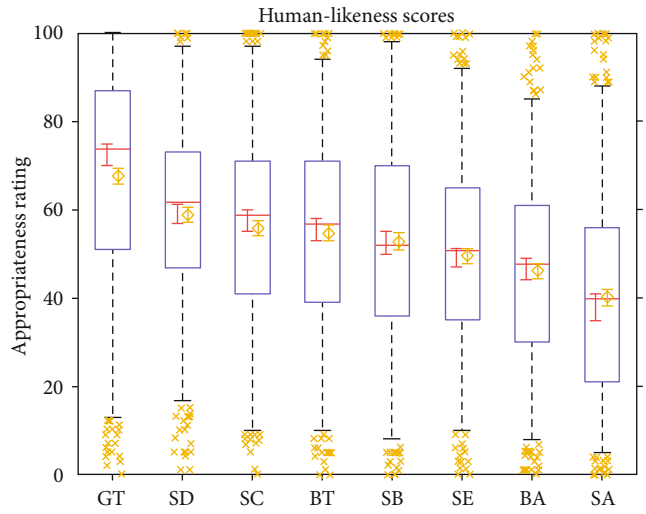

Figure 2: Human-likeness scores to every gesture synthesis systems. Here, the red line segment is the confidence intervals, Red line is the median value, yellow x represents outiler symbol, and yellow line in the rectangle is the mean value, and SA to SE are 5 different generation system, GT represents grand truth, BA and BT represent system with only audio and system with only text.

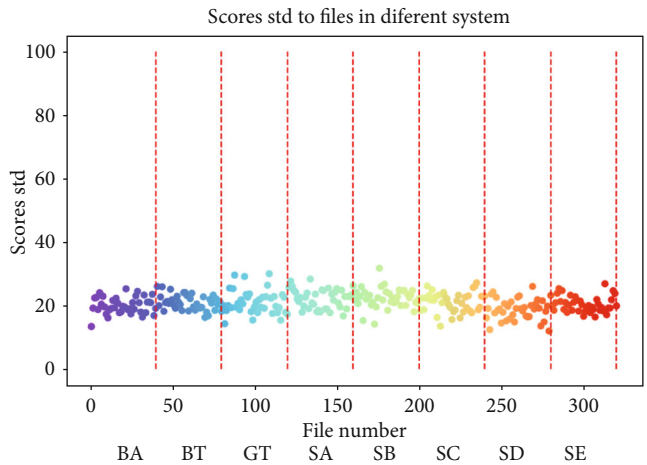

Figure 3: Standard deviation of scores given to all video segments of 8 synthetic system different color represents the different files, and the subscript of the area separated by the red dotted line represents which system the file comes from.

value of the score set can be used to represent the overall score set. Extract the average value of each score set, a total of 320 quantitative scores to form a new dataset, this dataset will be used as the label of the input sample.

Except being used as the label, the subjective data is also used to evaluate the accuracy of human evaluation. Here, we define a value *Em* to represent the average of the mean error between the artificial evaluation score and mean score, and a value *Es* to represent the standard deviation of the mean error between the artificial evaluation score and mean score. The calculation of *Em* is shown as Equation (1).

$$E_m = \frac{\sum_{i=0}^{n}\left(\sum_{j=0}^{m}\left(X_{ij} - \bar{X}_i\right)/m\right)}{n}. \tag{1}$$

The calculation of *Es* is shown as Equation (2).

$$E_s = \sqrt{\frac{\sum_{i=0}^{n}\left(\sum_{j=0}^{m}\left(X_{ij} - \bar{X}_i\right)/m - E_m\right)^2}{n}}. \tag{2}$$

Among this, Xij represents the value of artificial evaluation scores given to every video segments and $\bar{X}i$ is the mean value to every video segments, and n is the number of video segments and m is the scores given to every segments. Calculate these two values separately, *Em* is 8.739 and *Es* is 7.614. This will be used to evaluate the prediction level of the model by comparing with the mean value and standard deviation of model prediction error.

3.2. Structure Improved: Bi-LSTM(Attention). Considering the characteristics of learning multidimensional tensor form data, model learning can determine the accuracy of the system's prediction, and a single LSTM with a linear regression layer may not have good regression performance. In order to improve model learning ability, we introduce a structure improvement on the base-line model, which is based on a bidirectional LSTM network and a linear regression layer, adding an attention layer [11–13, 17, 18] between the two layers, and the architecture of the proposed model based on Bi-LSTM, RNN, and linear layers can be shown in Figure 1 (b).

In our model, Bi-LSTM needs to learn the joint coordinates in all previous frames. Therefore, the cell state may contain some key motion joint coordinates of the current frame for correct weighting. When inputting to a new frame, the repeated unit of Bi-LSTM forgets the useless joint coordinate information of the old frame. The key motion joint coordinates of the new frame are added to the cell state to replace the useless joint coordinate information it forgot. The output dimension of each repeated unit in each direction is [*xi*, *mi*, and *hn*], where the *hn* is the number of the hidden layer and *xi* is the number of input sample X. For every single sample, the output processed by Bi-LSTM in every direction has the dimension of [1, *mi*, and *hn*]. So, the final output combining two direction will be the size of [1, *mi*, and 2*hn*]. Due to the characteristics of LSTM, the output of the last unit refers to the key joint motion of the data in every previous units (frames). So, we just take the output of the last unit as the input of the linear layer, with dimension [*xi*, 1, 2*hn*], and for every single sample, the dimension we need is [1, 1, 2*hn*] that has two times of features than the output of single direction LSTM which has the size of [1, 1, *hn*]. Therefore, the output of the Bi-LSTM layer will be a vector with the size of [batch size, n step, n hidden∗ num of directions (=2)], which will be input into the attention layer. In the attention layer, the output from LSTM will first perform the addition of corresponding elements. After being transformed by the fully connected layer, the hidden layer vector will be used to calculate the similarity with the neighboring vector and converted into the importance of each batch data through softmax. The transformation steps are shown in Equation (3)

$$\alpha = softmax\left(w^T M\right). \tag{3}$$

Among the equation, w^T represents the parameters that need to be learned and M represents the hidden layer vector. α is used as the global weighted summation of hidden layer vector to obtain the vector that expresses the motion. After the iteration, this will be used by us as the evaluation score.

3.2.1. Algorithm Adjustment in Improved Model. When building the model, we mentioned the data importance calculation of the attention layer. In the calculation, the learning parameters and the hidden layer vector are directly matrix multiplied. This will inevitably cause some low importance weights to be increased in the calculation. In order to reduce the impact of this situation, we consider optimizing the algorithm. In the two-dimensional convolution of the image, the convolution of the input image and the edge detector convolution kernel can highlight the contour of the image [19]. We apply this idea to the calculation of data importance and propose the alternative calculation, Equation (4):

$$\alpha = softmax\left(\left(w^T * G1\right) \cdot \left(M * G2\right)\right). \tag{4}$$

Among them, $*$ represents convolution and $G1$ and $G2$ represent two different edge detector convolution kernels, respectively, The size of the convolution kernel is determined by the size of the matrix of w^T and M. After convolution filtering, the two two-dimensional matrices w^T and M are filtered out of weights other than the important weights. The result of the dot product is more prominent for important weights, while the impact of nonessential parts is minimized.

3.2.2. Superiority. Obviously, Bi-LSTM has one more backward layer than LSTM, that is, the input is also calculated from the reverse order. It can be seen that the forward layer and the backward layer are connected to the output layer, which contains 6 shared weights. In the forward layer, the forward calculation is performed from time 1 to time t, and the output of the forward hidden layer at each time is obtained and saved. In the backward layer, the backward calculation is performed from time t to time 1, and the output

of the backward hidden layer at each time is obtained and saved. Finally, at each moment, combine the output results of the forward layer and the backward layer at the corresponding time to obtain the final output [17, 18]. This shows that the data characteristics learned by Bi-LSTM are twice that of LSTM. After the output is folded, the weight of the same feature is twice the original, so the important information of the data can be more prominent.

The added attention layer is a weight parameter allocation mechanism, and the goal is to assist the model to capture important information. Given a set of <*key, value*> and a target vector query, the attention mechanism is to obtain the weight coefficient of each key by calculating the similarity between the query and each set of keys and then obtain the final attention by weighting and summing the value. This attention will be used to determine the data segment that the network focuses on learning.

The algorithm adjustment in the attention layer makes the weight coefficient distribution of important keys larger. Therefore, the attention to important joint motion features in gestures will be greater, which undoubtedly strengthens the model's learning of important features.

Theoretically, the proposed model has better learning ability and predicts and evaluates scores more accurately.

3.3. Evaluation. Because we treat this project as a sequence regression task, we choose mean square error as the evaluation metric to measure the similarity between the output and original scores [20]. MSE is the most direct measurement to calculate the difference between two sequences. The smaller MSE means two sequences are closer. However, this requires strict alignment of the two sequences. If the prediction is lagging or leading to the target, even if their scores are the same, it will cause large errors.

For this reason, we introduce our evaluation metric, which is to calculate the errors between the original scores and the corresponding predict score and find out the standard deviation and the maximum value of the errors. The model with the smaller maximum error value and standard deviation has the better predicted performance. We record this metric as the prediction evaluation.

Since the output scores and the actual scores are both a one-dimensional array, the Pearson correlation coefficient [21] and significance level P value [22] can also be introduced as the metrics to evaluate the performance of model's prediction. The Pearson's correlation coefficient is widely used to measure the degree of correlation between two variables. The larger its value, the more significant the correlation between two variables.

To discuss whether the two variables are related, the significance level must be discussed. It is meaningless to talk about the correlation coefficient regardless of the P value. The correlation between the two may only be caused by accidental factors, so we need to analyze the correlation between the two variables. The significance level of the relationship is judged. The P value is another basis for making inspection decisions. It is the probability which reflects the probability of occurrence of an event. Statistics according to the P value obtained by the significance test method, generally $P < 0.05$ is significant, $P < 0.01$ is very significant, which means that the probability of the difference between samples due to sampling error is less than 0.05 or 0.01.

Theoretically, the closer the predicted scores of the model are to the actual scores, the larger the Pearson correlation coefficient of the two sets of scores and the smaller the P value of the two sets of scores. Therefore, these two can also be used as a metric to evaluate the performance of the model. After the model is trained, the model will be evaluated with reference to the four evaluation metrics.

Table 1: Hyperparameter in model.

Regularization	Iteration	Hidden size	LR
L2	200	128	0.001

4. Experiments

4.1. Baseline LSTM Model. The implementation of the baseline model follows the architecture in Figure 1(a). Here, dataset of position was selected as the input sample X of the training model. To prevent the model from failing to learn the characteristics of all system synthetic data, when building the model, the 320 input sample X needs to be divided into eight data stacks according to the gesture generation system, and the eight data stacks are randomly divided into training set, validation set, and test set at a same ratio of 6 : 2 : 2; here, the random state was set to 1. Gather all training set data into a data set with a total of 192 samples as the input to train the model. By default, the hyperparameter set is shown in Table 1: The loss on training and validation sets of each epoch is recorded, being drawn in Figure 4(a).

By viewing the relationship between the loss values of the training set and the validation set with the epoch, we can find the larger the epoch, the greater the fluctuation of the loss value instead of convergence. The phenomenon may be caused by a high learning rate. The optimal number of iterations for the model without overfitting can be obtained in this figure, and at this stage, the lowest verification loss is 0.918, which is located at 59^{th} epoch that means the model reaches its best generalization performance. So, we take the network at this epoch as our best baseline model for subsequent evaluation. The training loss at this epoch is 0.888, which is at the same level as the validation set. Input the test set data into the model, we get the other evaluation result as shown in Table 2, first row. The comparison between the predicted score and the actual score is shown in Figure 4(b). Obviously, in the result of evaluation, the test loss of the baseline model reached 0.694, which is in the same order of magnitude as the training loss and verification loss and is slightly smaller, indicating that the baseline model has good generalization. The Pearson correlation coefficient is between 0.4-0.6, and the small P value indicates that the predicted score of the model has a middle correlation with the actual score. The standard deviation of the prediction error is 5.128, and the average error is 6.569. It is a little less than the value in manual evaluation which are 7.614 and 8.739, respectively. The variance of 18.717 is also

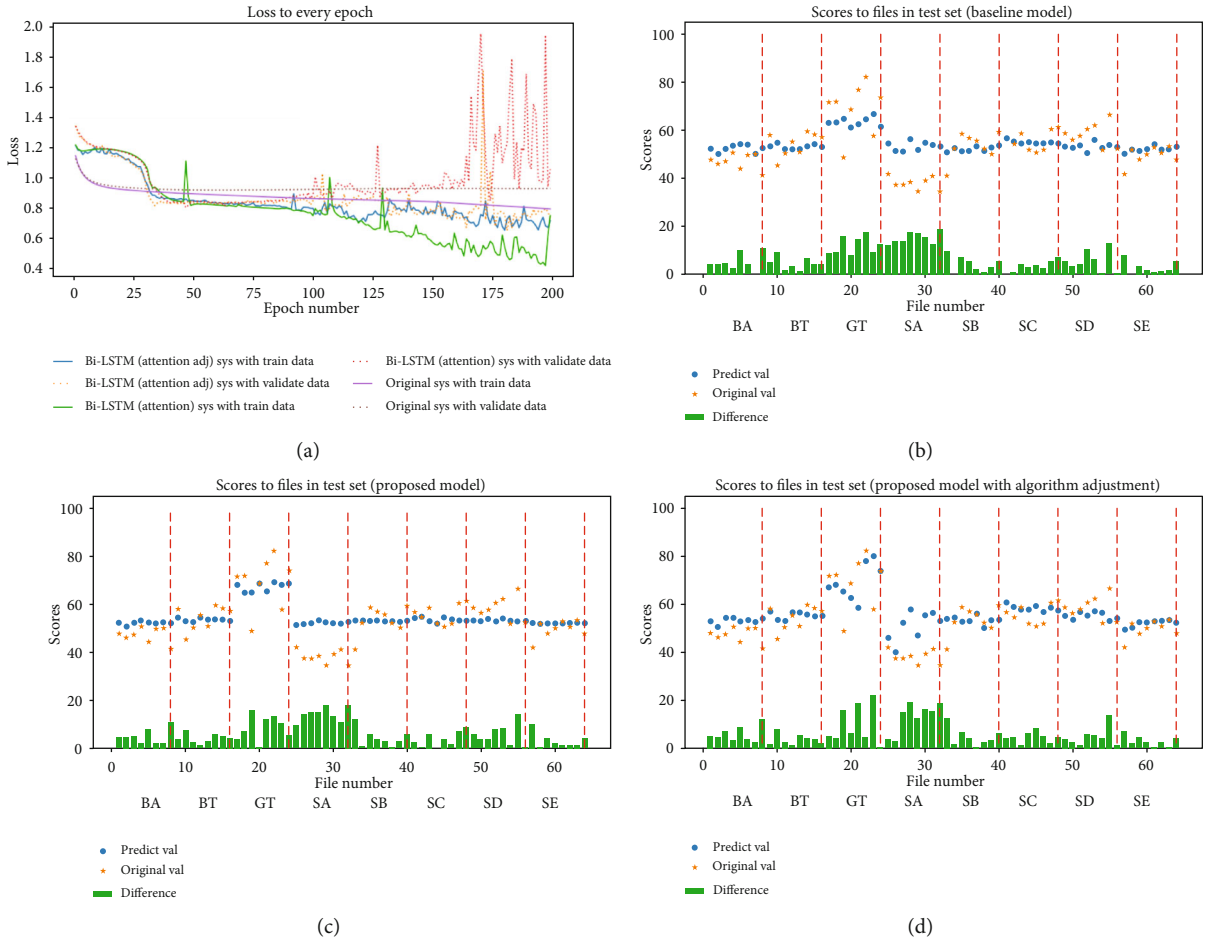

Figure 4: (a) Loss curve to every epoch, Bi-LSTM(attention adj) represent the proposed model with algorithm adjustment in attention layer. (b)Predict scores/original scores/errors to each files in test set(baseline).(c)Predict scores/original scores/errors to each files in test set(proposed). (d)Predict scores/original scores/errors to each files in test set(algorithm adjustment proposed).

Table 2: Evaluation of 3 models.

Model	MSEtest	PCC	P value	Error max	Error mean	Error std
Baseline	0.694	0.575	$6.864 * 10^{-7}$	18.717	6.569	5.128
Structure improved	0.592	0.659	$2.999 * 10^{-9}$	18.019	6.017	4.801
Algorithm adjustment	0.604	0.642	$1.012 * 10^{-8}$	20.953	5.975	5.266

less than that in manual evaluation. We believe that the overall performance of the model prediction has reached a level similar to the artificial evaluation, but the average error is still large. This means that the prediction score is not accurate enough. It can be seen from the prediction result figure that, except for the prediction score of grand truth, the other prediction scores all fall in the range of 50-60, and the prediction scores of video segments in different systems are close. This prediction trend is obviously inconsistent with the actual situation. That means the model needs to be improved. Therefore, the proposed model was established for evaluation.

4.2. Improved Model. The implementation of the proposed model follows the architecture in Figure 1(b). We use the same dataset to train the model and use the same model parameters in the subsection 4.1 for setting. It should be noted that the Bi-LSTM needs to be activated here, that is, set bidirectional = True. The loss on training and validation sets of each epoch is recorded, being drawn in Figure 4(a).

It can be observed that the MSE loss of the model on train set in the first 25 epoch iterations is slightly higher than the baseline, and the loss after multiple iterations is significantly lower than the baseline. The validation loss has the similar trend compared with the baseline. The validation

Table 3: Evaluation of 3 models under L1 regularization.

Model	Regularization	MSEtest	PCC	P value	Error max	Error mean	Error std
Model	L1	0.755	0.526	$8.183 * 10^{-6}$	20.470	6.846	5.351
Structure improved	L1	1.009	0.342	0.006	27.017	7.805	6.324
Algorithm adjustment	L1	0.677	0.591	$2.705 * 10^{-7}$	17.684	6.847	4.596

loss that the best iteration of the proposed model is 47^{th} epoch. At this epoch, both the train loss and the validation loss of the proposed model are smaller than the corresponding value of the baseline model. The other evaluation result of proposed model is shown in Table 2, second row.

Obviously, the MSE on test set 0.592 is smaller than the baseline, and the larger PCC 0.659 and smaller P value $2.999 * 10^{-9}$ indicate that the score predicted by the proposed model is more correlated with the actual score. Moreover, lower average error 6.017 and error standard deviation 4.801 also indicate better prediction accuracy. This confirms part of our idea, that is, using Bi-LSTM to replace the LSTM layer and adding the attention layer can indeed make the model have better learning capabilities. The weight multiplication and attention of important motion data enable the model to evaluate synthetic gestures more accurately.

But from Figure 4(c), we found the trend of the prediction scores almost be a straight line in SA to SE system, this is very strange, we know that the video segments of these different systems are different, and it is obviously undesirable for the proposed model to predict almost the same scores for these segments. This phenomenon is due to relatively similar synthetic gestures in these video segments, and the model assigns these features data to the same weight, thus causing a misjudgment.

In theory, our algorithm adjustment can make the proposed model avoid this mistake in weight assignment as much as possible. So we use the same parameter in section 4.1 to test the proposed model with algorithm adjustment. The figure of loss to epoch is also in Figure 4(a). It can be observed that the MSE loss of the model on train set and validation set has the similar trend to the original proposed model. But we can find the best epoch of this model is at 181^{th} epoch, much larger than that in the baseline and original proposed model. But the MSE on every set is lower than original baseline. At this epoch, the other evaluation result of proposed model is shown in Table 2, third row. It can be observed that, except for the average error 5.975, which is better, the other evaluations are not as good as the second model. Of course, all indicators are better than the baseline. But this does not mean that the algorithm adjustment is not effective. Combined with Figure 4(d), the trend of the prediction value is much closer to the actual value. This shows that the adjustment of the algorithm can indeed enable the model to learn more about the important motion features in the video segment. In the calculation, the weight assignment to the important target is more in place, although the cost is greater error dispersion, resulting in larger local prediction errors, but the average error will be reduced as much as possible. Therefore, we can believe that the algorithm adjustment is acceptable and can partially improve the prediction performance of the model.

4.3. Model Optimization

4.3.1. Data Regularization. Regularization is a method to improve model generalization to prevent overfitting [23, 24]. This experiment involves two regularization methods, L1 and L2.

The main idea of L2 regularization is to reduce parameter values, thereby reducing variance. This technique introduces an additional penalty term in the original loss function (L) and increases the sum of the square parameter (ω). But this penalty term may be too large that the network will try to minimize the loss function by making its parameters very close to zero, which is inconvenient. This is why we multiply this number by a small constant (λ), the value of which can be chosen arbitrarily. The loss function formula is Equation (5).

$$L_2(X, \omega) = L(X, \omega) + \lambda \sum \omega_i^2. \tag{5}$$

For L1 regularization, the network parameter in the penalty term is not squared, but its absolute value is used instead, the loss function formula is the Equation (6).

$$L_1(X, \omega) = L(X, \omega) + \lambda \sum |\omega_i|. \tag{6}$$

To test if the regularization will change the performance of the model, we just change the way of regularization, and other parameters is same to Section 4.1. Here is the evaluation to every model at L1 regularization (at their best epoch), shown in Table 3. It can be seen from the table that when using L1 regularization, the evaluation of the three models almost all got a lower level. It is worth noting that the evaluation of the original proposed model under L1 regularization is even worse than the baseline model, and it can hardly predict the score normally. We think this is because L1 regularization has the same penalty for all parameters, which can make some weights become zero, causing some motion feature data to not be learned by the model. For the model with algorithm adjustment, although the max value and variance of error become smaller, the average error increases, which will lead to nearly the same prediction scores for the different video segment of different systems mentioned above.

Therefore, in order to make the model have better prediction performance, the L2 regularization will be used when training the model.

Table 4: Evaluation of 3 models with data augmentation.

Model	Data augmentation	MSEtest	PCC	P value	Error max	Error mean	Error std
Baseline	Padding with last frame	0.622	0.580	$3.476 * 10^{-7}$	18.211	6.676	5.001
Structure improved	Padding with last frame	0.574	0.661	$9.857 * 10^{-8}$	17.846	5.965	4.228
Algorithm adjustment	Padding with last frame	0.591	0.657	$1.491 * 10^{-8}$	20.330	5.745	4.911
Baseline	Padding with zero	0.686	0.499	$3.412 * 10^{-6}$	20.218	7.019	5.976
Structure improved	Padding with zero	0.617	0.567	$1.61 * 10^{-6}$	19.130	6.333	5.494
Algorithm adjustment	Padding with zero	0.673	0.544	$8.758 * 10^{-7}$	23.653	6.131	5.765

4.3.2. Data Augmentation. Our optimization goal is to pursue the best point of our model with lower loss, which happens when the parameters are adjusted in the correct way. Obviously, if there are many parameters, you need to give the model a sufficient proportion of samples during training. Similarly, the number of required parameters is proportional to the complexity of the task. For the multifeature motion analysis data of this project, the directly trained model may be inaccurate. Therefore, it is necessary to introduce data enhancement methods to avoid model errors caused by insufficient data [25, 26]. The main method of data augmentation compared in this project is data padding, and the motion analysis data is cropped to 350 frames in the same time length which is the average number of frames. For data with less than 350 frames, frame supplementation is required. The two methods of supplementary frame are mainly compared. The first is to fill the supplementary frame with 0 data, and the second is to repeat the last frame of original data as the supplementary frame. Two kinds of data after padding are used to train models which are compared with the model trained by original data. The training data dimension after padding becomes [192, 350, and 174], while the original one is [192, fn, and 174], where fn is the frame number of each motion file.

The evaluation results of each model at their best epoch is shown as Table 4. It can be observed that compared with the model trained on the original data, the model trained with the data after padding does have a difference in prediction performance. Padding with last frame will result in better evaluation results for the trained model. And padding with zero achieves the opposite effect. Therefore, the best model can be optimized by using padding with last frame as the method of data augmentation.

4.4. Model Evaluation. After the above experiments, it can be observed that the three models have been optimized to achieve good prediction performance. Compared with the standard deviation and the average value of the prediction error, these three models have reached the similar level of manual evaluation.

The method to optimize the model is padding the data with its last frame and doing L2 regularization on the data. Three models all have better prediction performance on the test dataset after optimization.

The MSE of proposed model with algorithm adjustment after optimization on the test set is 0.591, the maximum prediction error of the test data set is 20.330, and the standard deviation is 4.911. The predicted score is highly correlated with the actual score, which is reflected in the Pearson correlation coefficient close to 0.7 and a very small P value.

Comparing the evaluation of the three models, using the same test set and model hyperparameter settings, it can be observed that maximum prediction errors of the baseline and original proposed model for the test data set are all around 18, which is very close to the error of human evaluation. But the average prediction error of proposed model with algorithm adjustment is smaller. The prediction trend is also more closer to the actual situation when using our algorithm adjustment. The overall improvement in performance is about 20% comparing with the base-line model.

The maximum prediction error of the proposed model with algorithm adjustment appears in the evaluation of grand truth gestures. We think this is because the gesture movement in the grand truth is much smoother without much obvious transition. This makes it difficult for the model to learn the feature of key motion data. The model's prediction scores for video segments of several other systems have tended to be similar with the actual scores. Therefore, it can be considered that the proposed model with algorithm adjustment can predict scores more accurately.

5. Conclusions

To build an automatic quality assessment system, we established baseline model with a single LSTM and linear layer and make structure and algorithm adjustment on it. Experiments have proved that the baseline can achieve the goal of predicting scores, but there is still space for improvement in performance, and the adjustment on model does reduce the prediction error.

For the Bi-LSTM(attention) model with algorithm adjustment we proposed, the moderate learning rate, data regularization, and correct type of data can help the model converge to a better local optimal faster while avoiding premature overfitting.

From the perspective of the network structure, this model has larger hidden dimensions and more model parameters than the baseline, which can make the prediction performance of the model better. At the same time, the model also has a good generalization ability, which is reflected in the model's test MSE lower than the model's validation and training MSE.

From an algorithm, this model has more reasonable weight distribution in the calculation process when learning important motion features comparing to the original Bi-LSTM(attention) model. This is reflected in the prediction results more similar to manual evaluation except on grand truth.

What is certain is that the algorithm adjustment we proposed does to some extent solve the problem of insufficient learning of key motion data features by the Bi-LSTM(attention) model, making the prediction trend much closer to reality, and the average prediction error is smaller, but at the same time, it also caused some negative effects to increase the prediction error for some video segments, especially, the model's scoring error for video segments of grand truth is relatively large.

However, our proposed model still has some problems. First of all, although the error variance of the prediction score is close to the manual evaluation, the model's evaluation error of the grand truth is significantly higher than that of other system synthetic gestures, which is unnatural. We guess that because grand truth gestures are more complex than synthetic gestures, each joint data in the action analysis data changes more frequently, resulting in that with the same number of iterations, and the parameters given by the model do not fully fit the data characteristics. This leads to higher error in the evaluation of grand truth gestures compared to synthetic gestures of other systems.

Secondly, due to the lack of text and speech materials for synthetic gestures, we only evaluate the human-likeness characteristics of synthetic gestures. However, manual evaluation will consider the relationship between synthetic gestures and text/speech materials, that is, the evaluation is based on appropriateness. Therefore, the system lacks the ability to automatically evaluate the appropriateness of synthetic gestures.

Also, compared with the baseline model, although the proposed model effectively improves the generalization and prediction accuracy on the test set, the computational complexity in the neural network becomes higher and the calculation time is greatly increased. This greatly limits the efficiency of the system in evaluating synthetic gestures.

Although only part of the goal has been achieved, we can still say that the model meets the established requirements and achieves the similar evaluation performance to manual evaluation, and even the error interval of the predicted score is smaller than that of manual evaluation.

5.1. Future Work. When training the model, we only consider evaluating the human-likeness of synthetic gestures. However, this is not the only criterion to evaluate the quality of synthetic gestures. Criteria such as appropriateness and style-control conditions are also used in manual evaluation. In future work, in order to make model evaluation as much as possible to achieve the same performance of manual evaluation, the model needs to take other centralized evaluation criterion into consideration, which requires a redesign of the program to ensure that the model can learn the data characteristics of other evaluation criterion.

Even if it has got a good evaluation performance on a single criterion of human-likeness, there are still many spaces for improvement in the model. From the perspective of learning ability, the use of CNN + Attention in model theoretically will make the model have better learning ability than single attention layer mentioned in the article. However, the experience of CNN is partial, and the viewing zone needs to be enlarged through the eye. However, the convolutional perception field of CNN is partial, and it is necessary to expand the field of view by superimposing the multilayer convolution area. Therefore, the model needs to be further optimized to achieve a balance between higher learning ability and better learning efficiency.

Regarding the model training results, this article has not yet solved the model's abnormal prediction error of grand truth gestures. If the conjecture is true, we need to redesign the network structure of the model, because increasing the number of iterations on this model will cause overfitting to the synthetic gesture data of other systems, which resulting in increased prediction errors. This needs to be verified by subsequent experiments.

Data Availability

The synthetic gesture analysis data and label data used to support the findings of this study are included within the article.

Conflicts of Interest

The author(s) declare(s) that they have no conflicts of interest.

References

[1] Y. Yuanzhi, *Implementation of a voice-driven 3D pronunciation action synthesis system*, Northwest University for Nationalities, 2021.

[2] S. Alexanderson, G. E. Henter, T. Kucherenko, and J. Beskow, "Style-Controllable Speech-Driven Gesture Synthesis Using Normalising Flows," *Computer Graphics Forum*, vol. 39, no. 2, pp. 487–496, 2020.

[3] L. Wang, L. S. Ma, and F. Kao-Ping, "Speech and text driven HMM-based body animation synthesis," US Patent 8-224652, 2012.

[4] D. Avola, L. Cinque, M. De Marsico, A. Fagioli, and G. L. Foresti, "LieToMe: Preliminary study on hand gestures for deception detection via Fisher-LSTM," *Pattern Recognition Letters*, vol. 138, pp. 455–461, 2020.

[5] G. Li, H. Tang, Y. Sun et al., "Hand gesture recognition based on convolution neural network," *Cluster Computing*, vol. 22, pp. 2719–2729, 2019.

[6] T. Kucherenko, P. Jonell, Y. Yoon, P. Wolfert, and G. E. Henter, "A large, crowdsourced evaluation of gesture generation systems on common data: The GENEA Challenge 2020," in *26th International Conference on Intelligent User Interfaces*, pp. 11–21, New York, NY, United States, 2021.

[7] L. L. P. Haoyang, "Real-time recognition of multi-dimensional feature gestures based on LSTM," *Computer science*, vol. 8, pp. 328–333, 2021.

[8] F. Wong, *Research progress and application of LSTM recurrent neural network, [M.S. thesis]*, Heilongjiang University, 2021.

[9] J. Mengmeng, X. Jinzhuang, and L. Ruipeng, "Gesture recognition algorithm of CNN combined with BI-LSTM hybrid model," *Laser Journal*, vol. 42, no. 6, pp. 88–91, 2021.

[10] G. Piaoyi et al., "Short-term power load forecasting method based on Attention-BiLSTM-LSTM neural network," *Computer Applications*, vol. 41, no. S1, pp. 81–86, 2021.

[11] H. Wanwei, L. Song, Z. Chaoqin, W. Sunan, and Z. Xiaohui, "Research on path optimization of service function chain based on graph attention network," *Journal of Electronics and Information*, 2022, http://kns.cnki.net/kcms/detail/11.4494.TN.20211122.1313.006.html.

[12] S. Hao, X. Yuhang, and C. Lian, "Research on facial expression recognition based on multi-scale fusion attention mechanism," *Microelectronics and Computer*, vol. 1-8, pp. 11–22, 2021.

[13] H. Ningsheng and J. Jingyu, "Parametric modeling and body size measurement method based on ConvLstm," *Information Technology*, vol. 9, pp. 89–94+100, 2021.

[14] L. Xiaoyang, *Research and implementation of 3D model animation based on motion capture data, [M.S. thesis]*, Zhengzhou University, 2016.

[15] H. Shumin, L. Yourong, and L. Lin, "A time series nonlinearity test method based on KS test," *Journal of Electronics and Information Technology*, vol. 4, pp. 808–810, 2007.

[16] H. Mei and Y. He, "Programming algorithm for testing normal distribution," *Journal of Changde Normal University (Natural Science Edition)*, vol. 2, pp. 23–25, 2001.

[17] L. Chaoran, X. Fei, F. Yaxiang, Z. Zhenyu, and Y. Guorun, "A simulation modeling method for lithium-ion batteries under high pulse rate conditions based on LSTM-RNN," *Chinese Journal of Electrical Engineering*, vol. 40, no. 9, pp. 3031–3042, 2020.

[18] H. Wang, "Speaker recognition based on deep two-way LSTM network," *Computer Engineering and Design*, vol. 41, no. 6, pp. 1768–1772, 2020.

[19] F. Wang, J. Yu, and Z. Qianshi, "Color edge detection based on adaptive directional derivative filter," *Computer Engineering*, vol. 1-9, pp. 11–23, 2021.

[20] L. Dongfang, Z. Jianjun, F. Haiqiang, and Z. Bing, "Quadratic optimization method of regularization parameters in the sense of mean square error," *Journal of Surveying and Mapping*, vol. 49, no. 4, pp. 443–451, 2020.

[21] Z. Feng, Z. Xie, and J. Cheng, "Combined weighting method based on vector Pearson correlation coefficient," *Firepower and Command Control*, vol. 40, no. 5, pp. 83–86, 2015.

[22] J. Hui and Z. Liling, "Hypothesis Testing and Recognition of P Value," *Environmental and Occupational Medicine*, vol. 34, no. 2, pp. 95–98, 2017.

[23] Z. Qian, M. Deyu, and Z. Xu, "L(1/2) Regularized Logistic Regression," *Pattern Recognition and Artificial Intelligence*, vol. 25, no. 5, pp. 721–728, 2012.

[24] Y. He and L. Baoqi, "A combined deep learning model learning rate strategy," *Acta Automatica*, vol. 42, no. 6, pp. 953–958, 2016.

[25] C. Zhikui, L. Ailing, and Z. Qingchen, "Incomplete data filling algorithm based on attribute importance," *Microelectronics and Computer*, vol. 30, no. 7, pp. 167-172–167-176, 2013.

[26] X. Jianxun, F. Wu, and C. Xie, "Applying data filling to alleviate sparse problems and achieve personalized recommendations," *Computer Engineering and Science*, vol. 35, no. 5, pp. 15–19, 2013.