

## Research Article

# Research on Disease Prediction Method Based on R-Lookahead-LSTM

Hailong Chen , Mei Du , Yingyu Zhang , and Chang Yang

*Department of Computer Science and Technology, Harbin University of Science and Technology, Harbin, Heilongjiang 150000, China*

Correspondence should be addressed to Hailong Chen; [hribustchl@163.com](mailto:hribustchl@163.com)

Received 26 January 2022; Revised 10 March 2022; Accepted 14 March 2022; Published 13 April 2022

Academic Editor: Thippa Reddy G

Copyright © 2022 Hailong Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cardiovascular disease is one of the most serious diseases that threaten human health in the world today. Therefore, establishing a high-quality disease prediction model is of great significance for the prevention and treatment of cardiovascular disease. In the feature selection stage, three new strong feature vectors are constructed based on the background of disease prediction and added to the original data set, and the relationship between the feature vectors is analyzed by using the correlation coefficient map. At the same time, a random forest algorithm is introduced for feature selection, and the importance ranking of features is obtained. In order to further improve the prediction effect of the model, a cardiovascular disease prediction model based on R-Lookahead-LSTM is proposed. The model based on the stochastic gradient descent algorithm of the fast weight part of the Lookahead algorithm is optimized and improved to the Rectified Adam algorithm; the Tanh activation function is further improved to the Softsign activation function to promote model convergence; and the R-Lookahead algorithm is used to further optimize the long-term memory network model. Therefore, the long- and short-term memory network model can be better improved so that the model tends to be stable as soon as possible, and it is applied to the cardiovascular disease prediction model.

## 1. Introduction

In recent years, the global information technology revolution has ushered in a new round of development climax with the vigorous development of emerging technologies such as cloud computing, big data, artificial intelligence, and the internet of things. At the same time, it has also injected new vitality into the development and transformation of traditional industries. At present, it is gradually being used in various industries such as transportation, logistics, education, and so on. In the field of medical and health care, the continuous accumulation of medical data makes the traditional medical and health care industry gradually develop to the stage of smart medical care. At this stage, the ability of computers to process data has grown rapidly by orders of magnitude, making the application scenarios of artificial intelligence in the medical and health industry more diverse, including medical image analysis, medical record and literature analysis, auxiliary diagnosis and treatment, drug

development, and disease prediction. The medical internet of things [1, 2] and many other aspects have promoted the formation of a patient-centered medical data network. With the recent explosion of artificial intelligence, machine learning as the main research method of artificial intelligence has also made great progress, with outstanding achievements in speech recognition, risk research, disease diagnosis, and so on. It has high accuracy in prediction, which also provides a new research direction for disease prediction.

The prediction model has gradually developed from the original prediction model based on expert rules to the prediction model based on statistical analysis and then to the prediction model based on machine learning, with increasing efficiency and accuracy. With the development of science and technology, prediction models for ischemic heart disease [3], liver disease prediction models [4], Bayesian-based disease prediction models [5], and acute coronary syndrome diagnosis and prediction models [6]

continue to emerge., which is an early disease prediction model, mainly through the establishment of prediction models through machine learning algorithms. At this stage, researchers mainly build models to predict diseases through a machine learning algorithm, and the accuracy and prediction effect need to be improved. Reference [7] proposed a hybrid method based on random forest and multivariate adaptive regression splines to establish a disease prediction model. Reference [8] established an MNN heart disease prediction model. Reference [9] proposed a new method based on heart rate signals. A new method for CAD diagnosis is based on feature extraction. Literature [10] proposed a disease prediction model using support vector machine classification algorithm and web service framework for medical data analysis. Literature [11] proposed a disease prediction based on dynamic sampling and transfer learning model. References [12, 13] proposed a multivariate time series prediction and time series knowledge graph link prediction model based on LSTM, respectively, which achieved great advantages in terms of computational cost and prediction effect. Reference [14] proposed an online medical decision support system for predicting chronic kidney disease. Reference [15] proposed an enhanced feature-level deep convolutional neural network model. The researchers went from initially using a machine algorithm to integrating statistical models and mathematical models into machine learning models, which further improved the prediction effect of the model. Reference [16] established two models to predict a blood disease. Reference [17] constructed a deep learning-based disease prediction model based on five different types of medical data. Reference [18] combined feature selection techniques with five classifications. Algorithms are used in combination. Literature [19] proposed an adaptive signal processing method based on the Levenberg–Marquardt filter of the function link artificial neural network. Literature [20] proposed an integrated deep learning method using deep learning and feature fusion model. More and more researchers are joining the study of prediction models. In order to improve the prediction effect of the models, many people use different models to predict the same disease or use different medical data sets to build a model. This to a certain extent improves the accuracy of the model. The neural network model has gradually emerged in the field of forecasting, but the parameters of the neural network are difficult to adjust, and the selection of optimization algorithms will also cause differences in forecasting results. Many prediction models only improve the structure of the neural network and do not analyze the difference in results caused by different optimization algorithms. Even when the input data set and model structure are exactly the same, the structure obtained by choosing different optimization algorithms is not the same. In view of the problem of low model accuracy caused by different optimization algorithms, this paper integrates the gradient descent optimization algorithm and the long-short-term memory network model to establish a combined disease prediction model, helping people become more aware of their physical condition, while also providing support for doctors to intervene in patients.

The second section of this paper expounds on the theoretical basis related to the model proposed in this paper; the third section introduces the basic process and main steps of the model establishment; and the fourth section carries out the experimental analysis, mainly including the comparative experiments of optimization algorithms and the lengths of different optimization algorithms. This paper compares and analyzes the algorithms proposed in this paper from different aspects through comparison experiments of time memory network models; the fifth section summarizes the full text.

## 2. Theoretical Basis

*2.1. Random Forest Feature Selection Algorithm.* The random forest algorithm [21] uses the Gini index method to evaluate the feature importance. First, the contribution of each feature vector in the data set to each tree in the random forest is judged, and then the average value is taken, and finally, the feature vectors are compared. The size of the contribution is arranged in descending order.

The calculation formula of the Gini index is shown in the following formula  $P_k$ :

$$\text{Gini}(p) = \sum_{k=1}^k P_k (1 - P_k). \quad (1)$$

Here,  $k$  represents the category and  $P_k$  represents the sample weight of the sample  $k$ .

The importance score of feature  $x_j$  on node  $m$  is shown in the following formula:

$$\text{VIM}_{jm}^{(\text{Gini})} = GI_m - GI_l - GI_r. \quad (2)$$

Here,  $GI_l$  and  $GI_r$  represent the Gini indices of the two new nodes before and after the split, respectively.

If the node where feature  $x_j$  appears in tree  $i$  is in set  $M$ , then the importance score of feature  $x_j$  on the  $i$ -th tree is shown in the following formula:

$$\text{VIM}_{ij}^{(\text{Gini})} = \sum_{m \in M} \text{VIM}_{jm}^{(\text{Gini})}. \quad (3)$$

Assuming that there are  $n$  trees in the random forest, the importance score is shown in the following formula:

$$\text{VIM}_j^{(\text{Gini})} = \sum_{i=1}^n \text{VIM}_{ij}^{(\text{Gini})}. \quad (4)$$

Finally, the obtained feature importance score is normalized, and the formula is shown in the following formula:

$$\text{VIM}_j = \frac{\text{VIM}_j}{\sum_{i=1}^c \text{VIM}_i}. \quad (5)$$

*2.2. LSTM Algorithm.* Aiming at the long-term dependence of the convolutional neural network model, Hochreiter and Schmidhuber improved on the basis of RNN and proposed a long short-term memory networks (LSTM) model, which better solves the gradient explosion and disappearance [22].

Compared with the ordinary convolutional neural network model, the LSTM model increases the cell state and introduces input gates, output gates, and forget gates to save and control information. The function of the cell state is to judge whether the information is useful; the input gate determines the retention of the current information; the forget gate determines the retention of the previous state; and the output gate determines the output information according to the current state of the neural network. As shown in Figure 1, compared with the RNN model, the model has a cell state  $C_t$  in addition to the hidden state  $h_t$  at each moment. For the  $t$ -th time step, the forget gate, the input gate, and the output gate, respectively, represent for  $f_t$ ,  $i_t$ , and  $o_t$ .

The core of the LSTM model is the cell state, which consists of an activation function sigmoid and a multiplier. The sigmoid function outputs a number in the range of (0,1), indicating how much information can pass through the cell state. The processing of data by the LSTM model mainly includes the following four steps:

#### Step 1. Forget the door

The forget gate controls the forgetting degree of information, and the forget gate controls whether the current cell state forgets the hidden cell state at the previous moment through the output probability of the sigmoid function. The forget gate takes the state output  $h_{t-1}$  of the hidden layer at the last moment and the current input  $x_t$  as input and obtains the output vector  $f_t$  of the forget gate, which determines how much state  $C_{t-1}$  is retained by the previous neuron, so the output value  $f_t$  of the forget gate determines the pair of degree of retention of information. The formula expression is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (6)$$

Here,  $f_t$  is the output value of the forget gate and  $\sigma$  is the sigmoid function.

#### Step 2. Input gate

The input gate is responsible for the input of current information, considering which new information to add to the cell state. First, the current input information is jointly determined by  $h_{t-1}$  and  $x_t$ . Then,  $h_{t-1}$  and  $x_t$  get the new candidate cell state  $C_t$  through the Tanh network layer, and the input gate assigns weights to the components in  $C_t$  to control how much new information is added to the network, and its weight range is generally between (0,1). The formula expression is as follows:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \text{tanh}(W_c \cdot [h_{t-1}, x_t] + b_c). \end{aligned} \quad (7)$$

#### Step 3. Cell state update

Update the cell state information  $C_{t-1}$  to obtain new cell state information  $C_t$ . The update formula is shown in the following formula:

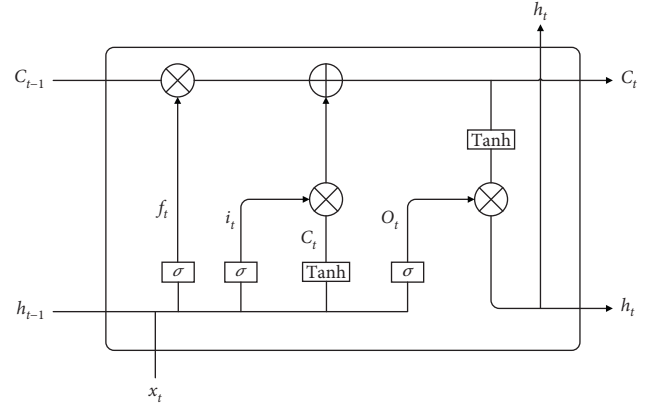


FIGURE 1: LSTM structure diagram.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t. \quad (8)$$

Here,  $\odot$  is the Hadamard product.

#### Step 4. Output gate

Using  $h_{t-1}$  and  $x_t$  to calculate the current state  $o_t$ , the new cell state  $C_t$  is adjusted by the Tanh activation function and multiplied by this vector, and finally, the cell output  $h_t$  of the current model is obtained. The formula expression is as follows:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t \odot \text{tanh}(C_t). \end{aligned} \quad (9)$$

When training the LSTM model, there are eight sets of parameters that need to be learned by the model, namely, four sets of weights  $w_f$ ,  $w_i$ ,  $w_o$ , and  $w_c$  and four sets of bias items  $b_f$ ,  $b_i$ ,  $b_o$ , and  $b_c$  corresponding to  $f_t$ ,  $i_t$ ,  $o_t$ , and  $C_t$ . For the RNN algorithm and the LSTM algorithm, the most commonly used BPTT (back-propagation through time) algorithm [23] is used for training. There are five main steps:

Step 1: initialize the weight parameters

Step 2: use the above formula to forward calculate the output value of each neuron

Step 3: calculate the error value of each neuron in reverse

Step 4: calculate the gradient of each weight parameter according to the corresponding error value

Step 5: use the optimization algorithm to update the weights and iterate continuously until the error converges to the specified threshold and stop training

### 2.3. Optimization

**2.3.1. Lookahead Algorithm.** Most of the gradient descent algorithms are improved by the SGD algorithm, including the Adagrad algorithm, the Adadelta algorithm, the RMSProp algorithm, the Adam algorithm, the Nadam algorithm, and the AdaBound and AMSBound algorithms [24]. These optimization algorithms improve the

exploratory training process by incorporating momentum methods or by optimizing the learning rate and finally allow the model to converge. The Lookahead algorithm [25] differs from the above algorithms in that it adopts a completely new design: two sets of weights are maintained and interleaved between them. Simply put, it allows the faster set of weights to keep exploring forward while leaving the slower set of weights behind, resulting in better long-term stability. The Lookahead algorithm flow is as follows:

Step 1: backup the existing parameter  $\theta_t$  of the model

Step 2: starting from parameter  $\theta_t$ , update  $k$  steps with the SGD algorithm to obtain a new weight  $\tilde{\theta}_t$

Step 3: update model weights  $\theta_t \leftarrow \theta_t + \alpha(\tilde{\theta}_t - \theta_t)$

The Lookahead algorithm maintains two sets of weights: fast weights and slow weights. First, an internal optimizer (generally a stochastic gradient descent optimization algorithm) is used to iteratively update the fast weights part  $k$  times, and then the slow weights part is updated in the direction of the last fast weights part. The fast weights part can move forward faster to better explore, and the slow weights part can play a role in maintaining stability. The two cooperate with each other to explore the entire space in a more detailed manner, enabling the Lookahead algorithm to explore faster and more stably and achieve convergence. When the fast weights part is slowly explored near the minimum value, the update of the slow weight part prompts Lookahead to explore better new regions, thereby improving the accuracy. The formulas of the fast weight part and slow weight part are as follows:

$$\text{fast weight: } \phi_{t,i+1} = \phi_{t,i} + A(L, \phi_{t,i+1}, d), \quad (10)$$

$$\text{slow weight: } \phi_{t+1} = \phi_t + \alpha(\phi_{t,k} - \phi_k). \quad (11)$$

**2.3.2. RAdam Algorithm.** The modified adaptive moment estimation algorithm, RAdam algorithm [26], is one of the latest optimization algorithms, which has the advantages of fast training speed and good convergence effect. Adam algorithm [27] cannot make accurate adaptive momentum selection due to lack of data in the early stage of training, the convergence speed is very fast when using the Adam algorithm to optimize, but it is easy to fall into the local optimum. Therefore, it is necessary to add a warm-up phase when the training is just started. However, the number of preheating needs to be adjusted manually, and the loss value is different for different times, and it will also be different on different data sets. Therefore, the RAdam algorithm is improved on the basis of the Adam algorithm. Based on the design of the rectifier function, the algorithm can dynamically and adaptively open or close the learning rate to adjust the gradient, which makes the initial learning rate more robust and avoids early training due to lack of the drastic change problem caused by learning enough data. The RAdam algorithm flow is as follows:

Step 1: initialize where the step size is  $\{\alpha_i\}_{i=1}^T$ , the exponential decay rate of moment estimation is  $\{\beta_1, \beta_2\}$ , the initial parameter is  $\theta_0$ , the random objective function is  $f_t(\theta)$ , the initial time step is  $t=1$ , the moment of the SMA is initialized to  $m_0 = 0$  and  $v_0 = 0$ , and the maximum length of the SMA is calculated as  $\rho_\infty = 2/(1 - \beta_2) - 1$ .

Step 2: calculate the gradient  $g_t$  at the  $t$ -th step, update the second-order moment estimate  $v_t$ , update the first-order moment estimate  $m_t$ , and calculate  $\rho_t$ .

Step 3: if  $\rho_t > 4$ , update parameter  $\theta_t$  by calculating adaptive learning rate and variance correction; otherwise, use nonadaptive momentum to update parameter.

Step 4: determine whether the parameters are converged. If the parameters have converged, stop training; otherwise, let  $t = t + 1$  and repeat steps 3 and 4 until the parameters converge.

### 3. Disease Prediction Model Based on R-Lookahead-LSTM

**3.1. Improved LSTM Algorithm.** In this section, the LSTM model is improved, and the Softsign activation function is used to replace the Tanh activation function in the input gate, which can make the model converge quickly. The improved LSTM model structure is shown in Figure 2.

The processing of data by the improved LSTM mainly includes the following four steps:

Step 1: control which historical information needs to be discarded by the cell state by forgetting the sigmoid activation function in the gate. The forget gate takes the state output  $h_{t-1}$  of the hidden layer at the previous moment and the current input  $x_t$  as input and outputs a vector  $f_t$ , which determines how much of the state  $C_{t-1}$  of the previous neuron to retain. The formula expression is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f). \quad (12)$$

Step 2: the input gate is responsible for the degree to which new information about the cell state is added. The information to be added first is determined by the state output  $h_{t-1}$  of the hidden layer at the previous moment and the current input  $x_t$ . Then  $h_{t-1}$  and  $x_t$  get the candidate cell state  $C_t$  through the Softsign network layer, and the input gate assigns each component in  $C_t$  a weight between 0 and 1 to control how much new information the network adds. The formula expression is as follows:

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \\ \tilde{C}_t &= \text{Softsign}(W_c \cdot [h_{t-1}, x_t] + b_c). \end{aligned} \quad (13)$$

Step 3: update the cell state information  $C_{t-1}$  to obtain new cell state information  $C_t$ . The update formula is as follows:

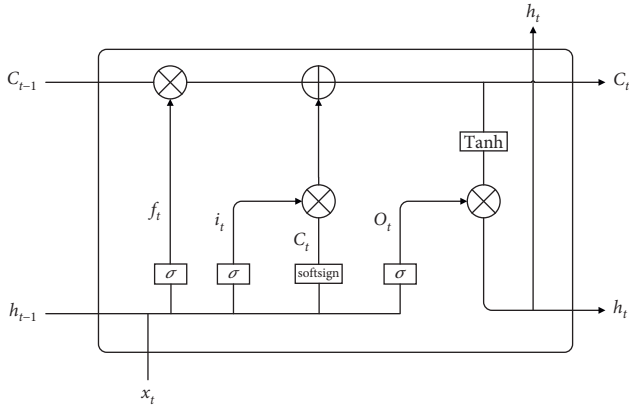


FIGURE 2: Improved LSTM model structure diagram.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t. \quad (14)$$

Step 4: the output gate is responsible for the determination of the output value  $h_t$ . Use  $h_{t-1}$  and  $x_t$  to calculate the state  $o_t$  after the information passes through the output gate;  $C_t$  is multiplied by the vector after being adjusted by the Tanh function; and finally, the unit output of the current neural network is obtained. The formula expression is as follows:

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \\ h_t &= o_t * \text{tanh}(C_t). \end{aligned} \quad (15)$$

When training the LSTM model, there are eight sets of parameters that need to be learned by the model, namely, four sets of weights  $w_f$ ,  $w_i$ ,  $w_o$ , and  $w_c$  and four sets of bias items  $b_f$ ,  $b_i$ ,  $b_o$ , and  $b_c$  corresponding to  $f_t$ ,  $i_t$ ,  $o_t$ , and  $C_t$ . The training process is as follows:

- Step 1: initialize the weight parameters
- Step 2: use the above formula to forward calculate the output value of each neuron
- Step 3: calculate the error value of each neuron in reverse
- Step 4: calculate the gradient of each weight parameter according to the corresponding error value
- Step 5: use the R-Lookahead optimization algorithm to update the weights and iterate continuously until the error converges to the specified threshold and stop training

**3.2. Improved Lookahead Algorithm.** To solve LSTM, it is necessary to continuously solve and update various weights and bias terms of the model to make it approach or reach the optimal value, so as to achieve the effect of minimizing the loss function. This chapter optimizes the LSTM model using the improved Lookahead optimization algorithm (R-Lookahead algorithm). The Lookahead algorithm maintains two sets of weights, including the fast weights part and the slow weights part. This section makes improvements to the fast weights section, using a modified adaptive moment

estimation algorithm to adjust the gradient. The RAdam algorithm can dynamically switch on and off the adaptive momentum so that the entire training process can be quickly stabilized, avoiding the phenomenon of violent oscillation caused by the limited number of samples in the early training period. The R-Lookahead algorithm is shown in Algorithm 1.

**3.3. R-Lookahead-LSTM Disease Prediction Model.** As the main algorithm of this section, the LSTM algorithm can solve the problem of gradient explosion and disappearance in the RNN model training process to a certain extent. The model structure of the LSTM algorithm is optimized and improved on the basis of the model structure of the RNN algorithm. This algorithm sets an output gate, an input gate, and a forget gate in each neuron of the RNN algorithm. For data samples, on the one hand, the LSTM algorithm improves the long-term dependence of the RNN algorithm to a certain extent. On the other hand, the Softsign activation function in the improved LSTM algorithm can improve the convergence speed of the model and make the model stabilize as soon as possible. However, due to the overall structure of the neural network model, no matter how it is optimized and reconstructed, in some application scenarios, its training results will still appear to be more or less locally optimal. In view of the above problems, this chapter uses the R-Lookahead algorithm to optimize and improve the LSTM model and proposes the R-Lookahead-LSTM model for predicting cardiovascular disease. The workflow of the R-Lookahead-LSTM model is shown in Figure 3, and the model building process is shown in Algorithm 2.

## 4. Empirical Analysis

**4.1. Data Preprocessing.** The cardiovascular disease data set selected in this paper includes three types of data: objective facts, examination results during physical examination, and information provided by patients, with 11 input variables and 1 target variable. This paper aims to predict whether a sample has cardiovascular disease through body-related features. If the sample has cardiovascular disease, the label is 1, and if the sample is healthy, the label is 0. This data set is a classic two-category data set, which meets the requirements for data sets in this paper.

**4.1.1. Construction of Strong Eigenvectors.** The purpose of this paper is to build a predictive model suitable for cardiovascular disease. If the input of the model only relies on the feature vector in the data set, there is a lack of strong features in the context of the prediction of the disease, and the prediction result is not ideal. In order to generate new strong features, based on the feature-derived method, this paper analyzes the existing feature vectors in the data set, mines the underlying laws and data structures of cardiovascular disease prediction problems, and constructs a series of new features. The process of constructing new features will be described in detail below.

Aiming at the prediction of cardiovascular disease, this paper firstly selects height and weight for targeted research, analyzes the relationship between height and weight and

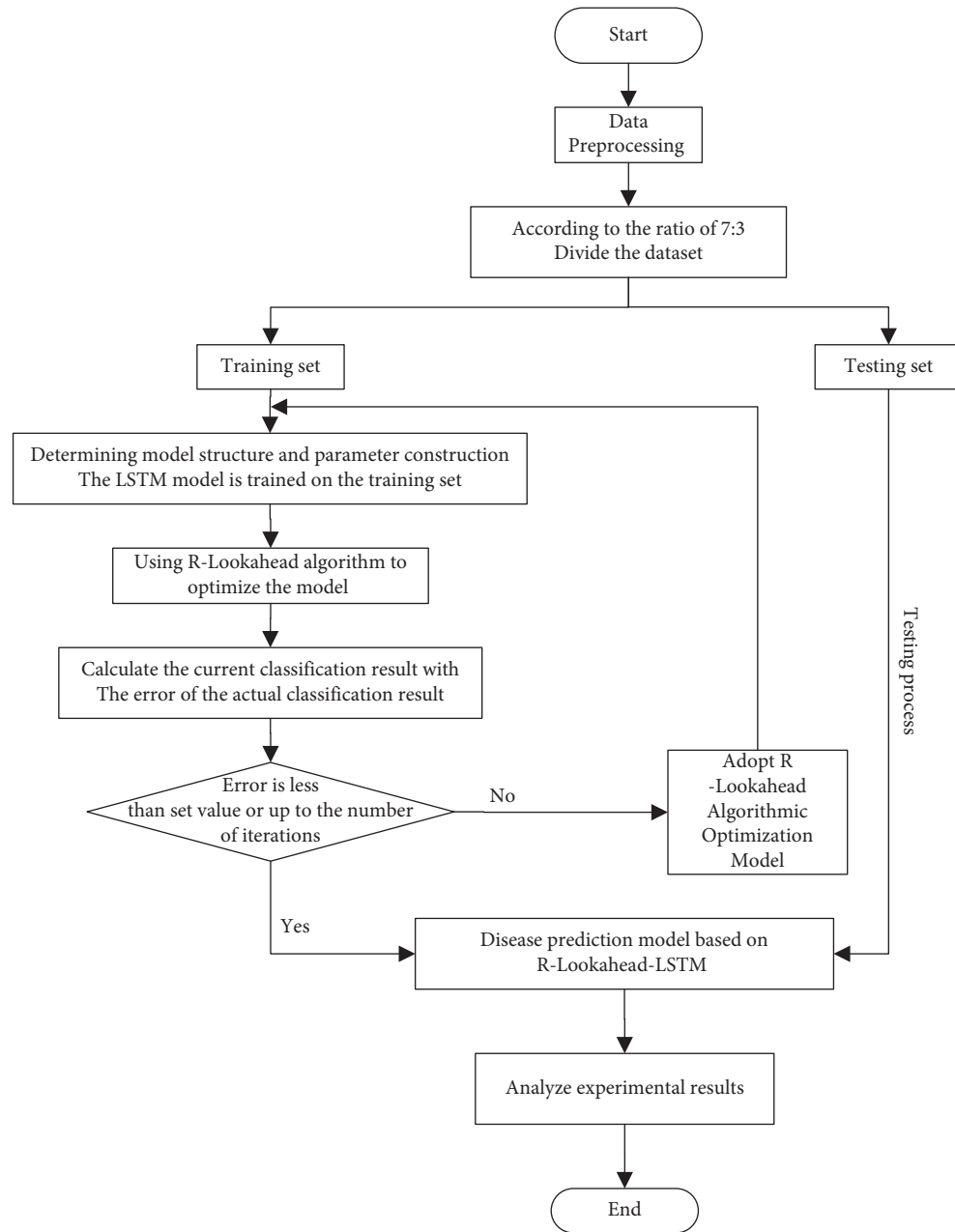


FIGURE 3: Flow chart of a disease prediction model based on R-Lookahead-LSTM.

whether or not the disease is diagnosed, and constructs new features based on their relationship. Figures 4 and 5 are scatter plots of height, weight, and disease, depicting the distribution of height and weight in the cardiovascular disease data set, which can reflect the relationship between height, weight, and disease, as well as the overall law. As you can see from the graph, for the same height, people with heavier weights are more likely to have the disease. Further analysis, obesity may be an important factor leading to cardiovascular disease. In order to make the prediction model fit this point more accurately, a new eigenvector body health index (BMI) is added to comprehensively consider

height and weight, and the calculation formula is shown in the following formula:

$$\text{BMI} = \frac{\text{weight}}{\text{height}^2} \times 100\%. \quad (16)$$

Mean arterial pressure (MAP) is the mean arterial blood pressure level throughout a cardiac cycle. Pulse pressure (PP) is the pressure difference between  $ap_{hi}$  and  $ap_{lo}$ . Studies have shown that MAP and PP may be risk factors for cardiovascular disease. In order to make the predictive ability of the model more accurate, new eigenvectors mean

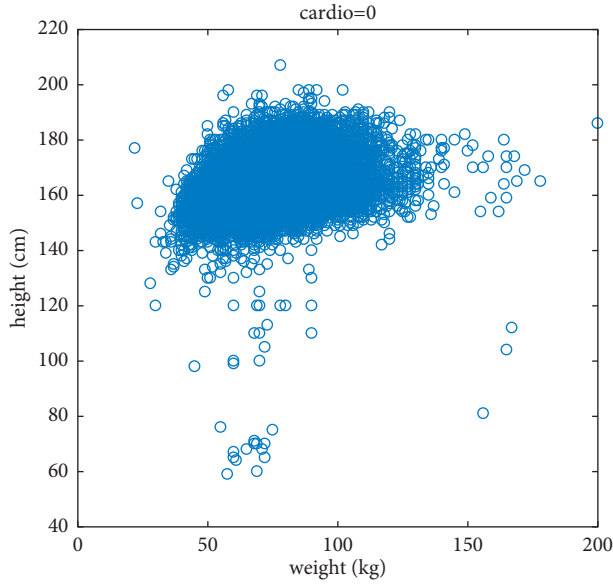


FIGURE 4: Scatter diagram of the relationship between height, weight, and healthy samples.

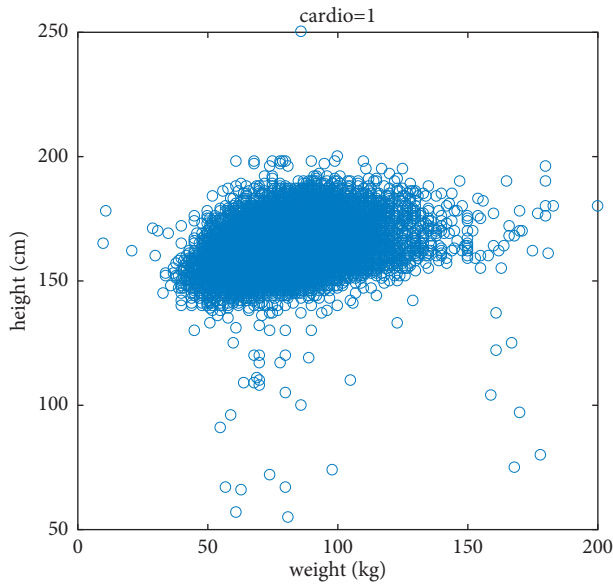


FIGURE 5: Scatter diagram of the relationship between height, weight, and sick samples.

arterial pressure (MAP) and pulse pressure (PP) are added, and the calculation formulas are shown in the following formula:

$$\text{MAP} = \frac{ap\_hi + 2ap\_lo}{3}, \quad (17)$$

$$\text{PP} = ap\_hi - ap\_lo. \quad (18)$$

**4.1.2. Data Processing.** Due to the huge amount of data in the cardiovascular disease data set and the nonuniform type of features, which makes the research work difficult, the data

needs to be processed first. This paper checks the duplicate values. After checking, it is found that there are 24 duplicate samples in the data set. The duplicate items have no effect on the training of the model, so the duplicate samples are deleted. A box plot was used to detect outliers. Looking closely at the data, we noticed that the youngest was about 29 years old, the youngest was 55 cm tall, the youngest weighed 10 kg, the tallest was 250 cm tall, and the heaviest weighed 200 kg, the smallest  $ap\_hi$  is  $-150$ , and the smallest  $ap\_lo$  is  $-70$ . In order to deal with such outliers, we delete samples with height and weight below 5% or above 95% and delete samples with negative blood pressure. The normal value for diastolic blood pressure is 60–80 mmHg, and the normal value for systolic blood pressure is 90–120 mmHg, so this paper will remove  $ap\_hi$  outliers over 200 and  $ap\_lo$  outliers less than 50. For the cholesterol and blood sugar fields, this subsection takes the form of the data set itself because the progression of the data also represents the level of the sample in this feature. At this point, the data set has been basically processed in this section, but there are still two problems. One is that the continuous variables and the categorical variables in the features do not belong to the same dimension. If the original data values are used directly, the features will affect the prediction results. The degree of influence is different, and the features are not comparable; second, some features are in the form of categorical variables, and the numerical size will mislead the model to a certain extent, so the data needs to be normalized and one-hot coding. In addition, in order to more intuitively see the relationship between features, this paper uses the Pearson correlation coefficient to draw a heat map.

(1) *Normalizing.* In large-scale data analysis projects, data often have different sources, and their dimensions and specifications are different and cannot be directly compared, so normalization processing is required to eliminate the resulting bias. After the original data is normalized, all indicators are in the same order of magnitude, and the input features and target predicted values approximately obey the normal distribution, which helps eliminate outliers and noise in the data and is suitable for comprehensive comparative evaluation. This section normalizes the “age,” “height,” “weight,” “zap\_hi,” “ap\_lo,” “MAP,” and “PP” fields to reduce the feature data of different magnitudes covering other features on the target effect of the function. The normalization formula is as follows:

$$y_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \quad (19)$$

(2) *One-Hot Encoding.* In machine learning algorithms, we often encounter categorical features. These feature values are not continuous but discrete and disordered, so it is necessary to digitize such data. Plotting the height distribution by violin drawing, we found that the height of class 2 is always higher than that of class 1, so class 1 represents women and class 2 represents men, and 2 is always numerically higher

than 1, which will mislead the model to some extent, so the gender field uses one-hot encoding in this paper.

(3) *Heat Map*. According to the size of the correlation coefficient corresponding to the different square colors in the heat map, the size of the correlation between the variables can be judged, and the relationship between the various features can be observed more intuitively. The calculation formula is as follows:

$$\rho_{X_1, X_2} = \frac{\text{Cov}(X_1, X_2)}{\sqrt{DX_1, DX_2}} = \frac{EX_1X_2 - EX_1 * EX_2}{\sqrt{DX_1 * DX_2}}, \quad (20)$$

where  $\rho$  is the correlation coefficient, Cov is the covariance, and  $E$  is the mathematical expectation.

From Figure 6, we can easily see the relationship between data and data, which lays the foundation for subsequent feature selection.

**4.2. Feature Selection.** After feature construction and data processing, this paper uses a random forest algorithm to model all samples for feature selection. This article uses the Scikit-learn library in Python. First, load the RandomForestClassifier module in the Scikit-learn library to model all the data. When the model is built, we use the feature\_importances\_ function in the module to output the importance of all features. This function will calculate and record the Gini coefficient change value of each node after feature splitting during the tree construction process. Finally, a unified normalization can be done to obtain the importance of each feature. The feature importance rankings are shown in Table 1.

Through the feature importance ranking made by the random forest feature selection algorithm, several important information can be found. First, the three new feature vectors MAP, PP, and BMI constructed in this paper are in the relatively high position, which shows that the constructed new feature vector plays a very important role in the prediction of cardiovascular disease; second, the feature importance ranking made by the random forest feature selection algorithm is basically the same as the result of the heat map, which shows that the important features selected by the random forest algorithm are very important. It is very convincing. In this paper, the top 12 features are selected as the input vector of the cardiovascular disease prediction model constructed in this paper. Removing the features with lower rankings through feature selection can speed up the running speed of the next prediction algorithm and at the same time improve the prediction accuracy of the obtained model and improve the model's overall performance.

### 4.3. Metric Indicator

**4.3.1. Confusion Matrix.** The confusion matrix is a metric for judging the results of a classification model and is part of model evaluation. Confusion matrices are mostly used to judge the pros and cons of classifiers and are suitable for classified data models. The confusion matrix is shown in Table 2.

True positive (TP) and true negative (TN) measure the ability of a classification model to predict whether a patient has a disease or not, and false positive (FP) and false negative (FN) identify the number of false predictions produced by the predictive model. The accuracy rate represents the overall predictive ability of the machine learning model and is used to measure the success of the predictions of the disease prediction model. The recall is used to measure the sensitivity of a disease prediction model with the aim of recalling potential cases. F-score is the weighted harmonic average of precision and recall, which is often used to evaluate the quality of classification models. F1 combines the results of precision and recall, and when it is higher, it can indicate that the experimental method is more effective. This paper uses different performance indicators to evaluate the model proposed in this paper, and the formula is shown in Table 3.

**4.3.2. MCC Value.** The Matthews correlation coefficient (MCC value) [28] is a contingency matrix method for calculating the Pearson product-moment correlation coefficient between actual and predicted values. The key advantage of the Matthews correlation coefficient is that the classifier must make correct predictions for most negative cases and most positive cases to get a high-quality score, independent of their proportion in the entire data set. The formula for calculating the MCC value is as follows:

$$\text{MCC} = \frac{\text{TP} * \text{TN} - \text{FP} * \text{FN}}{\sqrt{((\text{TP} + \text{FP}) * (\text{TP} + \text{FN}) * (\text{TN} + \text{FP}) * (\text{TN} + \text{FN}))}} \quad (21)$$

### 4.4. Analysis of Results

**4.4.1. Optimization Algorithm Experiment Comparison.** In order to show the advantages of the improved Lookahead algorithm proposed in this paper, RMSprop algorithm [29], Adam algorithm, RAdam algorithm, and Lookahead algorithm are used as comparison algorithms, and the parameters are adjusted for comparison experiments. The parameters of each optimization algorithm are set as shown in Table 4.

In order to compare the performance of the above optimization algorithms, this section tests the RMSprop, Adam, RAdam, Lookahead, and R-Lookahead algorithms and uses accuracy and loss as evaluation indicators. The accuracy graph and loss graph of the optimization algorithm are shown in Figures 7 and 8. The results show that with the increase of the number of iterations, the correct rate of each optimization algorithm keeps increasing, and the loss value keeps decreasing. Among them, the RAdam algorithm has been improved on the basis of Adam. It can dynamically open or close the adaptive learning rate to adjust the gradient according to the variance of the adaptive rate. Therefore, when the number of iterations is 500, the accuracy and loss are better than Adam's algorithm. The Lookahead algorithm maintains two sets of weights to achieve faster convergence.



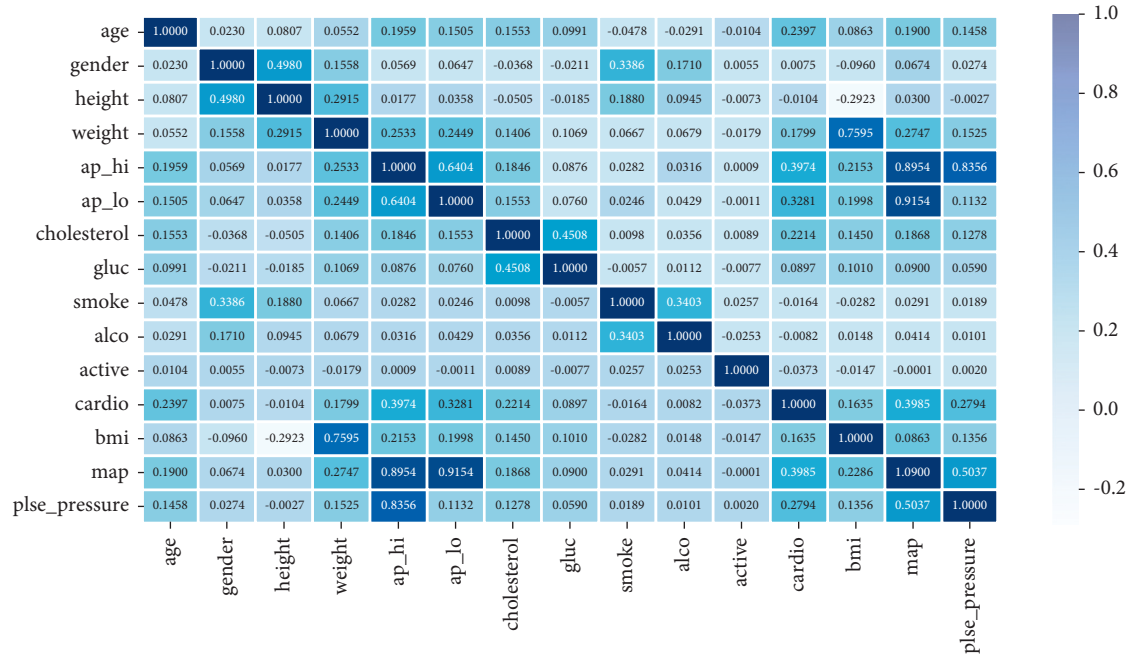


FIGURE 6: Correlation diagram of characteristic variables.

```

R-Lookahead algorithm
Require: Initial parameters  $\varphi$ , objective function  $L$ 
Require: Synchronization period  $k$ , slow weights step size  $\alpha$ , optimizer RAdam
for  $t = 1, 2, \dots$  do
    Synchronize parameters  $\theta_{t,0} \leftarrow \varphi_{t-1}$ 
    for  $i = 1, 2, \dots, k$  do
        Sample minibatch of data  $d \sim D$ 
         $\theta_{t,i} \leftarrow \theta_{t,i-1} + \text{RAdam}(L, \theta_{t,i-1}, d)$ 
    end for
    Perform outer update  $\varphi_t \leftarrow \varphi_{t-1} + \alpha(\theta_{t,k} - \varphi_{t-1})$ 
end for
return parameters  $\varphi$ 
    
```

ALGORITHM 1: R-Lookahead algorithm.

```

R-Lookahead-LSTM algorithm
Step 1: process the data, analyze the correlation of the data, and use the ensemble learning algorithm random forest algorithm to perform feature selection on the data, so as to further determine the feature vector required to build the model
Step 2: divide the above-processed data set into a training set and test set according to the ratio of 7:3
Step 3: determine the structure of the LSTM model for the data samples of the training set and determine the number of network layers and initialization parameters through experimental tests
Step 4: train the model and use the R-Lookahead optimization algorithm to optimize the loss function of the LSTM model
Step 5: use the test set data samples as the input data of the model to test the prediction effect of the R-Lookahead-LSTM disease risk prediction model
Step 6: use multiple indicators such as accuracy rate, recall rate, F1-score, specificity, and MCC value to evaluate the prediction effect of the model
    
```

ALGORITHM 2: R-Lookahead-LSTM algorithm.

TABLE 1: Ranking of feature importance.

Ranking	Feature	Feature importance coefficient
1	MAP	0.260602
2	ap_hi	0.155965
3	ap_lo	0.116014
4	PP	0.110476
5	age	0.094877
6	cholesterol	0.077568
7	weight	0.048473
8	BMI	0.038585
9	gluc	0.036257
10	smoke	0.016611
11	active	0.015657
12	alco	0.013322
13	gender	0.008640
14	height	0.006953

TABLE 2: Confusion matrix.

Confusion matrix	Predict positive class	Predict negative class
Actual positive class	TP	FN
Actual negative class	FP	TN

TABLE 3: Performance indicators.

Name	Formula
Accuracy	$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
Precision	$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
Recall	$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$
Specificity	$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$
F1-score	$F1 - \text{score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

TABLE 4: Optimization algorithm parameter settings.

Optimizers	learning_rate	beta_1	beta_2	Epsilon	$\gamma$	$k$	$\alpha$
RMSprop	0.001					0.9	
Adam	0.001	0.9	0.999	$1e-08$			
RAadam	0.0001						
Lookahead						5	0.5
R- Lookahead						5	0.5

At this time, the traditional Lookahead algorithm slow weight uses the SGD algorithm, which also achieves good results. The accuracy is 0.7747, and the loss is 0.4226. When the SGD algorithm in the slow weight part of the Lookahead algorithm is improved to Radam, this improves the speed to a certain extent and is more stable, and the loss value is also smaller to 0.3928. The specific results of the experiment are shown in Tables 5 and 6.

**4.4.2. Comparison of LSTM Models of Different Optimization Algorithms.** The optimization problem is one of the most important research contents in the field of machine learning. Even when the model structure and model input data are exactly the same, the results obtained by selecting

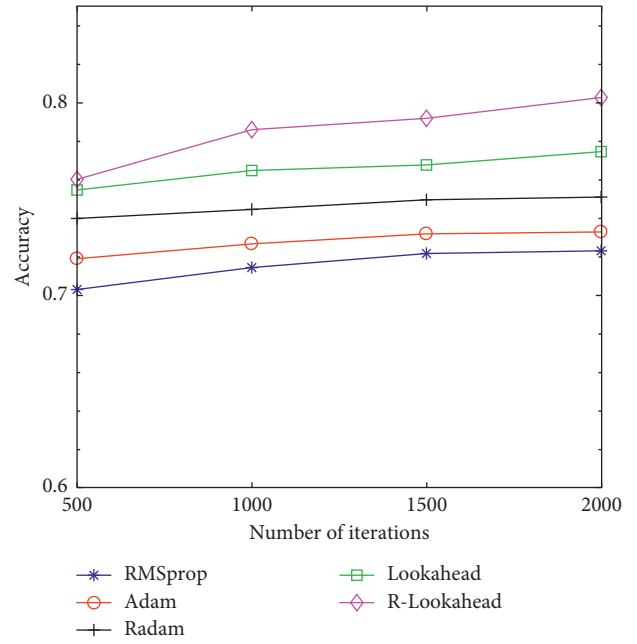


FIGURE 7: Accuracy graph of optimization algorithm.

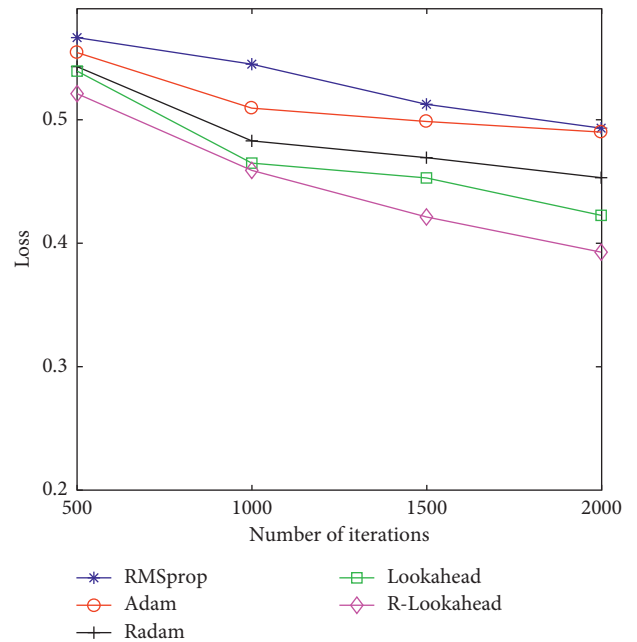


FIGURE 8: The loss graph of the optimized algorithm.

different deep learning optimization algorithms are different. In order to better compare the effects of RMSprop, Adam, Radam, Lookahead, and R-Lookahead algorithms on the LSTM model, this section integrates the above optimization algorithms with the models established in this section and analyzes LSTM models using different algorithms in detail. Experiments show that the Lookahead algorithm and the Lookahead algorithm using Radam have achieved good results, with the correct rate being 0.828296 and 0.857287, respectively; the traditional LSTM

TABLE 5: Accuracy comparison of the optimization algorithms.

Optimizers	Accuracy			
	500	1,000	1,500	2,000
RMSprop	0.7031	0.7145	0.7218	0.7232
Adam	0.7191	0.7268	0.7320	0.7330
RAdam	0.7400	0.7447	0.7497	0.7511
Lookahead	0.7548	0.7649	0.7678	0.7747
R-Lookahead	0.7603	0.7861	0.7920	0.8028

TABLE 6: Loss comparison of optimization algorithms.

Optimizers	Loss			
	500	1,000	1,500	2,000
RMSprop	0.5655	0.5450	0.5125	0.4932
Adam	0.5544	0.5094	0.4986	0.4900
RAdam	0.5430	0.4829	0.4693	0.4531
Lookahead	0.5394	0.4639	0.4529	0.4226
R-Lookahead	0.5211	0.4590	0.4213	0.3928

TABLE 7: Comparison of LSTM models with different optimization algorithms.

Classifiers	Measures					
	Accuracy	Precision	Recall	F1-score	Specificity	MCC
LSTM	0.739870	0.727797	0.776074	0.751161	0.702801	0.480341
RMSp-LSTM	0.770312	0.756012	0.806160	0.780281	0.733607	0.541419
Adam-LSTM	0.784808	0.769447	0.820487	0.794148	0.748276	0.570503
RAdam-LSTM	0.813800	0.796318	0.849140	0.821881	0.777614	0.628672
Lookahead-LSTM	0.828296	0.809754	0.863467	0.835748	0.792283	0.657757
R-Lookahead-LSTM	0.857287	0.836625	0.892120	0.863481	0.821622	0.715926

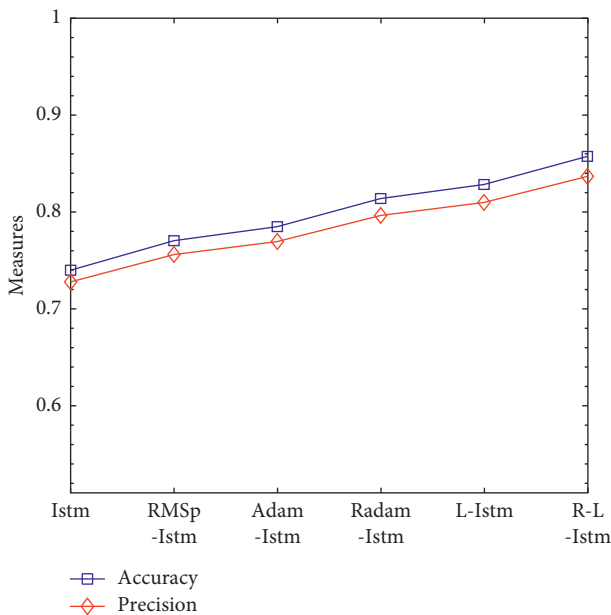


FIGURE 9: Accuracy and precision line graphs of LSTM models under different optimization algorithms.

model has a correct rate of 0.739870 and a recall rate of 0.776074; the R-Lookahead-LSTM model proposed in this chapter has a recall rate of 0.892120 and an MCC value of

0.715926, which indicates that the cardiovascular disease prediction model established in this chapter has good generalization ability and can make correct predictions for most positive and negative situations. The experimental results are shown in Table 7 and Figures 9–11.

### 5. Summary

The disease prediction model is proposed to help doctors make decisions more informatively and accurately. In order to scientifically and effectively assist doctors in medical decision-making, this paper proposes the R-Lookahead-LSTM model for disease prediction. In order to better highlight the prediction effect of the model proposed in this paper, this study compares the proposed model with LSTM, RMSprop-LSTM, Adam-LSTM, RAdam-LSTM, and Lookahead-LSTM in detail. Accuracy, recall, F1-score, MCC value, and other aspects of the prediction model were compared and analyzed. Compared with the traditional LSTM model, the accuracy rate of the proposed model increased by 0.117417; the precision rate increased by 0.108828; the recall rate increased by 0.116046; the F1-score increased by 0.112320; and the MCC value increased by 0.235585. And MCC values increased significantly. For a disease prediction model, obtaining a good accuracy rate indicates that the model has a good classification effect; obtaining a relatively high

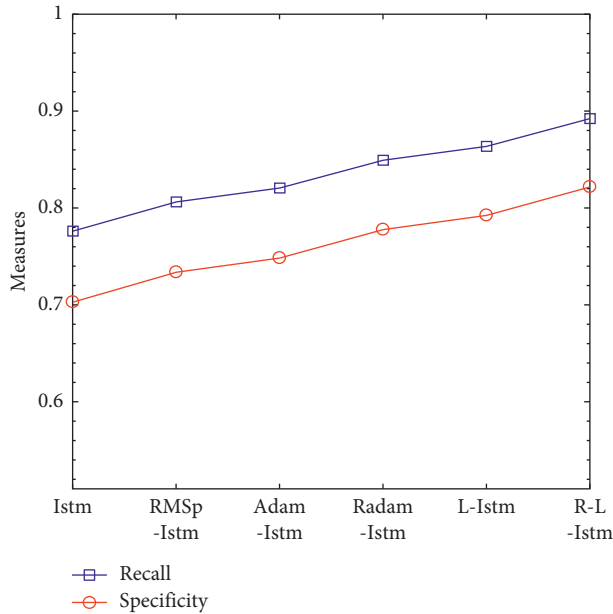


FIGURE 10: Recall rate and specificity line graphs of LSTM models under different optimization algorithms.

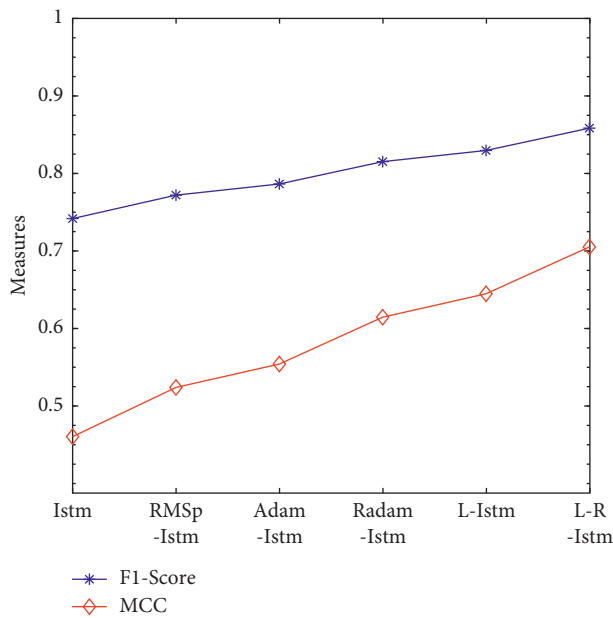


FIGURE 11: Line graph of F1-Score and MCC values of LSTM models under different optimization algorithms.

recall rate, that is, recalling possible case samples, has very important practical significance. The MCC value shows that the model proposed in this section can make correct predictions for most negative cases and most positive cases and achieves the desired effect. The whole experimental process clearly shows that the model proposed in this chapter has good classification performance and achieves the desired effect, providing a new method for the field of disease prediction.

## Data Availability

In this paper, we used the cardiovascular disease data set. The cardiovascular disease data set can be obtained from the following website: [https://www.kaggle.com/sulianova/cardiovascular-disease-dataset#cardio\\_train.csv](https://www.kaggle.com/sulianova/cardiovascular-disease-dataset#cardio_train.csv).

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This study was funded by the National Natural Science Foundation of China (NSFC) (Grant no. 61772160) and the Special Foundation of Scientific and Technological Innovation for Young Scientists of Harbin, China (Grant no. 2017RAQXJ045).

## References

- [1] X. Hu, C. Jin, A. Mamoun et al., "On the design of blockchain-based ECDSA with fault-tolerant batch verification protocol for blockchain-enabled IoMT," *IEEE Journal of Biomedical and Health Informatics*, vol. 12, no. 3, pp. 23–56, 2021.
- [2] W. Wang, C. Qiu, Z. Yin et al., "Blockchain and PUF-based lightweight Authentication protocol for wireless medical sensor networks," *IEEE Internet of Things Journal*, vol. 34, no. 2, 2021.
- [3] M. Kukar, I. Kononenko, C. Grošelj, K. Kralj, and J. Feticch, "Analysing and improving the diagnosis of ischaemic heart disease with machine learning," *Artificial Intelligence in Medicine*, vol. 16, no. 1, pp. 25–50, 1999.
- [4] Y. S. Kim, S. Y. Sohn, and C. N. Yoon, "Screening test data analysis for liver disease prediction model using growth curve," *Biomedicine & Pharmacotherapy*, vol. 57, no. 10, pp. 482–488, 2003.
- [5] C. M. Chu, H. J. Tscai, N. F. Chu et al., "The establishment of bayesian coronary artery disease prediction model," in *Proceedings of the AMIA Annual Symposium Proceedings American Medical Informatics Association*, pp. 925–930, Qazvin, Iran, 2005.
- [6] M. Green, J. Björk, J. Forberg, U. Ekelund, L. Edenbrandt, and M. Ohlsson, "Comparison between neural networks and multiple logistic regression to predict acute coronary syndrome in the emergency room," *Artificial Intelligence in Medicine*, vol. 38, no. 3, pp. 305–318, 2006.
- [7] D. Yao, J. Yang, and X. Zhan, "A novel method for disease prediction: hybrid of random forest and multivariate adaptive regression splines," *Journal of Computers*, vol. 8, no. 1, pp. 170–177, 2013.
- [8] S. Elango and J. Sundararajan, "MNN: Multiclass neural network classifier for cardiac disease prediction models," *Asian Journal of Research in Social Sciences and Humanities*, vol. 6, no. 1, pp. 293–309, 2016.
- [9] S. Patidar, R. B. Pachori, and U. Rajendra Acharya, "Automated diagnosis of coronary artery disease using tunable-Q wavelet transform applied on heart rate signals," *Knowledge-Based Systems*, vol. 82, pp. 1–10, 2015.
- [10] T. Zeng and Y. Zhang, "Medical data analysis and disease prediction model based on support vector machine," *Journal of Medical Imaging and Health Informatics*, vol. 6, no. 4, 2016.

- [11] M. M. Hu, X. Chen, and Y. Z. Sun, "Disease prediction model based on dynamic sampling and migration learning," *Chinese Journal of Computers*, vol. 42, no. 10, pp. 2339–2354, 2019.
- [12] D. H. Chen, S. N. Yin, and J. J. Le, "A link prediction model for time series knowledge graphs in the clinical field," *Computer Research and Development*, vol. 54, no. 12, pp. 2687–2697, 2017.
- [13] W. Zheng and J. Hu, "Multivariate time series prediction based on temporal change information learning method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 1, 2022.
- [14] A. Pramila and P. Eswaran, "An intelligent IoT with cloud centric medical decision support system for chronic kidney disease prediction," *International Journal of Imaging Systems and Technology*, vol. 30, no. 3, pp. 815–827, 2020.
- [15] Z. Zhang, Y. Qiu, X. Yang, and M. Zhang, "Enhanced character-level deep convolutional neural networks for cardiovascular disease prediction," *BMC Medical Informatics and Decision Making, Guangzhou*, vol. 20, no. 3, pp. 1–10, 2020.
- [16] G. Guncar, M. Kukar, M. Notar et al., "An application of machine learning to haematological diagnosis," *Scientific Reports*, vol. 8, no. 1, pp. 411–432, 2018.
- [17] M. A. Green, "Use of machine learning approaches to compare the contribution of different types of data for predicting an individual's risk of ill health: an observational study," *The Lancet*, vol. 392, no. 2, pp. 40–61, 2018.
- [18] R. Ahuja, S. C. Sharma, and M. Ali, "A diabetic disease prediction model based on classification algorithms," *Annals of Emerging Technologies in Computing*, vol. 3, no. 3, pp. 23–41, 2019.
- [19] L. Das, S. Nanda, and J. K. Das, "Hereditary disease prediction in eukaryotic DNA: an adaptive signal processing approach," *Nucleosides, Nucleotides & Nucleic Acids*, vol. 39, no. 8, pp. 1179–1199, 2020.
- [20] F. Ali, S. El-Sappagh, S. M. R. Islam et al., "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion," *Information Fusion*, vol. 63, pp. 208–222, 2020.
- [21] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognition Letters*, vol. 31, no. 14, pp. 2225–2236, 2010.
- [22] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [23] J. O. Orlando De Jesus and M. T. Hagan, "Backpropagation algorithms for a broad class of dynamic networks," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 14–27, 2007.
- [24] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," in *Proceedings of the International Conference on Learning Representations*, pp. 1–21, New Orleans, LA, USA, May 2019.
- [25] M. R. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead Optimizer: K Steps Forward, 1 Step Back," in *Proceedings of the Neural Information Processing Systems*, pp. 48–64, Vancouver, Canada, December 2019.
- [26] L. Liu, H. Jiang, P. He et al., "On the Variance of the Adaptive Learning Rate and beyond," in *Proceedings of the International Conference on Learning Representations*, pp. 21–35, Addis Ababa, Ethiopia, May 2019.
- [27] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Computer Science*, vol. 3, no. 2, pp. 45–59, 2014.
- [28] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, pp. 6–34, 2020.
- [29] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and F.-F. Li, "Every moment counts: dense detailed labeling of actions in complex videos," *International Journal of Computer Vision*, vol. 126, no. 2, pp. 375–389, 2018.