Hindawi
Journal of Spectroscopy
Volume 2022, Article ID 6875022, 10 pages
https://doi.org/10.1155/2022/6875022

Research Article

Component Prediction of Antai Pills Based on One-Dimensional Convolutional Neural Network and Near-Infrared Spectroscopy

Tuo Guo,[1] **Fengjie Xu**,[1] **Jinfang Ma**,[2] **and Fahuan Ge**[3]

[1]*School of Electronic Information and Artificial Intelligence, Shaanxi University of Science and Technology, Xian, Shaanxi 710021, China*
[2]*Opto-Electronic Department of Jinan University, Guangzhou 510632, China*
[3]*School of Pharmaceutical Sciences, Sun Yat-Sen University, Guangzhou 510006, China*

Correspondence should be addressed to Fahuan Ge; gefahuan@mail.sysu.edu.cn

Received 19 July 2022; Revised 24 October 2022; Accepted 22 November 2022; Published 5 December 2022

Academic Editor: Daniel Cozzolino

Convolutional neural networks (CNNs) are widely used for image recognition and text analysis and have been suggested for application on one-dimensional data as a way to reduce the need for preprocessing steps. In this study, the performance of one-dimensional convolutional neural network (1DCNN) machine learning algorithm was investigated for regression analysis of Antai pills spectral data. This algorithm was compared with other chemometric methods, including support vector machine regression (SVR) and partial least-square regression (PLSR) methods. The results showed that the 1DCNN model outperformed the PLSR and SVR models with similar data preprocessing for the three analytes (wogonoside, scutellarin, and ferulic acid) in Antai pills. Taking wogonoside as an example, the indices such as the correction coefficient of determination (R_v^2), the root mean-squared error of cross validation (RMSECV) for calibration set, the prediction coefficient of determination (R_p^2), and the root mean-squared error of prediction (RMSEP) obtained by PLSR modeling were 0.9340, 0.5568, 0.9491, and 0.5088; the indices obtained by SVR modeling were 0.9520, 0.4816, 0.9667, and 0.4117; and the indices obtained by 1DCNN modeling were 0.9683, 0.3397, 0.9845, and 0.2807, respectively. The evaluation metrics of 1DCNN are better than those of PLSR and SVR, and the prediction effect is the best, proving that 1DCNN has a good generalization ability. Especially with outliers of spectra, PLSR's R_p^2 decreased by 0.0181, SVR's R_v^2 decreased by 0.01, and 1DCNN's R_v^2 increased by 0.0009 and R_p^2 decreased by 0.0057. The evaluation indices of 1DCNN have no significant change in comparison with no outliers and can still show good performance, which reflects the inclusiveness of the 1DCNN model for outliers. Simultaneously, the feasibility and robustness of the 1DCNN model in the application of near-infrared spectroscopy was verified, which has a certain application value.

1. Introduction

Pharmaceuticals are special commodities directly related to the health and safety of each citizen, and each link in its life cycle requires strict quality control [1]. In recent years, counterfeit drugs mainly adding chemicals to the capsule shells of Chinese patent medicines or health foods to make their manufacturing and marketing methods more hidden. A survey published on the PSI's website in 2021 showed that the number of reported counterfeit drug incidents worldwide had risen 20-fold from 196 in 2002 to 4,334 in 2020. Quick and effective screening of drug quality has become an urgent problem to be solved. Common drug content analysis methods include thin-layer chromatography (TLC), gas chromatography (GC), high-performance liquid chromatography (HPLC), and DNA molecular markers [2, 3]. These methods typically require the use of various instruments and chemical reagents for destructive pretreatment of drugs, which is time-consuming and costly. Near-infrared spectroscopy (NIRS) is an alternative method for identifying pharmaceutical products and is a fast, simple, contamination-free, sample-free, and holistic analysis technology [4, 5].

Near-infrared spectroscopy technology has been widely used in the pharmaceutical industry, for it has abundant frequency doubling and frequency vibration information in molecular groups [6]. Since NIRS is an indirect analysis technique, it is imperative to find a suitable method for analyzing spectral data. Traditional quantitative analysis methods include multiple linear regression, principal component regression, partial least-square regression, artificial neural networks, and support vector machine regression [7]. However, in the actual spectrum acquisition process, it is inevitably affected by environmental factors, human error, setting noise interference, etc., resulting in the acquisition of outliers of spectra. Under normal circumstances, if there are outliers of spectra, the artificial eye can judge the large difference between the normal spectra, and outliers can be easily distinguished from inliers and manually eliminated. However, in the actual situation, many outliers often require professionals to find a suitable outliers discovery algorithm that is programmed to eliminate it. If outliers of spectra cannot be effectively eliminated in the modeling process, it will greatly impact the prediction effect of regression modeling. This study attempts to establish a more inclusive model of outliers by using the modeling algorithm of one-dimensional convolutional neural networks, conducting quantitative analysis, and applying it to the regression modeling and prediction of the active ingredients of Antai pills.

Kinds of machine learning algorithms have been combined with spectroscopic techniques to be used for classification or regression tasks in recent years [8]. Convolutional neural networks (CNNs) are a key concept in deep learning. Unlike traditional feature extraction methods [9–11], CNN does not require manual feature extraction and uses large amounts of data to achieve the desired results. Specifically, CNNs have demonstrated that deep learning can discover intricate patterns in high-dimensional data, reducing the need for manual effort in preprocessing and feature engineering [12]. CNN is useful for both one-dimensional and multidimensional scenes [13]. Compared with previous artificial networks, CNN does not consider the entire dataset but obtains the features of the data by considering local information. CNN trains faster and with fewer parameters, thus reducing the computational cost and power consumption. Recently, CNNs have been used for classification tasks in infrared (IR) [14], NIR [14], Raman [14–16], and laser-induced fluorescence (LIF) [17] spectral analyses and have been used for regression tasks in IR [18, 19] and NIR [18–23] spectral analyses. These studies indicated that, in some cases, the CNN model outperformed some traditional methods, such as PLS [20, 21, 23], SVR [20], and extreme learning machine (ELM) [23]. PLSR is one of the most commonly used multivariate analysis methods in spectroscopy [24], and SVR has been combined with spectroscopic since 2004 [25]. Zhang et al. proposed a new 1DCNN inception model and studied the performance of CNNs through classification analysis of spectral data. Experimental results show that this model outperformed previous methods such as PCA-ANN, SVR, and PLS while it predicts better results on four different raw datasets than the preprocessed version of those NIR spectra data [20].

This study proposes a quantitative analysis method based on near-infrared technology combined with 1DCNN and constructs a general and robust spectral data analysis model. The analysis results of this model were compared with those of PLSR and SVR, which verified the feasibility of the 1DCNN model in near-infrared technology. The tolerance of 1DCNN to outliers of spectra was verified by observing the differences in the prediction results of 1DCNN, PLSR, and SVR to outliers and inliers, which provides a new idea for solving the problem of outliers.

2. Materials and Methods

2.1. Experimental Environment. The hardware of the experimental environment was an Intel Xeon (R) Platinum 8124M CPU 3.00 GHz, 64 GB memory, and the GPU model was Nividia GeForce RTX 3060. The operating system version was Windows 10, and all experiments were completed using Python. The deep learning model uses the Keras 2.8.0 framework, and the back-end uses TensorFlow 2.8.0 to support the GPU. The PLSR and SVR algorithm codes are based on the scikit-learn 1.0.2 software package.

2.2. Sample Collection and Preparation. We used reference [26] for data collection. The data selected in this study were 101 from 21 batches of Antai pills produced in 2013, 2014, and 2015. These spectral data were measured in 2015 using a SupNIR1500 near-infrared spectrometer in the range of 1000-1800 nm with a 1 nm interval in diffuse reflection mode, and the contents of the three chemical components wogonoside, scutellarin, and ferulic acid in 21 batches of Antai pills were determined by high-performance liquid chromatography (HPLC) gradient elution. We observed outliers during the data acquisition process and conducted on inliers and outliers of spectra. Two datasets were provided, one of which contains five outliers and 96 inliers, 80 samples were used for the training set, and 21 samples for the prediction set. The other datasets have 96 inliers, which were split into training set of 76 samples and prediction set of 20 samples. The method of identifying outliers of spectra is described in Section 2.3.

2.3. Anomaly Spectral Identification. This study uses the Mahalanobis distance (MD) method based on principal component analysis (PCA) to detect outliers. First, normalize the original spectral data, and the data of different orders of magnitude are transformed into the same order of magnitude for comparison, which improves the data comparability. PCA is used for dimension reduction of the data, and the data are linearly mapped to the low-dimensional space to maximize the data difference in the low-dimensional representation. Finally, MD is used to find outliers, and samples over MD are necessarily outliers. The MD formula is as follows:

$$D(X) = \sqrt{(X-\mu)^{T} S^{-1} (X-\mu)}, \tag{1}$$

where X vector represents a spectrum, S represents a covariance matrix, and μ represents a vector composed of the mean values of all X columns. The obtained MD results are shown in Figure 1.

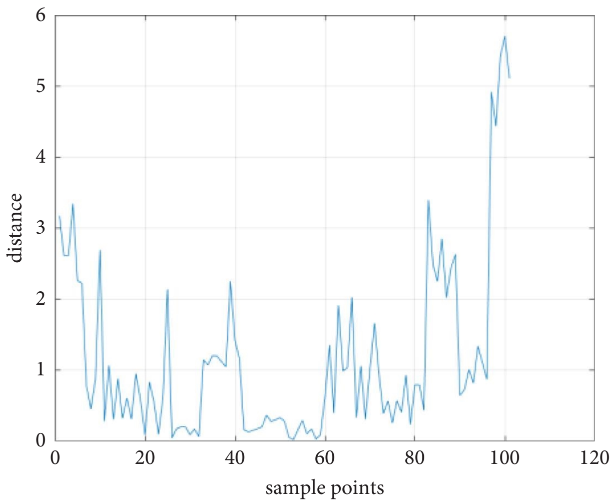


Figure 1: Mahalanobis distance.

It can be observed from Figure 1 that the values of the last five points are significantly higher than the other values so the last five spectra are judged as outliers.

2.4. Data Preprocessing. In data preprocessing, the data are imported first, and two classical methods, Savitzky–Golay smoothing (S-G) and Standard Normal Variate (SNV), are used to preprocess the near-infrared spectral data. The SG method for smoothing filtering improves the spectral smoothness and reduces noise interference; SNV is used to eliminate the influence of solid particle size, surface scattering, and optical path change on NIR diffuse reflectance spectra. The data were then preprocessed using the data standardization method. Standardize features by removing the mean and scaling to unit variance.

As shown in Figure 2, Figure 2(a) is the original spectra without outliers, Figure 2(b) is the spectra treated with SG and SNV, and Figure 2(c) is the spectra obtained after data standardization. Similarly, Figures 3 and 3(a) show the original spectra with outliers, Figure 3(b) is the spectra treated with SG and SNV, and Figure 3(c) is the spectra after data standardization.

2.5. Data Augmentation. Data augmentation is a common method for improving the image training of convolutional neural networks, which can be understood as simulating the changes of images, such as rotating the image 90° and zooming and shrinking the image. This change is easily understood by humans, but it confuses machine learning algorithms. By simulating various changes in the training set data, it may expand from a limited training set to generate training samples to prevent overfitting, which is suitable for small datasets. Data augmentation methods commonly used in the field of images are flipping, random rotation, scaling, clipping, shifting, and increasing Gaussian noise.

Data augmentation is also exceedingly significant for spectroscopy applications, where several translations of the spectrum may occur between measurements, such as frequency shifts, peak broadening, and intensity changes. We divided the samples into training and validation sets for this experiment. Among them, 80% of the samples were used as the training set, 20% of samples were used as the verification set, and data augmentation was used to increase the number of training sets. By randomly shifting the data by 0.1 times the mean, that is, to enlarge or reduce the mean by 0.1 times, and then shifting the slope by 0.05 times, that is, to randomly adjust the slope between 0.95 and 1.05 to increase the spectrum.

For the training set with outliers, this augmentation was repeated 15 times for each sample and an example output, and the sample size was expanded to 1200 samples. The other training set was enhanced 16 times to ensure the same sample size, and the sample size was expanded to 1216 samples.

2.6. One-Dimensional Convolutional Neural Networks. 1DCNNs are generally composed of input layers, convolutional layers, BN layers, fully connected layers, output layers, and other parts.

2.6.1. Convolutional Layer. The convolutional layer was composed of several convolution kernels. Using a convolution kernel to convolve the original data is equivalent to extracting the features of the original data that contain convolutional kernel features. The convolutional kernel size represents the size of each convolution of data, and the convolution kernel of size S has S weights. The convolution result is to multiply each data by the sum of weights, and the output result called feature mapping. Another parameter, the step size, represents the step size of the convolution kernel moving after a convolution. Figure 4 illustrates the execution process of the convolution kernel. The execution process of each convolution kernel is similar, only the weight changes. Set two convolution kernels, the input one-dimensional data size is 4, the convolution kernel size is 2, and the step size is 2. Formula (2) is the calculation method for the output size, where y is the output size, x is the input spectra, s is the convolution kernel size, and i is the step size.

$$y = \frac{x - s + 1}{i}. \tag{2}$$

2.6.2. Activation Layer. Convolutional neural networks can exploit different activation functions to express complex features. Each neuron accepts the previous layer of neurons as input and then passes the processed value to the next layer. In a multilayer neural network, there is an activation function for every pair of layers. The output of a neural network without an activation function is a linear combination of inputs with limited learning ability. Theoretically, a deep neural network with a nonlinear activation function can approximate any function, significantly improving the data-fitting ability of the neural network. Commonly used activation functions include the sigmoid, tanh, and ReLU

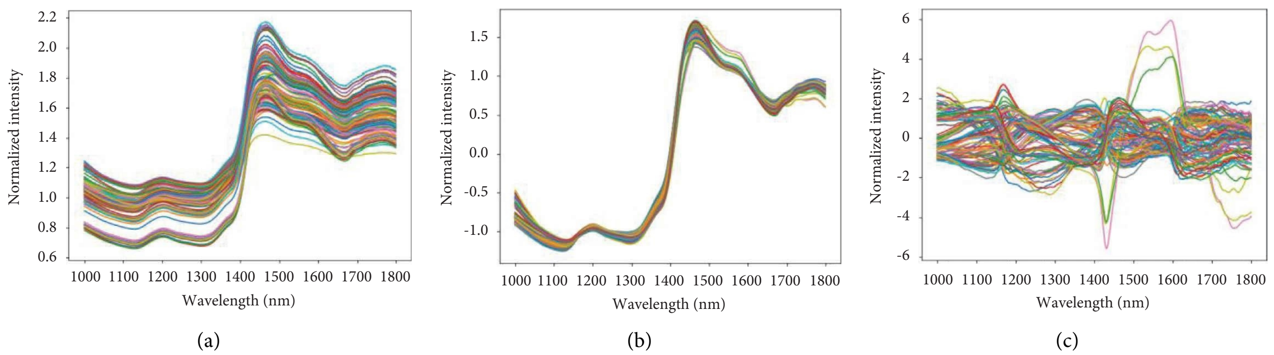


Figure 2: Normal spectra and the spectra after pretreatment of Antai pill samples: (a) original spectra, (b) spectra after SG and SNV treatment, and (c) spectra after data standardization.

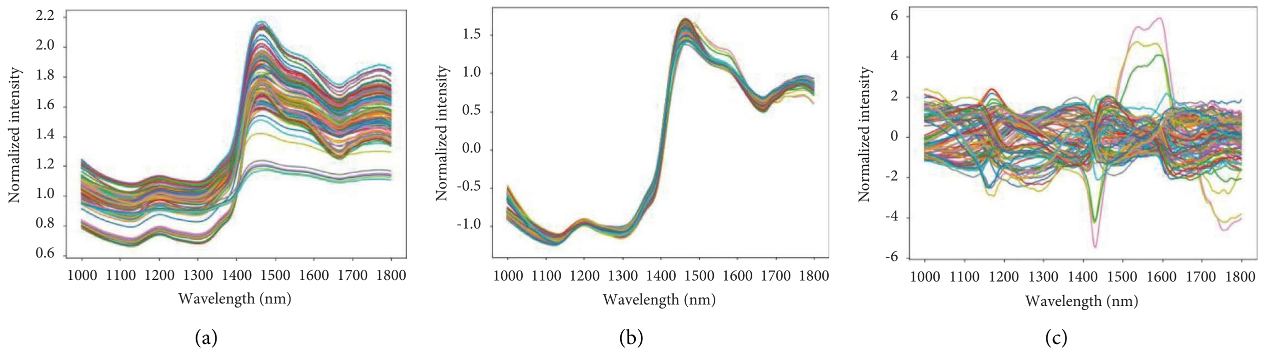


Figure 3: Original spectra with outliers and the spectra after pretreatment of Antai pill samples: (a) original spectra, (b) spectra after SG and SNV treatment, and (c) spectra after data standardization.

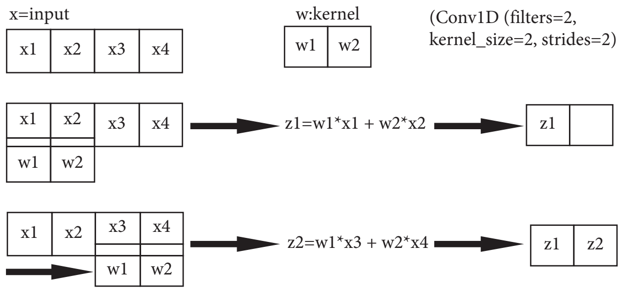


Figure 4: Convolutional layer execution process.

functions. The rectified linear unit (ReLU) function was used as the activation function in this experiment. A significant advantage of using the ReLU function is that it speeds up learning compared to sigmoid and tanh functions. The ReLU function sets all the negative data in the convoluted feature map to 0, and the non-negative numbers keep the gradient unchanged, alleviating the problem of gradient disappearance. In deep learning, ReLU is the most widely used activation function, and its formula is as follows:

$$y = \begin{cases} x, & x \geq 0, \\ 0, & x < 0. \end{cases} \tag{3}$$

2.6.3. BN Layers (Batch Normalization). Batch normalization allows us to use much higher learning rates and be less careful about initialization [27]. The BN layer tries to overcome the difficulty of model training caused by the deepening of neural network layers. Neural network structures are typically divided into input, output, and hidden layers. The hidden layer comprises all network layers between the input and output layers. When training a neural network, normalization is often used on the input data to improve the network training speed. For the hidden layer, it is necessary to use the BN layer to standardize the data passed by the previous layer to the current hidden layer, which maintains the input of each layer of the neural network the same distribution. The use of a BN layer often achieves better results.

2.6.4. Dropout Layer. The dropout layer temporarily drops the neural network unit of each fully connected layer randomly from the network, according to a certain probability during the training process of the deep neural network. Therefore, each batch trains a different network, which simplifies the structure of the neural network, thereby increasing the robustness of the network and reducing overfitting.

2.6.5. Flatten Layer. The flatten layer flattens the input data without affecting batch size and is usually followed by fully connected (FC) layers. Because multiple feature maps are output after the convolutional layer, these feature maps need to be converted into vector sequences to correspond to the FC layer.

2.6.6. Fully Connected Layer. Each node in the FC layer is connected to all the nodes in the previous layer to establish the mapping between extracted features and the output to play the role of a regression. The purpose of the convolutional and activation layers is to map the original data to the hidden layer, whereas the fully connected layer maps learned features to the sample label space.

2.6.7. Optimization. The training process of the neural network is the process of constantly updating the weight parameters, and the optimization algorithms are used to calculate this group of parameters. The neural network method is to initialize the weight parameters of each layer first and then calculate the loss function by forward calculating the output value of the network in the training process. If the loss is close to 0, the network is trained and no further weight update is required. Otherwise, the weight parameters are updated using back propagation. The best optimizer is selected for fast convergence and correct learning while adjusting the internal parameters to minimize the loss function. The commonly used optimization algorithms are SGD, Adam, AdaGrad, and RMSProp. This study uses a stochastic gradient descent (SGD) optimization algorithm. SGD randomly selects one sample at a time to update the parameters, which is fast. The SGD formula is as follows:

$$g_{SGD} = \nabla_\theta L\left(x^{(i)}, y^{(i)}, \theta_t\right), \tag{4}$$

$$\theta_{t+1} = \theta_t - \eta g_{SGD}, \tag{5}$$

θ represents the weight, t represents the t-th iteration, $L(\cdot)$ represents the loss function, ∇_θ denotes the partial derivative of loss function L, and η is the learning rate, which determines the amplitude of parameter change when updating the parameters. Equation (4) is the process of updating the weight parameters; each time selecting one sample to update the parameters can quickly update the gradient.

2.6.8. Huber Loss Function. The loss function is usually used as a learning criterion for optimization problems, and the distance between the predicted and real values is calculated using the loss function. When dealing with regression problems in neural networks, the average absolute error (MAE) or mean square error (MSE) is typically used. This study uses the Huber loss function to account for outliers, which is less sensitive than the MSE [28]. That is, compared with the MSE, it is more robust to outliers. It is based on the absolute error but becomes the mean square error when the error is small. It combines the advantages of mean absolute error (MAE) and mean-squared error (MSE). The formula for the Huber loss is as follows:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{for}|y - f(x)| \leq \delta, \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{otherwise.} \end{cases} \tag{6}$$

The use of Huber loss for training causes outliers to have a linear function, thus a much greater impact on the gradient. In the case that the sample is not an outlier, the function becomes a quadratic (this tolerance being the parameter, δ), at which point it essentially becomes the MSE [29]. Thus, it may potentially reach the minimum faster than MSE when handling outliers.

2.6.9. 1DCNN Modeling. 1DCNN training process of convolutional neural network consists of two stages. The first stage is the stage of data propagation, and the second stage is back propagation.

In the process of forward propagation, feature vectors are extracted from the input graphic data through multiple convolutional layers and transferred to the fully connected layer to obtain the recognition result. When the output result matches the expected value, the output result is generated. Otherwise, the back propagation process is performed. The error is calculated between the result and the expected value and then it is returned layer by layer to update the weight (see Algorithm 1).

The 1DCNN model structure proposed in this study is shown in Figure 5. It consists of 13 layers: an input layer, one Gaussian noise layer, one reshape layer, three convolutional layers, three batch normalization (BN) layers, one dropout layer, one flattening layer, one fully connected (FC) layer, and one output layer.

The parameters of the 1DCNN model structure are listed in Table 1. The unmarked parameters were the default parameters in TensorFlow. A brief description of each layer is as follows:

(1) The Gaussian noise layer. Regularization of the model is assisted by the influence of the Gaussian noise filter on the noise of the data, which is only effective in training. The standard deviation of the Gaussian noise is expressed by t.

(2) The reshape layer. For deep neural network, through a network layer to change the dimension of input data, NIRS data from two-dimensional adjustment to three-dimensional, the value of the third dimension is fixed to 1.

(3) The convolutional layer 1. Convolution is performed by one-dimensional convolution. Three 1D convolutional layers are used, and each layer with a ReLU activation. The number of convolution kernels is represented by k, the convolution kernel size by s, and the activation function by a. Convolutional layer 1 uses 8 convolution kernels, each of which is 32 in size.

```
INPUT: samples: number of samples
    epochs: the training times of all training samples.
    b: the number of samples selected in one training session.
Initialize(net)
for epoch = 1; epoch ≤ epochs; epoch++
    for size = 1; size ≤ math.ceil(samples/b); size++
        spectral data ⟵ uniformly random sample $b$ spectral data
        analytes ← uniformly random sample $b$ analytes
        $z$ ⟵ forward(net, spectral data)
        l ⟵ loss($z$, analytes)
        grad ⟵ backward(l)
        update(net, grad)
    end for
end for
```

Algorithm 1: train the one-dimensional convolutional neural network with stochastic gradient descent.

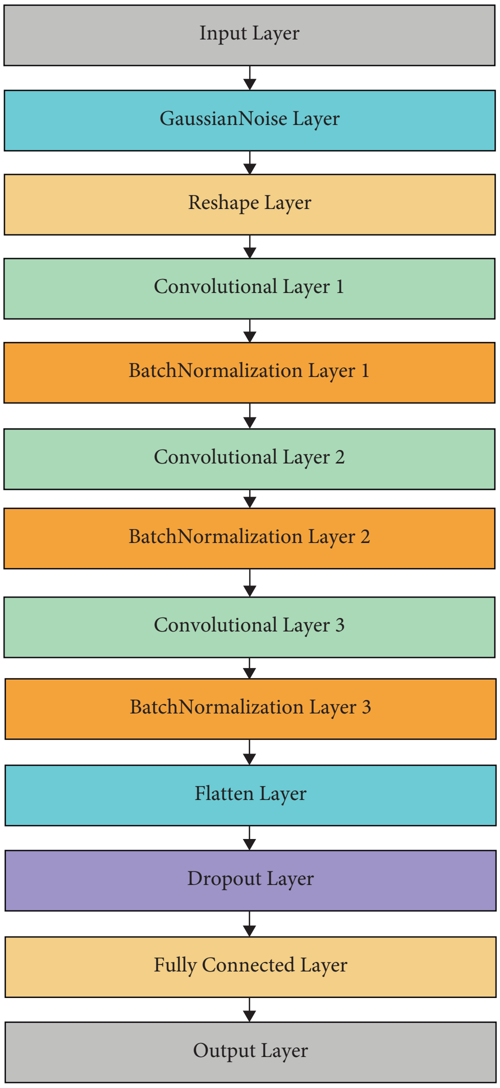
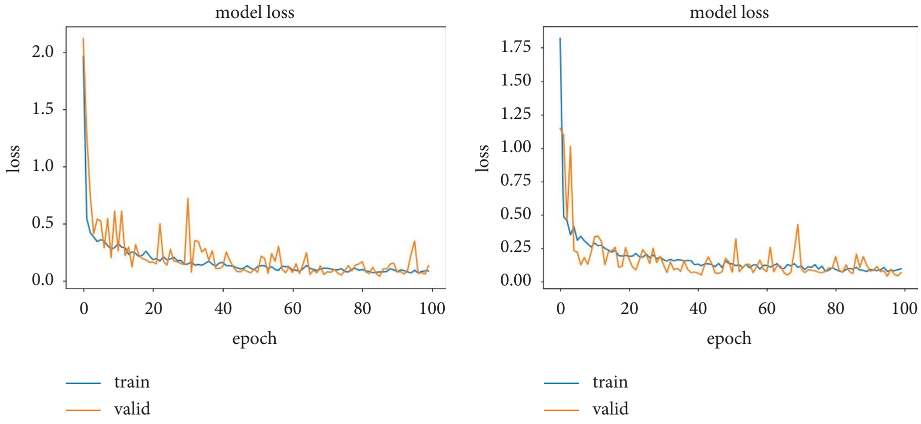


Figure 5: 1DCNN model structure.

Table 1: Network structure parameters of 1DCNN.

Network layer	Antai pills		
	Input	Output	Parameter
Gaussian noise	(None, 800)	(None, 800)	$t = 0.05$
Reshape	(None, 800)	(None, 800, 1)	
Conv1	(None, 800, 1)	(None, 769, 8)	$k = 8$, $s = 32$, $a =$ "relu"
BN1	(None, 769, 8)	(None, 769, 8)	
Conv2	(None, 769, 8)	(None, 738, 16)	$k = 16$, $s = 32$, $a =$ "relu"
BN2	(None, 738, 16)	(None, 738, 16)	
Conv3	(None, 738, 16)	(None, 707, 32)	$k = 32$, $s = 32$, $a =$ "relu"
BN3	(None, 707, 32)	(None, 707, 32)	
Flatten	(None, 22624)	(None, 22624)	
Dropout	(None, 34944)	(None, 34944)	$r = 0.5$
FC	(None, 128)	(None, 128)	$d = 128$, $a =$ "relu"
Output	(None, 1)	(None, 1)	$d = 1$, $a =$ "linear"



(a)

(b)

Figure 6: Loss curves of the training set and validation set: (a) loss curves of normal spectra, and (b) loss curves with outliers of spectra.

(4) The BN layer 1. After each convolution, the BN layer was used to normalize the output features to a mean value of 0 and variance of 1. Data standardization was realized, the training speed was improved, the convergence process was accelerated, and a large learning rate was allowed. BN layer 1 is the normalization of the data processed by convolutional layer 1.

(5) The convolutional layer 2. Convolutional layer 2 uses 16 convolution kernels, each of which is 32 in size.

(6) The BN layer 2. BN layer 2 is the normalization of the data processed by convolutional layer 2.

(7) The convolutional layer 3. Convolutional layer 3 uses 32 convolution kernels, each of which is 32 in size.

(8) The BN layer 3. BN layer 3 is the normalization of the data processed by convolutional layer 3.

(9) The flattening layer flattens the features extracted after convolution and readjusts the 3D input to the 2D data.

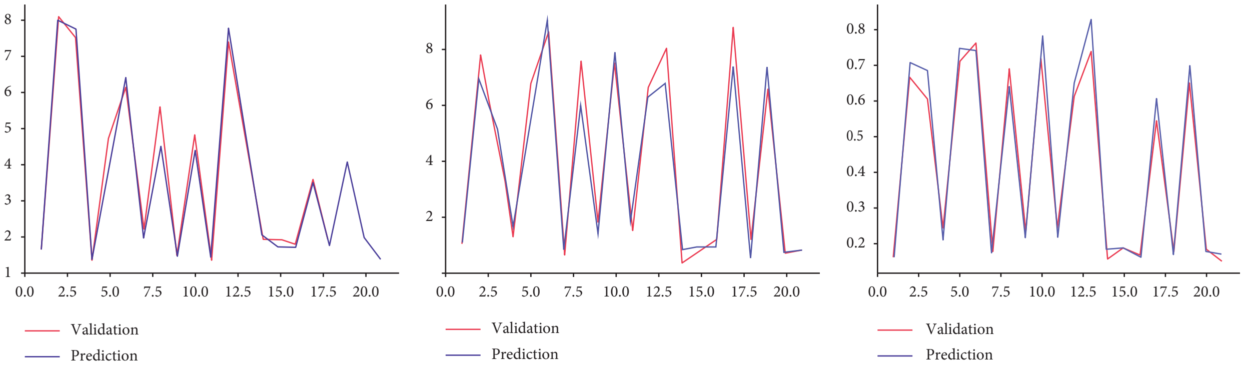
(10) The dropout layer. By randomly discarding neurons to improve the generalization ability of the model to prevent overfitting, the ratio rate of the input unit to be deleted is expressed by r.

(11) The FC layer is activated by the linear activation function and further compresses the nodes in the network. The spatial dimensions of the output are represented by d.

(12) The output layer maps the learned features to the sample markup space using the full connection layer of output node 1.

The preprocessed data were trained using a convolutional neural network with an SGD optimizer. The initial learning rates of the three chemical components were 0.01 (learning_rate = 0.01), 100 iterations (epoch = 100), and batch size was set to 16 (batch_size = 16).

Figure 6 shows the decline curve of loss of training set and validation set of analyte wogonoside in the training process of the 1DCNN model. a is the loss curve of normal spectra training; b is the loss curve with outlier training. Observe from the figure that in the loss curves, the loss of the



(a)

(b)

(c)

Figure 7: Real-predicted value curves: (a) real-predicted value curves of analyte wogonoside, (b) real-predicted value curves of analyte scutellarin, and (c) real-predicted value curves of analyte ferulic acid.

training set and the loss of the verification set have converged, the difference between the two is small, and the fit is successful.

3. Results and Discussion

3.1. Evaluating Indices. The evaluation indices are as follows:

3.1.1. Root Mean-Square Error (RMSE).

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\widehat{y}_i\right)^2}. \tag{7}$$

3.1.2. Determinant Coefficient (R^2).

$$R^2=1-\frac{\sum_{i=1}^{n}\left(y_i-\widehat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i-\overline{y}\right)^2}. \tag{8}$$

RMSE reflects the degree of deviation between the predicted and real values of the regression model and is sensitive to outliers. The smaller the RMSE value, the better is the accuracy of the prediction model for describing the experimental data. R^2 refers to the degree of fitting between predicted value and real value of the regression. If R^2 is close to 1, values can be accurately predicted, and the regression model fits better. Where n is the vector length, y_i and $\widehat{y}_i$ are the real and predicted values, respectively, and $\overline{y}$ is the average value of the real value.

3.2. Comparison between 1DCNN Model and Classical Regression Method. In this study, the 1DCNN model was trained for concentration prediction of three different analytes using Huber error loss function and 10-fold cross validation. The 1DCNN model predicted real-predicted curves for the three analytes without outliers which are shown in Figure 7. Figures 7(a)-7(c) represent the prediction results for wogonoside, scutellarin, and ferulic acid, respectively. The closer the two lines overlap, the smaller the prediction deviation. The results showed that the R^2 of above 0.965 was obtained for all analytes.

Tables 2 and 3 list the results of the 1DCNN model with and without outliers compared to the classical regression methods of PLSR and SVR. PLSR and SVR use similar preprocessing methods for data and the 1DCNN model. PLSR and SVR algorithms were implemented using the scikit-learn library in Python. The PLSR algorithm determines the optimal number of principal components through cross validation on training data. Finally, the number of principal components was ten, and the other parameters were used by the default parameters of the PLS regression method of the scikit-learn library. The SVR algorithm selects the Gaussian kernel function (kernel = "rbf"), penalty factor C was 1.0, and other parameters used the default parameters of the scikit-learn library SVR method. R_v^2 and RMSECV are the correction determination coefficient and correction root mean-squared error after the 10-fold cross validation of the calibration set, and R_p^2 and RMSEP are prediction decision coefficient and root mean-square error of the prediction set, respectively. As it can be seen from Tables 2 and 3, the prediction accuracy of the 1DCNN model is greatly improved for all analytes. When there were no outliers, RMSEP is reduced to 0.2807, 0.7129, and 0.0453, respectively. R_p^2 value is, respectively, increased to 0.9845, 0.9489, and 0.9663. This 1DCNN model applied here exhibits promising regression capabilities compared with PLSR and SVR models.

The 1DCNN model can also perform well when there are outliers. Taking analyte scutellarin as an example, compared with inliers, the PLSR model showed that R_v^2 decreased by 0.0222 and RMSECV increased by 0.0645; the SVR model showed a 0.0132 decrease in R_v^2 and 0.0531 increase in RMSECV; while the 1DCNN model showed a 0.0025 decrease in R_v^2 and 0.0382 increase in RMSECV. These results indicate that the 1DCNN model has high inclusiveness for outliers of spectra and maintains high prediction accuracy for a small number of outliers, demonstrating the sound performance of the 1DCNN model.

Table 2: Comparison of evaluation indices of the 1DCNN model, PLSR model, and SVR model with normal spectra.

Analyte	Model	R_v^2	RMSECV	R_p^2	RMSEP
Analyte 1	PLSR	0.9340	0.5568	0.9491	0.5088
	SVR	0.9520	0.4816	0.9667	0.4117
	1DCNN	0.9683	0.3397	0.9845	0.2807
Analyte 2	PLSR	0.9027	1.0077	0.8924	1.0346
	SVR	0.9137	0.9209	0.9421	0.7591
	1DCNN	0.9780	0.4950	0.9489	0.7129
Analyte3	PLSR	0.8834	0.0883	0.9056	0.0758
	SVR	0.9240	0.0707	0.8703	0.0888
	1DCNN	0.9659	0.0469	0.9663	0.0453

Table 3: Comparison of evaluation indices of the 1DCNN model, PLSR model, and SVR model with spectra with outliers.

Analyte	Model	R_v^2	RMSECV	R_p^2	RMSEP
Analyte 1	PLSR	0.9342	0.5534	0.9310	0.5591
	SVR	0.9420	0.5119	0.9793	0.3062
	1DCNN	0.9692	0.3283	0.9788	0.3100
Analyte 2	PLSR	0.8805	1.0722	0.9105	0.9423
	SVR	0.9005	0.9740	0.9516	0.6927
	1DCNN	0.9755	0.5332	0.9632	0.6042
Analyte 3	PLSR	0.8769	0.0894	0.8980	0.0804
	SVR	0.9091	0.0738	0.9496	0.0565
	1DCNN	0.9633	0.0616	0.9712	0.0427

4. Conclusions

This study proposes using one-dimensional convolution neural networks to process near-infrared spectral data, and the quantitative analysis technology of chemical composition was explored with Antai pills as the research object. We drew the following conclusions:

(1) Because the small number of prediction samples, it is easy to have the problem of overfitting or weak generalization ability, the data augmentation strategy is adopted to increase the sample size, and the data augmentation is realized by 0.1 times the mean of the random offset and 0.05 times the slope of the random offset. This method replicates the systematic errors in the spectral method and is suitable for training convolutional neural networks.

(2) The experimental results show that the performance of the 1DCNN method is good, and the prediction accuracy is superior to classical regression methods. It is feasible to quantitatively analyze the chemical composition of drugs using near-infrared spectroscopy combined with convolutional neural networks, which is suitable for large-scale, multi-variety, and multi-manufacturer drug tasks.

(3) The 1DCNN model maintains excellent performance with a few outliers, whereas the traditional regression algorithm is not as good as 1DCNN. This model provides a new approach to addressing the problem of spectra with outliers.

In the next study, massive near-infrared spectra will be used to build a more broaden and robust model. Meanwhile, although CNN has little concern on preprocessing and is time-saving, the parameters in network will be manually adjusted. The following novelty is to find a solution to optimize parameters automatically and widen the application of CNN in drug quality management.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that they have no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant no. 62031021) and Guangzhou Science and Technology Planning Project (20180310104).

References

[1] F. Yanchun and H. Changqin, “The application progress of near infrared technology in the field of drug circulation in China,” *Spectroscopy and Spectral Analysis*, vol. 34, no. 05, pp. 1222–1228, 2014.

[2] X. M. Liang, Y. Jin, Y. P. Wang, G. W. Jin, Q. Fu, and Y. S. Xiao, “Qualitative and quantitative analysis in quality control of traditional Chinese medicines,” *Journal of Chromatography A*, vol. 1216, no. 11, pp. 2033–2044, 2009.

[3] K. Y. Lam, G. K. Chan, G. Z. Xin et al., "Authentication of cordyceps sinensis by DNA analyses: comparison of ITS sequence analysis and RAPD-derived molecular markers," *Molecules*, vol. 20, no. 12, pp. 22454–22462, 2015.

[4] M. C. Sarraguca, P. R. Ribeiro, A. O. Santos, M. C. Silva, and J. A. Lopes, "A PAT approach for the on-line monitoring of pharmaceutical co-crystals formation with near infrared spectroscopy," *International Journal of Pharmaceutics*, vol. 471, no. 1-2, pp. 478–484, 2014.

[5] T. B. Bagchi, S. Sharma, and K. Chattopadhyay, "Development of NIRS models to predict protein and amylose content of brown rice and proximate compositions of rice bran," *Food Chemistry*, vol. 191, pp. 21–27, 2016.

[6] H. Yang, B. Hu, X. Pan et al., "Deep belief network-based drug identification using near infrared spectroscopy," *Journal of Innovative Optical Health Sciences*, vol. 10, no. 02, Article ID 1630011, 2017.

[7] L. Yin, J. Zhou, D. Chen et al., "A review of the application of near-infrared spectroscopy to rare traditional Chinese medicine," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 221, p. 117208, 2019.

[8] X. Yan, S. Zhang, H. Fu, and H. Qu, "Combining convolutional neural networks and on-line Raman spectroscopy for monitoring the Cornu Caprae Hircus hydrolysis process," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 226, p. 117589, 2020.

[9] T. Lindeberg, "Scale invariant feature transform," *Scholarpedia*, vol. 7, no. 5, p. 10491, 2012.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, vol. 1, pp. 886–893, 2005.

[11] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[13] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021.

[14] J. Acquarelli, T. van Laarhoven, J. Gerretzen, T. N. Tran, L. M. Buydens, and E. Marchiori, "Convolutional neural networks for vibrational spectroscopic data analysis," *Analytica Chimica Acta*, vol. 954, pp. 22–31, 2017.

[15] J. Liu, M. Osadchy, L. Ashton, M. Foster, C. J. Solomon, and S. J. Gibson, "Deep convolutional neural networks for Raman spectrum recognition: a unified solution," *Analyst*, vol. 142, no. 21, pp. 4067–4074, 2017.

[16] J. L. Dong, M. J. Hong, Y. Xu, and X. Q. Zheng, "A practical convolutional neural network model for discriminating Raman spectra of human and animal blood," *Journal of Chemometrics*, vol. 33, no. 11, p. e3184, 2019.

[17] F. Hu, M. R. Zhou, P. C. Yan et al., "Identification of mine water inrush using laser-induced fluorescence spectroscopy combined with one-dimensional convolutional neural network," *RSC Advances*, vol. 9, no. 14, pp. 7673–7679, 2019.

[18] S. Malek, F. Melgani, and Y. Bazi, "One-dimensional convolutional neural networks for spectroscopic signal regression," *Journal of Chemometrics*, vol. 32, no. 5, p. e2977, 2018.

[19] C. Yuanyuan and W. Zhibin, "Quantitative analysis modeling of infrared spectroscopy based on ensemble convolutional neural networks," *Chemometrics and Intelligent Laboratory Systems*, vol. 181, pp. 1–10, 2018.

[20] X. Zhang, T. Lin, J. Xu, X. Luo, and Y. Ying, "DeepSpectra: an end-to-end deep learning approach for quantitative spectral analysis," *Analytica Chimica Acta*, vol. 1058, pp. 48–57, 2019.

[21] C. H. Cui and T. Fearn, "Modern practical convolutional neural networks for multivariate regression: applications to NIR calibration," *Chemometrics and Intelligent Laboratory Systems*, vol. 182, pp. 9–20, 2018.

[22] C. Ni, D. Wang, and Y. Tao, "Variable weighted convolutional neural network for the nitrogen content quantization of Masson pine seedling leaves with near-infrared spectroscopy," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 209, pp. 32–39, 2019.

[23] Y. Y. Chen and Z. B. Wang, "End-to-end quantitative analysis modeling of near-infrared spectroscopy based on convolutional neural network," *Journal of Chemometrics*, vol. 33, no. 5, p. e3122, 2019.

[24] S. Wold, M. Sjöström, and L. Eriksson, "PLS-regression: a basic tool of chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 58, no. 2, pp. 109–130, 2001.

[25] U. Thissen, M. Pepers, B. Üstün, W. Melssen, and L. Buydens, "Comparing support vector machines to PLS for spectral regression applications," *Chemometrics and Intelligent Laboratory Systems*, vol. 73, no. 2, pp. 169–179, 2004.

[26] J. Ma, X. Wang, X. Xue, P. Yin, and F. Ge, "Near infrared nondestructive detection of the content of key quality control index components in Antai Pills," *Modernization of Traditional Chinese Medicine and Materia Medica-World Science and Technology*, vol. 20, no. 05, pp. 651–659, 2018.

[27] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International conference on machine learning*, pp. 448–456, 2015.

[28] J. J. Faraway, "Linear models with R," *Texts in Statistical Science Series*, vol. 63, p. 229, 2005.

[29] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.