Mathematical Biosciences and Engineering

MBE, 20(8): 14395–14413.
DOI: 10.3934/mbe.2023644
Received: 04 May 2023
Revised: 04 June 2023
Accepted: 11 June 2023
Published: 30 June 2023

http://www.aimspress.com/journal/MBE

Research article

Dose-effect relationship analysis of TCM based on deep Boltzmann machine and partial least squares

Wangping Xiong[1,2], Yimin Zhu[1], Qingxia Zeng[1], Jianqiang Du[1], Kaiqi Wang[1], Jigen Luo[1], Ming Yang[1,2,]* and Xian Zhou[1,]*

[1] School of Computer, Jiangxi University of Chinese Medicine, Nanchang 330004, China
[2] Key Laboratory of Modern Preparations Chinese Medicine, Jiangxi University of Chinese Medicine, Nanchang 330004, China

* **Correspondence:** Email: liumore@jxutcm.edu.cn; 20030731@jxutcm.edu.cn.

Abstract: A dose-effect relationship analysis of traditional Chinese Medicine (TCM) is crucial to the modernization of TCM. However, due to the complex and nonlinear nature of TCM data, such as multicollinearity, it can be challenging to conduct a dose-effect relationship analysis. Partial least squares can be applied to multicollinearity data, but its internally extracted principal components cannot adequately express the nonlinear characteristics of TCM data. To address this issue, this paper proposes an analytical model based on a deep Boltzmann machine (DBM) and partial least squares. The model uses the DBM to extract nonlinear features from the feature space, replaces the components in partial least squares, and performs a multiple linear regression. Ultimately, this model is suitable for analyzing the dose-effect relationship of TCM. The model was evaluated using experimental data from Ma Xing Shi Gan Decoction and datasets from the UCI Machine Learning Repository. The experimental results demonstrate that the prediction accuracy of the model based on the DBM and partial least squares method is on average 10% higher than that of existing methods.

Keywords: deep Boltzmann machine; deep learning; partial least squares; traditional Chinese medicine; drug dose-effect relationships

1. Introduction

Traditional medicine has a rich history spanning thousands of years, resulting in a vast and complex collection of data, including numerous laws and experiences. However, manually inducing insights from this data may not accurately reflect its scientific, objective, and comprehensive nature. Therefore, finding ways to leverage the benefits of technology in the modern era to excavate and discover insights from this data has become a pressing issue [1,2]. Traditional Chinese Medicine (TCM) is considered a treasure, and is known for its unique approach of using multiple herb components to target multiple aspects of diseases. However, conducting TCM research presents significant data-related challenges and analytical difficulties that must be carefully considered and explored [3,4].

In TCM research, experimental data often involves multiple dependent and independent variables, as well as nonlinearity, due to the individualized treatment approaches in TCM. As a result, there may be issues with multicollinearity among the data variables. Because of the small sample size and the nonlinear distribution of the characteristics of the medicinal properties and efficacy of TCM, the application of small-sample nonlinear data sets is very common [5]. At the same time, noise and outliers in TCM data also puts the robustness of the model to the test due to errors in instrumentation, operational errors during data collection, etc. [6]. Therefore, the analysis and mining of TCM data require higher technical tools and a more comprehensive view of the data; the rapid development of machine learning and artificial intelligence technologies provide us with powerful tools [7–9]. Using these tools, the potential patterns in TCM data can be better mined and discovered, providing a more scientific basis and support for the clinical application of TCM, as well as speeding up the TCM research and development (R&D) process and reducing costs and risks.

In such a context, the dose-effect relationship analysis of TCM using machine learning methods has become one of the current hot spots and difficulties in the field of TCM research [10]. In order to analyze the dose-effect relationship of TCM, overcoming the multicollinearity among variables and fully expressing the nonlinear characteristics of TCM data are the keys to obtaining more accurate analysis results [11–13].

Partial least squares (PLS) is a widely adopted multivariate statistical analysis method that enables linear regression modeling of multiple dependent variables on multiple independent variables [14–16]. It is particularly useful in cases where there is multicollinearity among the variables or where the number of samples is less than the number of variables. PLS can effectively address these challenges and provide reliable results in such situations. However, the principal component extraction of PLS is a linear dimensionality reduction method, which cannot fully express the nonlinear characteristics of TCM data.

Traditional methods for reducing the dimensionality of nonlinear data include kernel feature mapping (KFM), kernel independent component analysis (KICA), and kernel principal component analysis (KPCA). Qin et al. [17] proposed a novel approach that combines intrinsic dimension estimation with PLS to map the data into a high-dimensional linear space using kernel functions. This method has the advantage of avoiding the need to select principal components and can effectively capture the nonlinear features in the data. However, choosing the appropriate kernel functions can be challenging.

With the development of deep learning, neural networks have shown great advantages in nonlinear feature extraction, and many scholars have started to incorporate neural networks into PLS. Zhou [18] proposed to combine a fuzzy neural network with PLS to achieve nonlinear feature representation while avoiding the problem of principal component selection. However, the results of this model may be sensitive to the choice of the affiliation function. Zhu et al. [19] combined a restricted Boltzmann machine with a PLS method (RBM-PLS), thus avoiding the problem of principal component selection; however, the results may be sensitive to the choice of initial values. Xiong et al. [20] combined a deep confidence network with a PLS method (DBN-PLS) from the original data to extract nonlinear features and avoid the need to select the number of principal components. However, inappropriate parameter selection may cause the model to converge to a locally optimal solution.

The current research focuses on establishing a suitable model for the dose-effect analysis of TCM and improving the accuracy of the dose-effect analysis [21–23]. In this paper, we propose a model suitable for analyzing small sample nonlinear data sets based on deep Boltzmann machine (DBM) and PLS (DBM-PLS) for the dose-effect relationship analysis model of TCM. By combining the nonlinear feature representation capability of the DBM and the data dimensionality reduction capability of PLS, DBM-PLS can more fully exploit the nonlinear features in the data. This method has a strong nonlinear modeling capability and high prediction accuracy, which helps to understand and analyze the dose-effect relationship of TCM more accurately and further promotes the research and development of the dose-effect relationship of TCM.

The rest of this paper is organized as follows. In Section 2, the DBM-PLS model architecture and its underlying methodology are described in detail. The experimental data set description and preprocessing methods are given in Section 3. The evaluation criteria and model configurations are given in Section 4, and the results of the comparison of different models are analyzed. Finally, conclusions are drawn in Section 5.

2. Materials and methods

2.1. PLS

PLS combines a principal component analysis (PCA), a canonical correlation analysis (CCA), and a multiple linear regression (MLR), and is a regression modeling method of multiple dependent variables on multiple independent variables [14–16]. Compared with an ordinary least squares regression, PLS is particularly effective when there is either multicollinearity among variables or the number of samples is less than the number of variables, and it is easy to explain the regression coefficients of each variable [24].

The basic idea of PLS is to extract the principal components from the independent variable X and the dependent variable Y to reflect the combined variable information of the original variables [25]. Specifically, PLS first extracts the first principal components t_1 and u_1 in the independent and dependent variables such that the principal component variances $Var(t_1) \rightarrow max$, $Var(u_1) \rightarrow max$, and the correlation $Var(u_1) \rightarrow max$. Next, a linear regression model of the independent and dependent variable components is built using t_1 and u_1, and the residual matrix is calculated. The residual matrix is then subjected to a principal component extraction to obtain the second principal components t_2 and u_2, and the above steps are iteratively repeated. The number of extracted principal components is determined by cross-validation, and the final PLS regression model is established [26].

2.2. DBM

The DBM is a deep learning model, constructed based on the restricted Boltzmann machine

(RBM) [27,28]. In essence, DBM is a special kind of neural network that forms abstract high-level feature representations by combining the underlying features [29].

Specifically, the DBM can be viewed as a stack of multiple RBMs. Each RBM consists of a visual layer and a hidden layer, where the neurons in the visual layer are used for input data and the neurons in the hidden layer are used for feature learning [30]. It is worth noting that neurons between the same layers are not connected, while neurons between different layers are fully connected and independent of each other [28,29]. To construct a DBM model for feature extraction, we assume N hidden layers, with v and h representing the visual layer neurons and hidden layer neurons, respectively. The model construction involves randomly initializing the parameters θ of each RBM and unsupervised pre-training of the first RBM using the contrast scattering algorithm to obtain its parameters. Then, the hidden layer of each RBM is utilized as the visual layer of the next layer, and the pre-training of the subsequent RBM continues. This process is repeated until the last layer, and the parameters of the entire DBM are obtained. Following pre-training, the DBM is fine-tuned using the back-propagation method. Figure 1 depicts the structure of the DBM model with N hidden layers.

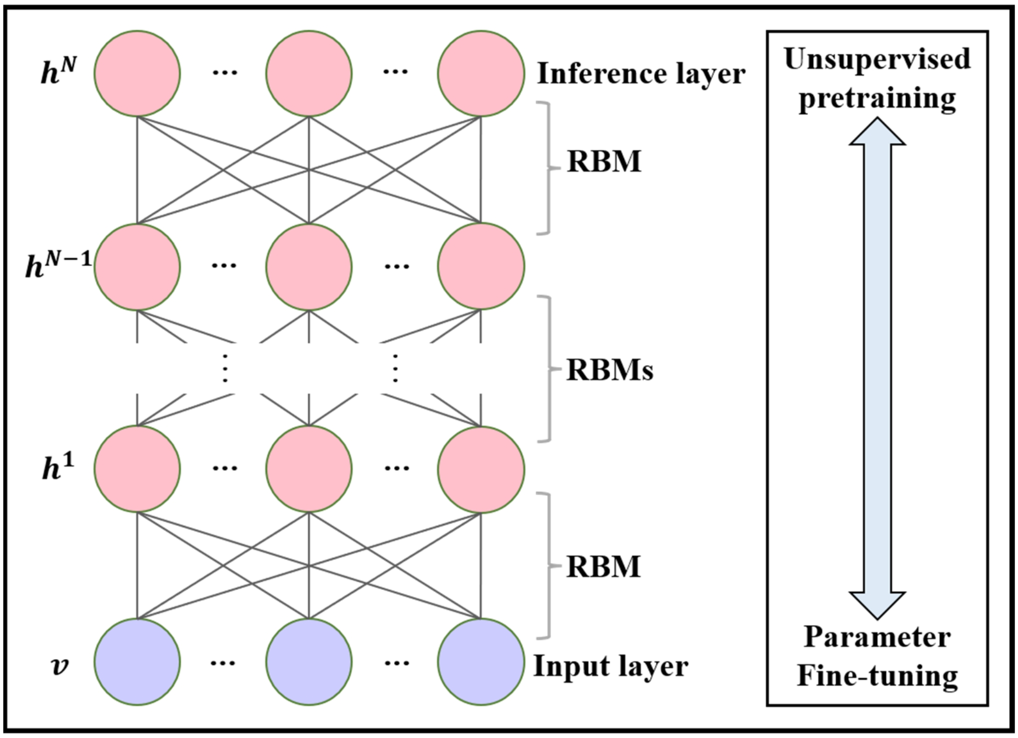

Figure 1. Structure diagram of DBM with N hidden layers.

Since the DBM consists of a stack of multiple RBMs, the main process of training the DBM is to train the RBM [31]. Let n be the number of neurons in the visual layer and m be the number of neurons in the hidden layer. Denote the state vector of neurons in the visual layer as v and the state vector of neurons in the hidden layer as h. Equation (1) represents the energy of the RBM for a given set of state vectors:

$$E(v,h|\,\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{i=1}^{n}\sum_{j=1}^{m} \omega_{ij} h_j v_i - \sum_{j=1}^{m} b_j\, h_j, \tag{1}$$

where the model parameters are $\theta = \{a_i, b_j, \omega_{ij}\}$, a_i is the bias of neuron i in the visual layer, b_j is the bias of neuron j in the hidden layer, and w_{ij} is the weight between neuron i in the visual layer and neuron j in the hidden layer. Using the energy function, we can express the joint probability distribution of v and h as Eq (2):

$$p(v, h|\,\theta) = \frac{e^{-E}(v, h|\theta)}{Z(\theta)}, \tag{2}$$

where $Z(\theta) = \sum_{v,h} e^{-E}(v, h|\theta)$ is the normalization factor. Eqs (3) and (4) describe the conditional probability of each visual layer neuron and hidden layer neuron being activated, based on the energy function:

$$p(v_i = 1|h) = sigmoid\left(a_i + \sum_j w_{ij}h_j\right), \tag{3}$$

$$p(h_i = 1|v) = sigmoid\left(b_j + \sum_i w_{ij}v_i\right), \tag{4}$$

where $sigmoid(x) = \frac{1}{1+e^{-x}}$.

When the DBM is trained and the parameters $\theta = \{a_i, b_j, \omega_{ij}\}$ are known, the original data is preprocessed to obtain $V = (v_1, v_2, \dots, v_n)$, which maps the visual layer data to the hidden layer data $H = (h_1, h_2, \dots, h_m)$, thus converting the original data into another nonlinear representation, and further using the hidden layer data $H = (h_1, h_2, \dots, h_m)$ as an input to form a deeper nonlinear representation of the data by propagating it layer by layer [32].

2.3. Model construction based on DBM and PLS

When using PLS for multivariate regression modeling, it is common to first extract the principal components of the dataset using PCA to reduce the dimensionality of the variables and eliminate redundant information [26]. However, PCA can only extract linear features in the dataset and cannot adequately represent nonlinear features, such as features common in TCM datasets.

To overcome this problem, DBM can be used to extract new features from the dataset. Feature extraction by DBM can be divided into two phases: pre-training and parameter tuning. In the pre-training phase, the RBM and the basic structure of the DBM are trained on the data, layer by layer. The first step of the pre-training process is to input the normalized dataset into the underlying RBM and train it with predefined parameters (denoted as θ) to obtain the first layer of the feature representation of the dataset (denoted as h_1). Then, h_1 is used as the input to the next layer of RBM, which is again trained with predefined parameters to obtain the more abstract feature representation h_2. Through continuous iterations, the feature representations of all layers (denoted as h_N) can be obtained, where h_N integrates different features abstracted from each layer of RBM. These features are deeper and more complex nonlinear representations of the original data, which can more fully represent the nonlinear features in the TCM dataset.

In the parameter tuning phase, the parameters of the DBM are tuned and optimized to obtain an improved performance. A nonlinear representation of the features is obtained by training and combined with CCA and MLR in PLS to build the prediction model. A cross-validation approach is used to evaluate the predictive power of the model, and model selection and parameter tuning are performed to obtain the best predictive performance.

In summary, by combining DBM and PLS methods, nonlinear features in TCM datasets can be better handled to obtain more accurate prediction models. The DBM-PLS method proposed in this paper is based on this theory, using DBM to extract nonlinear features in TCM datasets and combining PLS for regression modeling. Meanwhile, the deep learning model is optimized by a backpropagation algorithm to further improve the predictive capability of the model. Finally, the validation test data are used to evaluate the performance and prediction accuracy of the model.

In this paper, we study a multivariate data set involving an independent variable X and a dependent variable Y, where X is a matrix with n rows and p columns and Y is a matrix with n rows and q columns.

The specific construction process of the model is as follows:

1) Data preprocessing: the independent variable X and the dependent variable Y are normalized to obtain $E_0 = (e_1, e_2, \dots e_p)$ and $F_0 = (f_1, f_2, \dots f_q)$.

2) The independent variable E_0 undergoes DBM processing with the following steps:

A. In DBM, the size of the visual layer, denoted by the variable p, is determined based on the independent variable E_0, which consists of p components $(e_1, e_2, \dots e_p)$. The primary objective of this layer is to reduce the dimensionality of the input features. To achieve this, the number of neurons in the hidden layer, denoted by p_1, is usually set to be smaller than that of the visual layer (i.e., $p_1 < p$). This helps to extract and represent the essential information from the input in a more compact form.

B. The initialization parameters $\theta = \{a_i, b_j, w_{ij}\}$ in DBM are randomly assigned. The biases of the visual and hidden layers are represented by vectors $a = \{a_1, a_2, \dots, a_p\}$ and $b = \{b_1, b_2, \dots, b_p\}$, respectively. The weight matrix $W = \{w_{ij} | 0 \leq i \leq p, 0 \leq j \leq p_1\}$ connects the visual and hidden layers and contains weights w_{ij}. Random initialization of these parameters ensures that the DBM starts with a diverse range of values, allowing it to explore and learn a wide variety of features and representations.

C. The DBM takes the independent variable $E_0 = (e_1, e_2, \dots e_p)$ as an input to the visual layer. The DBM is trained layer by layer using an RBM. In this training process, the hidden layer of the previous RBM becomes the visual layer of the next RBM, and the output of the previous RBM is used as the input of the next RBM, continuing until the final layer is reached. This layer-wise training approach helps to avoid the problem of vanishing gradients and enables the DBM to learn complex hierarchical representations of the input data.

D. The final layer of the DBM, denoted by $T = \{t_1, t_2, \dots, t_k\}$, produces k principal components that are extracted from the independent variables. These principal components can then be used in PLS analysis to model the relationship between the input variables and the output variables. By using the principal components extracted from the DBM as inputs for PLS, we can obtain a more compact and informative representation of the input data, which can improve the accuracy and interpretability of the PLS model.

3) The principal components extracted by the DBM in step 2 are used as inputs to the PLS external model, and an MLR analysis is performed jointly with F_0. The coefficients generated from this analysis are used to construct an MLR equation for Y relative to X.

4) We determine whether the model accuracy is appropriate by evaluating four metrics, namely root mean square error (RMSE), mean absolute error (MAE), mean squared percentage error (MSPE), and coefficient of determination (R^2). If the model accuracy is appropriate, the algorithm will be terminated. Otherwise, the hyperparameters of the model are adjusted and the process is iterated until the accuracy requirements are met. This iterative approach allows the model to improve

its performance over time and reach the appropriate level of accuracy.

Algorithm 1 outlines the two components of the proposed DBM-PLS model: nonlinear feature extraction using DBM and regression using PLS. The DBM is capable of automatically extracting nonlinear features from the original data using its interlayer transfer property. Then, these features are used for regression using PLS. The proposed model combines the benefits of PLS, which can address issues such as multicollinearity in TCM data, while also addressing the problem of PLS's limited ability to capture the nonlinear characteristics of TCM data.

Algorithm 1. The model algorithm based on DBM and PLS

```
Input: TCM experimental data set ($D$);
Output: DBM-PLS equation.
Step 1: Normalized preprocessing of D to obtain ($E_0, F_0$)
Step 2: DBM processing
    Initialization of model parameter $\theta = \{a_i, b_j, w_{ij}\}$
    $Layersize = 1$
    $hidden\_layers\_sizes = [8,4,8]$
    Set a threshold for precision and accuracy conditions for the RBM loop
    While the precision condition is not met for the number of RBM layer sizes
      While the accuracy condition is not met for the number $Layersize$ of neurons in each layer
        For each layer z in $Layersize$
          Calculate the probability of the hidden layer neurons being activated, $P_z(h_i|v_i)$
          Extract a sample using Gibbs sampling: $v_{i+1} \sim P_z(v_{i+1}|h_i)$
          Using the activated hidden layer neurons probability $P_z(v_{i+1}|h_i)$, reconstruct the visual layer.
          Extract a sample using Gibbs sampling: $v_{i+1} \sim P_z(v_{i+1}|h_i)$
          Calculate the probability of activating hidden layer neurons $P_z(h_{i+1}|v_{i+1})$ using the sample $v_{i+1}$ extracted through Gibbs sampling.
          Update the weights: $b_i \leftarrow b_i + \lambda(v_i - v_{i+1})$
Step 3: PLS external regression model
    Extract eigenvalues $T = sigmoid(E_o \times w^t + a^t)$ from DBM model
    In the PLS external model, $T$ is used as input to the MLR and the regression analysis is performed jointly with $F_0$ to obtain the standardized regression coefficients
    Standardize regression coefficients and de-normalize to obtain the DBM-PLS equation
Step 4: End
```

Furthermore, the proposed model obtains the regression equations of the original variables through the use of CCA and MLR. The model is validated using test data and hyperparameters are adjusted to improve its performance. The overall objective of the DBM-PLS model is to adapt to small sample nonlinear data sets and to improve the accuracy of predicting quantitative-effective relationships in TCM. Figure 2 depicts the overall structure of the DBM-PLS model.

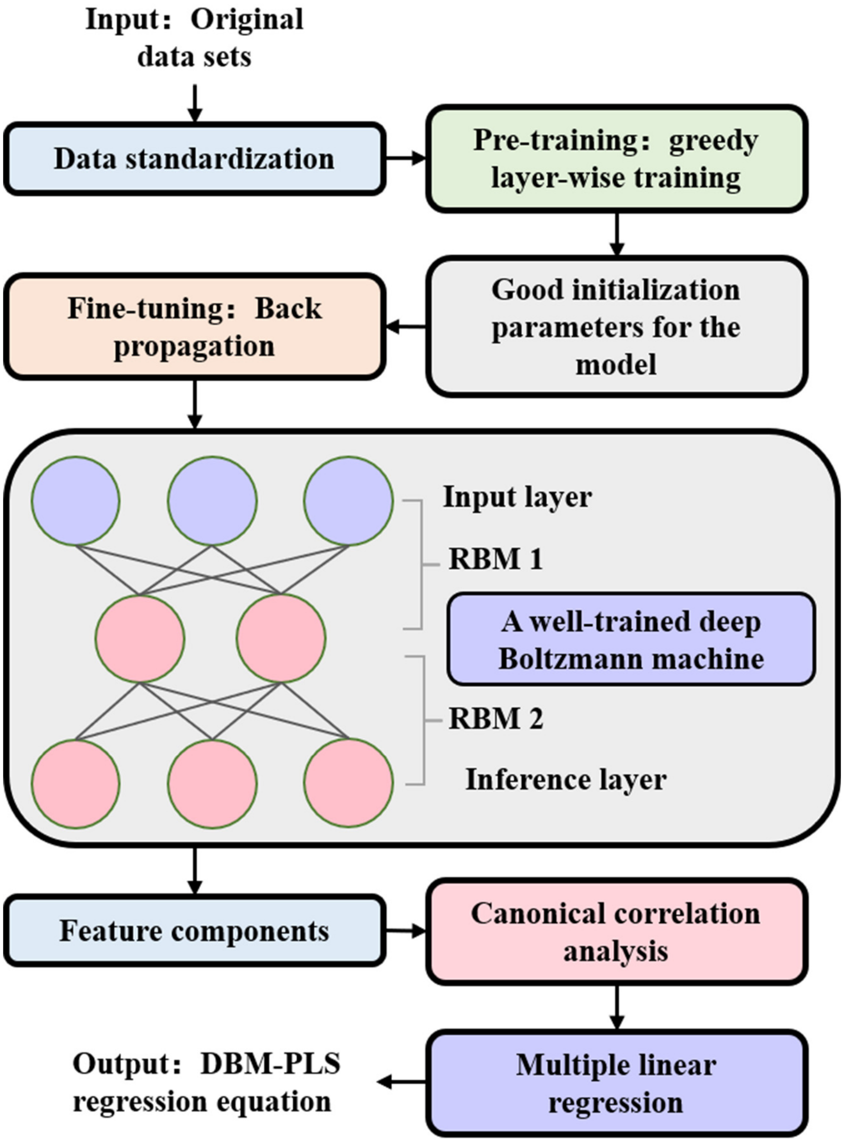
Input：Original data sets
Data standardization
Pre-training：greedy layer-wise training
Good initialization parameters for the model
Fine-tuning：Back propagation
Input layer
RBM 1
A well-trained deep Boltzmann machine
RBM 2
Inference layer
Feature components
Canonical correlation analysis
Multiple linear regression
Output：DBM-PLS regression equation

Figure 2. Overall structure of DBM-PLS model.

3. Experimental data and pre-processing

3.1. Description of experimental data

This study utilized experimental data primarily sourced from Ma Xing Shi Gan Decoction (MXD) trials for relieving asthma (MXD-R) and defervescence (MXD-D) conducted in the key laboratory of the Jiangxi University of Chinese Medicine. Additionally, the Parkinson's telemonitoring (PT) and physicochemical properties of protein tertiary structure (PPPTS) datasets from the UCI standard dataset [33,34] were incorporated. Table 1 provides detailed information and characteristics of each dataset.

In the experiment of the dose-effect relationship study of MXD, the effects of the active ingredients of 13 different doses of drugs in rat plasma on pharmacological indices were investigated. Each group of 10 rats were treated with a certain concentration of the drug ratios of MXD. The blood concentration and pharmacological indexes of each rat were measured, and the average blood concentration and average pharmacological indexes of each group of 10 rats were used as the final experimental data. The main active components in rat plasma were used as independent variables, namely Ephedrine (E), Pseudoephedrine (PE), Methylephedrine (ME), Amygdalin (A), Prunasin (P), Liquiritin (L), Liquiritigenin (LG), and Glycyrrhetinic Acid (GA). The dependent variables were the pharmacological indices examined, in the case of the data from the relieve asthma experiment, the asthma incubation period (AIP) and the duration of asthma (DA), and in the data from the defervescence experiment, the Prostaglandin E2 (PGE2), TRI temperature index, and the 6-h fever suppression rate. Some of the experimental data are shown in Table 2.

The PT and PPPTS datasets were selected from the UCI standard dataset for the medium sample dataset and the large sample dataset, respectively. For a detailed description, please refer to the website: https://archive.ics.uci.edu/ml/index.php.

Table 1. Data set description.

Data set name	Number of samples	Number of independent variables	Number of dependent variables
MXD-R	13	8	2
MXD-D	13	8	3
PT	5875	16	2
PPPTS	45730	9	1

Table 2. The experimental data set of MXD for relieving asthma.

Blood Concentrations (ng/ml)								Pharmacological indicators	
E	PE	ME	A	P	L	LG	GA	AIP (s)	DA (min)
25.4	26.9	8.76	92.28	93.35	3.59	0.47	3.97	64.2	8.05
18.3	25.7	2.02	9.97	35.45	32.22	5.46	32.21	89.78	7.9
33	69.9	4.64	0.39	0.27	46.06	7.03	48.48	62.88	7.48
18.9	35.5	1.07	1.04	0.66	15.31	5.34	14.83	54.43	7.37
…	…	…	…	…	…	…	…	…	…

3.2. Data standardization preprocessing

Data standardization is a necessary task when data mining. There may be order-of-magnitude differences in different datasets that can affect the results of the data analysis [35,36]. In order to eliminate data bias and to improve data consistency, the above statistics are standardized. The raw data were standardized using the Z-score method, which is also known as standard deviation standardization, and ensures that all indicators are on the same scale. The Z-score method used in this study ensures that the processed data have a mean of 0 and a variance of 1. The transformation function is represented by Eq (5):

$$x^* = \frac{x-\mu}{\sigma} \tag{5}$$

where $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$ denotes the mean value of each feature in data X and $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}$, denotes the standard deviation of each feature in data X.

Experimental data on the efficacy of MXD in relieving asthma were standardized, and the standardized results are shown in Table 3.

Table 3. Standardization processing results.

Blood concentrations								Pharmacological indicators	
E	PE	ME	A	P	L	LG	GA	AIP	DA
−0.616	−0.661	−0.561	1.743	3.115	−1.005	−1.223	−0.962	−0.619	0.454
−0.632	−0.664	−0.655	−0.330	0.812	1.196	0.914	0.984	1.466	0.231
−0.599	−0.546	−0.619	−0.571	−0.587	2.260	1.586	2.105	−0.727	−0.403
−0.631	−0.638	−0.668	−0.555	−0.572	−0.104	0.861	−0.214	−1.415	−0.575
…	…	…	…	…	…	…	…	…	…

4. Experiment and result analysis

4.1. Evaluation criteria and model configuration

To validate the feasibility and effectiveness of the DBM-PLS model, we compared it with a Lasso regression, a XGBoost regression, the original PLS method, the PLS with fused restricted Boltzmann machines (RBM-PLS) method, and the PLS with fused deep belief network (DBN-PLS) method. To obtain more accurate estimates of model performance, we trained and tested each method using the same data set. We evaluated these models by examining the RMSE, MAE, MSPE, and R^2. These four evaluation criteria can be represented by Eqs (6)–(9):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{6}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{7}$$

$$MSPE = \frac{100\%}{N}\sum_{i=1}^{N}(\frac{y_i - \hat{y}_i}{y_i})^2 \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{9}$$

where N is the number of test samples and y_i and $\hat{y}_i$ are the observed and predicted values, respectively.

In the subsequent experiments, we first divided the four datasets into a training set and a test set, where the test set accounts for 20% of the total. Second, we used the grid search cross-validation method on the training set for model training and fine-tuning, and we used RMSE as the evaluation function of grid search. Finally, we used the test set to evaluate the model performance.

The structure of the DBM-PLS and DBN-PLS models used in the experiments are set as follows. We set the number of layers of DBM and DBN to 3. The pre-training phase uses the contrastive divergence algorithm and the fine-tuning phase uses the BP algorithm. For different datasets, we used a grid search to determine the number of nodes in the hidden layer (num_hidden1 and num_hidden2), the number of pre-training iterations (pretrain_epochs), the pre-training learning rate (pretrain_lr), the number of fine-tuning iterations (train_epochs), and the fine-tuning learning rate (train_lr). To prevent model overfitting, we added L2 regularization to control the complexity of the model to avoid the overfitting problem, and the size of the L2 regularization term (alpha) is also determined by the grid search.

The other models in the experiment are set up as follows. We implemented Lasso and PLS regression models using Scikit-learn and determine the size of the regularization term (alpha) for Lasso regression and the number of principal components (n_components) for PLS by grid search. We implemented the XGBoost model using xgboost, and we determined the number of base learners (n_estimators) and the maximum depth of the tree (max_depth) by grid search. We added L2 regularization to avoid model overfitting, and the alpha size is also determined by grid search We used TensorFlow to implement the RBM-PLS model, and used the contrastive divergence algorithm for the pre-training and fine-tuning phases. We used a grid search to determine the sizes of num_hidden1, pretrain_epochs, pretrain_lr, train_epochs, train_lr, and L2 regularization terms (alpha) of the RBM. The parameter grids for all models, the specific parameters after grid search, and the cross-validation folds (cv) on different data sets are shown in Tables 4 and 5.

Table 4. Model parameters in the experiment.

Model	Data name	n_components	max_depth	n_estimators	alpha	cv
Parametric Grid	MXD-R	arrange (1, 9, 1)	arrange (2, 9, 1)	arange (10, 200, 1)	[0, 0.1, 1]	
	MXD-D	arrange (1, 9, 1)	arange (2, 9, 1)	arange (10, 200, 1)	[0, 0.1, 1]	
	PT	arrange (1, 17, 1)	arange (2, 11, 1)	arange (10, 400, 1)	[0, 0.1, 1]	
	PPPTS	arange (1, 10, 1)	arange (2, 11, 1)	arange (10, 400, 1)	[0, 0.1, 1]	
Lasso	MXD-R				1	10
	MXD-D				1	10
	PT				0.1	5
	PPPTS				0.1	5
XGBoost	MXD-R		3	50	0.1	10
	MXD-D		5	100	0.1	10
	PT		5	50	1	5
	PPPTS		7	200	0.1	5
PLS	MXD-R	1				10
	MXD-D	1				10
	PT	7				5
	PPPTS	9				5

Table 5. Model parameters in the experiment.

Model	Data name	num_ hidden1	num_ hidden2	pretrain_ epochs	pretrain_ lr	train_ epochs	train_ lr	alpha	cv
Paramet-ric Grid	MXD-R	arange (2, 9, 1)	arange (2, 9, 1)	arange (10,1000,1)	[0.01,0.1]	arange (10,200,1)	[0.01,0.1]	[0.01,0.1,1]	
	MXD-D	arange (2, 9, 1)	arange (2, 9, 1)	arange (10,1000,1)	[0.01,0.1]	arange (10,200,1)	[0.01,0.1]	[0.01,0.1,1]	
	PT	arange (2,17,1)	arange (2,17,1)	arange (10,1000,1)	[0.01,0.1]	arange (10,200,1)	[0.01,0.1]	[0.01,0.1,1]	
	PPPTS	arange (2,10,1)	arange (2,10,1)	arange (10,1000,1)	[0.01,0.1]	arange (10,200,1)	[0.01,0.1]	[0.01,0.1,1]	
RBM-PLS	MXD-R	8		265	0.1	90	0.1	0.1	10
	MXD-D	8		160	0.1	105	0.1	0.1	10
	PT	16		676	0.1	78	0.1	0.01	5
	PPPTS	9		788	0.1	93	0.1	0.01	5
DBN-PLS	MXD-R	5	8	104	0.1	31	0.1	0.1	10
	MXD-D	4	8	139	0.1	23	0.1	0.1	10
	PT	12	16	724	0.1	101	0.1	0.01	5
	PPPTS	6	9	847	0.1	114	0.1	0.01	5
DBM-PLS	MXD-R	7	8	218	0.1	61	0.1	0.1	10
	MXD-D	6	8	197	0.1	53	0.1	0.1	10
	PT	14	16	571	0.1	122	0.1	0.01	5
	PPPTS	7	9	821	0.1	89	0.1	0.01	5

4.2. Analysis of experimental results

During the experiments, we used four sets of experimental data to compare and validate the models. We used the grid search cross-validation method to search for the best parameters of the model for model optimization on each data set and compared the effects of the six methods on each data set. The experimental results are shown in Table 6.

Based on the experimental results in Table 6, we can observe that the DBM-PLS model overall outperforms the other models on the four data sets. The effect is more significant when the data feature space is nonlinearly structured and there is multicollinearity among the features. The specific analysis is as follows:

1) Compared to the non-PLS model, the DBM-PLS showed an average improvement of 58.01% in RMSE, 56.51% in MAE, 54.44% in MSPE, and 238.58% in R^2 compared to the Lasso regression and XGBoost. The Lasso regression tends to select one of the relevant features and ignore other

highly relevant features. It may either lead to oversimplification of the model or loss of important information and is relatively weak in handling data with multicollinearity. The XGBoost model is based on the integration of decision trees, each of which has its own rules and paths, leading to complexity in the interpretation of the overall model. In contrast, DBM-PLS combines the nonlinear feature extraction capability of DBM and the data dimensionality reduction capability of PLS to better capture complex patterns and associations in the data, while reducing redundant features and improving the model interpretation.

2) Compared to the PLS model, DBM-PLS showed an average improvement of 23.48% in RMSE, 16.38% in MAE, 1.54% in MSPE, and 54.79% in R^2 compared to PLS. Because PLS is a linear regression method, when there are complex nonlinear relationships in the data, PLS may not be able to capture these patterns, resulting in a degraded model performance. In contrast, the DBM-PLS model makes use of the powerful nonlinear feature extraction capability of DBM to extract nonlinear features from the feature space, which makes up for the weak nonlinear modeling capability of the PLS model.

Table 6. Comparison of experimental results.

Data name	Evaluation criterion	Lasso	XGBoost	PLS	RBM-PLS	DBN-PLS	DBM-PLS
MXD-R	RMSE	8.6891	8.0023	0.8446	0.8169	0.7757	0.6796
	MAE	5.1385	5.9201	0.5277	0.6042	0.4514	0.4196
	MSPE (%)	1.826	2.48	0.168	0.1302	0.1501	0.0882
	R^2	0.2189	0.1748	0.1904	0.5584	0.7654	0.8468
MXD-D	RMSE	4.6809	8.4201	0.8186	0.8371	0.6878	0.6245
	MAE	2.7462	5.4861	0.5528	0.531	0.4571	0.3861
	MSPE (%)	5.9886	40.7278	0.6717	0.5529	0.7271	0.2069
	R^2	0.2344	0.0068	0.5154	0.7461	0.8491	0.9004
PT	RMSE	9.5013	7.8411	12.6896	13.7871	10.7623	9.6851
	MAE	7.8078	6.1914	8.4432	9.704	7.7634	7.2998
	MSPE (%)	44.3284	29.3454	33.8978	41.9103	43.0811	36.305
	R^2	0.2776	0.2912	0.9081	0.8806	0.9263	0.9471
PPPTS	RMSE	5.1781	4.3577	1.1657	1.0154	0.9671	0.8854
	MAE	4.3562	3.1418	1.0809	0.9863	0.8573	0.7612
	MSPE (%)	80.0358	94.1851	33.4995	32.2665	28.8689	30.5812
	R^2	0.2778	0.4958	0.5477	0.6102	0.5853	0.6517

3) Comparing the improved PLS model, DBM-PLS showed an average improvement of 18.91% in RMSE, 15.98% in MAE, 9.01% in MSPE, and 13.36% in R^2 compared with RBM-PLS and DBN-PLS. All three models use neural networks to extract nonlinear features to make up for the weak nonlinear modeling ability of PLS. RBM-PLS uses RBM to extract nonlinear features; RBM is a shallow model, and the nonlinear feature extraction ability of a single RBM is relatively weak and

cannot extract the high-level features of the data. DBN-PLS uses DBN to extract nonlinear features and can extract high-level features of the data. Compared with DBN, each layer of DBM can learn feature representations of data independently, though DBN can only learn feature representations between neighboring layers. Therefore, DBM can learn higher-level and more abstract feature representations. This enables DBM to better capture nonlinear and complex relationships in the data and improve the effectiveness of feature extraction.

Combining the performance of the DBM-PLS model on each data set, the DBM-PLS model outperforms the other five models in general, indicating that the DBM-PLS model has the best adaptability to data with nonlinearity and the presence of multicollinearity. This is because the DBM-PLS model uses DBM to extract nonlinear features in the high-dimensional feature space to capture nonlinear relationships and complex patterns in the data. At the same time, it combines the data dimensionality reduction capability of PLS to reduce the dimensionality of the feature space and eliminate the influence of multicollinearity among features, thus improving the predictive power and interpretability of the model.

To visualize the performance of the DBM-PLS model, we plotted bar charts reflecting RMSE, MAE, MSPE, and R^2. Since the RMSE, MAE, and MSPE magnitudes are different for each data set, the results are mapped to the same magnitude for comparison purposes, as shown in Figure 3.

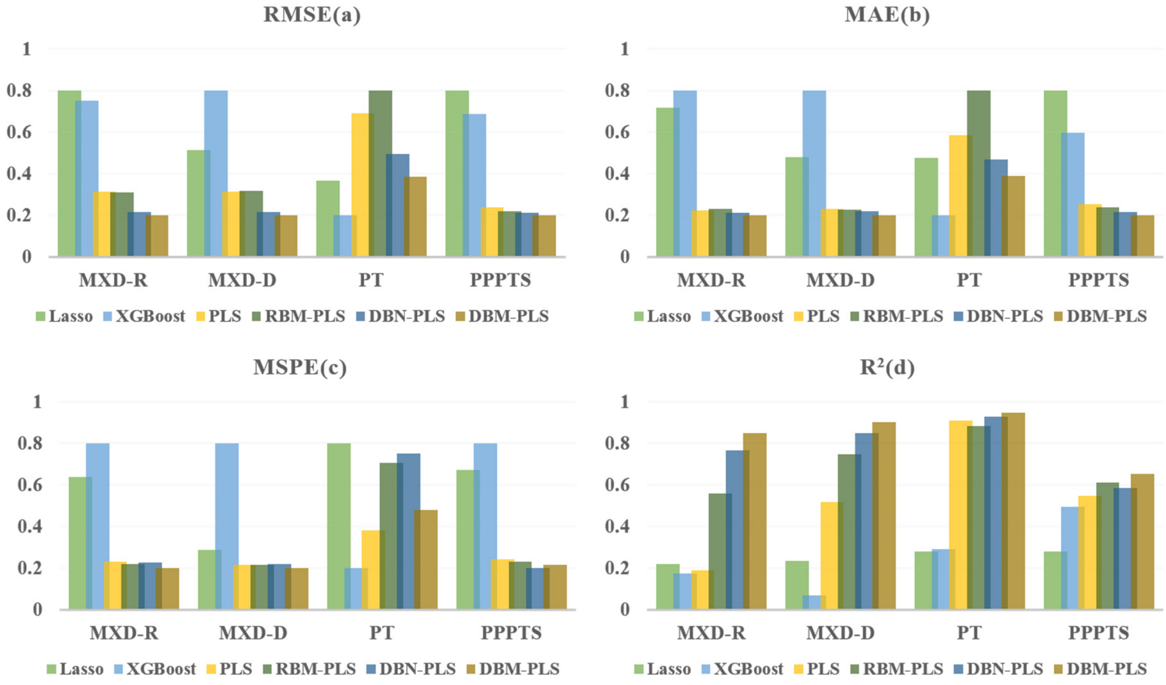

Figure 3. Model evaluation visualization results. (a) Comparison of RMSE of six models. (b) Comparison of MAE of six models. (c) Comparison of MSPE of six models. (d) Comparison of R^2 of six models.

As can be seen in Figure 3, the DBM-PLS model overall outperforms the other models overall on the four data sets. This is attributed to the powerful nonlinear modeling capability of DBM and the data dimensionality reduction capability of PLS. The deep structure of DBM enables it to learn deeper abstract feature representations. PLS can effectively reduce the data dimensionality and maintain the important features of the data after dimensionality reduction to retain the main information and solve the multicollinearity problem.

To examine the learning process of the DBM-PLS model, Figure 4 shows the descent process of RMSE on the MXD-R and MXD-D datasets, where the horizontal axis indicates the training epoch and the vertical axis indicates the RMSE. As can be seen in the figure, the RMSE drops sharply at the beginning and then gradually slows down to a lower minimum value of the loss function.

In summary, the DBM-PLS model shows good adaptability in dealing with small sample nonlinear data sets, can effectively capture nonlinear features in TCM data, and can reduce the effects of multicollinearity among features. Therefore, the DBM-PLS model can help us understand and analyze the dose-effect relationship of TCM more accurately and provide strong support for TCM research.

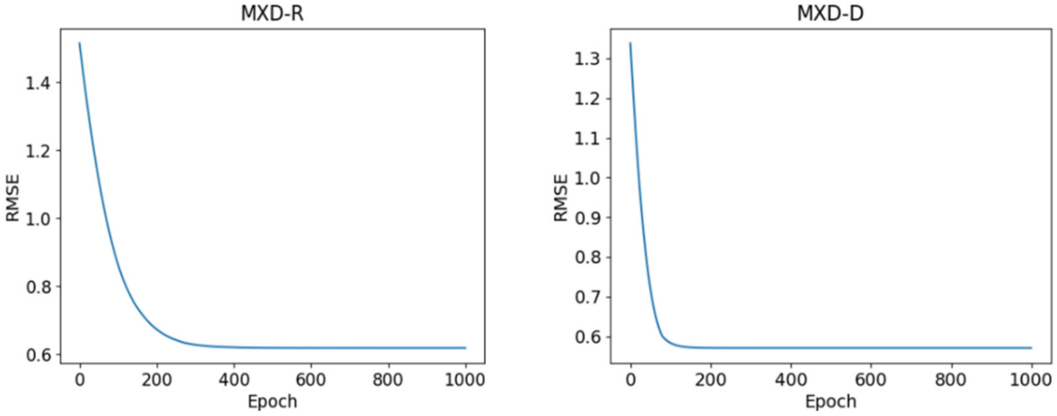

Figure 4. Descent of RMSE in DBM-PLS learning process.

5. Conclusions

Machine learning-based computational methods have many advantages in the dose-effect prediction of TCM, which can support and help in clinical drug use and TCM R&D. There are four main advantages.

1) Dose optimization: Traditional trial methods require significant time and resources to test the effects of various herbal doses. In contrast, machine learning-based computational methods can quickly screen a large number of dose combinations, thereby reducing trial costs and time.

2) Dose-effect law mining: Machine learning models can handle large-scale Chinese medicine data sets, including information on chemical composition, pharmacological effects, and clinical trial results of herbs. By analyzing these multidimensional data, the potential patterns and interactions of herbal dosage effects can be revealed.

3) Individualized prediction: Machine learning models can predict and evaluate the effects of herbal medicines at different doses based on individual characteristics and drug data. This provides valuable information to support individualized herbal medicine use and helps doctors and researchers make more accurate decisions.

4) Reduce trial and error costs: Machine learning models can provide accurate predictions and assessments of herbal dose effects, thereby reducing the number and cost of trials and clinical trials.

By screening out promising dose combinations in advance, the risk of unnecessary trials and R&D failures can be reduced.

This study presents a novel model based on DBM and PLS for analyzing dose-effect relationships in TCM. The proposed model integrates the strengths of DBM in nonlinear feature extraction and PLS in data dimensionality reduction. This is particularly suitable for small sample nonlinear datasets in TCM and has a high capability for nonlinear modeling and prediction accuracy. To validate the model's performance, we conducted experiments and obtained significant results. The experimental results demonstrate that the DBM-PLS model outperforms existing methods with an average prediction accuracy improvement of 10%.

Despite its effectiveness, the DBM-PLS model still faces some challenges that requires further consideration. The first issue is model selection, which involves determining the optimal number of layers, neurons per layer, learning rate, momentum, and number of epochs. These parameters have a significant impact on the performance of the model. We used an automated grid search hyperparameter optimization technique to search the parameter space to find the best combination. This approach can impose a large burden on computational resources, especially when the parameter space is large or multiple iterations of experiments are required. Therefore, for the dose-effect relationship analysis of TCM, some more efficient and intelligent model selection methods need to be developed to search for the optimal model. Secondly, in practical dose-effect relationship analysis of TCM, noise, and outliers in TCM data need to be addressed to improve data quality. Therefore, pre-processing methods need to be developed to enhance the robustness and prediction accuracy of the model. By addressing these challenges, the DBM-PLS model can be further improved and used effectively in the analysis of TCM dose-effect relationships.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (82274680, 82160955) and the university-level research team of Jiangxi University of Traditional Chinese Medicine for the innovation team of Chinese medicine preparation technology and equipment (CXTD22006).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. D. B. Singh, R. K. Pathak, D. Rai, From traditional herbal medicine to rational drug discovery: strategies, challenges, and future perspectives, *Rev. Bras. Farmacogn.*, **32** (2022), 147–159. https://doi.org/10.1007/s43450-022-00235-z

2. J. Yang, Y. Li, Q. Liu, L. Li, A. Feng, T. Wang, et al., Brief introduction of medical database and data mining technology in big data era, *J. Evidence-Based Med.*, **13** (2020), 57–69. https://doi.org/10.1111/jebm.12373
3. D. Ma, S. Wang, Y. Shi, S. Ni, M. Tang, A. Xu, The development of traditional Chinese medicine, *J. Tradit. Chin. Med. Sci.*, **8** (2021), 1–9. https://doi.org/10.1016/j.jtcms.2021.11.002
4. J. Sun, X. Meng, Application of big data technology in extracting information analysis of traditional Chinese medicine, *J. Phys.: Conf. Ser.*, **1881** (2021), 042050. https://doi.org/10.1088/1742-6596/1881/4/042050
5. X. Chu, B. Sun, Q. Huang, S. Peng, Y. Zhou, Y. Zhang, Quantitative knowledge presentation models of traditional Chinese medicine (TCM): A review, *Artif. Intell. Med.*, **103** (2020), 101810. https://doi.org/10.1016/j.artmed.2020.101810
6. H. Jiang, Y. Zhang, Z. Liu, X. Wang, J. He, H. Jin, Advanced applications of mass spectrometry imaging technology in quality control and safety assessments of traditional Chinese medicines, *J. Ethnopharmacol.*, **284** (2022), 114760. https://doi.org/10.1016/j.jep.2021.114760
7. L. Gan, X. Yin, J. Huang, B. Jia, Transcranial Doppler analysis based on computer and artificial intelligence for acute cerebrovascular disease, *Math. Biosci. Eng.*, **20** (2023), 1695–1715. https://doi.org/10.3934/mbe.2023077
8. X. Wang, X. Zhang, J. Li, B. Hu, J. Zhang, W. Zhang, et al., Analysis of prescription medication rules of traditional Chinese medicine for bradyarrhythmia treatment based on data mining, *Medicine*, **101** (2022), 31436. https://doi.org/10.1097/md.0000000000031436
9. Y. Yang, Y. Huang, L. Yang, H. Liu, Design of TCM research demand system based on data mining technology, in *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)*, (2023), 1–5. https://doi.org/10.1109/ICICACS57338.2023.10099881
10. J. Chen, J. Xu, P. Huang, Y. Luo, Y. Shi, P. Ma, The potential applications of traditional Chinese medicine in Parkinson's disease: A new opportunity, *Biomed. Pharmacother.*, **149** (2022), 112866. https://doi.org/10.1016/j.biopha.2022.112866
11. Y. Liu, T. Geng, Z. Wan, Q. Lu, X. Zhang, Z. Qiu, et al., Associations of serum folate and vitamin B12 levels with cardiovascular disease mortality among patients with type 2 diabetes, *JAMA Netw. Open*, **5** (2022), 2146124. https://doi.org/10.1001/jamanetworkopen.2021.46124
12. Y. Liu, B. Li, Y. Su, R. Zhao, P. Song, H. Li, et al., Potential activity of Traditional Chinese Medicine against Ulcerative colitis: A review, *J. Ethnopharmacol.*, **289** (2022), 115084. https://doi.org/10.1016/j.jep.2022.115084
13. M. E. McNamara, M. Zisser, C. G. Beevers, J. Shumake, Not just "big" data: Importance of sample size, measurement error, and uninformative predictors for developing prognostic models for digital interventions, *Behav. Res. Ther.*, **153** (2022), 104086. https://doi.org/10.1016/j.brat.2022.104086
14. H. Abdi, Lynne. J. Williams, Partial least squares methods: Partial least squares correlation and partial least square regression, *Comput. Toxicol.*, **930** (2012). https://doi.org/10.1007/978-1-62703-059-5_23
15. F. Schuberth, M. E. Rademaker, J. Henseler, Assessing the overall fit of composite models estimated by partial least squares path modeling, *Eur. J. Mark.*, **57** (2022), 1678–1702. https://doi.org/10.1108/EJM-08-2020-0586

16. Z. Shang, Y. Dong, M. Li, Z. Li, Robust feature selection and classification algorithm based on partial least squares regression, *J. Comput. Appl.*, **37** (2017), 871–875.
17. J. Qin, X. Yu, P. Zhang, M. Yang, An optimal band selection method for hyperspectral imagery based on kernel partial least squares, *J. Geomatics Sci. Technol.*, **30** (2013), 172–176.
18. L. Zhou, Research on Feature Extraction Method Based on Monlinear Partial Least Squares, Master's thesis, Nanjing University of Science and Technology in Nanjing, 2011.
19. Z. Zhu, J. Du, R. Yu, B. Nie, Partial least squares optimization method integrating restricted Boltzmann machine, *Comput. Eng.*, **43** (2017), 193–197.
20. W. Xiong, T. Li, Q. Zeng, J. Du, B. Nie, C. Chen, et al., Research on partial least squares method based on deep confidence network in traditional Chinese medicine, *Discrete Dyn. Nat. Soc.*, **2020** (2020), 4142824. https://doi.org/10.1155/2020/4142824
21. W. Dai, K. Feng, X. Sun, L. Xu, S. Wu, K. Rahmand, et al., Natural products for the treatment of stress-induced depression: Pharmacology, mechanism and traditional use, *J. Ethnopharmacol.*, **285** (2022), 114692. https://doi.org/10.1016/j.jep.2021.114692
22. Y. Fei, H. Cao, R. Xia, Q. Chai, C. Liang, Y. Feng, et al., Methodological challenges in design and conduct of randomised controlled trials in acupuncture, *BMJ*, **376** (2022), 064345. https://doi.org/10.1136/bmj-2021-064345
23. J. Yan, C. Peng, P. Chen, W. Zhang, C. Jiang, S. Sang, et al., In-vitro anti-Helicobacter pylori activity and preliminary mechanism of action of Canarium album Raeusch. fruit extracts, *J. Ethnopharmacol.*, **283** (2022), 114578. https://doi.org/10.1016/j.jep.2021.114578
24. Z. Xie, X. Feng, X. Chen, Partial least trimmed squares regression, *Chemom. Intell. Lab. Syst.*, **221** (2022), 104486. https://doi.org/10.1016/j.chemolab.2021.104486
25. Q. Zeng, *Research and Application of improved PLS in Traditional Chinese Medicine Data*, Master's thesis, Jiangxi University of Chinese Medicine in Nanchang, 2019. https://doi.org/10.27180/d.cnki.gjxzc.2019.000050
26. T. Li, *Study on Partial Least Squares Variable Screening Method for Chinese Dosage Effectiveness Data*, Master's thesis, Jiangxi University of Chinese Medicine in Nanchang, 2021. https://doi.org/10.27180/d.cnki.gjxzc.2021.000470
27. Y. Ichikawa, K. Hukushima, Statistical-mechanical study of deep boltzmann machine given weight parameters after training by singular value decomposition, *J. Phys. Soc. Jpn.*, **91** (2022), 114001. https://doi.org/10.7566/JPSJ.91.114001
28. N. Srivastava, R. R Salakhutdinov, G. E. Hinton, Modeling documents with deep boltzmann machines, preprint, arXiv: 13096865.
29. F. Taheri, K. Rahbar, P. Salimi, Effective features in content-based image retrieval from a combination of low-level features and deep Boltzmann machine, *Multimed. Tools Appl.*, (2022). https://doi.org/10.1007/s11042-022-13670-w
30. Y. Wang, F. Xu, J. Wang, X. Cui, T. Yi, Reconfigurable stochastic neurons for restricted boltzmann machine, *J. Phys.: Conf. Ser.*, **2347** (2022), 012014. https://doi.org/10.1088/1742-6596/2347/1/012014
31. N Zhang, S. Ding, J. Zhang, Y. Xue, An overview on Restricted Boltzmann Machines, *Neurocomputing*, **275** (2018), 1186–1199. https://doi.org/10.1016/j.neucom.2017.09.065
32. V. Duraisamy, A. Devi, S. Aggarwal, Multi disease prediction based on combined deep reinforcement Boltzmann machines, *AIP Conf. Proc.*, **2555** (2022), 020003. https://doi.org/10.1063/5.0108952

33. P. Rana, Physicochemical Properties of Protein Tertiary Structure, 2013. Available from: https://archive.ics.uci.edu/ml/datasets/Physicochemical+Properties+of+Protein+Tertiary+Structure
34. A. Tsanas, M. Little, P. McSharry, L. Ramig, Accurate telemonitoring of Parkinson's disease progression by non-invasive speech tests, *Nat. Prec.*, (2009). https://doi.org/10.1038/npre.2009.3920.1
35. H. Anysz, A. Zbiciak, N. Ibadov, The influence of input data standardization method on prediction accuracy of artificial neural networks, *Procedia Eng.*, **153** (2016), 66–70. https://doi.org/10.1016/j.proeng.2016.08.081
36. M. Shanker, M. Y. Hu, M. S. Hung, Effect of data standardization on neural network training, *Omega*, **24** (1996), 385–397. https://doi.org/10.1016/0305-0483(96)00010-2