Research on Diabetes Aided Diagnosis Model Based on Deep Belief Network

Zhijie Zhao[1,2(✉)], Yang Liu[1,2], Huadong Sun[1,2], Xiaowei Han[1,2], and Ran Wang[1,2]

[1] School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China
1249182744@qq.com
[2] Heilongjiang Provincial Key Laboratory of Electronic Commerce and Information Processing, Harbin University of Commerce, Harbin 150028, China

Abstract. Diabetes is a chronic disease that seriously endangers human health, which should be early detection, early diagnosis and early treatment by establishing prediction model. With the help of disease auxiliary diagnosis based on machine learning, the process of early diagnosis could be more reliable. Then, the patients have more chances of early treatment. Deep learning technology can take advantage of its own powerful feature learning ability to the application of disease auxiliary diagnosis, and has gained good results. This paper proposes a diabetes prediction model based on Deep Belief Network (DBN). The model is established by using Pima Indians Diabetes data set, combined with cross-validation, setting DBN structure and adjusting DBN network parameters. The experimental results show that the accuracy of the model is as high as 77.60% and the performance is good.

Keywords: Deep learning · DBN model · Auxiliary diagnosis · Diabetes prediction model

1 Introduction

Diabetes is one of the most serious and critical health problems facing the world in the twenty-first Century [1]. Traditionally, physicians mainly take years of accumulated personal experience and laboratory or instrumental indicators as the basis for diabetes mellitus diagnosis. Many subjective factors are in doping and easily misdiagnosed [2]. Prediction of diabetes mellitus can effectively solve the drawbacks of traditional methods and assist doctors in more comprehensive and reliable disease diagnosis by establishing a deep learning model [3, 4].

Establishing diabetes prediction model needs to consider the non-linear effect between the various pathogenic factors. Modeling methods such as statistics and machine learning, which are commonly used at home and abroad, are limited in the ability of expressing complex functions, and are more or less restricted [5]. Deep learning takes advantage of its own powerful feature learning ability, which is applied to the application of disease auxiliary diagnosis, has achieved good results. A deep belief networks model framework based on the heterogeneous electronic health records (EHRs) has been developed for identifying informative risk factors and predicting

M. Jia et al. (Eds.): WiSATS 2019, LNICST 281, pp. 229–240, 2019.
https://doi.org/10.1007/978-3-030-19156-6_22

osteoporosis, which performances well [6]. Chen and others have constructed a prediction model of thyroid nodules based on DBN, which has high accuracy in both non-sparse and sparse data sets, reaching 94% and 88.84% respectively [7]. Combined with genetic algorithm, an improved DBN model is proposed to predict coronary artery disease, which the prediction result of is good and the accuracy is as high as 89.24% [8]. Therefore, according to the diagnostic and data characteristics of diabetes mellitus, this paper constructs a diabetes prediction model by using DBN technology, and achieves a higher accuracy.

2 Deep Belief Network

In 2006, Hinton proposed a DBN structure and showed that each layer can perform unsupervised training again on the basis of the output of training results on the previous level, which clearly pointed out the effectiveness of unsupervised learning at all levels of training [9]. The typical DBN model is stacked by a series of Restricted Boltzmann Machine (RBM), which can solve the problem of slow convergence rate and easy to fall into local optimum, when the traditional back-propagation algorithm trains multilayer neural networks [10].

2.1 Restricted Boltzmann Machine

As the core of DBN model, RBM has a powerful architecture, which has two layers of network structure, namely visual layer and hidden layer. As shown in Fig. 1. According to the graph, the lower level $v = (v_1, v_2, \cdots, v_m)$ represents the visual layer formed by m visible nodes and the upper layer $h = (h_1, h_2, \cdots, h_n)$ represents the hidden layer formed by n hidden nodes. There is a weight connection between the visible layer node and the hidden layer node and there is no connection between the visible layer and the hidden layer unit. That is to say, each visual node is only affected by the n hidden nodes and independent of other visual nodes, which means that each visual node has only two states {0, 1}. The same is true for hidden nodes. The architecture features make RBM training easier. RBM training uses Contrastive Divergence (CD) algorithm, which is a fast learning algorithm of RBM proposed by Hinton in 2002. The training process is as follows [11–14].

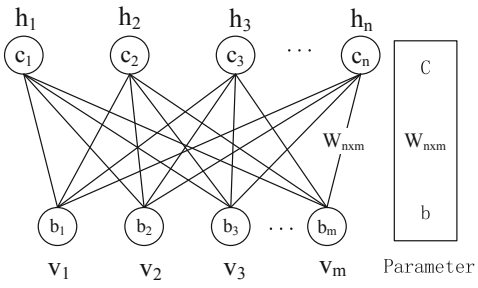

Fig. 1. Schematic diagram of RBM network structure.

Input: Training Sample x_0, Hidden Layer Node Number n, Learning Rate λ, Maximum Training Period T.

Output: Link Weight Matrix W, Visible Layer Bias Vector b, Hidden Layer Bias Vector c.

Training Phase: The state of initializing visible layer nodes is $v_1 = x_0$, and W, b 和 c are small random number.

For $t = 1{:}T$

For $j = 1{:}n$ # For All Hidden Nodes

$$P(h_{1j} = 1|v_1) = \text{sigmoid}\ (\text{c}_j\ +\ \text{sum\_}i(v_{1i} * W_{ij})) \tag{1}$$

For $i = 1{:}m$ # For All Visible Nodes

$$P(v_{2i} = 1|h_1) = \text{sigmoid}(\text{b}_i\ +\ \text{sum\_}j(W_{ij} * h_{1j})) \tag{2}$$

For $j = 1{:}n$ # For All Hidden Nodes

$$P(h_{2j} = 1|v_2) = \text{sigmoid}\ (\text{c}_j\ +\ \text{sum\_}i(v_{2i} * W_{ij})) \tag{3}$$

Update Weight and Bias

$$W = W + \lambda * (P(h_1 = 1|v_2) * v_1 - P(h_2 = 1|v_2) * v_2) \tag{4}$$

$$b = b + \lambda * (v_1 - v_2) \tag{5}$$

$$c = c + \lambda * (P(h_1 = 1|v_1) - P(h_2 = 1|v_2)) \tag{6}$$

2.2 DBN Structure and Training

DBN is a deep neural network composed of multi-layer RBM and a BP neural network. The basic structure is shown in Fig. 2. The DBN model is based on the joint distribution of data and features, the basic idea of it is to use the layer-by-layer greedy algorithm for hierarchical unsupervised learning of DBN, and then take advantage of the tagged data at the top level to conduct supervised learning and adjustment of the network. The training process is as follows [11–14].

(1) The method of unsupervised greedy layer by layer is used to pre train to obtain the weights of generated models. Unsupervised training for each layer of RBM is carried out. At this stage, the visual layer produces a vector v_1, which is mapped to the hidden layer vector value h_1, and uses the hidden layer vector value h_1 to reconstruct the vector value of the visual layer v_1 to get v_2, then the reconstructed v_2 mapping h_2 is used again, the process of which is called Gibbs sampling. This method can ensure that feature information is retained as much as possible when feature vectors are mapped to different feature spaces. The difference between the visible layer vector value and the hidden layer vector value is used to update the weight.

(2) After pre-training, the top layer associates the output of the lower layer with its memory. Each layer of RBM training can only ensure that the weight of its own layer achieves the best mapping of the layer feature vector. Therefore, the whole DBN needs to be adjusted from top to bottom according to the difference between the network output and the expected output. The BP network is added to the top layer of DBN, and the output vector of the last layer RBM is used as the input vector of BP network, then according to tagged data, the discriminative performance is adjusted through supervised training of the classifier by using BP algorithm.

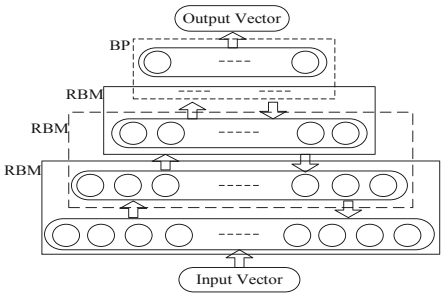

Fig. 2. Schematic diagram of DBN basic structure.

3 Prediction Modeling Process Based on Deep Belief Network

The data is divided into two parts: training set and test set. For training set, the prediction model of diabetes based on DBN is established, combined with empirical formula, setting and adjusting the parameters to determine the optimal network structure of DBN. For test set, The DBN diabetes prediction model is validated by it. The sensitivity and specificity are used to evaluate the performance of the model. They complement each other and complete the construction of DBN diabetes prediction model. The establishment process is shown in Fig. 3.

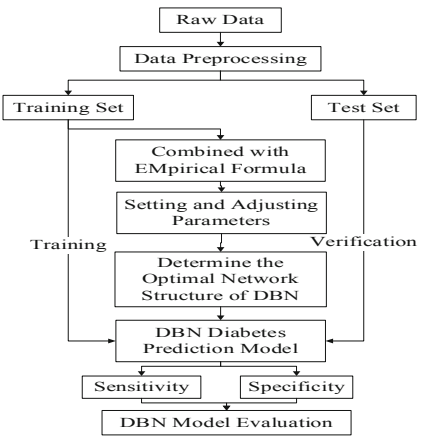

Fig. 3. Flow chart of diabetes prediction modeling based on DBN.

3.1 Training Set and Test Set

After considering the problem about the number of diabetes data sets and model training, this paper adopts K-fold Cross Validation method to determine the training set and test set, so as to avoid over-learning and under-learning effectively. The basic idea of K-fold Cross Validation is to divide the available data into K parts ($K \geq 3$), each representing a subset. Arbitrary K-1 subset is combined into a training set, and the remaining subset is used as a test set [15]. Thus, different training sets and corresponding test sets of K groups could be obtained.

3.2 DBN Network Structure

Only four important parameters including the number of input layer nodes, hidden layer nodes, hidden layer nodes and output layer nodes are identified to determine the DBN network structure before the prediction model can be established.

The number of input layer nodes is the attribute number of the dataset and the number of output layer nodes is equal to the class number in the dataset. What we need to pay attention to is the choice of DBN hidden layer and node number. A DBN model with multiple hidden layers performs better, but it doesn't mean that the more layers there are, the better [16]. Hidden layer nodes are the knowledge acquired by the DBN model through the data set, which can show the complex nonlinear relationship among them. The appropriate number of nodes should be selected to maximize the performance of DBN [17]. The empirical formula for selecting the number of hidden layers and nodes is adopted to determine its approximate range, which can avoid the blindness and increase the effectiveness of selection [12]:

$$S = \sqrt{mn} + \mathrm{k}/2 \tag{7}$$

Here, S is the number of nodes in the hidden layer, m is the number of nodes in the input layer and n is the number of nodes in the output layer. K is constant belonging to 1–10.

Under the condition that m, n, k are known, the number of nodes in the first hidden layer S_1 can be calculated to have p values according to formula (7), which are recorded as $S_1 = [S_{11}, S_{12}, S_{13}, \cdots, S_{1P}]$. Assuming that the number of nodes in the first hidden layer has been determined, the number of nodes in the first hidden layer is equal to the number of nodes in the input layer, for example, if $S_1 = S_{1P}$, then $m = S_{1P}$, and the number of nodes in the output layer n remains unchanged. According to formula (7), the number of nodes in the second hidden layer S_2 can be calculated to have q values, which are recorded as $S_2 = \left[S_{21}, S_{22}, S_{23}, \cdots, S_{2q}\right]$. By analogy, under the different values of the number of nodes in the first hidden layer S_1, the number of nodes in the second hidden layer S_2 can be calculated respectively, which should be merged, and are recorded as $S_2 = \left[S_{21}, S_{22}, S_{23}, \cdots, S_{2q}, \cdots, S_{2\,(q+a)}\right]$. This calculation idea of simultaneously determining the number of hidden layers and the number of corresponding hidden layer nodes can be continued and determined experimentally.

3.3 DBN Network Parameters

In the process of training DBN model, parameters need to be set and adjusted to improve the classification accuracy of it. On the basis of DBN network structure, hidden layer and output layer neuron activation function need to be set, and not only RBM training parameters, but also BP training parameters should be set up. DBN can be trained in batches, which is determined by the number of training set samples and each batch of training data. In addition, the setting of parameters including maximum number of network cycles, learning rate and momentum factor are also crucial.

3.4 DBN Network Training

Under the same training parameters, it is necessary to combine the p value of S_1 with the $q + a$ value of S_2, and K-time DBN training should be carried out by using the K-group training set and test set, totaling $K * p * (q + a)$ experiments. The average accuracy of each classification under K-time DBN training is calculated as the final classification accuracy. The higher the classification accuracy rate, the better the prediction effect of DBN model.

3.5 DBN Model Evaluating

The performance of DBN prediction model was evaluated by accuracy, specificity and sensitivity. The calculation of specificity and sensitivity needs to draw on the confusion matrix, as shown in Table 1. For a specific example, there will be four cases if the disease is positive class and no disease is negative class. If a positive class is predicted to be a positive class, it is True Positive. If a negative class is actually predicted to be a positive class, it is False Positive. If a negative class is predicted to be a negative class, it is True Negative. If a positive class is predicted to be a negative class, it is False Negative [18].

Table 1. Confusion matrix.

	Predict disease	Predict no disease
Actual disease	True Positive (TP)	False Negative (FN)
Actual no disease	False Positive (FP)	True Negative (TN)

The accuracy is the classification accuracy of the DBN optimal prediction model, expressed in Acc.

The specificity is the ratio of the actual uninfected people number in the predicted number of uninfected people to the total number of actual uninfected people, representing the generalization ability of the model, expressed in Spe.

$$Spe = \frac{TN}{TN + FP} \tag{8}$$

The sensitivity is the ratio of the actual infected people number in the predicted number of infected people to the total number of actual infected people, representing the accuracy of the model classification, expressed in Sen.

$$Sen = \frac{TP}{FN + TP} \tag{9}$$

In the medical diagnosis of diabetes mellitus, it is the primary task to diagnose patients with diabetes mellitus, that is, the greater the value of Sen, the better. The number of times that the normal person without diabetes is misdiagnosed as the disease, that is, the smaller the value of Spe, the better. Jointly evaluating the performance of the DBN diabetes prediction model by the above indicators is more comprehensive and objective.

4 Diabetes Prediction Simulation Experiment

4.1 Data Sources

The Data is about diabetes diagnosis information of native American women from Pima Indian heritage near phoenix, Arizona. They are above the age of 21 per capita. Eight risk factors related to diabetes are extracted, including six quantitative and continuous features, composed of a variety of clinical trial results, the remaining two quantitative and discrete features. The 9 features and sample data, including the classification codes, are shown in Tables 2 and 3 respectively. In Table 2, Body Mass Index (BMI) is calculated by formula, being recorded as $BMI = wei \div hei^2$, where *wei* represents weight, unit kg and *hei* represents height, unit m. In Table 3, Column 1 is the patient number. Columns 2–9 are the eigenvalues of the diabetes checking, as the attribute of the input data. The last column shows whether a given individual will suffer from diabetes within five years: 1 represents the positive in diabetes, that is, the individual will have diabetes in five years, a total of 268 cases. 0 represents the negative in diabetes, that is, the individual will not have diabetes within five years, a total of 500 cases. The Pima Indians Diabetes data set can be achieved in the UCI machine learning database.

Table 2. Data feature list (including category).

Feature number	Feature descriptions
Feature 1	Number of times pregnant
Feature 2	Plasma glucose concentration
Feature 3	Diastolic blood pressure (mm Hg)
Feature 4	Triceps skin fold thickness (mm)
Feature 5	2-h serum insulin (mu U/ml)
Feature 6	Body mass index
Feature 7	Diabetes pedigree function
Feature 8	Age in years
Classification codes	Binary class variable

Table 3. Original data sample (including category).

Number	Feature1	Feature2	Feature3	Feature4	Feature5	Feature6	Feature7	Feature8	Classification codes
1	6	148	72	35	0	33.6	0.627	50	1
2	8	183	64	0	0	23.3	0.672	32	1
3	1	85	66	29	0	26.6	0.351	31	0
4	1	89	66	23	94	28.1	0.167	21	0
…	…	…	…	…	…	…	…	…	…

4.2 Experiment Results and Analysis

The Referring to the Sect. 3 and making K = 3, 768 pieces of input data, which are normalized by [0, 1] of the interval normalization method, are divided into three parts. 256 pieces of each part represent a subset, and three groups of training sets and corresponding test sets are obtained.

Seeing the data characteristics of Pima Indians Diabetes, it is known that the number of input nodes m is 8 and the number of output categories n is 2. Using the formula (7) and the idea of the corresponding algorithm, which are mentioned in Sect. 3.2, experiments are conducted to determine that the number of hidden layers is 2 layers. The number of nodes in the first hidden layer S_1 has 6 values, recorded as $S_1 = [4, 5, 6, 7, 8, 9]$, and the number of nodes in the second hidden layer S_2 has 9 values, recorded as $S_2 = [2, 3, 4, 5, 6, 7, 8, 9, 10]$. The sigmoid function is set as the hidden layer and the output layer neuron activation function. The maximum number of network cycles is 1000 times and the training data per batch is 128. The value of learning rate is set to 1 and the value of momentum factor is set to 0, which are contained in the RBM training parameters. BP training parameters include learning rate, the value of which is set to 2, and momentum factor, the value of which is set to 0.9. Three DBN network training are conducted for three groups of training sets, and 6 values of S_1 and 9 values of S_2 are combined in two to conduct each DBN training, in total of 3 * 6 * 9 experiments. The experimental results are shown in Table 4.

Table 4. DBN experiment results.

S2	S1					
	4	5	6	7	8	9
2	0.7591	0.7630	0.7721	0.7669	0.7697	0.7656
3	0.7630	0.7630	0.7565	0.7552	0.7578	0.7682
4	0.7734	0.7682	0.7318	0.7617	0.7642	0.6966
5	0.7708	0.7643	0.7617	0.7617	0.7513	0.7656
6	0.7656	0.7747	0.7396	0.7669	0.7708	0.6862
7	0.7565	0.7096	0.7357	0.7552	0.7695	0.7318
8	**0.7760**	0.7617	0.7591	0.7669	0.7305	0.7708
9	0.7734	0.7656	0.7620	0.7539	0.7539	0.7500
10	0.7565	0.7396	0.7591	0.7578	0.7669	0.7305

Table 4 shows that under the same training parameters, the number of nodes in two hidden layers is different, resulting in different classification accuracy of DBN, but the overall effect of the model is better than 65%, and the classification accuracy of some DBN network structures is as high as 80%. When the number of nodes in the input layer is 8, the number of nodes in the first hidden layer is 4, the number of nodes in the second hidden layer is 8, and the number of nodes in the output layer is 2, the diabetes prediction model is best and the accuracy is 77.60%.

In the training process of DBN model, a training times-error graph is generated. As shown in Fig. 4. The horizontal axis represents the times of training, the vertical axis represents the reconstruction error generated by each training, and the three different colored curves represent training process of different training sets under 3 fold cross validation. It can be seen from Fig. 4 that the directions of three curves are roughly the same, indicating that the essential characteristics of different training data represented by the three curves are similar. With the increase of training times, each curve decreases sharply, and tends to be stable after the inflection point appears around the fiftieth training, which shows that the reconstruction error decreases rapidly before the inflection point and it still decreases, but the decrease is small after the inflection point.

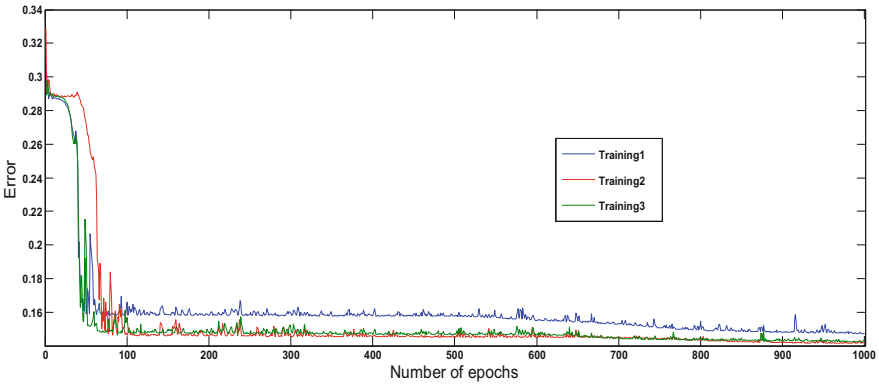

Fig. 4. Training times-error graph of DBN model.

4.3 Comprehensive Comparisons of Testing Results

For Pima Indians Diabetes data set of UCI machine learning database, some scholars also established classification models to predict diabetes, and compared it with the DBN prediction model built in this paper. As shown in Table 5.

Table 5. Accuracy comparison results of several models.

Authors	Year	Data size & class	Classification model	Accuracy (%)
Eggermont et al. [19]	2004	768 Controls: 500 Diabetes: 268	C4.5	71.60
Karthikeyani et al. [20]	2012	768 Controls: 500 Diabetes: 268	SVM	74.80
Karthikeyani et al. [21]	2013	768 Controls: 500 Diabetes: 268	LDA	74.40
Bozkurt et al. [22]	2015	768 Controls: 500 Diabetes: 268	AIS, ANN	76.00
Iyer et al. [23]	2015	768 Controls: 500 Diabetes: 268	DT, NB	74.79

As can be seen from Table 5, classification accuracy of the models established by some scholars range from 70% to 75%, and they are high. The classification accuracy of DBN diabetes prediction model proposed in this paper is slightly improved. BP Neural Network and Support Vector Machine (SVM) are used for comparative experiments to verify the validity of the DBN model prediction and enhance the credibility of the model. Sensitivity and specificity, which are mentioned in Sect. 3.5, are discussed. The experimental results are shown in Table 6.

Table 6. Performance evaluation of models.

Model	Acc (%)	Sen (%)	Spe (%)	Time (s)
DBN	77.60	55.96%	89.21%	0.36
BP	76.3	60.46%	84.82%	0.36
SVM	76.56	54.11%	89.28%	62.57

It can be seen from Table 6 that the accuracy of the three diabetes prediction models is above 75%, indicating that they can predict diabetes accurately for the experimental data. The accuracy of DBN model is the highest, reaching 77.60%, which shows that it performs better than the other two models in the term of prediction accuracy. Compared with sensitivity and specificity index, DBN diabetes prediction model is at a moderate level among them. The running time of DBN model is 0.36 s, which is greatly shortened comparing with SVM. The feature makes it more conducive to practical application in the field of diabetes auxiliary diagnosis. Overall, the DBN diabetes prediction model based on Pima Indians Diabetes data set is the best, with high prediction accuracy, good model effect and fast running time. The experimental results show that the modeling process of DBN on diabetes is feasible, and the established DBN model can predict diabetes effectively and perform well.

5 Conclusion

Using Pima Indians Diabetes data set, a DBN-based diabetes prediction model is established, the optimal network structure of it is 8-4-8-2, the accuracy of it is 77.60%, the sensitivity of it is 55.96%, the specificity of it is 89.21%, and the running time of it is 0.36 s. Compared with BP and SVM, DBN diabetes prediction model has the best prediction accuracy and running time, and has a moderate level among them in terms of sensitivity and specificity index. The test results are better. This study shows that the application of DBN model in the auxiliary diagnosis of diabetes mellitus is good, and it can provide a reference for the application of similar methods in China.

Acknowledgements. This research is supported by the Harbin Science and Technology Bureau outstanding subject leader fund project (2017RAXXJ055), the Nature Science Foundation of Heilongjiang Province (F2018020) and the Humanities and social sciences research projects of the Ministry of Education (18YJAZH128).

References

1. IDF Diabetes Atlas Eighth Edition poster Homepage. http://diabetesatlas.org/resources/2017-atlas.html. Accessed 24 Sept 2018
2. Liu, X., Jia, H., Li, A., et al.: Common methods and standards for screening and diagnosing diabetes mellitus. Med. Recapitul. **11**(12), 1104–1106 (2005)
3. Pérez-Gandía, C., Facchinetti, A., Sparacino, G., et al.: Artificial neural network algorithm for online glucose prediction from continuous glucose monitoring. Diabetes Technol. Therap. **12**(1), 81–88 (2010)
4. Gao, W., Wang, S., Wang, Z., et al.: Study on the application of artificial neural network in analysing the risk factors of diabetes mellitus. Chin. J. Epidemiol. **25**(8), 715–718 (2004)
5. Zhu, M., Wu, Y.: Research on image processing based on deep network. Electron. Technol. Softw. Eng. (5), 101–102 (2014)
6. Li, H., Li, X., Ramanathan, M., et al.: Identifying informative risk factors and predicting bone disease progression via deep belief networks. Methods **69**(3), 257–265 (2014)
7. Chen, D., Zhou, D., Le, J.: Thyroid nodule benign and malignant prediction based on deep learning. Softw. Algorithms **36**(12), 13–15 (2017)
8. Lim, K., Lee, B.M., Kang, U., et al.: An optimized DBN-based coronary heart disease risk prediction. Int. J. Comput. Commun. Control **13**(4), 492–502 (2018)
9. Hinton, G.E., Osindero, S., The, Y.W.: A fast learning algorithm for deep belief nets. Neural Comput. **18**(7), 1527–1554 (2006)
10. Wang, Z., Li, Y., Feng, X., et al.: Personalized information recommendation based on deep belief network. Computer Engineering **42**(10), 201–206 (2016)
11. Wang, F., Li, Q.: Research on face recognition algorithm based on improved deep belief networks. J. Lanzhou Jiaotong Univ. **35**(1), 42–48 (2016)
12. Yang, X.: Study of early warning for cerebrovascular risk based on deep beliefs networks. Beijing Jiaotong University (2016)
13. Sun, Z., Xue, L., Xu, Y., et al.: Overview of deep learning. Appl. Res. Comput. **29**(8), 2806–2810 (2012)
14. Yang, X.H., Zhong, N.Y.: Forecasting of hospital outpatient based on deep belief network. Comput. Sci. **43**(11A), 26–30 (2016)

15. Hu, J., Zhang, G.: K-fold cross-validation based selected ensemble classification algorithm. Bull. Sci. Technol. **29**(12), 115–117 (2013)
16. Gao, Q., Ma, Y.-M.: Research and application of the level of the deep belief network (DBN). Sci. Technol. Eng. **16**(23), 234–238 (2016)
17. Liao, Q., Zhang, J.: Optimization of DBN network structure based on information entropy. Inf. Commun. **1**, 44–48 (2018)
18. Ma, L.: Analyzing risk factors for multi-diseases with decision tree, logistic regression and improved neural network. Software **12**, 58–65 (2014)
19. Eggermont, J., Kok, J.N., Kosters, W.A., et al.: Genetic programming for data classification: partitioning the search space. In: ACM Symposium on Applied Computing, pp. 1001–1005 (2004)
20. Karthikeyani, V., Begum, I.P., Tajudin, K., et al.: Comparative of data mining classification algorithm (CDMCA) in diabetes disease prediction. Int. J. Comput. Appl. **60**(12), 26–31 (2012)
21. Karthikeyani, D.V., Begum, I.P.: Comparison a performance of data mining algorithms (CPDMA) in prediction of diabetes disease. Int. J. Comput. Sci. Eng. **5**(3), 205 (2013)
22. Bozkurt, M.R., Yurtay, N., Yilmaz, Z., et al.: Comparison of different methods for determining diabetes. Turk. J. Electr. Eng. Comput. Sci. **22**(4), 1044–1055 (2015)
23. Iyer, A., Jeyalatha, S., Sumbaly, R.: Diagnosis of diabetes using classification mining techniques. Int. J. Data Min. Knowl. Manage. Process **5**(1), 1–14 (2015)