Research on Diabetes Prediction Analysis Based on Improved ANN Algorithm

Yinggui Wang¹,a, Ben Wang²,b*, Jiazhi Di³,c, Jiehao Sun⁴,d, Xiya Wang⁵,e

¹School of Information Science and Engineering, Hangzhou Normal University, Hangzhou, Zhejiang, 311121, China
²School of Information Science and Engineering, Hangzhou Normal University, Hangzhou, Zhejiang, 311121, China
³School of Information Science and Engineering, Hangzhou Normal University, Hangzhou, Zhejiang, 311121, China
⁴Operation Manager, China Mobile Hangzhou R&D Center, Hangzhou, Zhejiang, 311121, China
⁵School of Information Science and Engineering, Hangzhou Normal University, Hangzhou, Zhejiang, 311121, China

aemail : 2276598112@qq.com,
b*Corresponding author’s e-mail : 20170056@hznu.edu.cn
cemail : 2020112011016@stu.hznu.edu.cn, demail : sunjh42a3@163.com,
eemail : 1647101212@qq.com

Abstract. Diabetes mellitus is a common chronic disease with a long phase of asymptomatic. This paper focuses on five classification algorithms in machine learning, which are MLP, SVM, KNN, DT and the improved algorithm of ANN. By adjusting appropriate parameters for mining and analysing diabetes data, classifier effect is analysed according to the performance indicators of accuracy, precision, recall, and F1-score. The suitable algorithm is researched for diabetes prediction, and provides ideas for mining and analysing other disease data from current medical industry.

1. Introduction
In recent years, the number of diabetes cases is still increasing globally, but there is no effective way to cure diabetes. Rapid and accurate detection of diabetes has become the primary task of controlling this condition. In the era of big data, machine learning and deep learning technologies are increasingly used to provide scientific assistance for medical diagnosis. At present, many researchers in the world have studied the classification prediction model of diabetes by machine learning algorithm, and further studied the prediction and classification of diabetes. Based on the electronic medical data, Simon G.J expanded the use of association rules to study diabetes-related factors and diabetes-susceptible people [1]. Vijayan analysed and compared the performance from many data mining algorithms to predict and diagnose diabetes, such as the expectation maximization (EM) algorithm in the clustering algorithm, the K-means algorithm, the classification algorithm, the k-nearest neighbor (KNN) algorithm. Its
experimental data are the Bima area Indian diabetes data set in the machine learning database of the University of California [2]. The American Diabetes Association and Eddie, the creator of the Archimedes model, have developed application software based on stochastic statistical theory. It predicts the probability that users will have type 2 diabetes and related complications in the future [3]. Li Juan combined 13 environmental factors and 5 genetic factors as predictors to construct a type 2 diabetes prediction model. The results show that the prediction of environmental and genetic factors were comprehensively considered. The performance of this model was better than the predictive model which only considered environmental factors [4]. Purushottam predicted diabetes risk of patients by using C4.5 algorithm and Partial Tree algorithm to automatically extract diabetes prediction rules [5]. Santhanam handled genetic algorithm to reduce the dimension of diabetes data set. It supported vector machine to predict diabetes [6]. Huang Yanqun established a personalized diabetes prediction model on patients' similarity [7]. Aiswarya utilized the J48 decision tree and Naive Bayes to classify the diagnosis of diabetes with the Pima Indian diabetes dataset. And the classification accurate ratio were 74.8%, 79.5% by the J48 decision tree and Naive Bayes method [8].

2. Data sets and data processing

2.1. Data sets description
This article uses a public data set on UCI: PIMA Indian diabetes dataset (PIDD). There are 768 data items in PIDD, each contains eight feature attributes and one classification label, the specific characteristic attributes are shown in Table 1.

| Selected feature attributes | Description                                                                 |
|-----------------------------|----------------------------------------------------------------------------|
| Pregnancies                 | Number of participant pregnancies                                          |
| Glucose                     | Blood glucose value, the blood glucose concentration value of the glucose tolerance test taken within two hours |
| Blood Pressure              | Diastolic blood pressure                                                   |
| SkinThickness               | Sebum thickness, the thickness of the subcutaneous fat of the triceps brachii muscle |
| Insulin                     | Serum insulin of two hours                                                 |
| Bmi                         | Body mass index                                                            |
| Diabetes Pedigree Function  | Genetic coefficient of diabetes mellitus                                   |
| Age                         | The age of the participants                                                |
| Outcome                     | Diabetes-like variables that are representative of patients who are diabetic and not representative of patients who aren’t diabetic |

2.2. Read the dataset
According to the requirements, the first 60% of the data set is the training part, the next 40% is the test set. The sample order cannot be changed. As the result, the first 460 data sets are divided into training sets, and the last 308 data sets are divided into test sets. Considering the inconvenience of text observation data, the text files are converted to csv files, named diabetes_train.csv and diabetes_test.csv. The pandas library and other necessary libraries in python are imported to read the dataset as shown in Table 2.

2.3. Data visualization
For better visualization, Notebook is utilized here for the operation. The matplotlib library in python and some advanced API in the library for the overall visualization of the dataset is necessary. The data set read here is the entire data set .csv, and the data is also read via pandas. As shown in Figure 1, the correlation coefficient matrix between the features is shown.
2.4. Data pre-processing
Considering that neural network, support vector machine and other algorithms are very sensitive to the scalability of data, we need to adjust the features to make them simply scale and move according to the features, so that the data is more suitable for these algorithms. The standardScaler and minmaxScaler functions from the sklearn library in Python are implemented for the purpose. Standardization means that the average value of each feature becomes 0 and the standard deviation becomes 1, so the features are comparable among different metrics. Meanwhile, the effect on the objective function is reflected in the geometric distribution rather than the numerical value, without changing the original data distribution. It makes each feature obey the standard normal distribution, thus eliminating the factors of different results due to dissimilar distribution of each feature.

Table 2 Import library functions and read datasets.

|   | Import pandas as pd          |
|---|------------------------------|
| 2 | import seaborn as sns       |
| 3 | import datetime             |
| 4 | import matplotlib.pyplot as plt |
| 5 | data_set = pd.read_csv('diabetes_train.csv') |
| 6 | data_set = pd.read_csv('diabetes_test.csv') |

3. Models and Methodologies

3.1. Model references

3.1.1. The basic classification models.
The sklearn package is demanded to download in python. Python library also integrates many classification algorithms, which can be used to build classification library model, such as MLP, SVM, KNN, DT. The implementation process is shown in the Table 3.

Table 3 Calling library functions.

| Feature: Call the classification model in the sklearn library | Import MLPClassifier from sklearn.neural_network | Transfer MLP |
|-------------------------------------------------------------|--------------------------------------------------|--------------|
| Import svm from sklearn                                     | Transfer SVM                                     |
| Import KNN from sklearn.neighbors                            | Transfer KNN                                     |
| Import DecisionTreeClassifier from sklearn.tree             | Transfer DT                                      |
3.1.2. **Improved ANN.**  
Firstly, weights and biases should be initialized. It initializes values randomly from -1 to 1 with a bias per node. During training samples, data are forward propagated from the input layer. According to the parameters of initialization and the activation function, the node outputs are calculated and sent to the output layer. Then reverse propagation is manipulated to calculate errors, and update the weights and biases. When the update of the weights falls below a certain threshold, the process will be terminated. The implementation process is shown in the Figure 2.

3.2. **Model parameters**

3.2.1. **MLP parameters.**  
Multilayer neural network (MLP) is inferior to other models in prediction accuracy, which may be due to different data scales. Deep learning algorithms similarly expect all input features to vary within the same scale. Ideally, the mean is 0 and the variance is 1. Therefore, the data must be renormalized to be able to meet these requirements. After renormalizing the data, the accuracy of training set and test set are 0.82 and 0.86 respectively. After increasing the number of iterations, the accuracy of the training and test sets become 0.87 and 0.75. The results show that increasing the number of iterations can only improve the performance of the training set, but has no effect on the test set. When the alpha parameter
is raised and the regularization of the weights is enhanced, the accuracy of the training and test sets changes to 0.80 and 0.79. At this point, it is no longer possible to improve the accuracy of test sets. Therefore, the best model is the default parameter depth learning model after data standardization. As shown in the Table 4.

**Table 4** Accuracy before and after MLP parameters.

|                        | Training set | Test set |
|------------------------|--------------|----------|
| Before adjustment      | 0.87         | 0.75     |
| Adjusted               | 0.82         | 0.86     |

3.2.2. SVM parameters.

The model of SVM has two important parameters: one is the regularization parameter C, which is 1 by default and limits the importance of each point, that is, the tolerance to errors. The higher the C, the more intolerant the error is; the smaller the C, the easier it is to underfit. Obviously the generalization ability becomes poor if the C is too large or too small. The other one is Gamma, which by default is the reciprocal of the number of features. It is used to control the width of the Gaussian kernel, so as to determine the distance between points. The larger the Gamma value is, the smaller the support vector is. There is a possibility that the training accuracy can be high while the test accuracy is not high, which is often called overtraining. The smaller the gamma value is and the larger support vectors is. With the increase of smoothing effect, the accuracy of training set and test set will be improved. In this paper, cross-grid validation is exploited firstly to select the best parameter pair among the set pairs. After determining the approximate range, the parameters need to be further readjusted, because the order of the samples in this experimental requirement can not be varied. The specific parameter adjustment process is shown in the Table 5.
### Table 5  SVM specific tuning process.

| C  | gamma | Training set accuracy | Test set accuracy |
|----|-------|-----------------------|-------------------|
| 1  | 0.1   | 0.78                  | 0.79              |
| 10 | 0.1   | 0.80                  | 0.82              |
| 10 | 0.2   | 0.81                  | 0.84              |
| 100| 0.2   | 0.82                  | 0.83              |
| 55 | 1     | 0.89                  | 0.88              |

![Figure 3](image-url)  
**Figure 3**  Accuracy under different k values.

#### 3.2.3. KNN parameters.

Here, the KNN model needs to be adjusted for the value of K which is the number of N_Neighbors. The most appropriate value of K can be found by plotting the relationship curve between training accuracy and test accuracy against different values of K. As shown in Figure 3, if the value of K is too small in the actual KNN model training, it means to use a small neighborhood for prediction. If the neighborhood happens to be a noise point, it may lead to over fitting. As the value of K increases, the training accuracy of the model had a rise followed a drop. When K=3, it reached the maximum accuracy for the first time. When the value of K exceeded 8 or 9, the model training accuracy started to show a reduce trend. The reason is that the long distance of samples will affect the prediction results and lead to the prediction error. Therefore, K=9 was selected here.

#### 3.2.4. DT parameters.

The main consideration of the decision tree is the number of layers, that is, the depth. If the depth of the tree is not limited, the depth of the tree will be large, so that the leaf nodes are pure enough to remember all the labels of the training data. The problems of overfitting and poor generalization ability are generated by this reason. Hense, pruning operations are mandatory. The default value of max_depth is none, and the pure nodes will be extended automatically. The max_depth was set to 4 in the lab.

#### 3.2.5. Improved ANN parameters.

The improved neural network requires modification of learning rate, regularization coefficient, and iteration number parameters. According to the a priori experience of MLP above, the number of nodes in the hidden layer is set to 300. The prediction accuracy in each case is shown in the Table 6 and Table 7.
Table 6  Improved ANN parameters and accuracy relationship.

| The number of iterations is 20000 | Learning rate 0.01 | Learning rate 0.001 | Learning rate 0.0001 |
|----------------------------------|--------------------|--------------------|--------------------|
| Regular coefficient 0.01        | 0.85               | 0.86               | 0.82               |
| Regular coefficient 0.001        | 0.85               | 0.86               | 0.82               |
| Regular coefficient 0.0001       | 0.80               | 0.89               | 0.83               |

As shown in Table 7, the most suitable learning rate is 0.001, the regularization coefficient is 0.0001. The data training is better because the number of nodes in the hidden layer is adapted below to improve the prediction by multiplying.

Table 7  Prediction accuracy under different parameters.

| Hide layer nodes | The number of iterations | Prediction accuracy |
|------------------|--------------------------|--------------------|
| 300              | 20000                    | 0.87               |
| 400              | 25000                    | 0.88               |
| 400              | 30000                    | 0.89               |
| 500              | 30000                    | 0.88               |
| 500              | 40000                    | 0.86               |

4. Model evaluation

For a comparative analysis of four traditional machine learning models and an improved ANN model, four metrics were used: Accuracy, precision, recall, and F1-Score. Accuracy rate indicates the proportional size of the number of correctly predicted samples to the number of all samples in the diabetes data test set. The precision rate represents the proportion for which the model predicts a positive case that is actually diagnosed with diabetes. The recall rate means the proportion of samples where the test set is actually positive as well as the model prediction. F1-Score can be interpreted as the summed average of precision and recall. In the real-world medical diagnosis, it is essential to identify as accurately as possible the patients who actually have diabetes. Therefore, the higher the accuracy and completeness rate, the better. The details are shown in Table 8.

5. Conclusion

In this paper, an improved ANN for diabetes prediction model is proposed. The improved ANN model performed well in the four indicators: accuracy, precision, recall rate and F1-Score. These factors reflect that the improved ANN model has better results in the prediction model. The accuracy on the validation set reached 89.2%, and the overall prediction model also showed strong generalization capability. However, the sample data is limited since there are 8 attributes involved in this classification. This reason limits the accuracy improvement to extra extent. It will be improved soon.

Table 8  Comparison of different models.

| Mode   | Accuracy | Precision | Recall | F1-score |
|--------|----------|-----------|--------|----------|
| MLP    | 0.86     | 0.83      | 0.83   | 0.83     |
| SVM    | 0.88     | 0.86      | 0.89   | 0.87     |
| KNN    | 0.86     | 0.82      | 0.83   | 0.81     |
| DT     | 0.87     | 0.84      | 0.86   | 0.82     |
| Improved ANN | 0.89 | 0.88 | 0.89 | 0.88 |

Acknowledgments

This paper was funded by projects: Zhejiang Province (LGF19F020011, LS21F020003). And it utilized the research platform and data in our teams [9].
References
[1] G.J.Simon, P.J.Caraballo, T.M.Therneau. (2015) Extending Association Rule Summarization Techniques to Assess Risk of Diabetes Mellitus. J. IEEE Transactions on Knowledge and Data Engineering, 27(1):130-141.
[2] Vijayan V., Ravikumar A. (2014) Study of Data Mining Algorithms for Prediction and Diagnosis of Diabetes Mellitus. J. International Journal of Computer Applications, 95(17):12-16.
[3] Eddy D M, Schlessinger L. (2003) Archimedes: a trial-validated model of diabetes. J. Diabetes Care, 26(11):3093-3101.
[4] Li Juan, Wu Jiang, Lu Li, Liu Donglei, Pang Xinghuo, Hu Yonghua. (2012) Based on support vector machine to establish a prediction model of environmental and genetic factors for type 2 diabetes. J. Chinese Journal of Disease Control, 16(02):171-175.
[5] Purushottam, Saxenak, Sharmar. (2015) Diabetes mellitus prediction system evaluation using C4.5 rules and partial tree. In: Ecological Entomology. India. pp. 1-6.
[6] SANTHANAM T, PADMAVATHI M S. (2015) Application of K-means and genetic algorithms for dimension reduction by integrating SVM for diabetes diagnosis. J. Procedia Computer ence, 47:76-83.
[7] HUANG Yanqun, WANG Ni, ZHANG Hui, et al. (2019) Diabetes mellitus. J. Journal of Medical Informatics, 40 (1):54-58.
[8] Aiswarya I, Jeyalatha S, Sumbaly R. (2015) Diagnosis of diabetes mellitus based on classification mining. J. Data Mining and Knowledge Management, 5(1):1.
[9] Chen, L.Q., Wang, B., Wang, Y.G., Wang, X.Y. (2021). Exploratory Data Analysis on the Usage of COVID-19 Vaccine. In: Proceedings of the 6th ISCIPT Symposium on Computer and Information Processing Technology. Changsha. pp. 101-105.