A Comparison of Support Vector Machines and Multilayer Perceptron on Heart Disease Prediction

By Ho Yin, Tam

Abstract:

This paper aims to evaluate and compare the classification performance of support vector machines and multilayer perceptron on heart disease diagnosis. Grid search with ten-fold cross-validation is applied to find the optimal hyperparameters. The grid search results show that the difference in the accuracy of different hyperparameters of support vector machines is subtle while the difference in the accuracy of different hyperparameters of multilayer perceptron is large. The best-trained models are then evaluated using the test set. The accuracy and area under curve (AUC) values are very close for the support vector machines and multilayer perceptron. However, with a slightly higher recall value, the multilayer perceptron is a preferable model to diagnose heart disease than the support vector machines.

Description and motivation:

Cardiovascular disease describes the conditions impacting the function of the heart or blood vessels. In 2019, over 18 million people died due to heart disease globally [1]. Hence, identifying if an individual has a cardiovascular disease has emerged as a critical issue as early treatment can alleviate symptoms and lower the risk of heart attack or stroke. In this report, support vector machines and multilayer perceptron are applied to the heart disease dataset to distinguish if a patient has heart disease. Then, the model performances are evaluated and contrasted based on training time and classification metrics, including accuracy, precision, recall, specificity, and F1 score. Also, the receiver operating characteristics (ROC) curves with the area under curve (AUC) values are plotted for the comparison of model performance.

Exploratory data analysis:

The heart disease dataset extracted from the Kaggle website contains 918 observations and 12 columns of which 11 columns are features and 1 column is the target label [2]. The features consist of both categorical variables and numeric variables. The categorical variables include the sex and the resting electrocardiogram results of the patients, while the numerical variables encompass the resting blood pressure and the maximum heart rate of the patients. The goal is to classify if the patient has a heart disease or is normal.

In the data cleaning and data exploratory process, though no missing data was found, 172 samples were revealed to have zero cholesterol and resting blood pressure which are anomalous. Therefore, these 172 samples are removed. Figure 1 describes that 356 patients have heart diseases, and 390 observations are normal, which accounts for 47.7% and 52.3% respectively. The dataset can be considered as a balanced dataset as the difference is small. The histograms and bar charts (Figure 2) depict the distributions of the features and the frequency of the features respectively. It is clear that the distribution of age and resting blood pressure are close to the normal distribution and the distribution of the old peak tends to have a right-skew distribution. The dataset is then encoded and standardized. The correlation matrix heatmap in Figure 3 illustrates the correlation between the features and the target label in which the variable of the slope of the peak exercise ST segment (ST slope) has the strongest correlation with the target label with a correlation coefficient of -0.6.

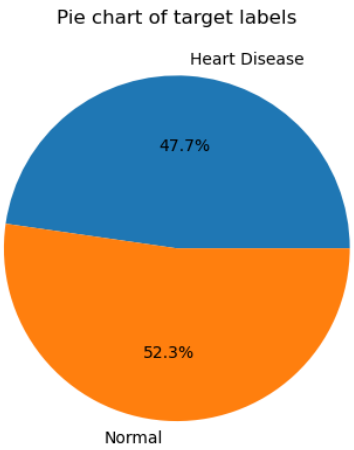

Figure 1

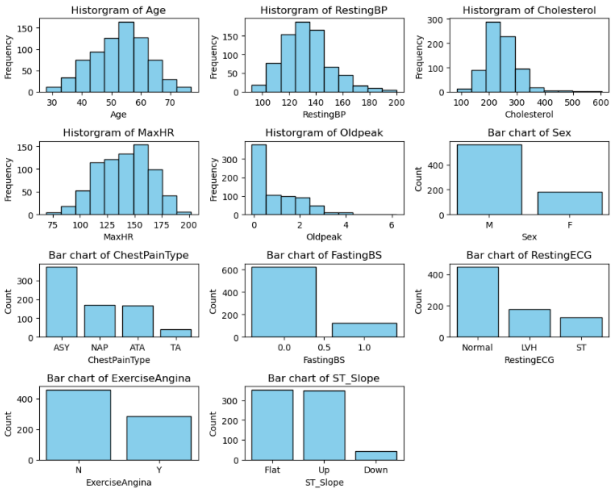

Figure 2

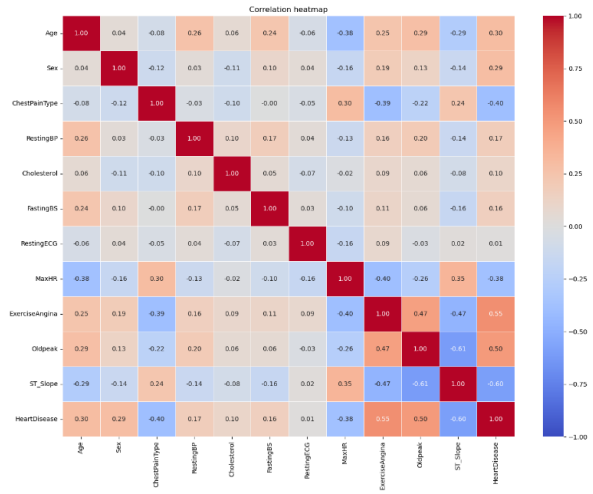

Figure 3

Summary of the two models with their pros and cons:

Support vector machines:

Support vector machines are supervised machine learning algorithms that can be used in classification and regression. For classification, it maximizes the margin between the hyperplane and the nearest point of any class. Not only linear classification tasks, but support vector machines also can perform non-linear tasks utilizing the kernel functions to map the input vectors into a high-dimensional feature space where a hyperplane can be constructed to separate the input into different classes. The advantages are that support vector machines can perform well in high-dimensional space and capture complex patterns due to a variety of kernel functions [3]. In contrast, the disadvantages are that support vector machines are computationally expensive due to hyperparameters tuning, and the long training time in large datasets due to the algorithmic complexity [4].

Multilayer perceptron:

Multilayer perceptron is an artificial neural network that consists of an input layer, one or more hidden layers, and an output layer. It can be used in supervised learning for classification and regression. In the forward pass, the input features propagate through layers until they arrive at the output layer where the output is produced. The output of the network is compared with the actual target label and the loss or error is computed. Through backpropagation, the weights are adjusted to minimize the loss function, and the model is trained [5]. Same as support vector machines, multilayer perceptron can also capture non-linear relationships between features and target variables using the non-linear activation function, such as the sigmoid and tanh functions. However, the multilayer perceptron is a black box that does not provide causality for events occurring in the system [6].

Hypothesis statement:

The first hypothesis is that the training time of the multilayer perceptron is longer than the support vector machines. The second hypothesis is that both multilayer perceptron and support vector machines perform well and achieve high scores in terms of accuracy, precision, recall, specificity, F1 score, and area under curve (AUC) value, due to their ability to capture non-linear complex structures in the data.

Description of the choice of training and evaluation methodology:

After the process of data cleaning, the remaining 746 observations are split into an 80% training set (596 observations) and a 20% test set (150 observations). To avoid overfitting, ten-fold cross-validation is employed during training for both the support vector machines and multilayer perceptron [7]. Also, grid

search is implemented to find the optimal hyperparameters for each model. The table of different combinations of hyperparameters will be shown and the model with the highest score will be chosen as the best-trained model for evaluating the performance using the test set. Moreover, the training time of the two models will be compared.

The performance of the well-trained model will be evaluated with the test set. The confusion matrix will be first displayed, and the classification metrics will be calculated, including accuracy, precision, recall, specificity, and F1 score. In addition, the receiver operating characteristics (ROC) curves with area under curve (AUC) value will be plotted to compare the performance of the well-trained models.

Choice of parameters and experiment results:

For support vector machines, the kernel function, degree of polynomial kernel function, and regularization parameter were chosen as the hyperparameters for tuning through grid search. For multilayer perceptron, the architecture consists of one input layer, one hidden layer, and one output layer. The rectified linear unit (ReLU) activation function was applied to the hidden layer and the SoftMax function was used in the output layer. Also, the binary cross entropy loss function was adopted and hence the model was trained via backpropagation to minimize the loss function. Also, the number of neurons in the hidden layer, learning rate, and momentum were chosen as the hyperparameters for tuning through grid search.

Table 1 and Table 2 summarize the grid search results. Table 1 shows that the support vector machines achieved its highest accuracy score of 0.85 with a polynomial kernel function, degree of 3 of the polynomial kernel function, and regularization parameters of 1. Table 2 displayed that the multilayer perceptron achieved its highest accuracy score of 0.84 with 100 neurons in the hidden layer, a learning rate of 0.1, and a momentum of 0.8. The best accuracy score of the two models is very close with a 2% difference. Re-running the grid search does not change the result of support vector machines but slightly changes the accuracy score of each multilayer perceptron model due to the initial random weights.

Another notable observation is that different hyperparameter combinations of support vector machines only yield a 4.7% difference between the highest and lowest accuracy scores. This illustrates that different combinations of hyperparameters contribute to the significant importance of accuracy rather than the hyperparameters themselves. In contrast, different hyperparameter combinations of multilayer perceptron have a significant impact on the accuracy scores with a 57% difference between the highest and lowest accuracy score. It is obvious that 1000 neurons in the hidden layer consistently have accuracy scores that are above 0.8, while the three lowest accuracy scores are only with 10 neurons in the hidden layer. This suggests that a larger number of neurons in the hidden layer tends to have a higher accuracy score as they can capture complex relationships between features.

Support Vector Machines			
Kernel	Degree	Regularization parameter	Score
poly	3	1	0.85404
poly	4	1	0.845621
poly	2	10	0.845537
poly	2	1	0.840621
rbf	4	1	0.840537
rbf	3	1	0.840537
rbf	2	1	0.840537
linear	4	1	0.84048
linear	3	1	0.84048
linear	2	1	0.84048
rbf	4	0.1	0.837175
rbf	3	0.1	0.837175
rbf	2	0.1	0.837175
poly	2	0.1	0.837147
linear	3	10	0.83709
linear	4	10	0.83709
linear	2	10	0.83709
poly	3	0.1	0.835621
poly	4	0.1	0.830621
linear	2	0.1	0.830395
linear	4	0.1	0.830395
linear	3	0.1	0.830395
poly	3	10	0.827203
rbf	2	10	0.825424
rbf	3	10	0.825424
rbf	4	10	0.825424
poly	4	10	0.815452

Table 1

Multilayer Perceptron			
Number of neurons in hidden	Learning rate	Momentum	Score
100	0.1	0.8	0.837232
10	0.1	0.9	0.832175
100	0.1	0.7	0.830508
1000	0.1	0.7	0.830452
100	0.1	0.9	0.828814
1000	0.01	0.7	0.828785
1000	0.01	0.8	0.828729
1000	0.01	0.9	0.827147
1000	0.1	0.8	0.827119
10	0.1	0.7	0.82709
10	0.1	0.8	0.825424
100	0.01	0.9	0.823814
1000	0.001	0.9	0.818701
1000	0.1	0.9	0.815565
1000	0.001	0.8	0.815452
100	0.01	0.7	0.813814
1000	0.001	0.7	0.812062
100	0.01	0.8	0.810226
10	0.01	0.9	0.793757
10	0.01	0.8	0.757034
100	0.001	0.9	0.734915
100	0.001	0.8	0.698079
10	0.01	0.7	0.68291
100	0.001	0.7	0.624548
10	0.001	0.9	0.604124
10	0.001	0.8	0.560226
10	0.001	0.7	0.533446

Table 2

Table 3 compares the training time of the support vector machines and multilayer perceptron in the grid search. The support vector machines were approximately 20 times faster than the multilayer perceptron to train as the training time of support vector machines and multilayer perceptron required 5.89 seconds and 118.47 seconds respectively. This supports the first hypothesis statement that the training time of the multiplayer perceptron is longer than the support vector machines.

	Support Vector Machines	Multilayer Perceptron
Training time (seconds)	5.89	118.47

Table 3

Analysis and critical evaluation of results:

Figure 4 and Figure 5 display the confusion matrices of support vector machines and multilayer perceptron respectively. Table 4 illustrates the classification performance on the test set by the two models, including accuracy, precision, recall, specificity, and F1 score. Both models perform well as all the performance metrics are all over 0.8. As the dataset contains balanced target labels of heart disease and normal, accuracy is a reliable measurement of the number of correctly identified observations out of the total number of observations. Support vector machines and multilayer perceptron achieve an accuracy of 0.86 and 0.85 respectively in which 86% and 85% of the observations are correctly classified. For the other performance metrics, including precision, recall, specificity, and F1 score, the differences are only approximately 5% or less which is very small. This supports the second hypothesis statement that both support vector machines and multilayer perceptron perform well.

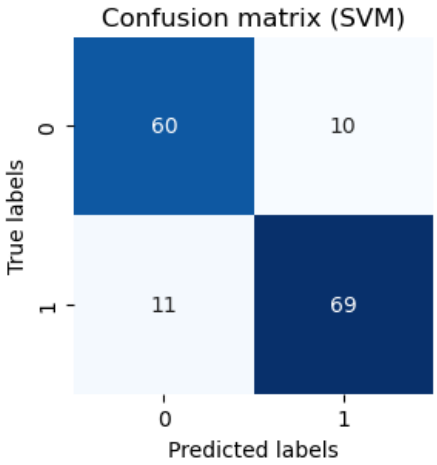

Figure 4

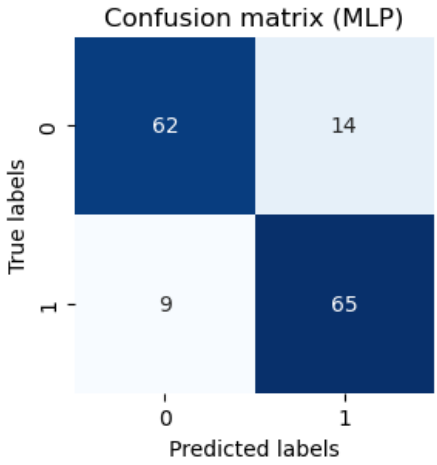

Figure 5

	Support Vector Machines	Multilayer Perceptron
Accuracy	0.86	0.85
Precision	0.87	0.82
Recall	0.86	0.88
Specificity	0.86	0.82
F1 score	0.87	0.85

Table 4

Figure 6 and Figure 7 depict the receiver operating characteristics (ROC) curves with the area under curve (AUC) values of the support vector machines and multilayer perceptron respectively. It is clear that support vector machines and multilayer perceptron perform well and have a very close area under curve (AUC) value of 0.93 and 0.94 respectively. This indicates that both models can effectively distinguish if the patients have heart disease or are normal. This also supports the second hypothesis statement that both support vector machines and multilayer perceptron perform well with similar results.

Though the accuracy and area under curve (AUC) value of the support vector machines and multilayer perceptron are very close, one important and interesting discovery is that the recall of multilayer perceptron is slightly higher than the support vector machines. Recall measures how well the model identifies the true positive cases out of all the actual positive cases. This indicates that the multilayer perceptron is more likely to identify all the actual positive cases or patients with heart disease than the support vector machines. It is a crucial metric when the task is associated with health issues and other problems with a high cost of error. Missing to identify a heart disease case can lead to the delay of treatment and further severe consequences for the health of the patients. Early diagnosis and therapy of heart disease can alleviate the symptoms and lower the risk of heart attack and stroke. Therefore, the multilayer perceptron can be a preferable model in distinguishing if a patient has heart disease or is normal compared to support vector machines.

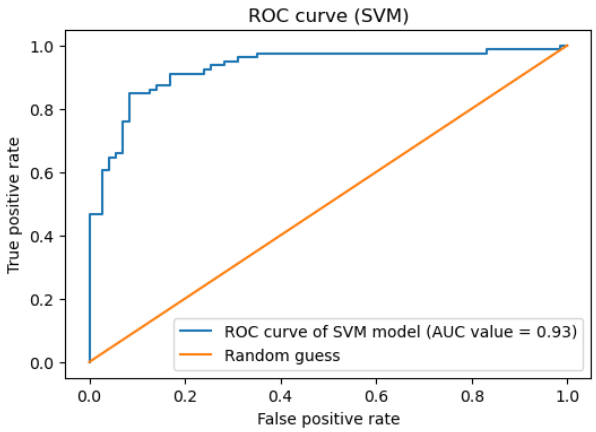

Figure 6

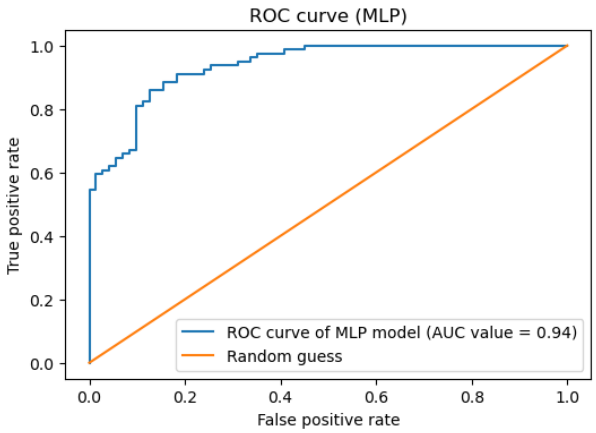

Figure 7

Conclusion, lesson learned and future work:

In summary, the result of the grid search with ten-fold cross-validation showed that the training time of the support vector machines is much faster than the multilayer perceptron, and the accuracy of both models is similar. The accuracy of the support vector machines depends on the combination of hyperparameters instead of the hyperparameters themselves. The accuracy of the multilayer perceptron highlights the larger number of neurons in the hidden layer tends to have a higher accuracy.

The evaluation process of the best-trained support vector machines and multilayer perceptron models using the test set revealed that both models performed well and demonstrated their strong ability to distinguish if a patient has heart disease or is normal. The performance metrics are all over 0.8. Both models have very close values of accuracy and area under curve (AUC) value. The differences in the other performance metrics, including precision, recall, specificity, and F1 score, are also relatively small. With a slightly higher recall value of 0.88, a multilayer perceptron is a preferable model for identifying heart disease due to its capability to classify all the actual positive cases or the patients with heart disease. When the task comes to health affairs or other issues with a high cost of error, such as diabetes diagnosis and cancer prognosis, a model with a higher recall value will be preferable. Missing to identification of the true positive cases may lead to the delay of treatment and serious consequences to the patients.

For future work, feature extraction techniques like principal component analysis (PCA) can be used to improve the performance of support vector machines and multilayer perceptron. It helps reduce the

dimensionality of the features by identifying a smaller set of uncorrelated features that capture most of the variance of the data. This reduces the complexity and lowers the risk of overfitting. Additionally, the ensemble learning techniques, such as bagging and boosting, can be explored to boost the performance of the models.

Reference:

[1] Saloni Dattani, Veronika Samborska, Hannah Ritchie and Max Roser (2023) - “Cardiovascular Diseases” Published online at OurWorldInData.org. Retrieved from: 'https://ourworldindata.org/cardiovascular-diseases'

[2] Fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction.

[3] Akkaya, Berke & Çolakoğlu, Nurdan. (2019). Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases.

[4] Cervantes, J., Garcia-Lamont, F., Rodríguez-Mazahua, L., & Lopez, A. (2020). A comprehensive survey on support Vector Machine Classification: Applications, challenges and Trends. Neurocomputing, 408, 189–215. https://doi.org/10.1016/j.neucom.2019.10.118

[5] Haykin, S. S. (2009). Neural networks and learning machines. Upper Saddle River, NJ: Pearson Education.

[6] Park, Y.S., Lek, S., Chapter 7 - Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling, in: Jørgensen, S.E. (Ed.) Developments in Environmental Modelling, Elsevier, 2016, pp. 123–140

[7] Géron, A. (2019). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems (2nd ed.). O’Reilly.

Glossary

Accuracy	The number of correctly classified observations (true positive and true negative) divided by the total number of observations.
Activation function	The function that allows a neural network to learn complex patterns in the data.
Area under curve (AUC)	A measure of how well the model distinguishes between positive classes and negative classes.
Backpropagation	An algorithm that computes the gradient descent and adjusts the weights to minimize the loss function in the neural network.
Binary cross entropy function	A loss function used in binary classification tasks in which the difference between the actual outcomes and the predicted outcomes are measured.
Classification	A supervised machine learning method in which the model classifies the input into a category or discrete values.
Confusion matrix	A table that summarizes the performance of a classification model.
Cross-validation	A technique to evaluate how well the model generalizes by training the model on the training set and evaluating using the validation set.
F1 score	The harmonic mean of precision and recall.
Features	The input variables or independent variables to a model.
Grid search	A technique to find the optimal hyperparameters for a model by trying all possible combinations of pre-defined hyperparameters and evaluating the performance of each combination.
Hyperparameter	The variables that are adjusted before training and control the learning of a model.
Kernel function	A method used in support vector machines which transforms the input vectors into a high-dimensional feature space so that they can be linearly separable.
Layer	A collection of neurons in the neural network and the common layers are the input layer, hidden layers and output layer.
Learning rate	A hyperparameter that controls the step size in each iteration towards the local minimum.
Momentum	A hyperparameter that helps accelerate convergence and overcome oscillation in the training process.
Neural network	A machine learning algorithm that models and learns complex relationships using at least one hidden layer and various activation functions.
Neuron	A basic unit or node that computes the weighted sum of the input values, passes the result to the activation function and generates an output value.
Overfitting	A situation in which a model performs well on the training data but poorly on the unseen test data.
Precision	The number of true positive classes divided by the number of positive classes predicted by the model (true positive and false positive).
Recall	The number of true positive classes divided by the number of actual positive classes (true positive and false negative).
Receiver operating characteristics (ROC)	A graph that plots the true positive rate against the false positive rate at different classification thresholds.
Rectified linear unit (ReLU)	An activation function that maps the output value to the input value if the input is greater than zero otherwise generates the output value to zero if the input is less than or equal to zero.
Regression	A supervised machine learning method in which the model predicts a continuous value given the input.

Sigmoid	An activation function that takes the input values and generates an output value between zero and one.
SoftMax	An activation function that takes the input values and generates a probability distribution over different classes.
Specificity	The number of true negative classes divided by the number of actual negative classes (true negative and false positive).
Target labels	The desired output predicted by a model.
Test set	A subset of a dataset to evaluate and test a trained machine learning model. The subset should be unseen in the training process.
Training set	A subset of a dataset to generate and train a machine learning model.

References:

https://developers.google.com/machine-learning

https://ml-cheatsheet.readthedocs.io/en/latest/index.html

https://www.w3schools.com/

Implementation details:

The data cleaning and exploration process are implemented in the 'heart_disease_EDA' Jupyter Notebook. No missing data is found via the function 'isnull()sum()'. However, the histogram shows that 172 observations have zero cholesterol and resting blood pressure which is abnormal. Hence, they are removed. Then the remaining data are encoded and standardized using the function of 'OrdinalEncoder()' and ''StandardScaler()' respectively. To ensure fairness in using the same data for training and evaluating, a training set and a test set are split and saved as CSV files for both support vector machines and multilayer perceptron.

The model training of support vector machines is implemented in the 'heart_disease_SVM_training' Jupyter Notebook. Different hyperparameters like kernel, degree of polynomial kernel function and regularization parameters are initialized in a dictionary and the function used for support vector machines is 'sklearn.svm.SVC'. To implement grid search with ten-fold cross-validation, the function 'GridSearchCV' takes the grid search hyperparameters dictionary, support vector machines function, and 10 as the parameters. Once the optimal hyperparameters are found, the best support vector machines model is saved as 'best_svm_model.joblib'.

The model training of multilayer perceptron is implemented in the 'heart_disease_MLP_training' Jupyter Notebook. The training data is imported through the pandas data frame and converted into the tensor data type. The Pytorch library is used to define the multilayer perceptron with one hidden layer. The 'nn.ReLU' function is used in the hidden layer and the 'nn.Softmax' function is used in the output layer. To leverage the power and convenience of grid search and cross-validation in 'Sklearn', the defined-multilayer perceptron with one hidden layer is wrapped into 'Skorch.NeuralNetClassifier'. Different hyperparameters like the number of hidden neurons in the hidden layer, learning rate and momentum are initialized in a dictionary. Using the GridSearchCV function, the grid search with ten-fold cross-validation is similar to the support vector machines. Once the optimal hyperparameters are found, the best multilayer perceptron model is saved as 'best_mlp_model.joblib'.

The model evaluation process for both support vector machines and multilayer perceptron is similar and implemented in the 'heart_disease_SVM_test' and 'heart_disease_MLP_test' Jupyter Notebook respectively. The main difference is that the test set imported as pandas data frame has to convert into tensor data type for the multilayer perceptron model. The two best-trained models are evaluated with the test set which are the same to ensure fairness. The confusion matrices, performance metrics, including accuracy, precision, recall, specificity and F1 score, and the receiver operating characteristics (ROC) curve with area under curve (AUC) value are displayed for comparison. Rerunning the code of support vector machines will provide the same result in training and testing. However, rerunning the code of multilayer perceptron may provide different results of grid search but similar performance due to the initial random weight.