Prediction of Heart Disease with Machine Learning

Jianxing Wu, Jieming Yu
SYDE 522 – Group 1, Option B
Systems Design Department
University of Waterloo, Canada
{j734wu, j537yu}@uwaterloo

***Abstract*—Cardiovascular disease is the most common cause of death worldwide over the past few decades. Early detection and continuous clinical monitoring are essential to reduce mortality rates. However, accurate detection of heart disease in all cases, along with synchronous and consistent monitoring by healthcare professionals, requires a significant time commitment as well as the support of experience and expertise. In the face of a large population of heart disease patients, such an allocation of medical resources is unrealistic. Thus, the classification of heart disease patients can conserve a considerable amount of resources, which is crucial in the diagnosis of cardiovascular disease. Numerous data mining techniques have been employed by researchers to aid healthcare professionals in the diagnosis of heart disease. For this task, many algorithms have been proposed in recent years. In this paper, we study different supervised machine learning techniques for the detection of heart disease and perform a procedural comparison of these methods. We utilize Logistic Regression, Linear Support Vector Machine, Decision Tree, Random Forest, and a 2-layer Neural Network on a heart disease dataset. The data used in this study is the Cleveland Clinic Foundation Heart Disease Data Set, available from the UCI Machine Learning Repository. This dataset records the clinical and non-invasive test results of 303 patients undergoing angiography at the Cleveland Clinic in Cleveland, Ohio. Principal Component Analysis and 5-fold cross-validation were applied to the algorithms to analyze their performance in heart disease detection. The Support Vector Machine outperformed the other algorithms, achieving the highest accuracy in correctly classifying instances, along with considerable precision and sensitivity.**

***Index Terms*—Cardiovascular disease, Logistic Regression, Support Vector Machines, Decision Trees, Random Forests, Neural Networks**

I. Introduction

Cardiovascular disease (CVD), which encompasses a range of conditions affecting the heart and blood vessels, remains one of the leading causes of mortality on a global scale. The rising prevalence of CVD, along with the increasing complexity of its associated risk factors, necessitates the development of efficient and accurate diagnostic and prognostic tools. Traditional diagnostic methods, which often rely on clinical judgment and a series of tests, are not only time-consuming and resource-intensive but also prone to human error, especially in cases involving subtle or complex presentations. The substantial volume of patient data generated on a daily basis further complicates this issue, accentuating the need for automated systems capable of efficiently processing large datasets to identify patterns and predict individual risks of CVD.

The advent of data mining and machine learning techniques presents a robust solution to these challenges. By harnessing sophisticated algorithms and statistical models, these methodologies strive to extract valuable insights from complex datasets, thereby enhancing diagnostic accuracy, enabling personalized risk stratification, and consequently facilitating more effective preventive and therapeutic strategies. Data mining techniques are capable of analyzing extensive volumes of patient data, encompassing a multitude of factors such as medical history, lifestyle choices, genetic predispositions, and physiological measurements, to discern latent patterns, correlations, and predictive relationships that may not be readily apparent through conventional methodologies. Machine learning algorithms, particularly those pertaining to supervised learning, can be trained on historical data to construct predictive models capable of estimating the probability of a patient developing cardiovascular disease (CVD) or experiencing adverse cardiovascular events.

Despite the considerable potential of data mining and machine learning in cardiovascular disease (CVD), a number of critical challenges persist. A primary obstacle is the inherent complexity of the disease process itself. Cardiovascular disease represents not a singular entity but rather a spectrum of conditions with varied etiologies, risk factors, and clinical manifestations. This heterogeneity complicates the development of universal predictive models that accurately reflect the nuances of individual patient cases. Another pivotal consideration is the quality and completeness of the data utilized for model development and evaluation. Data quality issues, such as missing values, inconsistencies, and inaccuracies, can substantially undermine the reliability and validity of the predictive models. Moreover, the ethical considerations related to the utilization of patient data for research and clinical decision-making must be meticulously addressed to ensure the responsible and unbiased application of these potent tools.

Extensive research has been dedicated to examining the application of data mining and machine learning techniques in predicting and diagnosing cardiovascular diseases (CVD). Numerous studies have utilized a variety of classification algorithms, such as support vector machines (SVMs), decision trees, naïve Bayes classifiers, and neural networks, across diverse datasets representing various populations and clinical contexts. These investigations have demonstrated encouraging outcomes concerning predictive accuracy; however, they also highlight certain limitations associated with specific datasets, algorithms, and methodological approaches. The tendency for overfitting in some models—where a model excels on training data but performs inadequately with unseen data—emphasizes the critical importance of employing robust evaluation methods, like cross-validation techniques, to ensure generalizability. Furthermore, the selection of appropriate features or variables that significantly contribute to predicting CVD risk remains a substantial challenge. Feature engineering and feature selection strategies are frequently utilized to enhance model performance, yet the optimal features may vary considerably depending on the particular characteristics of the dataset and the intended population.

Moreover, the interpretability of certain complex machine learning models, with particular emphasis on deep neural networks, constitutes a considerable impediment to their clinical adoption. While these models can attain high levels of predictive accuracy, their "black box" nature renders it difficult to comprehend the rationale underpinning the model's predictions. This lack of interpretability may obstruct the integration of these models into clinical practice, where clinicians necessitate clear elucidations of the diagnoses and prognoses delivered by the algorithms. Addressing this issue necessitates the advancement of more explainable machine learning algorithms, enabling clinicians to discern the determinants influencing the model's predictions, thereby potentially enhancing patient communication and collaborative decision-making.

The intricate nature of cardiovascular disease (CVD) necessitates a comprehensive approach to its prediction and diagnosis. The pathogenesis of the disease is modulated by a complex interplay of genetic predispositions, environmental influences, lifestyle behaviors, and various physiological conditions. Consequently, a solitary model that relies exclusively on clinical measurements may be inadequate. A more inclusive strategy that integrates multiple data sources, such as genetic data, lifestyle questionnaires, and continuous physiological monitoring, is likely to produce more precise and detailed predictions. The integration of these diverse data sources demands the formulation of sophisticated data fusion methodologies to maintain compatibility and consistency in analysis.

This paper seeks to enhance the existing body of research by addressing the previously identified limitations. It concentrates on a comparative analysis of various supervised machine learning techniques for cardiovascular disease (CVD) prediction, utilizing the widely recognized Cleveland Clinic Foundation Heart Disease dataset. The study will compare the performance of distinct algorithms, including logistic regression, support vector machines, decision trees, and ensemble methods, in the accurate prediction of CVD risk. Moreover, the paper will examine the influence of feature selection strategies on model performance and interpretability. The results from this study are expected to offer significant insights into the most effective and interpretable machine learning approaches for CVD prediction. A thorough evaluation of the models' performance using appropriate metrics will be conducted to evaluate the generalizability and reliability of the developed models. The outcomes of this study are envisaged to provide critical information for clinicians, researchers, and policymakers, aiding in the development of more effective and efficient strategies for the prevention and management of CVD. This comprehensive analysis aims to identify the most promising machine learning methodologies, offering a foundation for future progress and advancement in this essential field.

II. Background

The prediction of heart disease using data mining techniques has been an area of active research for the past two decades. A wide range of methodologies, including support vector machines, neural networks, regression analysis, decision trees, and naive Bayes classifiers, have been implemented utilizing patient data from diverse global sources.

The performances achieved using the corresponding dataset from this study are evaluated in this context. Researchers predominantly employ Support Vector Machine and Neural Network algorithms for classification tasks on this dataset, with a notable emphasis on feature selection methodologies being the primary area of research interest. Polat and Güneş [1] applied kernel F-score feature selection as a preprocessing step in conjunction with the Least Squares Support Vector Machine (LS-SVM), attaining an average accuracy of 77.78%. Tomar and Agarwal [2] employed the Least Squares Twin Support Vector Machine (LSTSVM) alongside the same feature selection method as Polat and Güneş, resulting in an average classification accuracy of 90%. Wang et al. [3] utilized linear kernel SVM classifiers, achieving an accuracy of 83.37%. Lee [4] achieved an accuracy of 87.4% utilizing a Neural Network with an innovative supervised feature selection method based on the bounded sum of weighted fuzzy membership functions (BSWFM) and Euclidean distances. Buscema et al. [5] reported an accuracy of 84.14% using an artificial neural network (ANN) trained with their proposed Training with Input Selection and Testing Algorithm (TWIST), a novel train-test split approach. The Single Hidden Layer Feedforward Neural Networks (SLFNs), applied by Subbulakshmi et al. [6], achieved an accuracy of 87.5%. Given the inherent interpretability of tree models and their capability to ascertain feature significance, they are extensively employed for addressing the classification challenges associated with heart diseases. Chaki et al. [7] used C4.5 Decision Tree and achieved an accuracy of 77.56%. Dangare and Apte [8] built a Decision Tree using the J48 algorithm and obtained a classification accuracy of 96.66%. Elyan and Gaber [9]adopted class decomposition to Random Forest and obtained an accuracy of 82.64%.

A. Logistic Regression

Logistic regression is a classification algorithm derived from the field of statistics used for binary classification. Logistic regression applies the sigmoid function along with a threshold to categorize input data into one of two possible categories or classes. The sigmoid function [10], which is an S-shaped curve, returns a probability score between 0 and 1.

$$\sigma(z) = \frac{1}{1+e^{-z}}, \quad \text{where } z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$

A threshold value is set; probability scores above the threshold are considered as one class _1, while those below are categorized as the other class _2.

B. Tree-based Models

Decision tree is supervised learning algorithms used for both classification and regression tasks. [11] It is a tree-like model where each internal node represents a test on an attribute, each branch represents the outcome of the test, and each leaf node represents a class label (classification) or a value (regression). The learning process constructs the tree by recursively partitioning the data based on the attribute that best separates the classes or minimizes prediction error.Decision trees are particularly valued for their interpretability; the tree structure facilitates straightforward visualization and comprehension of the decision-making process. However, they are prone to overfitting, particularly with deep trees, leading to poor generalization to unseen data. [12] Random Forest is an ensemble learning method, meaning it combines multiple decision trees to produce a more accurate and robust prediction than a single tree could achieve on its own.

C. Support Vector Machine (SVM)

Support Vector Machines aim to find the optimal hyperplane that maximally separates data points of different classes. [13] This is achieved by identifying support vectors—the data points closest to the hyperplane. For non-linearly separable data, kernel functions map the data into a higher-dimensional space where separation becomes linear. SVMs are powerful and effective but suffer from limitations: high computational cost for large datasets, sensitivity to hyperparameter tuning (kernel choice, regularization), and difficulty handling datasets with many irrelevant features or noise. The choice of kernel and feature selection method are crucial and needs careful consideration based on the data's characteristics. [14]

D. Neural Networks (NN)

Neural Networks mimic the human brain's structure, using interconnected nodes (”neurons”) organized in layers. [15] Input data is fed into the input layer, processed through hidden layers where weights and activation functions transform the data, and finally outputted. Learning involves adjusting weights to minimize prediction error using backpropagation, an iterative algorithm that propagates errors backward through the network. ANNs excel at complex pattern recognition. However, they are ”black boxes,” lacking interpretability, and prone to overfitting with insufficient data. Training can be computationally expensive and requires careful hyperparameter tuning.

III. Method

In this section, we will further explore the three different categories of machine learning algorithms (SVM, Tree-based Models, Neural Network) that we mentioned in our literature review in the “Background” section, elaborating on the state-of-the-art techniques in each category.

1. SVM

Polat and Güneş used kernel F-score feature selection (KFFS) on the dataset before feeding it to their SVM model. Different from the traditional F-score, kernel F-score are able to handle nonlinear relationship between features and target by incorporating kernel methods. Specifically, the input space is mapped with RBF or Linear kernel function, thus transformed from non-linearly separable dataset to linearly separable dataset. After calculating F-score, any feature with F-score higher than average F-score will be selected. The kernel F-score feature selection eliminated redundant features from high dimensional feature space. The authors also used a variation of SVM called least square SVM (LS-SVM), which has been proved successful in the field of pattern recognition and regression problems. Proposed by Suykens and Vandewalle [16], LS-SVM is based on the principle of structural risk minimization. As for the results, RBF kernel F-score feature selection improved the classification accuracy of LS-SVM by 3.7% on the Cleveland Heart Disease dataset, reaching 83.7%.

Tomar and Agarwal applied the feature selection method proposed by Polat and Güneş to Least Squares Twin Support Vector Machine (LSTSVM). Twin Support Vector Machine (TSVM), proposed by Jayadeva et al., is a solution of binary classification to reduce computational complexity of traditional SVM. By the factor of 4 [17], thereby being four times faster than traditional SVM. LSTSVM is based on TSVM with better generalization ability and faster computational time. Regarding their result, their model achieved 90% of test accuracy on the Cleveland Heart Disease dataset.

2. Tree-based Model

Dangare and Apte used J48 algorithm to build their Decision Tree model. According to their paper, J48 algorithm gave the highest accuracy on the train set compared with other decision tree algorithms, including CARAT, ID3, C4.5 and CHAID. Specifically, the J48 algorithm uses information gain as its splitting criterion. It recursively partitions the data until a stopping criterion is met, such as maximum depth, all instances in a node belong to the same class. It can also tackle overfitting with pruning techniques, reducing branches that cannot significantly improve performance. Using J48, their Decision Tree achieved an unprecedented testing accuracy of 96.66% on the Cleveland Heart Disease dataset. Despite the promising result, replication on their study is necessary to substantiate their findings.

Elyan and Gaber adopted class decomposition to improve the performance of their Random Forest. Their modification allows Random Forest to be applicable to any classification problem, including single classifier. They first applied k-means clustering to subjects in each class to create a new class-engineered dataset. Then, they apply Random Forest to it. This process iterates while tuning the number of clusters (k parameter).Their Random Forest achieved a test accuracy of 82.64% on the Cleveland Heart Disease dataset.

3. Neural Networks

Lee created a novel supervised feature selection method combined bounded sum of weighted fuzzy membership functions (BSWFM) with Euclidean distance to reduce computational load and improve model accuracy be removing irrelevant features. The process is as follows. During the learning process, all BSWFMs are normalized, and their centers of gravity are determined. Features are removed based on the proximity (Euclidean distance) of their BSWFM centers of gravity. Features with closer distances are removed first, as larger distances represent better feature differentiation and ranking. Therefore, a set of minimum number of features with highest classification performance is selected. As for the test results, Lee achieved a test accuracy of 87.4% with Neural Network on the feature selected Cleveland Heart Disease dataset.

IV. Results & Analysis

In this Section, we will discuss the implementation of our study. Starting from the dataset description and preprocessing, to training and testing of machine learning models.

1. Description of the Dataset

The dataset used in our study was downloaded from UC Irving Machine Learning Repository, Heart Disease Database [18]. The dataset was first introduced in a study by Detrano et al. in 1989, published in American Journal of Cardiology. It has 76 attributes, while most published research refers to using a subset of 14 of them. Additionally, the Cleveland dataset, in particular, is the only one that has been used by machine learning researchers so far. Therefore, because our goal was to implement and evaluate existing techniques, we chose to use the same Cleveland dataset as that used in previous studies.

The Cleveland dataset has 303 entries and 14 columns, 13 of which are features, including resting blood pressure, serum cholesterol, fasting blood sugar, exercise-induced angina and so on. The target column, referring to the presence of heart disease, has the integer values of 0, 1, 2, 3 and 4. The number 0 represents no presence of heart disease and 1 to 4 represent different severity of heart disease. Our study, same as existing experiments, was focused on binary classification, attempting to distinguish presence (values: 1, 2, 3, 4) from absence (value: 0).

2. Experimental Details

In this section, we will introduce the complete process and details of our experiment, from data preprocessing to testing. While the implementation for all models followed the same general process in order to make comparisons, the hyperparameter tuning for each model is different.

2.1 Preprocessing

Our data preprocessing phase has three steps to get the original dataset ready for training.

The first step is replacing missing values. After going through the dataset, missing values were found in either ”ca” or ”thal” column and rows 87, 166, 192, 266, 287, 302. We replaced the missing values with the mode in its corresponding column.

The second step is changing the labels. As we plan to do binary classification for the presence of heart disease, we changed the labels ”1, 2, 3, 4” representing different levels of severity to ”1” meaning the presence of heart disease. “0” still represents the absence of heart disease. We plotted a bar chart to visualize the distribution of our dataset, which is shown in Figure 1 below.

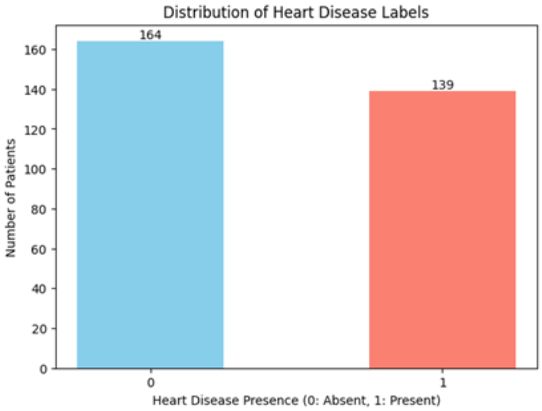


Fig. 1. Dataset label distribution

As we can see, our dataset is slightly imbalanced in which heart disease was present in 54% (164 out of 303) of the samples while absent in 46% (139 out of 303). According to a review by Kotsiantis et al. [19], imbalanced dataset may cause the classifiers to classify data to the majority class. This is a major problem in anomaly detection where datasets where the unequal distribution could be 100:1, 1000:1 etc. [20]. In our case, our imbalance was far from 100:1. Therefore, we chose to preserve all the entries avoiding information loss for our small dataset.

TABLE I
BEST HYPERPARAMETER COMBINATIONS

Model	Hyperparameters
Logistic Regression	C = 1
SVM (SGDClassifier)	alpha = 0.1, eta0 = 0.1, learning_rate = 'constant'
Decision Tree	criterion = 'gini', max_depth = 4, min_samples_leaf = 3, min_samples_split = 10
Random Forest	max_depth = None, min_samples_leaf = 5, min_samples_split = 12, n_estimators = 220
Two-layer Neural Network	alpha = 0.01, dropout_rate = 0.2, learning_rate = 0.01, hidden_neurons = 15

Before step 3 and 4, we split the dataset to train and test set, accounting for 75% and 25% of the whole dataset respectively.

The third step is data scaling. We used normalization to scale all the features to the same range of 0 to 1. This could prevent the dominance of features with larger values.

The fourth step is dimension reduction. We applied PCA which kept 80% of variance to the scaled features, reducing the dimensionality from 13 original features to 6 principal components. Each principal component is a linear combination of the original features.

We would like to stress that data scaling and dimension reduction was applied to train and test sets separately. Specifically, take PCA as an example, we imported PCA from Scikit-learn and trained it only on the train set of our data. Then we applied it to both train and test sets to reduce their dimension. In this way, the information in test set will not be leaked before we test the models.

2.2 Training

After preprocessing, we were ready to train the models on our data. We implemented overfit check before starting a full-scale training. Then we applied K-fold cross-validation and grid search to tune the hyperparameters. Finally, the models with optimum hyperparameters were trained on the full train set.

2.2.1 Overfit Check

We applied overfit check to all models before full-scale training to detect implementation errors, avoiding the waste of time on pointless training. We picked 5% from the training data and run our models with zero regularization. Loss of zero was achieved on Logistic Regression, SVM, Decision Tree and Random Forest. For the two-layer Neural network, loss was less than 0.01 and a 100% train accuracy was achieved on the small portion of train set.

2.2.2 Hyperparameter Tuning

We applied Grid Search with Cross Validation as our hyperparameter tuning technique. The process is as follows.

The train set was split into five folds with equal size and for each model, we designed a specific hyperparameter grid, including all the hyperparameters we plan to tune and their respective values that we would be experimenting on. For each hyperparameter combination, the model was trained on 4 folds then evaluated on the remaining fold. This was repeated 5 times, with each fold serving as validation once. The best hyperparameter combination was the one with the highest average validation accuracy across all k folds. Finally, a new model was trained using the entire dataset and the best hyperparameter combination. The best hyperparameter combination for each model is shown in Table 1.

2.2.3 Difference with Previous Studies

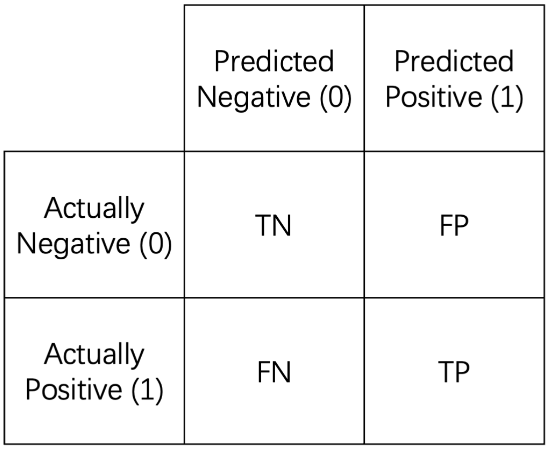
Our hyperparameter selection is based on the model description on Scikit-learn and several existing papers, each implementing several classification models on the Cleveland Heart Disease Dataset. The main differences are as follows.

For SVM, the model used by Kibria and Matin [21] is "SVC" with nonlinear kernel while we used "SGDClassifier" with hinge loss which gives a linear SVM as we plan to compare it with Logistic Regression, which is also a linear model.

For Neural Network, we built a two-layer fully connected neural network same as Kibria and Matin while using Adam optimizer instead of their Stochastic Gradient Descent (SGD).

3. Results

After the models were trained on the entire train set using the best hyperparameter combination that we found, test results were calculated based on the performance on test set. The models were evaluated based on several metrics and comparison was made.



	Predicted Negative (0)	Predicted Positive (1)
Actually Negative (0)	TN	FP
Actually Positive (1)	FN	TP

Fig. 2. Confusion matrix (illustration)

TABLE II
MODEL EVALUATION (SCORES ARE GIVEN AS PERCENTAGES)

	Tuning Time	Validation Accuracy	Train Accuracy	Test Accuracy	Test Precision	Test Recall	Test Specificity
Logistic Regression	0.16s	81.93	81.94	82.89	81.58	83.78	82.05
SVM	2.80s	83.25	80.62	85.53	86.11	83.78	87.18
Decision Tree	6.67s	82.36	86.78	82.89	90.00	72.97	92.31
Random Forest	196.82s	84.56	88.11	82.89	85.29	78.37	87.18
Neural Network	3 hours	85.02	83.70	80.26	84.38	72.97	87.18
Average	-	83.42	84.23	82.89	85.47	78.37	87.18

3.1 Metrics for Evaluation

The performance of each model was measured by validation and training accuracy and a confusion matrix on the test result. The confusion matrix (Figure 2) summarizes the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions made by a classifier. In our case, positive is presence of heart disease and negative is absence of heart disease. Three metrics are further calculated including Accuracy (TP+TN/TP+TN+FP+FN), Precision (TP/TP+FP), Recall (TP/TP+FN) and specificity(TN/TN+FP).

Our primary focus was on the recall as interpreted from the confusion matrix, prioritizing the correct identification among the patients with heart disease. This is crucial for minimizing missed diagnoses in practice.

3.2 Results and Comparison

The results are tabulated in Table 2. As we can see, SVM was the most effective model with the highest test accuracy (85.53%) and recall (83.78%). Logistic Regression had the same recall as SVM but a lower test accuracy (82.89%). In addition, Tree models, including Decision Tree and Random Forest, had higher training accuracy but lower test accuracy compared with SVM, suggesting a slight overfitting on train set. Moreover, Decision Tree got the highest Specificity (92.31%) but lowest Recall (72.97%), showing that this model is more sensitive to the absence of heart disease (label: 0) than its presence (label: 1). For all five models, the average recall (78.37%) was also lower than the average specificity (87.18%). This could be caused by the imbalance of our dataset containing slightly more entries without heart disease, causing the models, on different tendencies, to classify data to the majority class (label: 0). Lastly, the Two-layer Neural Network produced the least satisfying result despite of the longest tuning time .

4. Limitations and Challenges

All five models were experimented on the same train and test split on a relatively small dataset. Thus, we cannot rule out the possibility of ”lucky split” for high performance models. Computational power is the major challenge when tuning the Neural Network. A single run of Grid Search and Cross Validation tuning four hyperparameters (81 combinations) took more than three hours, restricting us from exploring Neural Networks with more hidden layers and neurons.

V. CONCLUSIONS

Our study was focused on implementing existing machine learning techniques, to classify the presence of heart disease on the Cleveland Heart Disease dataset. The features of the dataset were normalized before dimensionally reduced with PCA, preserving 80% of variance. The training section was composed of overfit check for implementation errors and Cross Validation combined with Grid Search for hyperparameter tuning. Each model with optimum hyperparameter was then trained on the entire train set. Finally, all models were tested on the test set, and evaluated based on hyperparameter tuning time, validation and training accuracy and confusion matrix on test results.

Five models are implemented, including Logistic Regression, linear SVM, Decision Tree, Random Forest and two-layer Neural Network. According to our evaluation, linear SVM was the most effective model with the highest accuracy and recall. We also tested linear SVM, Decision Tree and Random Forest without PCA and the test accuracy of Random Forest notably improved.

We acknowledge that our dataset was constructed in 1989, more than 20 years ago and the size of 303 entries was relatively small. Larger datasets with new patient information could be used to train models with better practicality.

Additionally, the linear models (Logistic Regression, SVM) generally performing better than nonlinear models (Decision Tree, Random Forest, two-layer Neural Network) suggested a potential linear relationship between the features and presence of heart disease. However, further studies on Deep Learning models such as LSTM, CNN are necessary to verify this finding.

REFERENCES

[1] Kemal Polat and Salih Güneş. A new feature selection method on classification of medical datasets: Kernel f-score feature selection. *Expert Systems with Applications*, 36(7):10367–10373, 2009.

[2] Divya Tomar and Sonali Agarwal. Feature selection based least square twin support vector machine for diagnosis of heart disease. *International Journal of Bio-Science and Bio-Technology*, 6:69–82, 04 2014.

[3] Shi jin Wang, Avin Mathew, Yan Chen, Li feng Xi, Lin Ma, and Jay Lee. Empirical analysis of support vector machine ensemble classifiers. *Expert Systems with Applications*, 36(3, Part 2):6466–6476, 2009.

[4] Sang-Hong Lee. Feature selection based on the center of gravity of bswfms using newfm. *Engineering Applications of Artificial Intelligence*, 45:482–487, 2015.

[5] Massimo Buscema, Marco Breda, and Weldon Lodwick. Training with input selection and testing (twist) algorithm: A significant advance in pattern recognition performance of machine learning. *Journal of Intelligent Learning Systems and Applications*, 5:29–38, 02 2013.

[6] C.V. Subbulakshmi, S.N. Deepa, and N. Malathi. Extreme learning machine for two category data classification. In *2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pages 458–461, 2012.

[7] Dipankar Chaki, Amit Das, and Zaber Moinul. A comparison of three discrete methods for classification of heart disease data. *Bangladesh Journal of Scientific and Industrial Research*, 50:293, 12 2015.

[8] Chaitrali Dangare and Sulbha Apte. Improved study of heart disease prediction system using data mining classification techniques. *International Journal of Computer Applications*, 47:44–48, 06 2012.

[9] Eyad Elyan and Mohamed Gaber. A fine-grained random forests using class decomposition: an application to medical diagnosis. *Neural Computing and Applications*, 27:1–10, 11 2016.

[10] Abdelhamid ZAIDI. Mathematical justification on the origin of the sigmoid in logistic regression. *Central European Management Journal*, 30(4):1327–1337, 2022.

[11] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1:81–106, 1986.

[12] Tom Dietterich. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, 27(3):326–327, 1995.

[13] Vikramaditya Jakkula. Tutorial on support vector machine (svm). *School of EECS, Washington State University*, 37(2.5):3, 2006.

[14] Arti Patle and Deepak Singh Chouhan. Svm kernel functions for classification. In *2013 International Conference on Advances in Technology and Engineering (ICATE)*, pages 1–9, 2013.

[15] AD Dongare, RR Kharde, Amit D Kachare, et al. Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(1):189–194, 2012.

[16] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9:293–300, 1999.

[17] Jayadeva, R. Khemchandani, and Suresh Chandra. Twin support vector machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, 2007.

[18] Andras Janosi, William Steinbrunn, Matthias Pfisterer, and Robert Detrano. Heart Disease. UCI Machine Learning Repository, 1989. DOI: https://doi.org/10.24432/C52P4X.

[19] Sotiris Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30:25–36, 11 2005.

[20] Akila Somasundaram and U. Srinivasulu Reddy. Data imbalance: Effects and solutions for classification of large and highly imbalanced data. In *Proc. of 1st International Conference on Research in Engineering, Computers and Technology*, 01 2016.

[21] Hafsa Binte Kibria and Abdul Matin. The severity prediction of the binary and multi-class cardiovascular disease - a machine learning-based fusion approach. *Computational Biology and Chemistry*, 98:107672, 2022.