An Efficient Hybrid Classifier for Prognosing Cardiac Disease

Richa Sharma
Amity School of Engineering and Technology, Amity University, Noida, Uttar Pradesh, India.
E-mail: richasharma649@gmail.com

Shailendra Narayan Singh
Amity School of Engineering and Technology, Amity University, Noida, Uttar Pradesh, India.
E-mail: snsingh36@amity.edu

Received September 30, 2021; Accepted December 21, 2021
ISSN: 1735-188X
DOI: 10.14704/WEB/V19I1/WEB19338

Abstract

Machine learning (ML) is a powerful tool which empowers the practitioners for predictions upon any existing or real- time data. Here, the Machine first understands the valuable patterns from the dataset and then uses that information to make predictions on the unknown data. Further, classification is the commonly used machine learning approach (ML-Approach) to make such predictions. The objective of this work aims to design and development of an ensemble classifier for prognosing cardiovascular disease (heart disease). The developed classifier integrates Support Vector Machine (SVM), K–Nearest Neighbor (K-NN), and Weighted K-NN. The applicability of ensemble classifier is evaluated on the Cleveland Heart disease dataset. Some other classifiers such as Logistic Regression (LR), Sequential Minimal Optimization (SMO), K-NN+Weighted K-NN are also implemented on the same dataset to make the performance analysis. The results of this study depict the significant improvement in the Sensitivity and Specificity parameter.

Keywords

Cardiovascular Disease (CV-Disease), Classification, Heart Disease Prediction, Support Vector Machine, K – Nearest Neighbor.

Introduction

CV-Disease is the most prominent bases of mortality around the globe. Heart diseases mainly affect humans during old or middle age, and in most cases, it leads to high complication (Centers for Disease Control and Prevention, 2008; Natl Vital Stat Rep, 2012). The statistics given by the world health organization depict this CV-Disease is responsible for 24 percent of deaths happing's in India (Mackay J, 2004; Chauhan Shraddha, 2015; Kaan

Uyar Ahmet Ilhan, 2017). On the other hand, these diseases are the primary cause for half of the deaths in other developed countries as the United States (Patil SB, 2009). Further, the statistics about the deaths due to CV-Disease show that approximately 17 million humans die every year across the world (Fida Benish; Vasighi Mahdi, 2013). CV-Disease seems to be a hazardous disease based on the above numbers. Therefore, CV-Disease / heart disease prediction at the earlier stage will surely help the medical practitioners save valuable human life (B. Al-Hamadani, 2016; Z. Al-Makhadmeh, 2019). Identifying heart disease is quite challenging due to numerous determinant conditions associated such as age, diabetes, sex, high B.P (Blood Pressure), family history, pulse rate, obesity, cholesterol, physical inactivity, and many other risk attributes (Lee Heon Gyu, 2007; Sudhakar K, 2014; Nahar J, 2013; Ali Khazaee, 2019). It is not feasible to estimate the severity of problem manually on the basis of numerous risk attributes (F. Ali, 2019; F. Ali, 2017; A.N.G. Bhuvaneswari, 2013; R. Davoodi, 2018). However, ML approaches are many practical to predict any disease in the early stages (C.B.C. Latha, 2019). Various classification mechanisms are adapted to make the early prediction of heart disease among people. K-NN, Naïve Bayes, SVM, Genetic Algorithm, and Decision Trees, etc., are the most commonly used classification for disease prediction (S. Mohan,2019). Further the research paper is described as follows: Section 2 highlights the existing approaches for heart disease prediction; Section 3 represents the developed ensemble classifier. Section 4 contains description of experiments performed on the Cleveland heart disease dataset. The Conclusion of research work and the future scope are mentioned in section 5.

Literature Review

This section highlights the existing ML-Approaches used by different authors in their researches available in the literature. Some taxonomy like classification of heart disease, heart disease prediction, classification approaches for cardiac disease prediction, etc., are focused to find the relevant studies. Various research studies are summarized for further research.

Unable to predict the cardiovascular disease in the past time is the main reason for excessive mortality rate because of heart disease worldwide. Many authors argued that ML-Approaches could act as a powerful tool to overcome this problem in this context. However, it is not possible to predict heart diseases using ML-Approaches with 100% accuracy. Still, these approaches exhibit high accuracy during disease prediction. Some ML approaches used for disease prediction in the past are SVM, Logistic regression, Naive Bayes, SQM, Decision tree, etc. (Xing Yanwei, 2007) compared the accuracy of different approaches such as SVM, Neural network, and decision tree. The authors claimed SVM as the best prediction approach with good accuracy in comparison with other methods used in this study. (Thenmozhi K, 2014) compared various classification approaches and suggested using decision tree classifiers for heart disease prediction. The author observed decision tree classification as high accuracy along with it's simple implementation. Similarly, (Jyoti Soni Ujma Ansari, 2011) argued classifier Naive Bayes produces most efficient prediction when integrated along neural network and decision tree.

Later, some authors developed an ensemble classifier to achieve high accuracy during heart disease prediction (Singh Jagwant, 2016). As progress goes on, the researchers move towards using nature-inspired optimization algorithms and the available classifiers (Kaan Uyar Ahmet Ilhan, 2017). Some of the nature-inspired algorithms used for heart disease prediction are the Genetic Algorithm, Particle Swarm Optimization, Firefly, etc. (Kaan Uyar Ahmet Ilhan, 2017; Ali Khazaee, 2013; Verma L, 2016; N.C. Long, 2015). Meanwhile, some authors suggested using Fuzzy Set Theory (FST) along-with the existing classifiers and nature-inspired algorithms. The use of the FST results in the development of fuzzy-based models such as co-active neuro-fuzzy inference system (CANFIS), Adaptive Neuro-Fuzzy Inference System (ANFIS), etc., for prognosing cardiac problem (Latha Parthiban, 2008; G. Manogaran, 2018). Deep study of the available literature reveals that many approaches have been used for heart disease prediction. Although, most of the methods are capable of such predictions. Yet, high accuracy is the most concern issue. So, presented research work aims to design and implement an ensemble classifier that definitely improves the prediction accuracy. The developed classifier is based on SVM followed by K-NN and weighted K-NN.

Material and Methods

This section aims at the development of an ensemble classifier using SVM, K-NN, and weighted K-NN. Further, it is divided into 3 subsections as (i) SVM, (ii) K-NN, (iii) weighted K-NN, and (iv) proposed ensemble classifier

Support Vector Machine (SVM)

SVM is one among powerful supervised learning methods which are applied over several regression and classification-based problems. Over the past period, SVM has been implemented to achieve noble generalization performances in a wide variety of classification problems. SVM’s works on the notion of decision planes, and these planes decides the decision boundaries (P.Y. Lau, 2005; S. Mohanapriya, 2013; M.M. Subashini, 2016). A plane which separates the input data having non-identical class representatives. Figure 1 shows an example with objects belonging to either class A or B. Here, the boundary is defined by the separating line, and the right-side things belong to class A while the left side things belong to class B. Figure 2 shows the basic concept behind the working of SVM. The original input objects are mapped or repositioned with the help of kernel. A group of mathematical functions is called as kernels, and rearrangement of things is called mapping. Commonly used kernels include (M.M. Subashini, 2016; J. Sachdeva, 2011):

Linear Kernel Function: $K(m, n) = m. n$

Radial Basis Function (Gaussian) Kernel: $K(m, n) = e^{-} \frac{|m-n|^2}{2\sigma^2}$
Polynomial Kernel: $K(m, n) = (m * n + 1)^d$

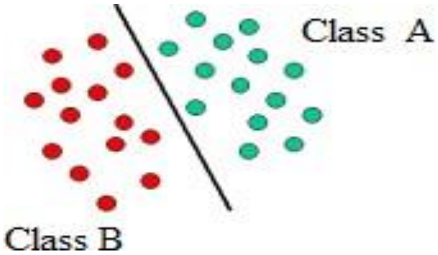

Fig. 1 Linear classifier

The depicted objects on right side of figure 2 are separable linearly in this new arrangement, avoiding separation by designing the complex curve shown on left side of figure. In this scheme, aim is identifying an ideal boundary line that can easily separate two classes. i.e., we need to map the high dimensional nonlinear feature space to the linearly separable input space and identify a separating hyper-plane or boundary having maximum margin between the two classes in their feature space. Figure 3 shows SVM separating class A and class B (S. Chandra, 2009; N. Zhang, 2009).

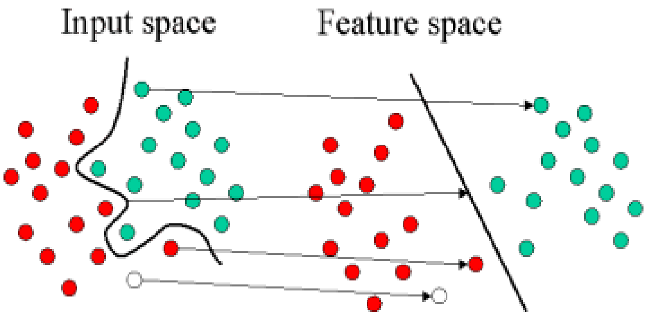

Fig. 2 SVM Separation

SVM can be optimally used for multiclass classification problems. SVM is one of the classifiers with the distinct characteristic of finding the optimal boundary or hyper-plane such that the expected error is minimized. In SVM, instead of reducing the empirical risk calculated from the training dataset, structural risk minimization is performed to achieve good generalization. The width of a margin between the classes defines the optimization criterion, while SVM training focuses to identify the hyper-plane separation with the maximum margin value.

The appropriate kernel function is crucial while training the SVM for classification problems because it is the kernel that drives the SVM and allows the mapping of nonlinear input feature space to a higher dimensional linear feature space. Here, we have used the linear kernel function for the classification.

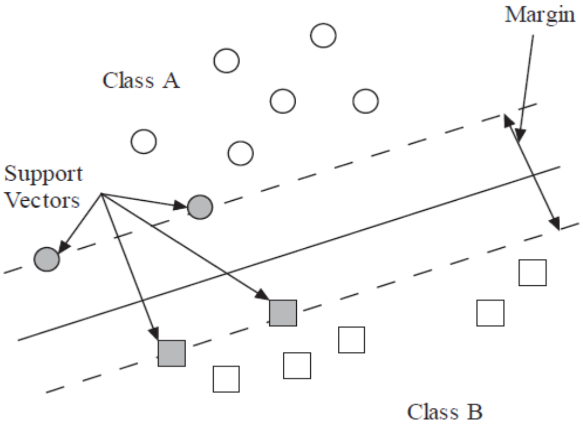

Fig. 3 SVM separating class A with Class B

SVM is a linear classifier which uses a linear discriminant function, Let X be an unknown sample or feature classifier decides whether this unknown instance belongs to Class A or Class B.

Linear discriminant function $g(x) = W^t x + b$

Where,
W: weight vector b: bias term
x: input feature vector
t: transpose of weight vector

In 2-D space a straight line can separate the classes, for that $g(x) = W^t x + b = 0$, where as in 3 D space planes are used for representation but when the dimensions are increased more than three concept of hyperplane is introduced. Where W is the perpendicular to hyperplane and it determines the direction of hyperplane in space of dimensions say d, b determines the location of hyperplane in space of dimension d, b bias the position of hyperplane in space of dimension d.

If $x1$ is at the positive side of hyperplane then,

$$g(x1) = W^t x1 \ + b > 0$$

If $x1$ is at the negative side of hyperplane then,

$$g(x1) = W^t x1 \ + b < 0$$

And if $x1$ lies on hyperplane then,

$$g(x1) = W^t x1 \ + b = 0$$

SVM divides the hyperplane in two half's one is positive and other is negative If, $W^t x1 + b > 0$ it is classified in Class A, $x1 \in$ Class A

Else, $W^t x1 + b < 0$ it is classified in Class B, x1 ∈ Class B

In the next step we have to train 'W' and 'b' and for the training purpose we need to consider training samples, every training sample trains iteratively. We start training 'W' and 'b' by taking initial values of it. If it is a sample of class A it gives $W^t x1 + b > 0$ which is fine but if not, we must modify the values of 'W' and 'b' accordingly, similarly when it gives $W^t x1 + b < 0$ it is considered in class B if not again we have to adjust the values of 'W' and 'b' in such a way that it moves to the negative side of hyperplane.

A desirable separating plane is required which is not biased to either of the class. It is safe if the boundary distance is large, we need to maximize the distance between the separating planes.

$W^t . xi + b > 0$ if $xi \in$ Class A
$W^t . \text{xi} + \text{b} < 0$ $if\ xi \in Class\ B$
(xi, yi), when xi belongs to class A then belonging $yi\ is$ positive, similarly when xi belongs to class B then belonging
yi is negative, yi can be either +1 or -1.

$yi\ (wi \cdot xi \ + b) > 0$, this is always positive even if xi belongs to class B.
Consider, p as an unknown feature vector
If $w.\ p + b > 0$ it goes to class A If $w.\ p + b < 0$ it goes to class B
Aim is to maximize the distance of separation boundary to each feature vector. For this instead of writing $W^t x + b > 0$, we write $W^t x + b > \vartheta$.

Where ϑ is margin which measures the distance from xi to separating boundary.
$w.x + b = 0$, distance from plane to x is $\frac{w\cdot x+b}{|w|} \geq v$
$w.\,x$ + b ≥ ϑ. |w|, by scaling ϑ. |w|=1
$w.\,x$ + b ≥ 1 $if\ x \in Class\ A$
$w.\,x$ + b ≤ −1 $if\ x \in Class\ B$

yi ($wi \cdot xi$ + b) ≥ 1, if xi $is\ support\ vector$ = 1 $and\ if\ xi$ $is\ not\ support\ vector$ ≠ 1, Dependent feature vectors are called support feature vectors which is the closest vectors near the hyperplane belongs to each class.

yi ($w \cdot xi$ + b) = 1, if xi is support vector. To maximize the margin, we should minimize ‘W’ and maximize 'ϑ’.
Maximize W

$\phi(W) = W^tW = \frac{1}{2}W.W$, for this constant is required
yi ($wi \cdot xi$ + b) = 1 is constant for maximizing W.

Now, the constant optimization problem is converted to constant free optimization problem Lagrange’s multiplier.

L(w, b)= ½(w.w)=∑$\alpha i[yi[wi.\ xi + b] - 1]$, minimize ‘w’ and maximize ‘b’.
$\alpha i\ \ is\ Lagrange's\ multiplier$

$L(w,b) = \frac{1}{2}(w.w) - \sum \alpha_i y_i (w_i \cdot x_i) - \Sigma \alpha_i y_i b + \sum \alpha_i$
$\frac{\partial L}{\partial b} = -\sum x_i y_i = 0$
$\sum_{|=1}^{m}$ αiyi = 0 …. (1), m denotes values of feature vector
$L(w,b) = \frac{1}{2}(w.w) - \sum \alpha_i y_i (w_i \cdot x_i) - \Sigma \alpha_i y_i b + \sum \alpha_i$
$\frac{\partial L}{\partial w} = -\sum \alpha_i y_i x_i = 0$

$$W=\sum_{|=}^{m}\alpha_i y_i x_i \qquad (2)$$

m denotes the value of samples used in training

Put eq(1) and eq(2) in original Lagrange’s expression

$$L = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\alpha_i \alpha_j y_i y_j \left(x_i . x_j\right), \alpha_i \geq 0, \sum_{i=1}^{m} \alpha_i y_i = 0$$

$$D(z) = \operatorname{sgn}\left(\sum_{i_2=1}^{m} \alpha_j y_j x_j \cdot z + b\right)$$

if sgn is positive z is classified to class A and if sgn is negative z is classified to class B

When features are not linearly separable, we have to cast those in higher dimensional space by Radial basis function or kernel.

$$b = \frac{1}{2}[min\left(\sum_{i|yi=+1} \alpha_i y_i \left(x_i x_j\right)\right) + max\left(\sum_{i|yi=-1} \alpha_i y_i \left(x_i x_j\right)\right)]$$

This value of b goes to D(z) to classify unknown feature vector z. When we have more than 2 class problem, we require multiple SVM’s.

Pseudo-code: Support Vector Machine Classifier Input: Cleveland Heart disease dataset D
Output: Performance Measures based on Confusion Matrix,
Train Dataset Size = 0.7
Test Dataset Size = 0.3
X: Number of Samples
Y: Labels, where Y ∈ {Class Normal or Class Abnormal} Model: SVM classifier model
Test the model Calculate Scores
Compute performance Measures Validate Model

Conventional K-NN Classifier

It is among widely used algorithm of classification bring in light by Hodges and Fix (F. Ali, 2017). It is an instance-based machine learning technique, and since it doesn't use any assumptions regarding the distribution of data, it is also known as non-parametric lazy algorithm. Concerned classification technique opted the nearest training examples in the feature space and estimated K nearest neighbors (F. Ali, 2019). During this step, commonly used class among the K neighbors is allocated a course for the new data. However, the main disadvantage is the huge memory is required for storing the whole sample, which in turn increases the response time on a sequential computer (N. Zhang, 2009). Despite this significant memory requirement issue, it proved good classification problems on various datasets (Lee, 2000).

KNN classifies the new instance values based on similarity measures. In an instance space each instance has it's corresponding (x,y) values whenever a new instance enters in a instance space it finds the closest instance value of y and x. for test instance, it finds the most similar neighbor instance.

In training phase, model learns by saving examples in some data structure in this phase it mainly aims to store the instances in next phase prediction phase, test instance finds the K-training examples, consider test instance xt it finds the training example ($x1$, $y1$) which is closest to xt. Now predict $y1$ as the output yt. In general, instead of considering single training example we consider k training samples.

{($x1$, $y1$), ($x2$, $y2$), ……….., (xk, yk)} for classification among y1, y2 ,……., yk predicts most frequent majority class. For regression average among y1, y2 ,……., yk is considered.

Distance function is used to calculate the distance of instance from nearest neighbor, in general standard distance function is used.

$$xi = (xi1, xi2, xi3 \dots \dots \dots \dots xiN)$$
$$xj = (xj1, xj2, xj3 \dots \dots \dots \dots xjN)$$

Euclidian Distance (xi , xj)=

$$\sqrt{\sum_{n=1}^{m} \left(x_{im} - x_{jm}\right)^2}$$

It selects the point which has smallest Euclidian distance.

Pseudo-code K-NN classification algorithm: Input: Cleveland Heart disease dataset D

Output: Performance Measures based on Confusion Matrix, Train Dataset Size = 0.7

Train Dataset Size = 0.7

Test Dataset Size = 0.3

Model: K-NN Classifier Model

```
For all the unknown data samples (i)
For all the known data samples (k)
Estimate the distance between the i and k,
End For
Discover the k-smallest distances
Prioritize the corresponding known data samples
```

Assign unknown data sample (i) to the frequently appearing class End For

Compute performance Measures Validate Model

Construction of Weighted K-NN Classifier

In this step, a weighted k-Nearest Neighbor Classifier is built. The decision rule of the weighted K-NN method is based on a metric called item strength rather than a majority vote. Any test point to be evaluated is allocated the class label with the highest summation of item strength. Here, the training data samples are broken into smaller subsets. A classifier model is designed for every subgroup, and thereafter a weighting approach used for classification of testing samples. On the basis of estimation, weights on the testing data will determine the performance of the classifiers. Henceforth, the weighted-k-NN is otherwise termed as the 'memory-based classifier'. Since it deals with the continuous attributes, so distance estimation in between the data is easily intended using the Euclidean distance formula. Let the samples in the first subset is represented as $(x1,\ x2,\ \dots\ \dots,\ xp)$, and the samples in second subset is represented as, $(y1,\ y2,\ \dots\ \dots\ ,\ yq)$. Then, the Euclidean distance between two subsets is given as,

$$Distance = \sqrt{(x_1 - y_1)^{2} + (x_2 - y_2)^2 + \cdots + (x_p - y_q)^2} \qquad (1)$$

Each data in the subsets are analyzed using the above equation. It is found that the most significant values create an issue with the smallest values. For example, cholesterol value ranges from 100 to 190, whereas; the age value ranges from 40 to 80. Thus, the cholesterol value is normalized, and therefore, it helps to calculate the distance between the subsets easily. The weighting technique is employed to aggregate the distance values obtained from subsets to make proper decision-making processes. The decision is made by aggregating all the weights, and thus, the class having maximal value of weight is considered the concluded prediction value.

Pseudo-code weighted K-NN classification algorithm: Input: Cleveland Heart disease dataset **D**

Output: Performance Measures based on Confusion Matrix, Train Dataset Size = 0.7

Train Dataset Size = 0.7

Test Dataset Size = 0.3

Model: Weighted K-NN Classifier Model

For all the unknown data samples (i)

For all the known data samples (k)

```
Estimate the distance between the i and k,
End For
Class prediction, using distance-weighted voting.
End For
Compute performance Measures Validate Model
```

Ensemble KNN Classifier

As discussed in step (a), the outliers of the SVM classifier will be further sorted out by the ensemble of weighted k- NN and the k-NN classifiers. Here, final output obtained from the hyperplane of SVM is fed into the weighted k-NN and the conventional k-NN classifiers to achieve the higher rate of performance of SVM classifiers in respect of accuracy and better decision-making process.

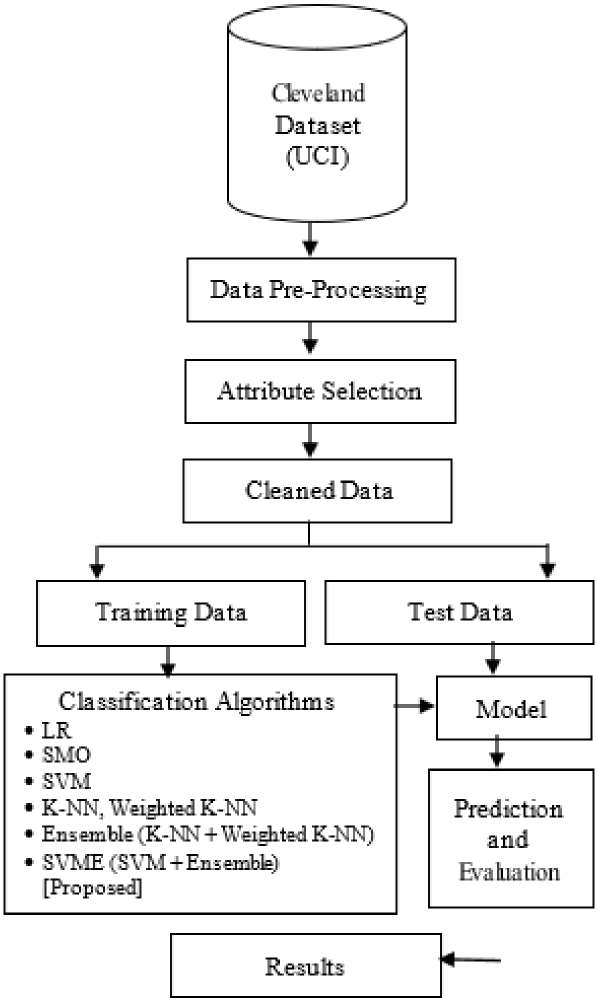

Figure 4 Working of Proposed ensemble Classifier

Proposed Algorithm

Platform used for implementation: jdk 1.8 (version 8)

```
Input: KNN, WKNN, SVM, ENSEMBLE
Output: SVME
Assumption -> Start with KNN(k)=3, KNN(k)=5, WKNN=5 Import libSVM

Step 1: Initialize KNN[k] = KNN [3]
Step 2: Initialize KNN[k] = KNN [5]
Step 3: Initialize WKNN[k] = WKNN[5]
Step 4: Var Margin=0.3
Step 5: SVME (Margin, Ensemble)
If (Margin>0.3)
call SVM
else
ENSEMBLE
Step 6: Initialize ENSEMBLE = [KNN, KNN, WKNN] For KNN(k)=3
KNN(k)=5
WKNN(K)=5    //k=5 by majority vote
Step 7: Execute SVME (Margin, Ensemble) -> SVME (0.3, {KNN[3], KNN[5],WKNN[5]})
Step 8: An SVME Ensemble with KNN=3, WKNN=5, SVM=5 and Margin=0.3 yielded the best accuracy.
```

Note: Above mentioned algorithm has been tested with different values for k and margin, best result is obtained with k=5 and margin value 0.3

Performance Analysis of Proposed Ensemble Classifier

This section discussed in this paper aims to effectively evaluate the proposed hybrid classifier, i.e., the integration of SVM, K-NN, and weighted K-NN. This section is further sub-divided into four sub-sections as described below.

Dataset Description

The Cleveland heart disease dataset is taken from the UCI repository for evaluating the proposed ensemble classifier. This dataset consists of 75 attributes concerned with heart disease. The detailed analysis shows the missing values for some of the attributes in this dataset. So, data pre-processing is done to get the refined dataset for further processing. A sampling process is used to pre-process the dataset in consideration. Now, identification of relevant and irrelevant attributes from the Cleveland dataset is performed. Finally, 13 attributes out of 75 are shortlisted, as provided in table 1.

Table 1 Selected Attribute description from the Cleveland Heart disease dataset

S. No.	Attribute Name	Attribute Type	Range of Values	Description
1	Age	Numeric	29 to 79	Patient's Age in years
2	Trestbps	Numeric	94 to 200	Resting Blood Pressure in mm Hg
3	Chol	Numeric	126 to 564	Serum Cholesterol in mg/dl
4	Thalach	Numeric	71 to 202	Maximum achieved Heart rate
5	Ca	Numeric	0 to 3	Number of major vessels colored by fluoroscopy
6	OldPeak	Numeric	1 to 3	ST depression induced by exercise relative to rest
7	Cp	Encoded String	1, 2, 3, 4	Type of Chest Pain [Typical type 1 Angina – 1, Atypical Type Angina – 2, Non-angina pain – 3, Asymptomatic – 4]
8	Slope	Encoded String	1, 2, 3	Slope of the peak exercise ST segment
9	Thal	Encoded String	3, 6, 7	Normal – 3, Fixed Defect – 6, Reversible Defect – 7
10	Restecg	Encoded String	0, 1, 2	Resting Electrocardiographic results
11	Sex	Binary	0, 1	Patient's Gender [Female – 0, Male – 1]
12	Fbs	Binary	0, 1	Fasting Blood Sugar in mg/dl
13	Exang	Binary	0, 1	Exercise Induced Angina

Performance Measure

Performance measures are the matrices that describe the overall performance of the classifier. The four performance measures as Sensitivity, Specificity, Classification Accuracy, and F-Measure, are considered in this study. All these performance measures work on the confusion matrix. This matrix consists of the measures as true positive (TP), false positive (FP), true negative (TN), and false negative (FN). The description of the four performance measures considered in this study for evaluating the efficiency of various classifiers is given in table 2.

Table 2 Performance Measures Taken in this Study

S. No.	Performance Measure	Description	Mathematical Model
1	Sensitivity	It is the division of real positive values that accurately predicts the positive	Sensitivity = (TP) / (TP + FN)
2	Specificity	It is the division of predicted negative values that accurately calculates real negative class.	Specificity = (TN) / (TN + FP)
3	Classification Accuracy	It is accurately predicted values to the total observation. It defines the ability to distinguish normal and abnormal cases.	Accuracy = (TP + TN) / (TP + TN + FP + FN)
4	Balanced F-measure	It conveys the balance between precision and recall.	F-measure: 2*((precision * recall) / (precision + recall))

10-Fold Cross-Validation

The data samples in consideration is distributed into training data samples and testing data samples using 10-fold cross-validation technique for evaluating hybrid model's performance stability. Sensitivity, specificity, and accuracy measures are used to validate the performance of introduced model. Sensitivity measure is determinant of positive sample values that are correctly classified as positive (e.g. the ratio of cardiac patient are classified as diseased). Specificity measure is determinant of negative sample values that are correctly classified as negative (e.g., the balance of healthy people that are classified as healthy). Accuracy is determined when the samples in consideration are classified correctly.

Performance Evaluation of Proposed Ensemble Classifier

At this step, various classifiers have been implemented on the taken dataset. Some algorithms perform well in terms of one performance measure but not so much good on other performance measures. Aim of ensemble classifiers is to enhance the overall performance of weak classifiers. The present work aims to use an ensemble classifier based on SVM followed by K-NN and weighted K-NN. 10-fold cross-validation technique is used for the performance evaluation of different classifiers. The classifiers, namely 'SMO' and 'LR' identify the 218 and 219 correct instances out total 303 instances present in the dataset. On the other hand, 'SVM' and 'Weighted K-NN' perform well compared to 'SMO' and 'LR'. Further, It is observed that the proposed ensemble classifier out- perform among all the classifiers considered here in term classification achieved. The performance statistics of various classifiers such as ‘SVM’, ‘LR’, ‘SMO’, ‘K-NN’, ’Weighted K-NN’, ‘K-NN+ Weighted K-NN', and proposed ensemble classifier based on individual performance measure is shown in figure 5, 6, 7, and 8. The compiled comparative statistics of different classifiers based on the four performance measures here are given in table 3.

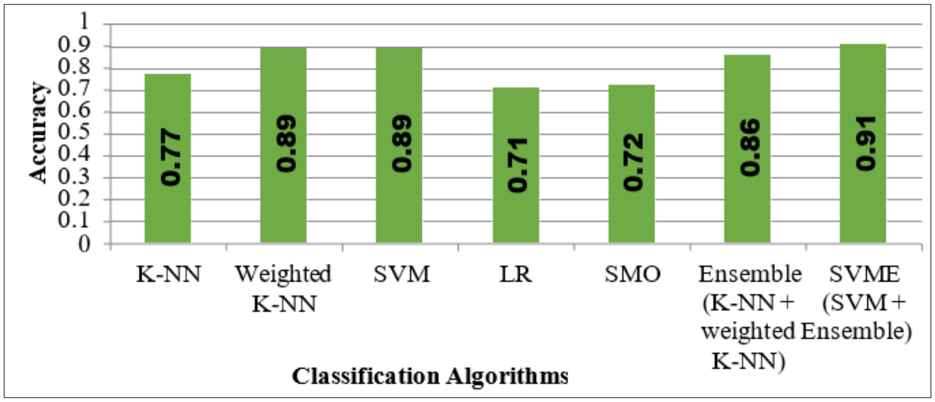

Figure 5 Comparative Performance of different Classifiers based on ‘Accuracy’

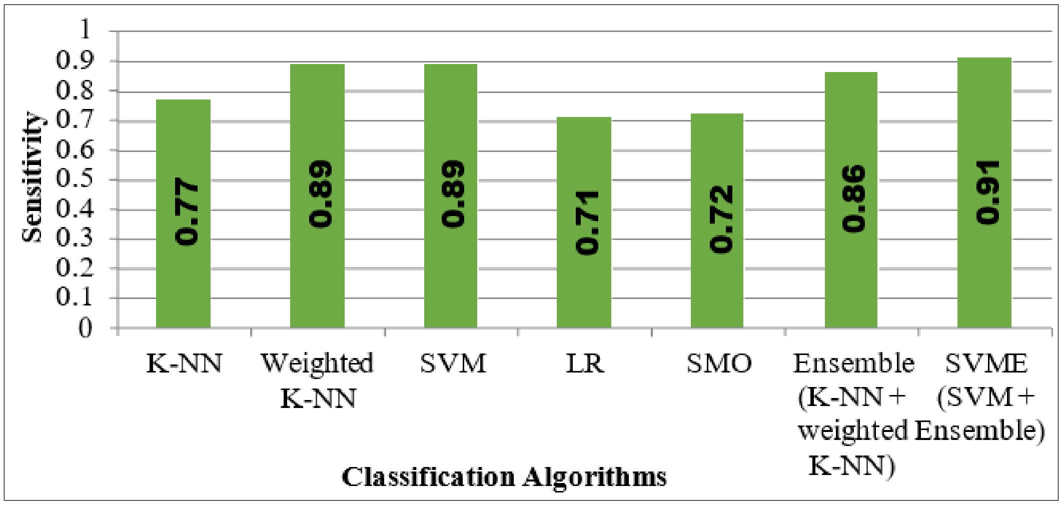

Figure 6 Comparative Performance of different Classifiers based on 'Sensitivity'

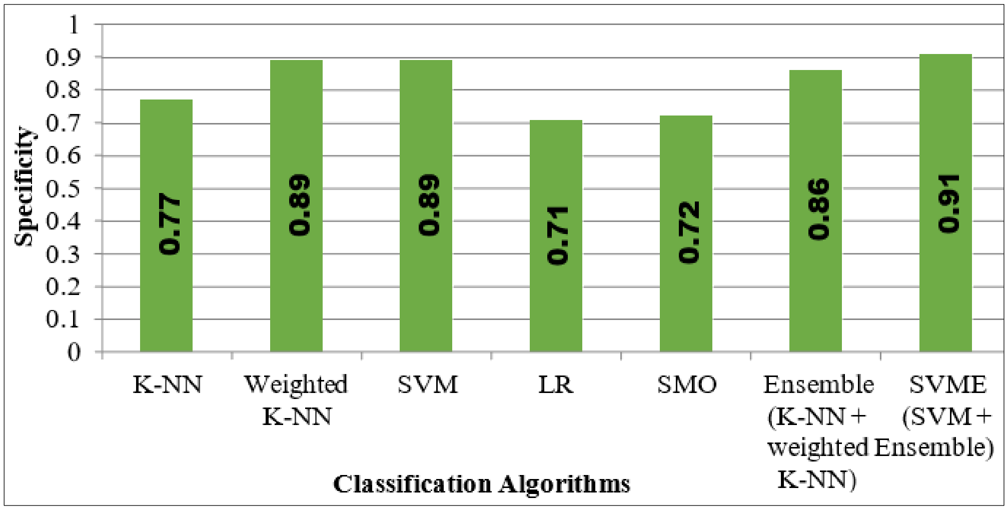

Figure 7: Comparative Performance of different Classifiers based on 'Specificity'

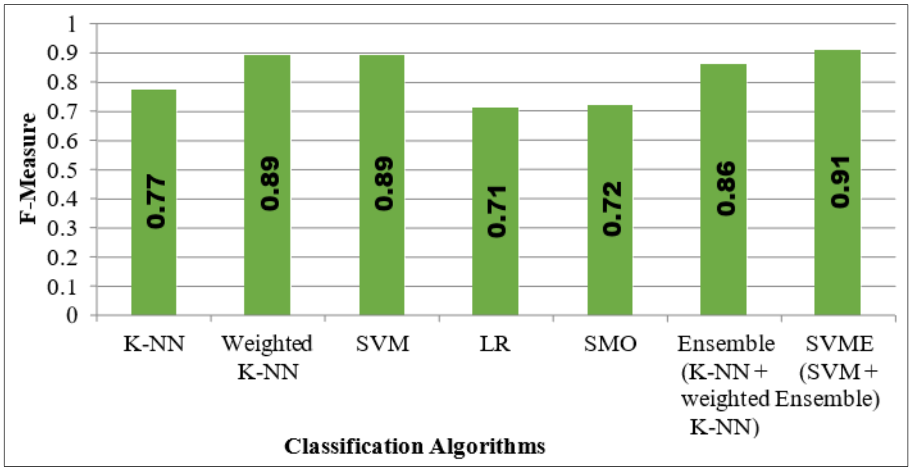

Figure 8 Comparative Performance of different Classifiers based on 'F-Measure'

Table 3: Comparative Performance Statistics of Different Classifier

Sr. No.	Classifier	Accuracy	Sensitivity	Specificity	F-Measure
1	K-NN	0.77	0.78	0.89	0.77
2	Weighted K-NN	0.89	0.89	0.97	0.89
3	SVM	0.90	0.88	0.88	0.89
4	LR	0.72	0.70	0.84	0.71
5	SMO	0.74	0.74	0.84	0.72
6	Ensemble (K-NN + weighted K-NN)	0.86	0.83	0.94	0.86
7	SVME (SVM + Ensemble)	0.91	0.93	0.96	0.91

Result and Discussions

An ensemble classifier combining the SVM and (K-NN and weighted K-NN) is proposed and implemented on a heart disease dataset. The following results are obtained from this study.

1. The proposed ensemble classifier provides 91% accuracy, 91% Sensitivity, 96% Specificity, and 91% F-measure. It means that the chances of error occurrence with this ensemble classifier are significantly less.
2. The performance statistics of various classification algorithms given in table 3 depicts that the presented ensemble classifier enhances the performance of classification results than the other classifiers as 'LR', 'SMO', 'K-NN', and 'K-NN + weighted K-NN' on all four performance measures. Although it is observed after evaluation that the classification accuracy and sensitivity of the projected ensemble classifier are pretty similar to the SVM classifier, yet the proposed ensemble classifier performs well in terms of specificity and F-measure. On the other hand, the comparison statistics of the weighted K-NN classifier and proposed ensemble classifier depicts the excellent performance of the weighted K-NN classifier only based on specificity. The performance comparison statistics of different classifiers in table 3 illustrate the overall high performance of the proposed classification algorithm.

Conclusion and Future Scope

A novel classification algorithm is proposed for prognosing cardiovascular disease. The proposed classification algorithm/classifier is the hybridization of SVM, K-NN and weighted K-NN. The measure estimates the performance of presented hybrid algorithm is analyzed by implementing it on UCI heart disease dataset. The results obtained show the high prediction accuracy of the algorithm in consideration as compared to the other existing algorithms. Identifying the relevant and irrelevant attributes form the heart disease dataset taken in this study having a significant role in the overall performance of the developed classifier. Limitation of presented work is that the performance evaluation of all the classifiers for predicting cardiovascular disease is done with ten cross-fold validation. The present work can be enhanced in some aspects as (i) variation in cross folds and (ii) implementation of more classifiers.

References

Centers for Disease Control and Prevention (CDC). Deaths: leading causes for 2008. Natl Vital Stat Rep June 6, 2012; 60 (No. 6).

Mackay, J., & Mensah, G. (2004). *Atlas of heart disease and stroke.* Non serial Publication.

Chauhan, S., & Aeri, B.T. (2015). The rising incidence of cardiovascular diseases in India: Assessing its economic impact. *J Prev Cardiol, 4*(4), 735-40.

Uyar, K., & İlhan, A. (2017). Diagnosis of heart disease using genetic algorithm based trained recurrent fuzzy neural networks. *Procedia computer science, 120,* 588-593.

Patil, S.B., & Kumaraswamy, Y.S. (2009). Extraction of significant patterns from heart disease warehouses for heart attack prediction. *IJCSNS, 9*(2), 228-235.

Fida, B., Nazir, M., Naveed, N., & Akram, S. (2011). Heart disease classification ensemble optimization using genetic algorithm. *In IEEE 14th International Multitopic Conference*, 19-24.

Vasighi, M., Zahraei, A., Bagheri, S., & Vafaeimanesh, J. (2013). Diagnosis of coronary heart disease based on 1H NMR spectra of human blood plasma using genetic algorithm-based feature selection. *Journal of Chemometrics, 27*(10), 318-322.

Al-Hamadani, B. (2016). An Emergency Unit Support System to Diagnose Chronic Heart Failure Embedded with SWRL and Bayesian Network. *International Journal of Advanced Computer Science and Applications, 7*(7), 446-453.

Al-Makhadmeh, Z., & Tolba, A. (2019). Utilizing IoT wearable medical device for heart disease prediction using higher order Boltzmann model: A classification approach. *Measurement, 147.*

Lee, H.G., Noh, K.Y., & Ryu, K.H. (2007). Mining biosignal data: coronary artery disease diagnosis using linear and nonlinear features of HRV. *In Pacific-Asia Conference on Knowledge Discovery and Data Mining,* 218-228.

Sudhakar, K., & Manimekalai, D.M. (2014). Study of heart disease prediction using data mining. *International journal of advanced research in computer science and software engineering, 4*(1), 1157-1160.

Nahar, J., Imam, T., Tickle, K.S., & Chen, Y.P.P. (2013). Computational intelligence for heart disease diagnosis: A medical knowledge driven approach. *Expert Systems with Applications, 40*(1), 96-104.

Khazaee, A. (2013). Heart beat classification using particle swarm optimization. *International Journal of Intelligent Systems and Applications, 5*(6).

Ali, F., El-Sappagh, S., & Kwak, D. (2019). Fuzzy ontology and LSTM-based text mining: a transportation network monitoring system for assisting travel. *Sensors, 19*(2).

Ali, F., Islam, S.R., Kwak, D., Khan, P., Ullah, N., Yoo, S.J., & Kwak, K.S. (2018). Type-2 fuzzy ontology–aided recommendation systems for IoT–based healthcare. *Computer Communications, 119,* 138-155.

NG, B. A. (2013, December). An intelligent approach based on Principal Component Analysis and Adaptive Neuro Fuzzy Inference System for predicting the risk of cardiovascular diseases. *In Fifth International Conference on Advanced Computing (ICoAC),* 241-245.

Davoodi, R., & Moradi, M.H. (2018). Mortality prediction in intensive care units (ICUs) using a deep rule-based fuzzy classifier. *Journal of biomedical informatics, 79,* 48-59.

Latha, C.B.C., & Jeeva, S.C. (2019). Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked, 16*.

Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE access, 7,* 81542-81554.

Xing, Y., Wang, J., & Zhao, Z. (2007). Combination data mining methods with new medical data to predicting outcome of coronary heart disease. *In International Conference on Convergence Information Technology (ICCIT 2007),* 868-872.

Thenmozhi, K., & Deepika, P. (2014). Heart disease prediction using classification with different decision tree techniques. *International Journal of Engineering Research and General Science, 2*(6), 6-11.

Soni, J., Ansari, U., Sharma, D., & Soni, S. (2011). Predictive data mining for medical diagnosis: An overview of heart disease prediction. *International Journal of Computer Applications, 17*(8), 43-48.

Singh, J., & Kaur, R. (2016). Cardio vascular disease classification ensemble optimization using genetic algorithm and neural network. *Indian Journal of Science and Technology, 9,* S1.

Verma, L., Srivastava, S., & Negi, P.C. (2016). A hybrid data mining model to predict coronary artery disease cases using non-invasive clinical data. *Journal of medical systems, 40*(7), 1-7.

Long, N.C., Meesad, P., & Unger, H. (2015). A highly accurate firefly based algorithm for heart disease prediction. *Expert Systems with Applications, 42*(21), 8221-8231.

Parthiban, L., & Subramanian, R. (2008). Intelligent heart disease prediction system using CANFIS and genetic algorithm. *International Journal of Biological, Biomedical and Medical Sciences, 3*(3).

Manogaran, G., Varatharajan, R., & Priyan, M.K. (2018). Hybrid recommendation system for heart disease diagnosis based on multiple kernel learning with adaptive neuro-fuzzy inference system. *Multimedia tools and applications, 77*(4), 4379-4399.

Lau, P.Y., Voon, F.C., & Ozawa, S. (2006). The detection and visualization of brain tumors on T2-weighted MRI images using multiparameter feature blocks. *In IEEE Engineering in Medicine and Biology 27th Annual Conference,* 5104-5107.

Mohanapriya, S., & Vadivel, M. (2013). Automatic retrival of MRI brain image using multiqueries system. *In International Conference on Information Communication and Embedded Systems (ICICES),* 1099-1103.

Subashini, M.M., Sahoo, S.K., Sunil, V., & Easwaran, S. (2016). A non-invasive methodology for the grade identification of astrocytoma using image processing and artificial intelligence techniques. *Expert systems with Applications, 43,* 186-196.

Sachdeva, J., Kumar, V., Gupta, I., Khandelwal, N., & Ahuja, C.K. (2011). Multiclass brain tumor classification using GA-SVM. *In Developments in E-systems Engineering,*182-187.

Chandra, S., Bhat, R., & Singh, H. (2009). A PSO based method for detection of brain tumors from MRI. *In World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 666-671.

Zhang, N., Ruan, S., Lebonvallet, S., Liao, Q., & Zhu, Y. (2009). Multi-kernel SVM based classification for brain tumor segmentation of MRI multi-sequence. In 16th IEEE International Conference on Image Processing (ICIP), 3373-3376.

Lee, I.N., Liao, S.C., & Embrechts, M. (2000). Data mining techniques applied to medical information. *Medical informatics and the Internet in medicine, 25*(2), 81-102.

Acharya, U.R., Sree, S. V., Chattopadhyay, S., Yu, W., & Ang, P.C.A. (2011). Application of recurrence quantification analysis for the automated identification of epileptic EEG signals. *International journal of neural systems, 21*(03), 199-211.

Alpaydin, E. (1997). Voting over multiple condensed nearest neighbors. *In Lazy learning,* 115-132.

Liang-yan, S., & Li, C. (2009). A Fast and Scalable Nearest Neighbor Based Classifier for Data Mining. *In IEEE Global Congress on Intelligent Systems.*