Heart Disease Prediction Using Adaptive Infinite Feature Selection and Deep Neural Networks

Sudipta Modak, *Member, IEEE and Esam Abdel-Raheem, Senior Member, IEEE, Luis Rueda, Senior Member, IEEE*
Department of Electrical and Computer Engineering
School of Computer Science
University of Windsor
401 Sunset Ave, Windsor, ON N9B 3P4, Canada
E-mail: {modak, lrueda, eraheem}@uwindsor.ca

***Abstract*—Prediction of heart disease is one of the most important fields of study in modern science. By studying data such as cholesterol levels, blood sugar, and blood pressure, heart disease can be predicted. In recent years, several machine learning techniques have been used to aid in fast prediction by learning from the data. However, the prediction accuracy still remains low. This is due to lower number of records contained in the databases available. In this paper, we propose a new method of heart disease prediction using a modified variation of infinite feature selection and multilayer perceptron. The method shows a high accuracy of 87.70%, a high F1-score of 87.21%, a high sensitivity of 88.50%, a high specificity of 87.02%, and a high precision in prediction of 86.05%. on the Cleveland, Hungarian, Switzerland, Long Beach, and Statlog datasets. For evaluation purposes, we have combined all the datasets together and then divided the combined dataset into training and test samples with a 20 % percent of the samples allocated for testing.**

***Index Terms*—Heart disease prediction, Infinite feature selection, Multilayer perceptron, Neural Networks**

I. Introduction

Automatic disease detection, classification, and prediction have been important areas of research for several decades. For this purpose, several algorithms have been developed to aid doctors with accurate predictions of several types of diseases. One such field is that of heart disease, which is one of the biggest causes of death in the modern world [1]. By studying the patterns of the electrocardiogram (ECG) signals and correlating them to existing data, common anomalies in the heart can be identified. Several techniques for the detection of QRS complex exist with high accuracy such as the works in [2]–[5]. However, the problem with the prediction of heart diseases remains quite challenging due to the low number of records contained in the available databases.

By learning from the data available, machine learning models can predict these diseases at early stages. The attributes taken into account for such techniques can be obtained from an individual's body such as electrocardiogram, blood pressure, sugar levels, age, sex, cholesterol levels, etc [6]. However, there can be redundant features present in the datasets. These redundant features make predictions inaccurate and use up precious memory and time. A significantly large amount of data has been collected by the healthcare industry from previous cases of heart-related disease from patients all over the world [1]. These datasets contain hidden information that is directly related to the condition of the heart and needs to be identified. Due to the presence of such a huge quantity of data, it is impossible to manually analyze them and create methods for prediction [6]. Therefore, machine learning techniques are required to deal with such data to predict diseases at early stages.

The work of [7] compares six different types of algorithms, including Linear, Quadratic, Cubic and Medium Gaussian support vector machines (SVM), as well as Decision Tree and Ensemble Subspace Discriminant for prediction accuracy. Deep learning has been used in the work of [8], where different combinations of a number of hidden layers and the number of epochs have been tested to learn which combination produces the best accuracy of prediction. Heart disease prediction using artificial neural networks can be found in [9]. This method uses six different classifiers to test the data and employs deep neural networks (DNN) for classification to achieve high accuracy in prediction. Similar algorithms can be found in [10], [11].

Feature selection techniques for heart disease prediction have been used in the works of [12]–[14]. In [12] an optimized version of genetic algorithms with SVM to achieve good accuracy in prediction, while in [13], a brute-force approach was used to select relevant features. That technique takes a small subset of features with at least three features and evaluates the combination of such features on several classifiers such as Logistic Regression (LR), k-nearest neighbour, decision tree, Naive Bayes, SVM, neural network, and vote. An integer-coded genetic algorithm has been used to select features as well [14]. That method aids the SVM-based prediction of heart disease and improves accuracy. A combination of the genetic algorithm and recursive feature elimination followed by a random forest classifier for better prediction has been presented in [15]. The work of [15] uses a genetic algorithm and recursive feature elimination to select relevant features from the data sets along with different classifiers for classification. Based on that work, the random forest classifier provides the best results when combined with the hybrid feature selection technique used in [15]. The work of [16] investigates the performance of several classifiers such as

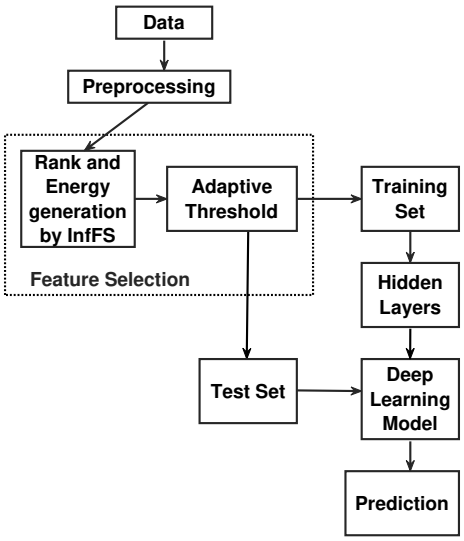

Fig. 1. Block diagram for the proposed method.

decision trees, Naive Bayes, k-nearest neighbor, and neural networks on heart disease prediction by varying the number of features provided as inputs. According to [16] the Naive Bayes classifier works best if the number of features is low.

This paper presents a new technique for prediction of heart disease using an advanced version of Infinite Feature Selection (InfFS) [17] and deep neural networks (DNN). The method is evaluated on five different datasets, namely, Cleveland, Hungary, Switzerland, Statlog, and Long Beach V [18]. A comparison with state-of-the-art methods in the field is included in this context as well.

II. Materials and Methods

The proposed method involves four stages, namely, preprocessing, which reads the raw data and converts it into usable quantities; a feature selection stage that selects a suitable subset of features and eliminates the redundant ones; a deep learning stage that is used to learn from the training datasets; finally, a prediction stage that predicts the outcomes from the test dataset. The block diagram for the entire process is summarized in Fig. 1.

A. Dataset Description

The datasets used in the experiments contain 1,190 records from five distinct databasets. It has 14 distinct features of which eight are categorical features and six are numeric features. The features are age, sex, chest pain type (cp), resting blood pressure (trestbps), serum cholesterol (chol), fasting blood sugar greater than 120 mg/d (fbs), resting electrocardiographic results (restecg), maximum heart rate achieved (thalach), exercise-induced angina (exang), ST depression induced by exercise relative to rest (oldpeak), the slope of the peak exercise ST segment (slope), number of major vessels colored by fluoroscopy (ca), thallium scan (thal), and class attribute (num). Out of these 14 features 13 of them are taken as inputs to the proposed algorithm and class attribute (num) is taken as the output. The aim is to design an algorithm that takes the 13 attributes and predicts the output. Since the data is labeled, the algorithm is regarded as a supervised learning algorithm.

B. Pre-processing

The algorithm begins with combining all five datasets into one complete dataset. Any record with missing values was eliminated from across the five databases and therefore only 1,025 records out of 1,190 were used in this work. The reason for this is that, if more data is fed into a neural network then it can learn better. Similarly, any missing values will hamper the process of learning as it creates ambiguity in the learning mechanism. The output is also changed to binary output, that is anyone with no disease is regarded as a '0', and an individual with heart disease is regarded as '1'. Originally, the data had classes from 0 to 3 which indicated the type of heart disease present in the individual with '0' being the absence of any heart disease and '1,' '2,' and '3' being the presence of three different types of diseases. For our work, we have considered only the binary case of not having any disease as '0' and the presence of disease as '1.'

C. Feature Selection

A dataset of data might have hundreds of features of which many of them can be uncorrelated to the output of the data. The main objective of feature selection (FS) algorithms is to pick out a subset of variables from the input that directly influence the outcome of the data while reducing the noise and filtering out the unwanted variables from the data. For each dataset, a feature selection algorithm needs a particular selection criterion that can evaluate the applicability of each of the features on the output classes. Once this measure is calibrated, the irrelevant features are identified one by one and eliminated if they do not satisfy the conditions being imposed.

In our research, we have employed an improved version of InfFS to select a distinct subset of feature from the dataset. The method was initially developed in [17] and is used to map the features on an affinity graph as nodes and then connects them. It then considers moving from feature to feature and by doing so, it creates a path by selecting several features as a subset of the original list of features. Given a list of features, F, the algorithm considers two aspects of the features, the vertices, V, and the edges, E. V contains a set of values that represent a feature distribution for each of the values in F, while E represents the relations between two features in the feature distribution space [17]. An adjacency matrix A, containing all the pairwise energies, which is the maximal feature dispersion and correlation between two features [17], is formulated. Once all the features are mapped onto the graph based on their weights in A, several pathways are selected with more than two features at each iteration. These features are all connected nodes and the energies of each of these paths are calculated as follows:

$$a_{i,j} = \alpha\sigma_{i,j} + (1-\alpha)c_{i,j}\,, \tag{1}$$

$$\xi_\gamma = \prod_{k=0}^{l-1} a_{v_k, v_{k+1}}\,, \tag{2}$$

where α is a loading coefficient, σ is the standard deviation, and ξ_γ accounts for the pairwise energies of all the feature pairs that compose the path [17]. Here, l is the length of the path, i and j are the positions of the feature, while a is the feature. The cycles are recorded in R_l, that are computed as follows:

$$R_l(i,j) = \sum_{\gamma \epsilon P^l_{(i,j)}} \xi_\gamma = A^l(i,j)\,, \tag{3}$$

where $P^l_{(i,j)}$ contains all the paths of length l between i and j. The single feature energy score $s(i)$ is given by:

$$S(i) = \sum_{j \epsilon V} R_l(i,j) = \sum_{j \epsilon V} A^l(i,j)\,, \tag{4}$$

and is equal to $R_l(i,j)$. The vector **S** stores the feature energies individually. The feature energies are then arranged in decreasing order with the feature having the highest energy first and stored in vector **M**.

Once all the feature rank energies are calculated, the algorithm automatically selects the number of features to keep, and hence some features are deemed redundant and are eliminated. A vector (B) is formulated, which stores the square of the individual rank energies. Equation 5 shows how each element in **B** is calculated.

$$B(k) = M(k)^2\,. \tag{5}$$

Here, k is the element number in both **B** and **M**. Finally, a threshold of T is used to decide how many features to keep for the classification step. This threshold is calculated as follows:

$$T = C \sum_{k=1}^{n} B(k)\,, \tag{6}$$

where n is the number of elements in **B** and C is a constant taken as 0.325 for the entire dataset.

If the individual energy of a feature exceeds the threshold T, then that feature is kept, or else it is discarded. The remaining features are then arranged in the order of their energy values in vector **M**. The number feature kept is denoted by N.

D. Classification

Artificial neural networks (ANN) are just like the neurons in a human brain [9]. Neural networks consist of several components such as neurons, synapses, weights, and biases. Deep neural networks (DNN) are an extension to ANN where multiple hidden layers are included to maximize the accuracy of decisions. These networks are feedforward networks, where the data always propagates in the forward direction, that is, in the direction of the output from the input and does not loop backward. Figure 2 shows an example of a DNN with one input and one output layer and two connected hidden layers.

For our research, we have used two hidden layers in the deep learning platform. The learning algorithm takes N inputs from the FS stage and formulates the same number of neurons in the input stage. The training dataset is then fed to the neural network in batches of size 32. The activation function used for the input stage is the rectified linear activation function (Relu), which is a piecewise linear function that will transfer the input directly to the output if it is positive. However, if the input is negative then it will pass zero to the output. The hidden layers consist of eight neurons each and the activation function of the first hidden layer is also kept as 'relu'. The second hidden layer is initialized with the activation function of 'sigmoid', which is a logistic function that is represented by the following equation:

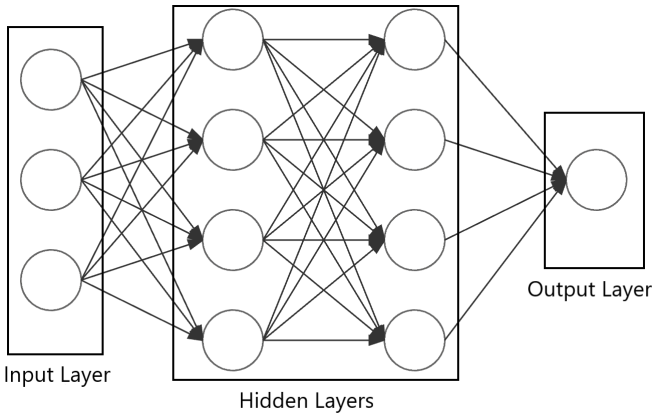

Fig. 2. Schematic view of a deep neural network.

$$\phi(x) = \frac{1}{1 + E^{-x}}\,, \tag{7}$$

where E is Euler's number and x is the element number in the input range of the sigmoid function. The dropouts for all three layers, that is the input layer and the two hidden layers are 0.25 each.

The activation function used for the output layer is 'softmax' which is a normalized exponential function and provides the probability of obtaining a '0' and a '1' as the output. Equation 8 represents the 'softmax' function.

$$\gamma(\vec{z}) = \frac{e^z}{\sum_{q=1}^{P} e^{z_l}}\,. \tag{8}$$

Here, P is the number of classes in the multi-class classifier, $\vec{z}$, input vector z standard exponential function for the input vector, and z_l standard exponential function for the output vector.

III. Experiment and Results

First, the datasets are combined into one database and then the combined database is divided into training and test datasets. The reason for this is that, the datasets all come from the same repository which is the UCI and according to other methods such as the one in [15] only the fourteen features mentioned in the dataset description of this paper are selected for training and testing. These fourteen features

are common for all five datasets and therefore they can be combined together into one comprehensive database. Now, the combined database is divided into separate segments for training the DNN and then testing the model. This is done by using five-fold cross-validation, that is the database is divided into five equal segments and out of the five one segment is kept as the test set and the rest four are used for training the neural networks. This process is repeated five times with a separate segment each time used for testing and the rest being used for training to cross-validate the results. Out of the 1,025 instances, each time, 205 samples are kept as the test samples which are later used to generate the metrics for the algorithm. Once, the design of the neural network is finalized using the above procedure, it is finally tested on the Cleveland database to ensure a fair comparison with other methods included those in the literature. The main metrics are $accuracy$, $sensitivity$, $specificity$, $precision$, and $F1$ score which are represented by the following equations,

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}, \tag{9}$$

$$sensitivity = \frac{tp}{tp + fn}, \tag{10}$$

$$specificity = \frac{tn}{tn + fp}, \tag{11}$$

$$precision = \frac{tp}{tp + fp}, \tag{12}$$

$$F1 = \frac{tp}{tp + 0.5 * (fp + fn)}, \tag{13}$$

where tp is the number of true positives, tn is the number of true negatives, fn is the number of false negatives, and fp is the number of false positives.

Table I shows the results on the test samples of five different folds. The values of $accuracy$, $sensitivity$, $specificity$, $precision$, and $F1 - score$ are included in the table. Figure 3 shows the test and training accuracy trends. The algorithm requires approximately 50 epochs to reach the desired weights for the DNN classifier, therefore there is not much need to increase the number of epochs as it might lead to overfitting. As shown in the graph, the training accuracy is close to 86% and stabilizes at this value at approximately epoch 45. The testing accuracy stabilizes at approximately 85 % which is roughly the same epoch number as the training accuracy. The figure shows the accuracy versus the number of epochs for Fold 5.

In addition, the proposed method is compared to five state-of-the-art methods in the field. The performances of these methods are collected from the papers mentioned in the literature. For a fair comparison, we have only used the Cleveland dataset, which has also been used by all methods for testing and comparison of metrics. The results are presented in Table II, while Figure 4 displays the comparison more vividly. It is observed that the proposed method outperforms the methods of Latha et al. [5], Gokulnath et al. [12], and

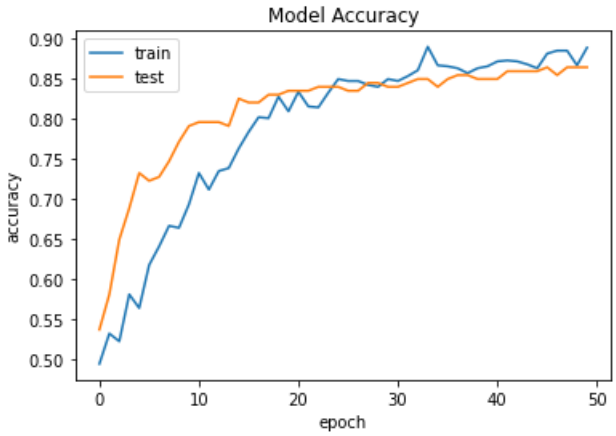

Fig. 3. Accuracy versus Epochs.

Rani et al. [15] in terms of accurate predictions. Similarly, the accuracy achieved by the proposed method is approximately equal to the method Amin et al. [13]. However, the method of Bharti et al. [9] shows higher accuracy than the proposed method. This method is evaluated further using merits such as sensitivity and specificity. Figures 5 and 6 illustrate the comparison of both quantities between the proposed method and the one in [9]. It is noticeable that the proposed method outperforms the method of Bharti et al. [9] in both cases with higher sensitivity and specificity. Furthermore, the method of Bharti et al. uses three dense layers with 128, 64, and 32 units in layers 1, 2, and 3, respectively. This increases the number of computations and time for processing significantly and makes the model quite complex. On the other hand, the proposed algorithm uses only 11, 8, and 8 units for the input, hidden layer 1 and hidden layer 2, respectively, and so has lesser time complexity. Considering this, it is safe to say that the proposed algorithm performs better than all methods included in the context.

IV. Conclusion

We have presented a new method of heart disease prediction using adaptive infinite feature selection and deep neural networks. The proposed method has shown high accuracy of prediction on five datasets which include Cleveland, Hungary, Switzerland, Statlog, and Long Beach V. The proposed method has been tested on different folds of data and shows high accuracy in terms of prediction of heart disease. In the future, the proposed method can be tested with eigenvector centrality for feature selection and more neural network-based classifiers can be implemented such as graph neural networks. The work can also be extended to the prediction of more acute and chronic diseases such as anemia, diabetes, and tumors.

Acknowledgement

This work has been partially supported by a grant provided by the Natural Science and Engineering Council of Canada (NSERC). The authors would also like to thank the Office

TABLE I
METRICS OF EVALUATION ON THE TEST SET PER FOLD.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
Accuracy (%)	90.24	88.29	90.24	83.41	84.91	87.70
Sensitivity (%)	92.63	84.69	92.55	84.38	88.24	88.50
Specificity (%)	88.18	91.59	88.29	82.57	84.47	87.02
Precision (%)	87.13	90.22	87.00	81.00	84.91	86.05
F1-Score (%)	89.80	87.37	89.69	82.65	86.54	87.21

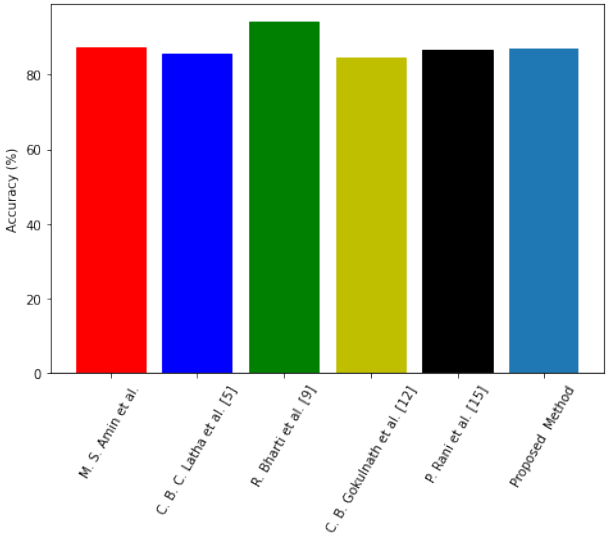

Fig. 4. Comparison of accuracy with state-of-the-art methods.

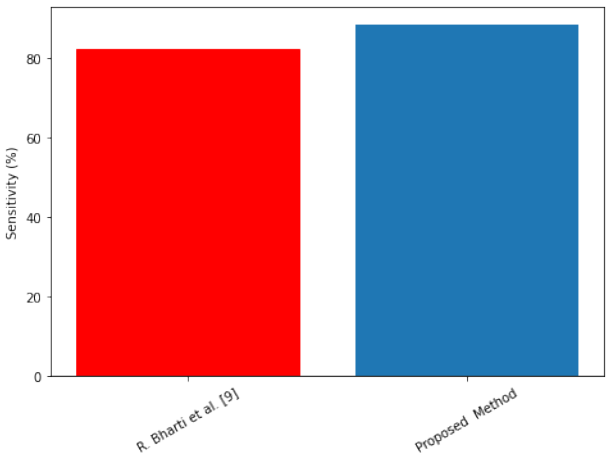

Fig. 5. Comparison of sensitivity with Bharti et al. [9].

TABLE II
COMPARISON WITH THE STATE-OF-THE-ART METHODS ON CLEVELAND DATABASE.

Method	Year	Accuracy (%)
M. S. Amin et al. [13]	2019	87.41
C. B. C. Latha et al. [6]	2019	85.48
R. Bharti et al. [9]	2021	94.20
C. B. Gokulnath et al. [12]	2019	84.40
P. Rani et al. [15]	2021	86.60
Proposed Method	**2021**	**87.13**

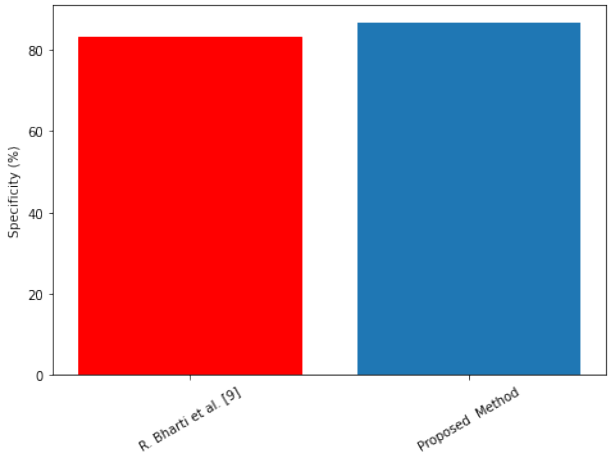

Fig. 6. Comparison of specificity with Bharti et al. [9].

of Research and Innovation Services of the University of Windsor.

REFERENCES

[1] R. Chitra and V. Seenivasagam, "Review of heart disease prediction system using data mining and hybrid intelligent techniques," *ICTACT journal on soft computing*, vol. 3, no. 04, pp. 605–09, 2013.

[2] S. Modak, L. Y. Taha, and E. Abdel-Raheem, "A novel method of qrs detection using time and amplitude thresholds with statistical false peak elimination," *IEEE Access*, vol. 9, pp. 46 079–46 092, 2021.

[3] S. Modak, E. Abdel-Raheem, and L. Y. Taha, "A novel adaptive multilevel thresholding based algorithm for qrs detection," *Biomedical Engineering Advances*, p. 100016, 2021.

[4] S. Modak, L. Y. Taha, and E. Abdel-Raheem, "Single channel qrs detection using wavelet and median denoising with adaptive multilevel thresholding," in *2020 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, 2020, pp. 1–6.

[5] Z. Zhang, Q. Yu, Q. Zhang, N. Ning, and J. Li, "A kalman filtering based adaptive threshold algorithm for qrs complex detection," *Biomedical Signal Processing and Control*, vol. 58, p. 101827, 2020.

[6] C. B. C. Latha and S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informatics in Medicine Unlocked*, vol. 16, p. 100203, 2019.

[7] S. Ekız and P. Erdoğmuş, "Comparative study of heart disease classification," in *2017 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*. IEEE, 2017, pp. 1–4.

[8] P. Ramprakash, R. Sarumathi, R. Mowriya, and S. Nithyavishnupriya, "Heart disease prediction using deep neural network," in *2020 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2020, pp. 666–670.

[9] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, "Prediction of heart disease using a combination of machine learning and deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[10] A. Khemphila and V. Boonjing, "Heart disease classification using neural network and feature selection," in *2011 21st International Conference on Systems Engineering*. IEEE, 2011, pp. 406–409.

[11] Y. E. Shao, C.-D. Hou, and C.-C. Chiu, "Hybrid intelligent modeling schemes for heart disease classification," *Applied Soft Computing*, vol. 14, pp. 47–52, 2014.

[12] C. B. Gokulnath and S. Shantharajah, "An optimized feature selection based on genetic approach and support vector machine for heart disease," *Cluster Computing*, vol. 22, no. 6, pp. 14 777–14 787, 2019.

[13] M. S. Amin, Y. K. Chiam, and K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telematics and Informatics*, vol. 36, pp. 82–93, 2019.

[14] S. Bhatia, P. Prakash, and G. Pillai, "Svm based decision support system for heart disease classification with integer-coded genetic algorithm to select critical features," in *Proceedings of the world congress on engineering and computer science*, 2008, pp. 34–38.

[15] P. Rani, R. Kumar, N. M. S. Ahmed, and A. Jain, "A decision support system for heart disease prediction based upon machine learning," *Journal of Reliable Intelligent Environments*, pp. 1–13, 2021.

[16] T. J. Peter and K. Somasundaram, "An empirical study on prediction of heart disease using classification data mining techniques," in *IEEE-International conference on advances in engineering, science and management (ICAESM-2012)*. IEEE, 2012, pp. 514–518.

[17] G. Roffo, S. Melzi, and M. Cristani, "Infinite feature selection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4202–4210.

[18] D. Dua, C. Graff *et al.*, "Uci machine learning repository," 2017.