ENSEMBLE METHODS FOR HEART DISEASE PREDICTION

TALHA KARADENİZ

NOVEMBER 2022

ÇANKAYA UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

Ph.D THESIS IN

COMPUTER SCIENCE AND ENGINEERING

ENSEMBLE METHODS FOR HEART DISEASE PREDICTION

TALHA KARADENİZ

NOVEMBER 2022

ABSTRACT

ENSEMBLE METHODS FOR HEART DISEASE PREDICTION

KARADENİZ, TALHA

PhD in Computer Science and Engineering

Supervisor: Prof. Dr. H. HAKAN MARAŞ

Co-Supervisor: Assist. Prof. Dr. HALİT ERGEZER

November 2022, 69 pages

This work consists of automatic heart disease prediction ensemble methods; this critical human health task is performed using several new algorithms. First, we introduce a weak classifier based on the randomness analysis of binary sequences. Second, we present another classifier in which the shrunk covariance estimation is utilised during the training and prediction phases. Third, we present a classifier in which Gaussian probabilities are summed via a kurtosis and KS-test importance scheme. Finally, a two-fold ensemble implementation is created by fusing logistic regression and our majority voting density estimation classifier. This final classifier is compared with state-of-the-art methods, and the sensitivity, specificity, accuracy and optimised precision are reported.

Keywords: Bagging, Ensemble methods, Base estimator, Classification, Heart disease prediction

ÖZET

KALP HASTALIĞI TAHMİNİ İÇİN ENSEMBLE METOTLAR

KARADENİZ, TALHA
Bilgisayar Bilimleri ve Mühendisliği Doktora

Danışman: Prof. Dr. H. HAKAN MARAŞ
Ortak Danışman: Dr. Öğr. Üyesi HALİT ERGEZER
Kasım 2022, 69 SAYFA

Bu çalışma otomatik kalp hastalığı tahmini için ensemble metotları içermektedir; bu kritik sağlık işlemi birçok yeni algoritma ile gerçekleştirilmiştir. Birincisi, ikili dizilerin rastgelelik analizine göre bir taban tahmincisi geliştirilmiştir. İkincisi, sıkıştırılmış kovaryans tahmini metotlarına dayalı başka bir sınıflandırıcı tanıtılmıştır. Üçüncüsü, kurtosis ve KS-test önem şemasına göre şekillenen bir sınıflandırıcı geliştirilmiştir. Son olarak, lojistik regresyon, çoğunluk oy uygulamasına ve olasılık yoğunluk tahminine dayalı sınıflandırıcı şemalarımız ile birleştirilmiştir. Bu son sınıflandırıcı, state-of-the-art metotlar ile karşılaştırılmiş ve elde edilen isabet oranları raporlanmıştır

Anahtar Kelimeler: Torbalama, Ensemble metotlar, Taban tahmincisi, Sınıflandırma, Kalp hastalığı tahmini

ACKNOWLEDGEMENT

This thesis is dedicated to my family; my brothers, my father and my mother. Thanks go to them for their endless support. Additionally thanks to Yusuf "oblomov" Karacaören for his help and advices. Special thanks to my supervisor Prof. Dr. H. Hakan Maraş for the excellent guidance and providing me with an excellent atmosphere to conduct this research. My special gratitude also goes to the rest of the thesis committee - Assist. Prof. Dr. Halit Ergezer, Assoc. Prof. Dr. İhsan Tolga Medeni, Assist. Prof. Dr. Gül Tokdemir - for the encouragement and insightful comments. Since life would be boring without cyberspace allies, greetings also go to my buddies from demoscene and sourtimes: anesthetic, cheja, janli, scg and vulpius.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

ANN	: Artificial Neural Network
BMVNC	: Bagged Majority Voting Nearest Centroid
BRVC	: Bagged Reference Vector Classifier
BSCC	: Bagged Shrunk Covariance Classifier
BagMOOV	: Bootstrap Aggregation with Multi-Objective Optimised Voting
CAFL	: Chaos Firefly Attribute Reduction and Fuzzy Logic
CANFIS	: Coactive Neuro-fuzzy Inference System
CART	: Classification and Regression Tree
CHD	: Coronary Heart Disease
CIFE	: Conditional Infomax Feature Extraction
DEMVB	: Density Estimation Based Majority Voting Bagging Classifier
DMX	: Data Mining Extension
DNN	: Deep Neural Network
DT	: Decision Tree
GA	: Genetic Algorithm
GKMVB	: Gaussian Probability and Kurtosis based Majority Voting Bagging Classifier
HDPS	: Heart Disease Prediction System
IWRF	: Inner-product Wavelet Transform Random Forest
kNN	: k-Nearest Neighbor

LM	: Linear Model
LR	: Logistic Regression
LTM	: Logistic Tree Model
MKMVB	: Maxwell Distribution and KS-Test Majority Voting Bagging Classifier
MLP	: Multilayer Perceptron
MRMR	: Minimum Redundancy Maximum Relevance
NB	: Naive Bayes
PCA	: Principal Component Analysis
RF	: Random Forest
SPECT	: Single-photon Emission Computed Tomography
SVM	: Support Vector Machine
UCI	: UC Irvine
XGBFN	: XGBoost + Neighborhood Component Analysis + Fuzzy Rough Set Prototype Selection + Random Oversampling

CHAPTER I

INTRODUCTION

5.1 HEART DISEASE PREDICTION

Medical diagnosis is the process of determining the status of a patient via the physician's experience and available data [1]. When one is focused on heart disease, mechanising this procedure is known as automatic heart disease prediction. Deriving a machine learning model by using medical parameters or features extracted from medical data to estimate risk levels is important for such a task [2]. The accuracy of the method is also crucial [1]. Today, the availability of large amounts of data makes it possible to analyse patterns in order to create machine learning models.

5.2 MOTIVATION

Our main motivation behind performing this research was of course to improve human health. On the other hand, presenting a new classifier was also an intriguing challenge. By concentrating on ensemble methods, we have devised new weak classifiers. Despite their simple nature, when they are followed by feature selection methods, they yield promising results.

5.3 ROUTE

In the Background Study chapter, we talk about ensemble methods, base estimators and logistic regression. In the Literature Review chapter, we concentrate on the heart disease prediction literature, in addition to artificial neural networks (ANNs), support vector machines (SVMs), naive Bayes (NB) methods and two state-of-the-art methods: `a firefly-based algorithm for heart disease prediction' [3] and `a novel ensemble for heart disease prediction' [4].

In the Proposed Technique chapter, we provide the algorithms of our weak classifier and describe the pipeline of the study, which consists of robust scaling, feature selection and classification.

In the Results section, metrics are described and a comparison of our method with state-of-the-art methods is performed. In the Conclusion chapter, naturally we summarise our experiments and sketch out potential future work.

CHAPTER II

BACKGROUND STUDY

6.1 ENSEMBLE METHODS

In this chapter, we discuss ensemble methods, base estimators and logistic regression. We first give the definition of an ensemble method and mention stacking. Then, we discuss the integration of weak classifiers into the framework. Finally, a logistic regression formulation is presented to complete the overall scheme.

A learning machine is a mechanical node of observation-to-output derivation. It deduces a decision after being fed a set of data. The data, most of the time, are in the form of a fixed-dimensional vector set. Ensemble methods gather information from several learning machines to form a more reliable decision [5]. Thus, one may say that an ensemble method is a meta-machine collecting `opinions' [5] from underlying machines.

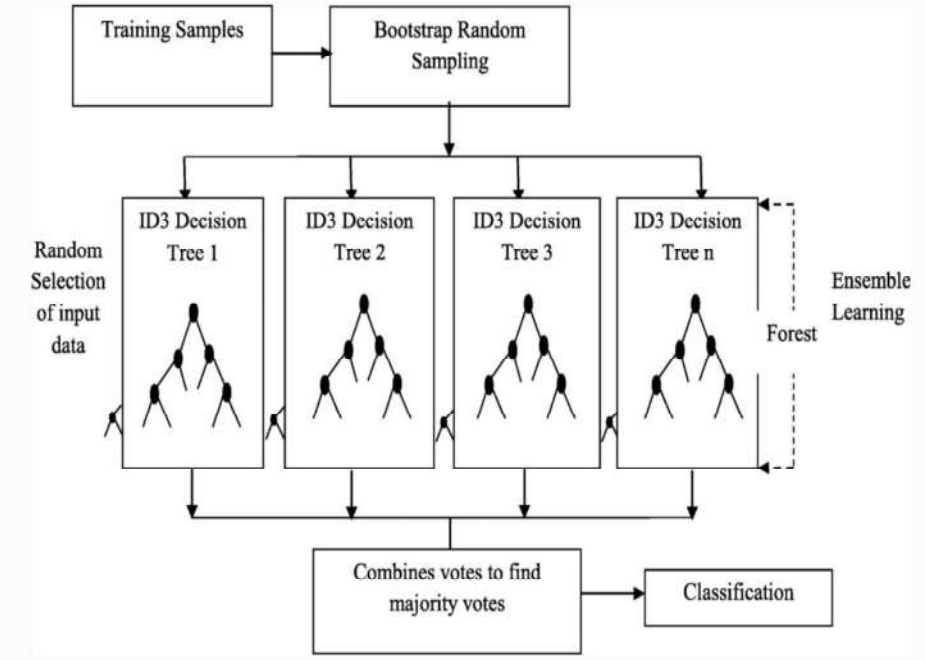

Figure 2.1: Random forest [6]

6.2 BAGGING

Bagging was proposed to reduce the variance of a predictor[7]. It stands for `bootstrap aggregating'. The word `bootstrap' references the sampling routine, while `aggregating' refers to the voting procedure. Given a base estimator, bagging yields **n_estimators** instances, where **n_estimators** is the total number of predictors. Each of these instances is trained on a different bootstrap sample [8]. Prediction is done according to the majority voting principle; i.e. the class that receives the greater number of votes from the base estimators is the winner of the prediction routine and thus is the output of the classification procedure.

The FIT() and PREDICT() algorithms for a bagging classifier are given below.

Algorithm 1 Bagging

procedure FIT($X, y, n\_estimators, base\_estimator$) ▷ Training dataset (X) and class labels (y), number of base estimators and a base estimator instance
    $base\_estimators \leftarrow list()$
    $i \leftarrow 0$
    while $i < n\_estimators$ **do**
        $X', y' \leftarrow resample(X, y)$ ▷ Bootstrap sample from X and y
        $clf \leftarrow fit(base\_estimator, X', y')$ ▷ Fit a base estimator to sample
        $append(base\_estimators, clf)$ ▷ Add to the list of estimators
        $i \leftarrow i + 1$
    end while
    return $base\_estimators$
end procedure
procedure PREDICT($x, base\_estimators$) ▷ Test vector x. $base\_estimators$ is the output of the FIT() procedure
    $s\_0 \leftarrow 0$ ▷ Vote sum for class 0
    $s\_1 \leftarrow 0$ ▷ Vote sum for class 1
    $n\_estimators \leftarrow len(base\_estimators)$
    while $i < n\_estimators$ **do**
        $label \leftarrow predict(base\_estimators[i], x)$ ▷ Run predict routine of the base estimator
        if $label == 0$ **then**
            $s\_0 \leftarrow s\_0 + 1$
        else
            $s\_1 \leftarrow s\_1 + 1$
        end if
        $i \leftarrow i + 1$
    end while
    if $s\_0 > s\_10$ **then**
        $label \leftarrow 0$
    else
        $label \leftarrow 1$
    end if
    return $label$
end procedure

Figure 2.2: Bagging

This algorithm is of course a simplified template of bagging. Today, there are implementations in which the number of features used by base estimators can be specified. Additionally, bagging ensembles can still be improved by new developments, as one can see from the results and analysis given by [9].

6.3 BASE ESTIMATORS

These underlying machines are nodes of learning; they are also known as predictors, base estimators or `weak classifiers'. Stacking predictor responses generally yields a much better performance than using a single weak classifier [10]. Averaging better-than-chance weak learners is known as boosting [11]. Stacking introduces a meta-classifier [11] that is trained on the base classifier responses to obtain better results. Random subspaces benefit from the idea of restricting the dimensionality of each base learner to a random subspace [11]. Our work combines bagging, random subspaces and stacking methods to predict heart disease.

6.4 LOGISTIC REGRESSION

Suppose that our dependent variable is binary; i.e. it is either 0 or 1. If we model it linearly, we obtain the ordinary least squares approach: $E(y) = \pi = \alpha + \sum \beta_k X_k$ [12], where X_k represents the independent variables (attributes) and $E(y) = \pi$ is the dependent variable. OLS has some disadvantages: namely, the use of a linear function, the violation of the pseudo-isolation condition and `error heteroskedasticity' [12].

A linear function, that is, a function of the form $\alpha + \sum \beta_k X_k$ does not need to be restricted to values of 0 and 1, so it is not compatible with a binary dependent left-hand side [12].

The violation of the pseudo-isolation condition means that there is a correlation between the error term and the regressor [12]. The choice for the distribution of the error term determines the analysis type [12]; a logistic distribution implies a logistic regression. If the distribution is normal, then one has a probit analysis [12].

In [13], it was shown, using a one-dimensional counter-example, that it is not possible to realise the zero mean and constant variance error assumption. Additionally, in [13], a real-life dataset is analysed and neglecting of a strong relation between an attribute and the regressand by linear prediction is shown. On the other hand, probit analysis covers this relation and yields `higher estimates of fit' [13].

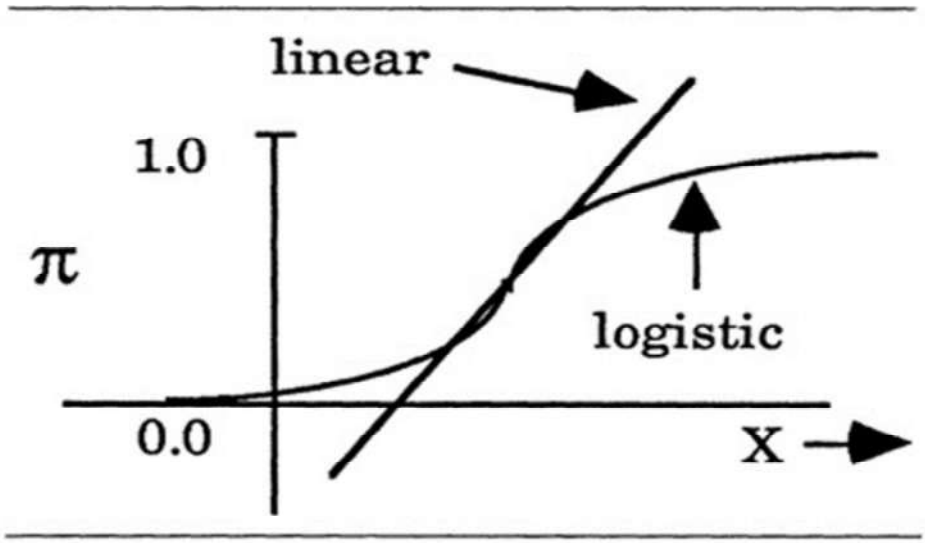

Figure 2.3: Linear vs. Logistic [12]

One advantage of logistic regression is that it is based on a `latent variable' approach [12]. Assume that the observed $Y$ is a `reflection' of the continuous Y^*, where $Y = 1$ if $Y^* > 0$ and $Y = 0$ otherwise [12]. Then, it can be said that $Y^* = \alpha + \sum \beta_k X_k + \epsilon$. If it is supposed that ϵ has a logistic distribution, then the appropriate analysis tool is logistic regression [12].

The probability that $Y = 1$ is $P(Y = 1) = \pi = \frac{exp(\alpha + \sum \beta_k X_k)}{exp(\alpha + \sum \beta_k X_k) + 1}$ [12]. After linearising, we get $log\left(\frac{\pi}{1-\pi}\right) = \alpha + \sum \beta_k X_k$, and the model parameters are estimated via the maximum-likelihood method [12].

CHAPTER III

LITERATURE REVIEW

7.1 HEART DISEASE PREDICTION

This chapter is about studies on heart disease prediction and traditional algorithms on automatic classification. We first provide a picture of the current literature on heart disease prediction and then move on to state-of-the-art statistical classification. Then, we concentrate on specific contemporary heart disease prediction methods.

In [14] a hybrid approach is proposed. These researchers performed experiments on the Cleveland dataset to report their method's performance. First, they removed some of the features manually and excluded missing-value observations from the dataset. Second, they determined the component candidates of their classification system by considering state-of-the-art classification schemes, including decision trees (DT), a linear model (LM), support vector machines (SVMs), random forest (RF), naive Bayes (NB), artificial neural networks (ANNs) and k-nearest neighbour (kNN) clustering. By clustering the data into partitions and testing the candidates, they found that RF and LM outperform the other methods.

Then, they combined these two methods to form their HRFLM method, which had an accuracy of 88.7%. Although it is a valuable empirical study concerning the combination of classifiers, this work suffers from the limitedness of the utilised dataset; only the Cleveland dataset is included.

The authors of [15] engineered a `heart disease prediction system' (HDPS) constructed using, as in [14], the Cleveland dataset (303 instances, 13 features). Their system is one of the relatively old decision support systems in the literature; it has an accuracy of 80%. They utilised ANNs to predict heart disease. Their neural network has a 3-layered structure consisting of input (13 neurons), hidden (6 neurons) and output (2 neurons) layers. They trained their system 100 times to study the performance of the method. A nice property of their research is that a GUI was built

that is ready to be used. The critique that applied to [14] can be repeated here, as the researchers considered only one dataset. Additionally, their algorithm presentation is too verbal, and a reader can easily be confused by the overall picture that is presented. For example, they use an abbreviation, `LAV', which seems to stand for something mysterious.

The authors of [16] also utilised the Cleveland dataset to provide classification results. They considered 3 state-of-the-art methods to predict heart disease: DTs, NB and ANNs. The power of this work relies on the fact that the researchers not only provide the success rates of the underlying classical models but also give examples of professionally interpretable data visualisations. A critique concerning the lack of a comparison with state-of-the-art methods can be neglected this time because this is also one of the oldest systems proposed in the literature. An additional property of their framework is the integration of the Data Mining Extension (DMX) query language. This allows an expert to use the system and obtain information about the data. One point that caught our attention is that the Cleveland dataset normally has 303 observations, but in this work, a higher number of observations (909) is reported: this is due to the consideration of all of the data located at https://archive.ics.uci.edu/ml/datasets/heart+disease that are marked as `Cleveland'. Studies concentrating on 303 observations actually utilise the `Cleveland' file of the archive above. Another point concerns the pre-processing step: all the numerical values are converted into categorical values in this work.

The authors of [17] created one of the early pipelines in the literature: it is based on a `coactive neuro-fuzzy inference system' (CANFIS) for predicting heart disease. Their CANFIS architecture has 5 layers: Premise Parameters, Firing Strength, Normalised Firing Strength, Consequence Parameters and Overall Output. The work combines the proposed architecture with genetic algorithms (GAs) to find the best parameters. Experiments are conducted on the Cleveland dataset and a very low mean square error is noted: 0.000842. GAs depend on three operations: selection, crossover and mutation. After an initial population is provided, the system evolves towards a better solution via the provided operators. The implementation platform is NeuroSolutions, an ANN environment developed by NeuroDimension. One advantage of the system is that it can categorize the following heart disease types: type 1, type 2, type 3 and type 4. Unfortunately, this report is restricted to only one dataset.

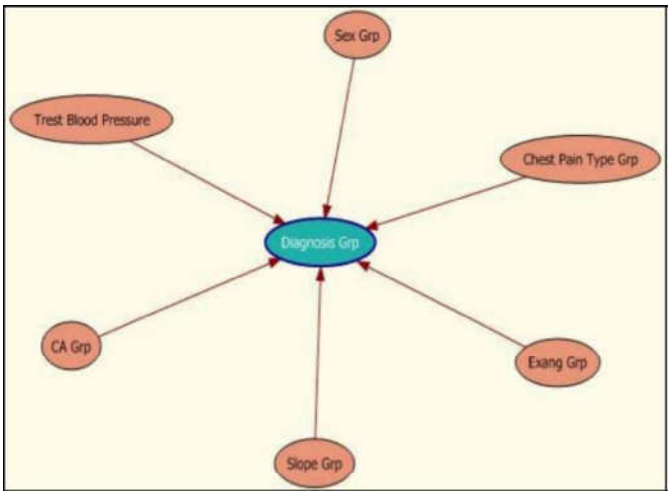

Figure 3.1: A DT dependency network from [16]

The authors of [17] created one of the early pipelines in the literature: it is based on a `coactive neuro-fuzzy inference system' (CANFIS) for predicting heart disease. Their CANFIS architecture has 5 layers: Premise Parameters, Firing Strength, Normalised Firing Strength, Consequence Parameters and Overall Output. The work combines the proposed architecture with genetic algorithms (GAs) to find the best parameters. Experiments are conducted on the Cleveland dataset and a very low mean square error is noted: 0.000842. GAs depend on three operations: selection, crossover and mutation. After an initial population is provided, the system evolves towards a better solution via the provided operators. The implementation platform is NeuroSolutions, an ANN environment developed by NeuroDimension. One advantage of the system is that it can categorize the following heart disease types: type 1, type 2, type 3 and type 4. Unfortunately, this report is restricted to only one dataset.

The authors of [18] concentrated on the ANN, DT and NB methods to evaluate the performance of data mining techniques on heart disease prediction. They conducted experiments on a UC Irvine Machine Learning Repository (UCI) dataset using WEKA, MATLAB and TANAGRA. MATLAB was used for fuzzy logic experiments. In addition to these, they also performed experiments concerning clustering-based classification. On the Cleveland dataset, they reported an accuracy of 86.5\% for naive Bayes. They applied a GA to select 6 features from the overall dataset and found that 6 features were relevant. Their pipeline also had an imputation component, where the missing values were replaced using appropriate strategies. An accuracy of 86.5% was obtained after 70%/30% training/testing splitting was used. Without this splitting, they reported a 100% accuracy for the ANN, but this may be

misleading since no training/testing splitting or cross-validation was performed. Therefore, the conclusion of the work is somewhat confusing since a kNN method can be by definition a winner when training/testing splitting is not used.

The authors of [19] utilised UCI data to establish a system supporting heart disease prediction. After 80%/20% training/testing splitting, they reported an accuracy of around 89% for their NB classifier. Alternative methods such as sequential mining optimisation (SMO), a Bayes net (BN) [20] and an ANN (multilayer perceptron (MLP)) were also examined. The most accurate and efficient choice was, according to their results, the NB classifier. One advantage of this work is that they present a fully functional system, that is, a system accompanied by data mining and encryption, to predict heart disease. The Advanced Encryption Standard (AES) is their method of choice for their encryption component. Although NB classifiers do not need pre-processing steps such as scaling, other additional pipeline components can be considered to improve the overall performance of this system. A feature selection step, for example, can be beneficial in terms of both accuracy and efficiency.

The authors of [21] conducted experiments on the Cleveland dataset and utilised WEKA and KEEL [22] to implement their methods. They formulated a decision tree to perform the task of heart disease prediction. Their system has 3 parameters: confidence, minimum item sets and a threshold. To formulate the best rules, they plugged a hill climbing algorithm into their framework. After 10-fold cross-validation, they found that their `efficient' system outperformed classical methods such as SVM and kNN (in particular, they tested 1NN). The novelty of this work is that they constructed an alternative decision tree algorithm that can outperform not only C4.5 but also ANN methods like MLPs. Their explicit formulation of the rules and the pruned rules is another advantage. In this way, the work contributes to the literature on interpretable classification. On the other hand, adding a bagging or boosting component to their system could improve the performance since ensemble methods have such proven accuracy-boosting abilities. Feature selection could also be applied to increase the overall accuracy.

The work [23] is a popular modern work that examines state-of-the-art classification methods. These methods are the NB, DT, kNN and RF methods. They found that 1NN is the most accurate method on the Cleveland training dataset (the one with 303 instances). On the test dataset, NB is the most accurate method since it has an accuracy of around 88.1%. Their pre-processing component involves a missing-

value replacer, but the authors do not explicitly state which one was used. The implementation tools were WEKA and Python. The training/testing split ratio is not given. The authors conclude that, to obtain higher accuracy rates, one needs to develop more complex methods. The work has a clean and simple summary of the methods they use. They also summarise the current literature by giving the accuracies of some high-performing models. In the article, it is concluded that kNN with k = 7 was the winner, but when we look at the test results, the winner seems to be naive Bayes. On the other hand, they cite the work of [24] by summarising their results: NB -- 84.5%, SVM -- 84.5% and functional trees (FTs) -- 84.5\%. When we first saw this triple tie, this seemed peculiar (imagine that 3 models being tested have exactly the same accuracy) and we intended to verify the results. In [24], the SVM is actually reported to have an accuracy of 85.1%.

The authors of [25] performed experiments on the Cleveland and Statlog datasets. They set Cleveland as the training set and Statlog as the testing set. The compared methods are the NB, DT and ANN (MLP) methods. They add two more attributes to the data: obesity and smoking. They found, by performing experiments on the datasets, that these additional features improve the overall accuracy of heart disease prediction. Without the added features, the accuracy scores for NB, DT and ANN are 94.4%, 96.7% and 99.3%, respectively. With the added features, the new accuracy scores are (in the same order) 90.4%, 99.6% and 100%. Their pre-processing step has a missing value replacement component. The implementation was done in WEKA. The architecture of the winning ANN is not given. Although Cleveland/Statlog training/testing splitting is somewhat tricky or `weird', this method provides one of the highest scores obtained in the heart disease prediction literature, followed by the CANFIS method of [17].

The work [26] resembles [18]. The authors summarise the experiments performed on the (extended) Cleveland dataset using a 1:1 training/testing split ratio. NB, kNN, DT and classification via clustering are the methods they considered. The implementation frameworks are WEKA and TANAGRA. Unlike [18], rough set theory is tested and some results are reported. Although there is a `rule dump', explicit measurements of the accuracy and other metrics are missing. The claim is that rough-set-theory-based feature selection outperforms the alternatives. Association rule discovery is also mentioned; four constraints are introduced to reduce the rule set. The

conclusion of the work is that DT outperforms other methods such as kNN, NB and classification via clustering.

The author of [27] summarises their IoT-based heart disease prediction system. They test their system on 3 datasets: the Hungarian, Framingham and Public Health datasets. Normally, the application of a deep neural network to tabular data requires additional steps, such as encoding or transformations [28]. Here, a direct implementation is mentioned, which could be revolutionary, but the author does not note the literature on the relationship between deep neural networks and tabular data. Since one needs spatial or time series data to make convolution work, how the pipeline is constructed is somewhat blurry in the paper. Is it transformation based? Is it encoding based? On the other hand, is it something else entirely? If we put aside the questions about the connection of the work to the general deep neural network literature, we see that, after the integration of adaptive elephant herd optimisation (AEHO), the accuracy of the framework is high: it achieves accuracies of 93.3%, 98.2% and 97.6% for the Hungarian, Framingham and Public Health datasets, respectively.

The work [29] is a survey of heart disease prediction. It is not an extended and full-scope review but it provides some insight into the methods utilised in the literature. It begins with feature extraction (principal component analysis (PCA)) and feature selection (correlation-based feature selection (CFS)), and then it proceeds to the classification methods: NB, SVM, kNN, DT and RF. The corresponding reported highest accuracies are (on different datasets) 84.2%, 99.0%, 83.2%, 92.2% and 97.0%. The authors also speak briefly about stacking. They conclude that there are many methods for predicting heart disease via machine learning. It is noted that while alone DT was not able to satisfactorily complete the task, when PCA was added to the framework, good results could be achieved. Additionally, they underline the ability of RF and ensemble methods to solve the problem of overfitting. They comment on the fast computation and success of NB.

The authors of [30] provide a sex-specific coronary heart disease (CHD) mathematical model. This model is older and from a time when there were no SVMs. They used the Framingham dataset, and the utilised variables were age, blood pressure, total cholesterol, HDL-C, diabetes and current smoking status. The Cox regression coefficients are considered in the work and the model is validated for different populations (the main model induction was done using a white, middle-class

population). For white and black populations, the estimates were reasonably good, but for Japanese-American and Hispanic men, and for Native American women, there was an overestimation in the prediction of CHD. This was resolved after the process of recalibration. The authors concluded that the proposed method was reasonably accurate and could also be applied to other ethnic groups after recalibration.

The authors of [31] utilised the Cleveland dataset. The compared methods were RF, DT, LR and NB. No feature selection or data transformation was applied. Experiments were performed after an 80%/20% training/testing split. The metrics employed are the accuracy, precision, recall and F-score. Confusion matrices are considered and accuracy values are given. According to the study, the winning method is RF with a 90.2% accuracy. RF is the winner according to the precision, recall and F-score measures as well, with values of 90.4%, 88.2% and 90.1%, respectively. The explanation of RF in this work is somewhat incorrect because RF does not create just one decision tree but benefits from several decision trees trained on data. The authors conclude their work by stating a plan for coding a web interface for their system. This study suffers from the use of only one dataset and the lack of appropriate feature selection and extraction methods.

The authors of [32] propose a method called the heart disease prediction model (HDPM); their pipeline has outlier detection, data balancing and classification steps. Outliers are detected via the density-based spatial clustering of applications with noise (DBSCAN), data balancing is performed using the synthetic minority over-sampling technique-edited nearest neighbour (SMOTE-ENN) method and classification is done using XGBoost. They report high values of the accuracy and other metrics. The accuracy scores are 95.9% and 98.4% for the Statlog and Cleveland datasets, respectively. This work is a state-of-the-art work in which each step is clearly formulated and analysed. The compared methods are NB, LR, MLP, SVM, DT and RF. The implementation libraries are sklearn, imbalanced-learn and XGBoost. The authors also present a web application based on their method.

The authors of [33] conducted experiments on their own dataset and integrated DTs and kNN to predict heart disease. The attributes that were considered were age, gender, blood pressure, pulse rate and cholesterol. The reported accuracy is 80.0% but no training/testing split ratio or cross-validation scores are given. They observe that the accuracy score increases when the number of attributes is increased. This work is very limited in terms of generalisation since the dataset was manually prepared. There

is also no data pre-processing and feature selection. The main contribution of the study is the presentation of a practical system for heart disease prediction. On the other hand, additional metrics such as the specificity and sensitivity are missing.

The work [34] is a comparison of the classification methods KStar, DT, SMO, BN and ANN (MLP). It differs from the surveys that we have talked about up to now due to its depiction of AUC curves in addition to the accuracy. According to this paper's findings, BN is the most successful method; it yields AUC values of 90.2% and 88.1% for the Statlog and `Collected Data' datasets, where `Collected Data' refers to the data the authors collected from Enam Medical Diagnosis Centre. The results are obtained after 10-fold cross-validation. This is the first heart disease prediction study in which the KStar algorithm is tested. KStar differs from standard kNN due to its use of the entropy-based distance. In terms of accuracy, SMO is the winner but in terms of the AUC, BN is more successful. In this work, the authors also report the training times of the models: the ANN is the slowest model and KStar is the fastest model.

The author of [35] performed experiments on their own dataset, which had 7339 observations; the data were retrieved from PGI, Chandigarh. This is the biggest dataset in the literature considered so far. Normally, this dataset has 15 dimensions but the author reduced the number of dimensions to 8 using best fit search. This work is also one of the studies in which feature selection is considered seriously and applied. In terms of the accuracy, the DT is the winner with an accuracy of 95.6% after 10-fold cross-validation, and in terms of the F-measure, the ANN is the winner with an F-measure of 97.4%. NB is also tested and its reduced-feature-setting accuracy score is 92.4%. The author concludes that the performance of the DT can be explained by its ability to capture `simple datasets'. WEKA is the platform on which the experiments are conducted. This work can be improved by testing more feature selection and classification methods on the proposed dataset.

The authors of [36] report an accuracy of 100.0% on the Cleveland dataset (the authors calls it a `public dataset' and state that the number of observations is 303, so we have deduced that the dataset is the Cleveland dataset). The utilised method is a backpropagation ANN and the framework is WEKA. In pre-processing, a missing value replacement routine is applied, and the experiment is conducted with a 40\%/60\% training/testing split configuration. Although the score is notable, it may be a `peeking at the test data' [37] result since the number of parameters is high and

the training/testing split is fixed. The pre-processing step is said to have a filtering step to exclude irrelevant data, but details are not given.

The authors of [38] run 10-fold cross-validation experiments on the Cleveland dataset, and the tested methods are the DT, logistic tree model (LTM) and RF. This is the first study that uses an LTM to predict heart disease. The authors of the work reported accuracy values of 83.4%, 76.3% and 80.0% for the DT, LTM and RF, respectively. No pre-processing or feature selection methods were performed and the Cleveland dataset is the only dataset considered. One of the claims in the conclusion is that the DT is the winner, but according to the tables presented before the conclusion, the LTM seems to be the winner. Since there are two tables that contradict each other, it is highly possible that there is a mistake in the LTM results. On the other hand, this study has the advantage of reporting a relatively detailed analysis of the DT with regard to the pruning process. The verbal description of the other algorithms is satisfying as well.

The authors of [39] test the SVM, DT, LR and kNN methods on the Cleveland dataset. They prefer a training/testing ratio of 73%/37%. The work also includes an introductory taxonomy of machine learning that describes supervised, unsupervised and reinforced methods. There is a nice balance of visual and verbal ingredients in the text. The reported accuracy scores are 83%, 79%, 78% and 87%, which indicates that kNN is the winner. Although there is a section called `data balancing' in which the class sample counts are given, there is no clear statement on the method used. There is also no feature selection or extraction. Additionally, no other metrics, such as the sensitivity and specificity, are reported. Despite its suitable visual-verbal representation, the study suffers from a lack of generalisation to multiple datasets. Additionally, there are bold claims like `x proved that...' which refer to empirical studies. The experiments are conducted in the Jupyter Notebook environment.

The work [40] is an extensive evaluation study on heart disease prediction. The tested methods are SVM, LR, deep neural networks (DNNs), DT, NB, RF and kNN. The Cleveland and Statlog datasets are used. All of the classifiers are presented in a formal and decent way to summarise the algorithm. Several different metrics are utilised, including the accuracy, sensitivity, specificity, precision, negative predictive value (NPV), F1-score and Matthews correlation coefficient (MCC). Testing is done via 5-fold and 10-fold cross validation. In 10-fold cross validation, with respect to the F1-score, SVM is the winner for both the Cleveland and Statlog datasets. In terms of

the accuracy, DNN is the winner for the Statlog dataset and SVM is the winner for the Cleveland dataset. The implementation ranges from mobile to web-based technologies and the coding language is Python. The authors compare their results with existing studies and claim that they have achieved better scores. Although the work is very detailed and many classification measures are employed, it is not clear how their systems differ from the existing systems. Clearly, it is not because of the data pre-processing part, because that part only involves the missing value replacement of at most 6 observations. Since there are no feature selection and feature transformation steps in the pipeline, one can deduce that this success is due to the libraries used, but no details are given about this subject.

The work [41] is a study on the (extended) Cleveland dataset in which the NB, DT and ANN classifiers are compared. For the default setting, in which WEKA routines are used on a 13-feature dataset, the accuracies are 86.53%, 89.0% and 85.53% for NB, DR and ANN, respectively. The methodology is not clear; no training/testing split or k-fold cross-validation was applied, so we assume that the results are due to the training data only, which somewhat invalidates the reliability of the research; this resembles the situation in [33]. One advantage of the study is that there is a GA component that is tested with NB, DT and clustering (it is mysterious that classification via clustering suddenly appears in the paper after the GA). For the GA, 6 features are selected with a crossover probability of 0.6 and a mutation probability of 0.033. In combination with the GA, the DT is the most successful method, yielding an accuracy score of 99.2%; NB achieved an accuracy of 96.53%. Clustering here achieves a relatively low score of 88.3\%, which is better than the scores achieved by NB and the ANN without the GA. There is additionally a description of the a priori and maximal frequent itemset algorithm (MAFIA) but this is again used in a `pop-up' manner in the paper because there are no empirical details concerning this algorithm.

The authors of [42] propose a system built upon NB. The Cleveland dataset is used and various training/testing ratios are utilised. For a testing size of 240, the overall accuracy is 89.6% (five-category classification: no risk, low risk, moderate risk, high risk and very high risk). For testing sizes of 290 and 276, the accuracies are 89.0% and 88.8%, respectively. The only data source is the Cleveland dataset and no feature selection scheme is applied. After a brief definition of machine learning terminology (supervised learning and unsupervised learning), the authors describe the

dataset and report their results for various numbers of test observations. The pre-processing step involves the removal of 6 missing-value vectors. This study is limited in various ways. No alternative methods are tested and no automatic pre-processing algorithm is employed.

The authors of [43] also propose a decision support system that uses NB. The Cleveland dataset is used. The organisation of the article is educational rather than research-focused. The step-wise explanation of NB is enriched by probability calculation details such as the use of the Gaussian distribution and categorical data handling. However, the methodology is missing; no measurements, comparisons or other empirical elements are reported. The Cleveland dataset is the only data source, and NB is the only method used. No pre-processing is applied, and no feature selection is performed, similar to [42]. Nevertheless, the introductory manual calculation on a computer-buying dataset has educational value.

The work [44] is an extensive work that concentrates on accuracy improvements via feature selection. Although the authors refer to the Cleveland dataset as UCI in the paper, there are also places in the study where the Cleveland dataset is referred to as a separate entity, which is confusing. There is also some confusion regarding the list of classifiers; a `Logistic Regression (SVM)' expression exists, but this contradicts the very definitions of each of these classifiers. Is it an SVM with a special kernel? Is it a hybrid LR-SVM? These questions cannot be directly answered using the paper since these explanations are missing. We assume that `LR (SVM)' stands for SVM and report that their most successful result was obtained by the minimum redundancy maximum relevance (MRMR) SVM; a score of 84.9% is reported. This score is above the MLP/SVM scores of other works cited here (84.2%). NB and RF are also tested together with MRMR, yielding approximately the same accuracy score: 84.2%. The DT and LR methods are also used. The tool the experiments were carried out on was RapidMiner. The pipeline in this paper also involves various pre-processing methods, including cleaning, transformation, reduction and binning, but the details are not given.

The work [45] is a comprehensive review of the subject of heart disease prediction. Following the definition of the tasks to be performed using machine learning, the authors present tables containing the paradigms of automatic heart disease prediction. They guess that future trends will involve soft computing, in which `multi-agent technologies' are employed. A list of popularly used tools is given; it includes

WEKA, TANAGRA, MATLAB, Orange, .NET and RapidMiner. The listed methods include the DT, NB, SMO, AdaBoost and ANN methods. According to the study findings, the ANN had an accuracy of 100.0% and the DT had an accuracy that was above 99.6% (on 15 features).

In [46], an MLP system is applied to predict heart disease. The precision and recall values are 0.91 and 0.89, respectively. The IDE used is PyCharm, and the selected machine learning library is sklearn. No pre-processing methods, such as scaling and dimension reduction, are carried out before classification. On the other hand, to exclude expensive lab measurements, the authors proposed a subset of 7 features: age, sex, blood pressure, heart rate, diabetes, hyper-cholesterol and body mass index. They also recommend the usage of several sensors for the practical application of their system, including AliveKor, MyHeart, HealthGear and Fitbit.

The authors of [47] evaluated J48, Reptree, NB, a Bayesian net and Classification and Regression Tree (CART) on the South Africa dataset. The accuracy values of the 10-fold cross validation are 0.991, 0.991, 0.972, 0.981 and 0.991 for J48, Reptree, NB, the Bayesian net and CART, respectively. The considered attributes of the dataset are the following: gender, age, chest pain type, blood pressure level, cholesterol, heart rate, smoking status, blood sugar and electrocardiogram (ECG). The environment in which the experiments were performed was WEKA. This work also utilised only one dataset and no data pre-processing was applied. Additionally, an explanation or summary of the classification techniques is missing.

Table 3.1: Literature Results

Team	Dataset	Accuracy
Bhatla et al. [18]	Cleveland	86.5
Repaka et al. [19]	Cleveland	89.0
Shah et al. [23]	Cleveland	84.5
Dangare et al. [25]	Cleveland	100.0
Khan et al. [27]	Hungarian	93.3
Rajdhan et al. [31]	Cleveland	90.2
Fitriyani et al. [32]	Chandigarh	95.9
Taneja et al. [35]	Cleveland	95.6
Singh et al. [36]	Cleveland	100.0
Methaila et al. [41]	Cleveland	99.2
Medhekar et al. [42]	Cleveland	89.6
Bashir et al. [44]	Cleveland	84.2

Ensemble methods have been successfully applied to heart disease prediction [2]. In [1], a decision support system is introduced that uses majority voting ensembles.

The authors of [4] established an ensemble of `heterogeneous' classifiers to build an effective system. In [48], another ensemble was found to provide accurate medical diagnoses.

The authors of [49] suggest using an MLP to model heart disease data. They report high accuracy results, indicating the effectiveness of neural network schemes in this domain. The authors of [50] introduce random forest swarm optimisation and test it on the Eric and Spectf datasets. In [51], a `feature boundaries' one-class classification method is applied and compared with other one-class classification models. The work [52] provides an informative survey of heart disease prediction.

Although deep learning [53] is mostly known for image and video input, there exist applications involving tabular data [54]. Exploring the power of neural networks in this domain, the authors of [55] tested such deep learning methods. The authors of [56] constructed a supervised neural network to detect heart disease based on fuzzy sets.

Support vector machines (SVMs) [57] are another widely applied classification method. The authors of [58] selected features via an SVM to estimate the heart disease risk. They reported specificity, sensitivity and accuracy scores on different datasets. The authors of [59] employed `mean Fisher-based feature selection' and `accuracy-based feature selection' before using a radial basis function SVM. The authors of [60] took advantage of `hybrid' forward feature selection, which was followed by SVM categorisation.

7.2 ARTIFICIAL NEURAL NETWORKS

Artificial neural networks are inspired by human brain biology. Inspired by a natural neuron, an artificial neuron has inputs, weights and activation functions [61]. The inputs are like synapses, weights resemble signals and activation involves a mathematical function that calculates the response of the neuron [61]. After their first appearance [62], ANNs with various architectures were designed; these architectures range from backpropagation ANNs [63] to deep learning systems [53]. In this section, backpropagation ANNs are considered and briefly explained.

ANNs have input, hidden and output layers [61]. The input layer is where the data are fed into the system. As the name implies, it represents the input of the network. A hidden layer is an intermediate [61] collection of nodes in which additional calculations are performed. By node, we mean a neuron or other complex unit (maybe

even a network itself [61]). The output layer provides the results of the system. A backpropagation ANN is a supervised mode l[61], which means that one provides input and output tuples to the machine to train the framework. The name `backpropagation' comes from the idea of reducing the error term by `propagating' the error backwards, while the layers feed information forward. The activation function is a weighted sum [61]

$$A_j(\overline{\mathrm{x}}, \overline{\mathrm{w}}) = \sum_{i=1}^{n} x_i w_{ji} \tag{3.1}$$

Rather than using the identity function, one generally selects the sigmoidal function [61] due to its nice mathematical interpretation:

$$O_j(\overline{x}, \overline{w}) = \frac{1}{1 + e^{A_j(\overline{x}, \overline{w})}} \tag{3.2}$$

The error function is then [61]:

$$E_j(\overline{x}, \overline{w}, d) = \left(O_j(\overline{x}, \overline{w}) - d_j\right)^2 \tag{3.3}$$

The error of the network is the sum over all output units:

$$E(\overline{x}, \overline{w}, d) = \sum_j \left(O_j(\overline{x}, \overline{w}) - d_j\right)^2 \tag{3.4}$$

The adjustment of the weights is done via gradient descent according to the following relation [61]:

$$\Delta w_{ji} = -\eta \frac{\delta E}{\delta w_{ji}}. \tag{3.5}$$

7.3 SUPPORT VECTOR MACHINES

Support vector machines, which were originally proposed by Vapnik et al. [64], are a generalisation of linear classifiers to possibly infinite-dimensional spaces via kernel tricks [65]. By a linear classifier, we mean a classifier in which the decision function is of the form [57] $f(x) = sign(\langle w, x \rangle) + b)$.

Here, $x \in R^n$ is an input vector and w is the weight vector. The optimal hyperplane is defined `as the one with the maximal margin of separation between the two classes' [57]. w can be uniquely represented as a linear combination of input vectors lying on the margin (support vectors) [57]: $w = \sum_i v_i x_i$. Thus, all information that is used to categorise the patterns is stored in a subset of the training set. Therefore, the final decision function is

$$f(x) = sign\left(\sum_i v_i \langle x_i, x \rangle + b\right) \tag{3.6}$$

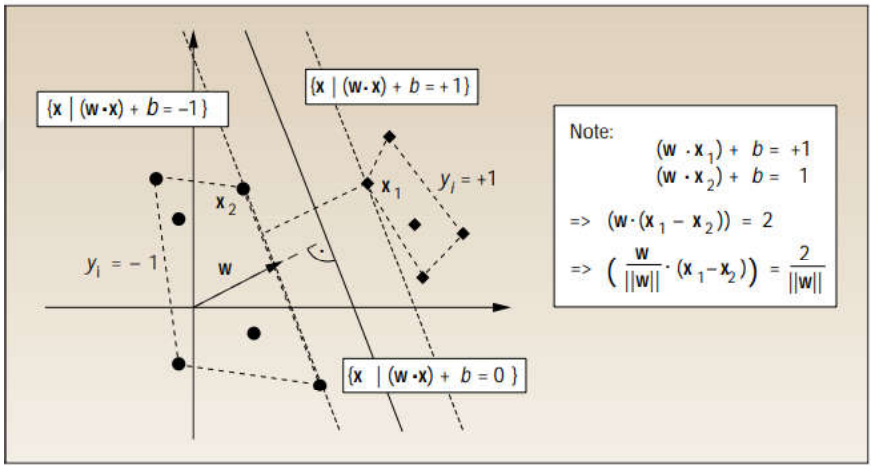

Figure 3.2: An SVM optimal hyperplane [66]

Moreover, if one defines a mapping by taking the input space as the domain and the feature space as the range [67] $\Phi: R^n \to F$ and lets $k(x, y) = \langle \Phi(x), \Phi(y) \rangle$ [67], then the decision function in the feature space can be represented as

$$f(x) = sign\left(\sum_i v_i k(x_i, x) + b\right) \tag{3.7}$$

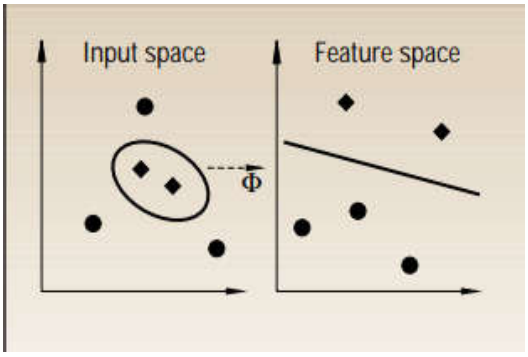

Figure 3.3: Kernel-induced feature space [66]

Φ or k is chosen such that the classes are separated linearly in the feature space. This is sometimes called implicit mapping [67], because one does not need the explicit representation of the features but only the kernel values -- that is, the inner products on the feature space. A linear kernel is induced via identity mapping: $\Phi(x) = x$. A polynomial kernel is of the form $\langle x, y \rangle^d$ (here, d is the degree). A radial basis function (RBF) kernel, on the other hand, has the form $k(x, y) = exp(-\|x - y\|/2\sigma^2)$.

Additionally, kernels are not restricted to fixed-dimensional input spaces; one can define kernels on vector sets [68] and on strings [69].

7.4 NAIVE BAYES

The naive Bayes classifier is backed by Bayes' Theorem, and the probability of an input x belonging to a given class c_j is calculated with the following formula [70]:

$$P(x|c_j) = P(x_1|c_j) \times P(x_2|c_j) \times \ldots \times P(x_N|c_j). \tag{3.8}$$

Then, the decision function is

$$\arg\max_j P(x|c_j). \tag{3.9}$$

In the case of continuous variables, $P(c_j)$ is usually calculated by assuming a normal distribution [71]:

$$P(c_j) = \frac{1}{\sqrt{2\pi}\sigma_j} exp\left(\frac{-1}{2}\frac{(x_j - \mu_j)^2}{\sigma_j^2}\right). \quad (3.10)$$

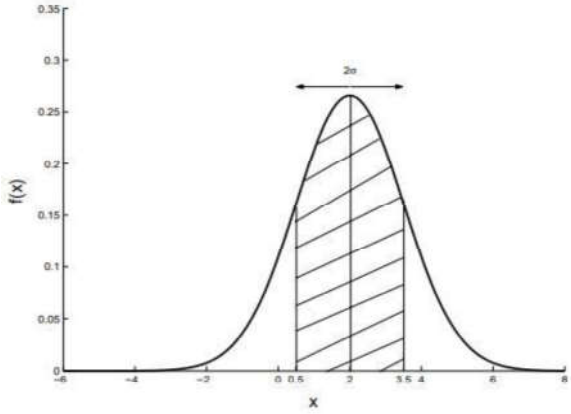

Figure 3.4: A Gaussian probability distribution function [72]

In [73], a heart disease prediction system is developed by building a 15-dimensional dataset via questionnaires and naive Bayes. The authors of [42] proposed a naive Bayes system based on the Cleveland dataset (a 15-dimensional training dataset). The authors of [74] applied k-means clustering before naive Bayes classification.

7.5 CHAOS FIREFLY ATTRIBUTE REDUCTION AND FUZZY LOGIC

One of the state-of-the-art approaches to heart disease prediction is introduced in [3]. In this approach, the firefly algorithm (FA) and rough set feature reduction are followed by type-2 fuzzy logic classification. There are three main properties of the FA [75]:

- All fireflies are unisex; hence, any two fireflies may be attracted to each other.
- For any two fireflies, the one with less brightness will move towards the other one. If, for a firefly, there is no brighter firefly, then its movement will be random in space.
- Brightness is calculated using the fitness function.

Attraction is calculated using the formula:

$$\beta = \beta_0 \times e - \gamma r_{ij},$$

(3.11)

where r_{ij} is the distance between two fireflies i and j [3]. β_0 is the attraction parameter defining the attractiveness at $r = 0$ [3] and γ is the light absorption coefficient [75]. The Gaussian map for the attraction parameter is [3], [75]

$$\beta_0(t+1) = \begin{cases} 0, & \text{if } \beta_0(t) = 0 \\ \frac{1}{\beta_0(t)} - \left[\frac{1}{\beta_0(t)}\right], & \text{otherwise} \end{cases}. \quad (3.12)$$

The authors of [3] combined this FA approach with rough sets [76] and `an interval type-2 Takagi–Sugeno–Kang fuzzy logic system' [3] to diagnose heart disease. A similar approach is used in [77] on the Cleveland, Hungarian and Switzerland datasets. The authors of [78] fused fuzzy logic with decision trees [79, 80] to detect coronary heart disease. The authors of [17] implemented a `coactive neuro-fuzzy inference system' (CANFIS) and tested their framework on the Cleveland dataset. The authors of [81] built a pipeline consisting of a genetic algorithm and fuzzy logic.

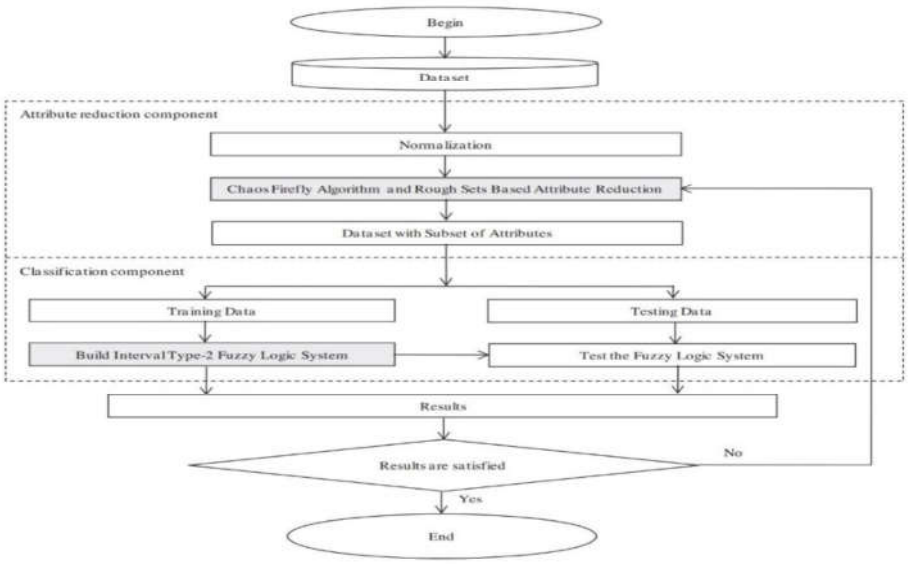

Figure 3.5: Chaos Firefly Attribute Reduction and Fuzzy Logic (CAFL) architecture

7.6 BAGMOOV

BagMOOV stands for `bootstrap aggregation with multi-objective optimised voting' and it was suggested in [4]. This extensive work considers 5 datasets (SPECT, SPECTF, Heart disease, Statlog and Eric) and reports accuracy, sensitivity, specificity and F-measure results after 10-fold cross-validation. The authors incorporate the idea of multi-objective optimised voting [82] using an ensemble of 5 base estimators: NB, linear regression [83], quadratic discriminant analysis [84], an instance-based learner [85] and SVMs. They introduced a weight calculation for each base estimator to obtain the best final decision. Of all the methods discussed so far, BagMOOV is the most reminiscent of our methods. The fundamental difference is that BagMOOV uses state-of-the-art classifiers as base estimators and modifies the `bagging' part by introducing a weighting scheme, whereas our methods concentrate on creating new base predictors and keep the classical bagging scheme.

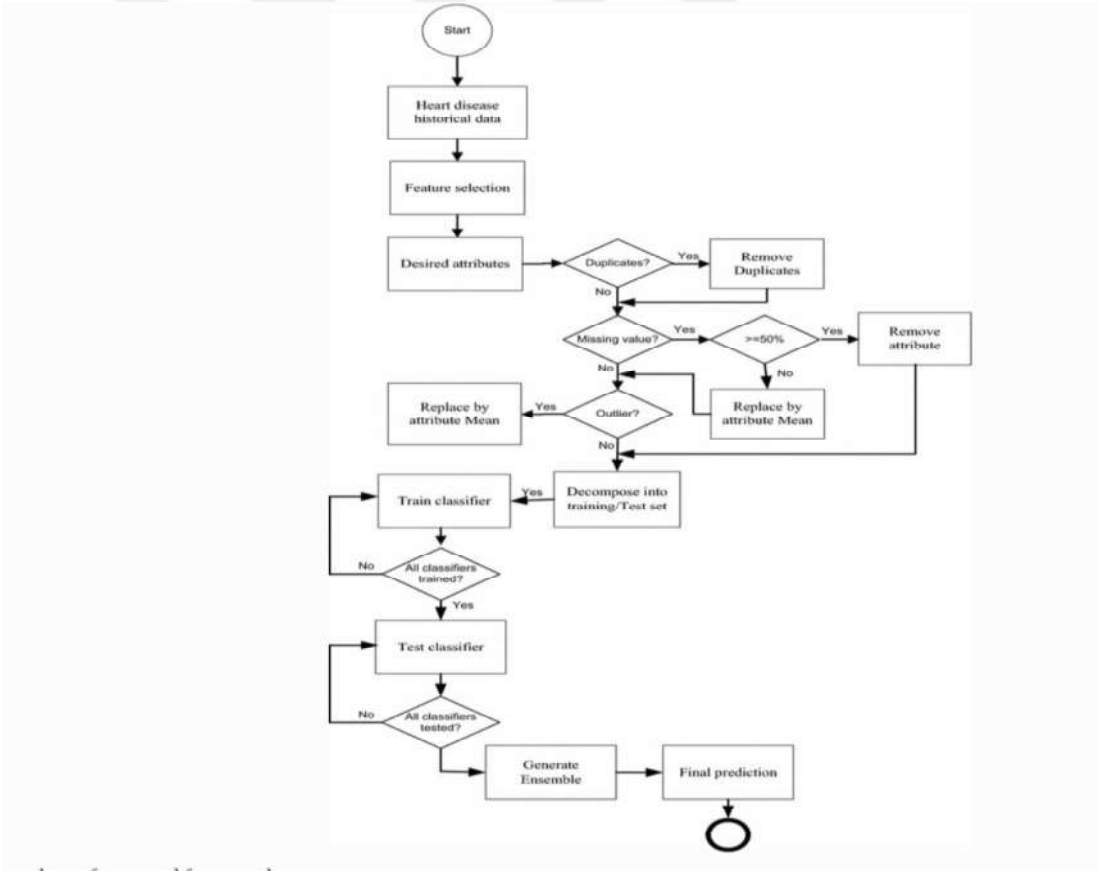

FIgure 3.6: BagMOOV architecture [4]

An early bagging approach to heart disease can be seen in [86]; these researchers work on the Cleveland dataset and compare their results with decision trees. The authors of [87] proposed a model that integrates fuzzy logic, bagging and gradient boosting [88] approaches. The authors of [89] analysed the ensemble method performance when classifiers are applied after swarm optimisation [90] feature selection. The author of [91] showed that ensemble methods outperform standalone base classifiers in the detection of coronary heart disease.

CHAPTER IV

PROPOSED TECHNIQUE

8.1 COMPONENTS

Although the name of this chapter implies `one technique', here we introduce the components of our `proposed techniques', since we propose several methods. Actually, our work can be seen as the `evolution' of a method; it begins with high accuracy and precision-recall values, develops towards balanced accuracy and sensitivity-specificity values and finally ends with an optimised precision (OP) [92] that is approximately 12% better than that of state-of-the-art methods. Moreover, two methods are introduced to predict heart disease using a popular dataset (Cleveland).

First, we begin with the bagged reference vector classifier (BRVC) and bagged shrunk covariance classifier (BSCC). Second, we continue by developing the Gaussian probability and kurtosis-based majority voting bagging classifier (GKMVB). Third, we introduce the Maxwell distribution and KS-test-based majority voting bagging classifier (MKMVB). Fourth, the density estimation majority voting bagging classifier (DEMVB) is proposed. Fifth, a bagged majority voting nearest centroid algorithm (BMVNC) is proposed. Despite the fact that our contributions are mainly `base estimators', there are actually components of transformation, normalisation and feature selection since each of these processes contributes to the success of the system. Therefore, we will talk about this as well. Sixth, a method is applied after extensive pre-processing steps, including scaling, neighbourhood component analysis, recursive cross-validated feature selection (as in GKMVB, MKMVB and DEMVB), fuzzy rough set-based instance selection and random oversampling. Additionally, a gradient boosting algorithm is applied after the same steps.

8.1.1 BRVC & BSCC

A reference vector classifier analyses the randomness of binary label sequences associated with the sorted array of distances to a vector. Let u be an observation. The

distances of other observations to u are calculated and sorted. Then, the binary sequence, which is the label sequence associated with the distance sequence, is analysed and assigned a randomness value [93]. The more non-random this number is, the more important the observation is. n such important observations, their distances and the associated label sequences are kept to form a decision function.

Algorithm 2 Reference Vector Classifier (RVC) Training

procedure FIT(X, y, n) ▷ Dataset, class labels and number of reference vectors
    $y\_all \leftarrow list()$ ▷ Binary sequence list
    $dist\_all \leftarrow list()$ ▷ Distance measurement list
    $r\_all \leftarrow list()$ ▷ Randomness values for each binary sequence
    $distances \leftarrow euclidean\_distances(X)$ ▷ Matrix of distances between observations
    $i \leftarrow 0$
    while $i < len(X)$ **do**
        $dist \leftarrow distances[i, :]$ ▷ i-th row of distance matrix
        $(dist\_y, dist\_1) \leftarrow extract\_binary\_seq(dist, y)$ ▷ Find binary sequences and distances associated with this observation; these are the corresponding labels when distances are sorted
        $r \leftarrow extract\_randomness(dist\_y)$ ▷ Calculate randomness value
        $append(r\_all, r)$
        $append(dist\_all, dist\_1)$ ▷ Append row
        $append(y\_all, dist\_y)$
        $i \leftarrow i + 1$
    end while
    $(ref\_vecs, ref\_labels, ref\_distances) \leftarrow get\_ref\_vecs($
    $X, y\_all, r\_all, dist\_all, r$
    ▷ This is done by sorting according to r values

return $(ref\_vecs, ref\_labels, ref\_distances)$
end procedure

Figure 4.1: RVC training

Algorithm 3 Reference Vector Classifier (RVC) Prediction

procedure PREDICT($x, n, m, ref\_vecs, ref\_labels, ref\_distances$) ▷ ref_vecs, ref_labels and $ref\_distances$ are results from the FIT() procedure, x is the unseen test observation

$label\_0 \leftarrow 0$

$label\_1 \leftarrow 0$

$i \leftarrow 0$

while $i < n$ **do**

$i \leftarrow i + 1$

$dist \leftarrow euclidean\_distance(x, ref\_vecs_i)$ ▷ Distance from x to i-th reference vector

$idx \leftarrow binary\_search(dist, ref\_distances_i)$ ▷ Search for closest value in reference distances

$patch \leftarrow get\_subseq(ref\_labels_i, idx, m)$ ▷ Get subsequence of length $2m + 1$ around idx

if $sum(patch) < m$ **then**

$label\_0 \leftarrow label\_0 + 1$

else

$label\_1 \leftarrow label\_1 + 1$

end if

end while

$y \leftarrow 0$

if $label\_0 > label\_1$ **then**

$ret\_val \leftarrow 0$

else

$ret\_val \leftarrow 1$

end if

return ret_val

end procedure

Figure 4.2: RVC prediction

FIT() represents the training phase of the algorithm. n is a hyperparameter: the number of reference vectors. $extract\_randomness$ refers to the randomness calculation function. We have benefitted from using the exponential of the autocovariance [94]. PREDICT() is the function that is used to find the label of a given test vector. For each reference vector u, the distance of the test vector x is computed and located in the corresponding sequence. Then, its location in the aligned binary sequence is found to calculate the vote of the reference vector: if there are more $1s$

than 0s in the neighbourhood, then the reference vector returns 1; otherwise, it returns 0. The votes are summed and compared to return the label of the test vector.

The final estimator is BRVC, which is the bagged variant of the reference vector classifier; i.e. each base estimator in the bagging classifier is an instance of the RVC.

In the literature, there have been applications of shrunk covariance matrices, including portfolio optimisation [95] and trajectory classification [96]. In this work, we employ two shrunk covariance estimation methods, namely Ledoit-Wolf [97] and Graph-Lasso [98], to sum weighted Mahalanobis distances [99]. The weighted Mahalanobis distances to the class means are summed and compared to determine the class of a test vector.

Algorithm 4 Shrunk Covariance Classifier (SCC) Training

procedure FIT(X, y) ▷ Training dataset (X) and class labels (y)
  $prec\_0 \leftarrow inv(GraphLasso(X))$ ▷ Fit a Graph-Lasso and find inverse
  $prec\_1 \leftarrow inv(Ledoit - Wolf(X))$ ▷ Fit a Ledoit-Wolf and find inverse
  $mean\_0 \leftarrow get\_class\_mean(X, y, 0)$ ▷ Retrieve mean for class 0
  $mean\_1 \leftarrow get\_class\_mean(X, y, 1)$ ▷ Retrieve mean for class 1
  return $(prec\_0, prec\_1, mean\_0, mean\_1)$
end procedure

Figure 4.3: SCC training

Analogous to BRVC, BSCC represents the bagged shrunk covariance classifier, where each base estimator is an instance of the SCC.

8.1.2 GKMVB & MKMVB

GKMVB analyses each component (attribute, feature) separately and collects votes for each class to predict the label of a vector. Of course, for this to happen, it needs fundamental statistics about each feature; here, these are the mean, variance, kurtosis and KS-test [100] statistics. One-dimensional classification is performed using a variant of the Gaussian probability. Actually, for a given distribution of location μ and scale σ, the probability density function (pdf) has the form [72]

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right). \tag{4.1}$$

Algorithm 5 Shrunk Covariance Classifier (SCC) Prediction

procedure PREDICT($x, prec\_0, prec\_1, mean\_0, mean\_1$) ▷

$prec\_0, prec\_1, mean\_0$ and $mean\_1$ are the results of the FIT procedure, x is the test vector

$a\_0 \leftarrow mahalanobis(x, mean\_0, prec\_0)$ ▷ Find Mahalanobis distance w.r.t. Graph-Lasso precision matrix

$b\_0 \leftarrow mahalanobis(x, mean\_0, prec\_1)$ ▷ Find Mahalanobis distance w.r.t. Ledoit-Wolf precision matrix

$a\_1 \leftarrow mahalanobis(x, mean\_1, prec\_0)$

$b\_1 \leftarrow mahalanobis(x, mean\_1, prec\_1)$

if $a\_0 + 0.25 * b_0 < a\_1 + 0.25 * b_1$ **then**

$ret\_val \leftarrow 0$

else

$ret\_val \leftarrow 1$

end if

return ret_val

end procedure

Figure 4.4: SCC prediction

We instead used

$$g(x) = \frac{1}{\sqrt{2\pi\sigma^4}} exp\left(\frac{-(x-\mu)^2}{2\sigma^4}\right). \tag{4.2}$$

Despite the existence of robust kurtosis estimation methods [101], a sample estimation of the kurtosis [102] is preferred since this yielded better results:

$$\kappa = \frac{\widehat{\mu_4}}{\widehat{\sigma^2}} - 3, \tag{4.3}$$

where $\widehat{\mu_4}$ is the sample moment defined by the formula

$$\widehat{\mu_r} = n^{-1} \sum_{i=1}^{n} (x_i - \bar{x})^r, \tag{4.4}$$

and $\widehat{\sigma^2}$ is the sample variance.

Algorithm 6 Proposed Base Estimator Method I: GKMVB Training

procedure FIT(X, y) ▷ Dataset, class labels
  $means\_0 \leftarrow calc\_means(X, y, 0)$ ▷ Means of features for class 0
  $means\_1 \leftarrow calc\_means(X, y, 1)$ ▷ Means of features for class 1
  $vars\_0 \leftarrow calc\_variances(X, y, 0)$ ▷ Variances of features for class 0
  $vars\_1 \leftarrow calc\_variances(X, y, 1)$ ▷ Variances of features for class 1
  $kurtosis \leftarrow calc\_kurtosis(X)$ ▷ Kurtosis of each feature
  $ks \leftarrow ks\_test(X)$ ▷ KS test result of each feature
return $(means\_0, means\_1, vars\_0, vars\_1, kurtosis, ks)$
end procedure

Figure 4.5: GKMVB training

MKMVB differs from GKMVB in terms of the probability density function that is used (it should be noted that our formula in (4.2) is not actually a pdf but instead is a `quasi'-distance). The Maxwell-Boltzmann pdf($x, \theta > 0$) [103] is

$$f(x) = \frac{4}{\sqrt{\pi}} \frac{1}{\theta^{3/2}} x^2 exp(-x^2/\theta). \tag{4.5}$$

Its cumulative distribution function (cdf) is [103]

$$F(x) = \frac{1}{\Gamma(3/2)} \Gamma\left(\frac{3}{2}, \frac{x^2}{\theta}\right), \tag{4.6}$$

where Γ is the incomplete gamma function

$$\Gamma(a, x) = \int_0^x u^{a-1} e^{-u} du. \tag{4.7}$$

Algorithm 7 Proposed Base Estimator Method I: GKMVB Prediction

procedure PREDICT($x, c\_0, c\_1, means\_0, means\_1, vars\_0, vars\_1, kurtosis, ks$)
▷ $means\_0, means\_1, vars\_0, vars\_1, kurtosis$, and ks are results of the FIT procedure. c_0 and c_1 are method parameters.
    $s\_0 \leftarrow 0$ ▷ Initialize votes for class 0.
    $s\_1 \leftarrow 0$ ▷ Initialize votes for class 1.
    $i \leftarrow 0$
    $D \leftarrow dim(x)$ ▷ Number of dimensions
    while $i < D$ **do**
        $val\_0 \leftarrow calc\_dens(x[i], means\_0[i], vars\_0[i])$ ▷ Given mean and variance, calculate density according to Equation ??
        $val\_1 \leftarrow calc\_dens(x[i], means\_1[i], vars\_1[i])$ ▷ Repeat for class 1
        if $val\_0 > val\_1$ **then** ▷ Feature class based probabilities are compared
            $s\_0 \leftarrow c\_0 + exp(-kurtosis[i]) + ks[i]$ ▷ Kurtosis and KS-statistic added
        else
            $s\_1 \leftarrow c\_1 + exp(-kurtosis[i]) + ks[i]$
        end if
        $i \leftarrow i + 1$
    end while
    $y \leftarrow 0$ ▷ Class of x
    if $s\_0 > s\_1$ **then**
        $y \leftarrow 0$ ▷ Class of x to 0.
    else
        $y \leftarrow 1$ ▷ Class of x to 1.
    end if
return y
end procedure

Figure 4.6: GKMVB prediction

Algorithm 8 Maxwell-Boltzmann Distribution-based Majority Voting Classifier Training

procedure FIT(X, y) ▷ Dataset, class labels
    $rv\_0 \leftarrow list()$ ▷ Maxwell-Boltzmann random variables for class 0
    $rv\_1 \leftarrow list()$ ▷ Random variables for class 1
    $ks \leftarrow list()$ ▷ KS-test result of each feature
    $D \leftarrow dim(X)$ ▷ Number of dimensions
    $i \leftarrow 0$
    while $i < D$ **do**
        $m\_0 \leftarrow frv(X, i, 0)$ ▷ Fit Maxwell random variable for class 0
        $m\_1 \leftarrow frv(X, i, 1)$ ▷ Fit Maxwell random variable for class 1
        $ks\_0 \leftarrow cks(X, i, 0, m\_0)$ ▷ Compute KS statistic for class 0
        $ks\_1 \leftarrow cks(X, i, 0, m\_1)$ ▷ Compute KS statistic for class 1
        $append(rv\_0, m\_0)$ ▷ Append values to lists
        $append(rv\_1, m\_1)$
        $append(ks, ks\_0 + ks\_1)$
        $i \leftarrow i + 1$
    end while
return (rv_0, rv_1, ks)
end procedure

Figure 4.7: MKMVB training

8.1.3 DEMVB

Applying standard probability density functions and weighting using the KS-test statistic may give good results. However, what about tackling the problem from a different perspective? That is, what happens if we try to approximate the cumulative distribution function and then take its derivative to obtain the density function? In this

method, kurtosis and the KS statistic are automatically excluded since $F(x)$ is approximated by the cumulative step function $S_N(x)$.

Two strategies are applied: the first is DEMVB-I. In this strategy, a polynomial $P(x)$ is fitted to $S_N(x)$, and its derivative $P'(x)$ is computed to approximate the density function. Second, in the DEMVB-II strategy, the derivative of $F(x)$ is directly calculated using $S_N(x)$ and the central difference formula. From a theoretical viewpoint, for a value $x_i < x < x_{i+1}$, where x_i and x_{i+1} are the nearest sample values, setting $F(x)$ to i/N is not satisfying since the left and right limits of $F(x_i)$ are not equal, which implies that the derivative does not exist. This is also true for a linear approximation of $F(x)$ that is calculated using $S_N(x)$. However, in practice, after the conduction of experiments, we have seen that a central difference formula can outperform the state-of-the-art methods. This, of course, does not mean that the proposed density estimation method is the best method available, as there are many sophisticated methods for estimating the density. By state-of-the-art results, we mean the overall results achieved after density estimation-based one-dimensional classification, majority sums and bagging in the context of heart disease prediction.

Algorithm 9 Maxwell-Boltzmann Distribution-based Majority Voting Classifier Prediction

procedure PREDICT($x, c\_0, c\_1, rv\_0, rv\_1, ks$) ▷ rv_0, rv_1, and ks are results from the FIT procedure. c_0 and c_1 are method parameters.
    $s\_0 \leftarrow 0$ ▷ Initialise votes for class 0
    $s\_1 \leftarrow 0$ ▷ Initialise votes for class 1
    $i \leftarrow 0$
    $D \leftarrow dim(x)$ ▷ Number of dimensions
    while $i < D$ **do**
        $val\_0 \leftarrow pdf(x[i], rv\_0[i])$ ▷ Given random variable, calculate probability density of $x[i]$
        $val\_1 \leftarrow pdf(x[i], rv\_1[i])$ ▷ Repeat for class 1
        if $val\_0 > val\_1$ **then** ▷ Feature class-based probabilities are compared
            $s\_0 \leftarrow c\_0 + exp(-|kurtosis[i]|) + ks[i]$ ▷ Kurtosis and KS statistic added
        else
            $s\_1 \leftarrow c\_1 + exp(-|kurtosis[i]|) + ks[i]$
        end if
        $i \leftarrow i + 1$
    end while
    $y \leftarrow 0$ ▷ Class of x
    if $s\_0 > s\_1$ **then**
        $y \leftarrow 0$ ▷ Class of x is 0
    else
        $y \leftarrow 1$ ▷ Class of x is 1
    end if
return y
end procedure

Figure 4.8: MKMVB prediction

8.1.4 BMVNC

BMVNC represents the bagged variant of the majority voting nearest centroid (MVNC) method. MVNC is essentially a `biased' variant of the nearest centroid classifier. The distance to the class mean is added to the distance from the class mean to the overall mean. Weighting is again done using a kurtosis exponentiation, with a small change; this time, the absolute value is not taken.

Algorithm 10 MVNC Training

```
procedure FIT(X, y)                                   ▷ Dataset, class labels
    means_0 ← list()                                  ▷ Means of features for class 0
    means_1 ← list()                                  ▷ Means of features for class 1
    means ← list()                                    ▷ Means of features
    kurtosis ← list()                                 ▷ Kurtosis of each feature
    c_0_tmp ← cc(y, 0)                                ▷ Compute class counts
    c_1_tmp ← cc(y, 1)
    c_0 ← c_0_tmp/(c_0_tmp + c_1_tmp)                 ▷ Calculate class ratios
    c_1 ← c_1_tmp/(c_0_tmp + c_1_tmp)
    D ← dim(X)                                        ▷ Number of dimensions
    i ← 0
    while i < D do
        mean_0 ← cfm(X, i, 0)                         ▷ Compute mean of i-th feature for class 0
        mean_1 ← cfm(X, i, 1)                         ▷ Compute mean of i-th feature for class 1
        mean ← cfm(X, i)                              ▷ Compute mean of i-th feature
        kurt ← cfk(X, i)                              ▷ Compute kurtosis of i-th feature
        append(means_0, mean_0)                       ▷ Append values to lists
        append(means_1, mean_1)
        append(means, mean)
        append(kurtosis, kurt)
        i ← i + 1
    end while
return (c_0, c_1, means_0, means_1, means, kurtosis)
end procedure
```

Figure 4.9: MVNC training

Algorithm 11 MVNC Prediction

```
procedure PREDICT(x, c_0, c_1, means_0, means_1, means, kurtosis)   ▷
c_0, c_1, means_0, means_1, means, and kurtosis are results from the FIT pro-
cedure.
    s_0 ← 0                                           ▷ Initialise votes for class 0
    s_1 ← 0                                           ▷ Initialise votes for class 1
    i ← 0
    D ← dim(x)                                        ▷ Number of dimensions
    while i < D do
        val_0 ← |x[i] − means_0[i]| + |means_0[i] − means[i]|   ▷ Calculate
distances
        val_1 ← |x[i] − means_1[i]| + |means_1[i] − means[i]|   ▷ Repeat for
class 1
        if val_0 > val_1 then      ▷ Feature class-based distances are compared
            s_0 ← c_0 + exp(−kurtosis[i])                 ▷ Kurtosis added
        else
            s_1 ← c_1 + exp(−kurtosis[i])
        end if
        i ← i + 1
    end while
    y ← 0                                             ▷ Class of x
    if s_0 > s_1 then
        y ← 0                                         ▷ Class of x is 0
    else
        y ← 1                                         ▷ Class of x is 1
    end if
return y
end procedure
```

Figure 4.10: MVNC prediction

8.1.5 IWRF

The inner-product wavelet random forest (IWRF) method involves the combination of a wavelet transform scheme and random forest classification. The algorithm exploits the feature extraction capabilities of the discrete wavelet transform via inner-product associated binary sequences. The idea depends on expressing a vector using its inner-product associated binary (label) sequences (which resembles the binary sequence generation of RVC). For each vector u, first, the inner-product collection $I = \{\langle u, v_1\rangle, \langle u, v_2\rangle, \ldots \langle u, v_N\rangle\}$ is calculated. Let I_s be $\left(\langle u, v_{i_1}\rangle, \langle u, v_{i_2}\rangle, \ldots \langle u, v_{i_N}\rangle\right)$ where $\langle u, v_{i_1}\rangle$ is the largest element of I, $\langle u, v_{i_2}\rangle$is the second largest element of I and so on. That is, I_s is the reversely sorted tuple obtained by I. If $y_u = \left(y_{i_1}, y_{i_2}, \ldots, y_{i_N}\right)$ are the corresponding class labels of $v_{i_1}, v_{i_2}, \ldots, v_{i_N}$ and cA_u, cB_u are the results of $dwt(y_u)$, then the transformed features are the first k elements of CA_u. Here, k is a method parameter.

Algorithm 12 IW

procedure FIT($X, y, wavelet, k$) ▷ Dataset, class labels
  return $(X, y, wavelet, k)$ ▷ FIT simply records the dataset and parameters
end procedure
procedure TRANSFORM($x, X, wavelet, y, k$) ▷ $X, y, wavelet$ and k are results from the FIT procedure.
  $inner \leftarrow cip(x, X)$ ▷ Calculate inner products
  $inner\_y \leftarrow extract\_binary\_seq(inner, y)$ ▷ Find binary sequences associated with this observation; these are the corresponding labels when inner products are sorted
  $ca, cb \leftarrow dwt(inner\_y, wavelet)$ ▷ Compute DWT
  return $ca[:k]$ ▷ Return first k elements of ca
end procedure

Figure 4.11: IWRF

8.1.6 XGBFN

XGBFN stands for XGBoost [104] neighbourhood component analysis + fuzzy rough set prototype selection + random oversampling. XGBFN is a direct application that resembles [32]. It is a robust pipeline; that is, its reported metrics are averaged after a series of runs (the random state of each stochastic method is set to a number in a run, and this is repeated 5 times to find the average measurements). XGBoost is probably `the king of tabular data' since a notable number of winners in Kaggle

competitions used XGBoost to crunch the sample matrices. Unlike in [32], we intended to apply the method to the three UCI datasets, the Cleveland, Hungarian and Switzerland datasets, following the methodology of [105].

Neighbourhood component analysis seeks a linear transformation A such that in the transformed space, one obtains a more accurate classification; specifically, the average leave-one-out (LOO) classification is maximised. The effect of linear transformation on the final space is dealt with by considering the whole transformed space via stochastic gradient descent [106].

Fuzzy rough set prototype selection (FRPS) is a sophisticated prototype selection algorithm in which fuzzy logic and rough set theory are unified to select the best prototypes. First, a quality measure called the ordered weighted average (OWA) is introduced and a wide range of thresholds are selected. For each threshold, the corresponding subsets in which the quality of the instances is higher than the threshold are formed [107]. The LOO accuracy of each subset is recorded; that is, the whole dataset is classified with respect to the subset and the resulting accuracy is calculated. Then, the maximal accuracy subset is returned.

Random oversampling is a relatively simple method in which (in our case) the minority class is resampled to balance the dataset.

8.1.7 PREPROCESSING & FEATURE SELECTION

Before GKMVB, MKMVB and DEMVB classification, a quantile transformer [108] and a robust scaler [109] are applied. The quantile transformer maps the input data into the desired distribution (for GKMVB, MKMVB and DEMVB-I, a uniform distribution is chosen, whereas for DEMVB-II, the normal distribution is chosen) and the robust scaler considers the median and interquartile range (IQR) to scale the data. A standard scaler [108] is utilised before BMVNC. A minmax scaler [109] is used before BRVC and BSCC.

For feature selection, BRVC and BSCC are followed by conditional infomax feature extraction (CIFE) [110, 111] and SelectKBest [112] (which is analysis of variance (ANOVA)-based), respectively. Thirty-three features were selected before BRVC and 10 before BSCC. Recursive Feature Elimination with Cross Validation (RFECV) is the method of choice for GKMVB, MKMVB and DEMVB for the Spectf/Statlog pipelines. For the application of DEMVB to the Eric dataset, a quantile

transformer, followed by a standard scaler, is used before SelectKBest. The number of selected features was 6 for this scenario.

8.1.8 PARAMETERS & PIPELINES

For BRVC and BSCC, a grid-search cross-validation is conducted to find the set of bagging classifier parameters: namely, the number of estimators, the maximum sample ratio for sampling and the maximum feature ratio for subspace selection.

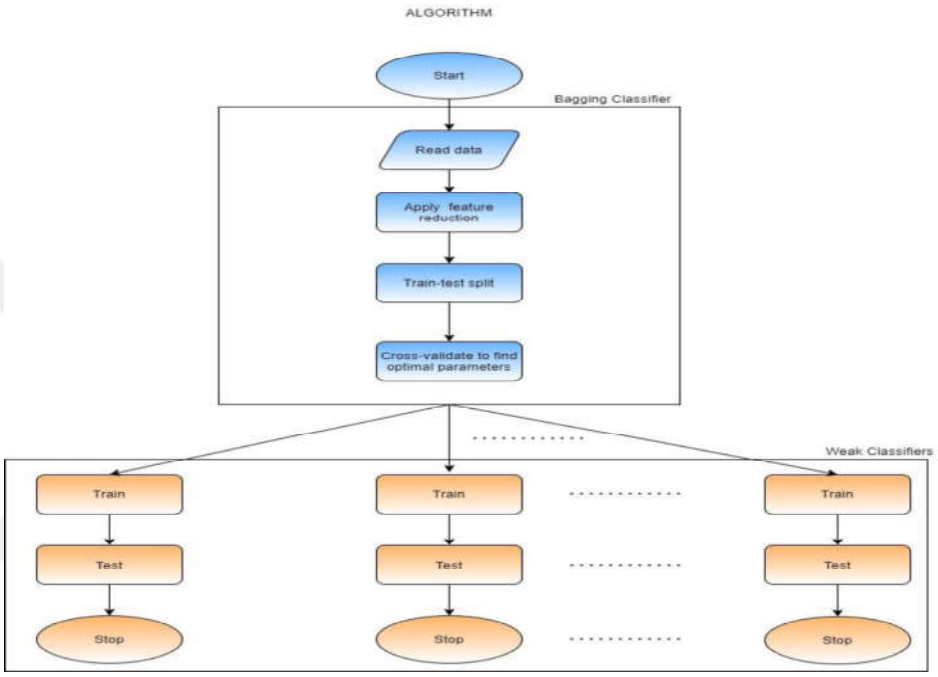

Figure 4.12: BRVC and BSCC pipeline

For GKMVB, MKMVB and DEMVB, no cross-validation is applied. The performance of BMVNC is directly measured via 10-fold cross-validation, as in [4]. The GKMVB, MKMVB and DEMVB pipeline can be seen in Figure 4.13.

For DEMVB-I on Spectf, $c\_0 = c\_1 = 0$ and the polynomial degree is set to 3. For DEMVB-I on Statlog, c_1 is set to 1.5 and c_0 is set to 1.0. The quantile transformer output distribution was set to the default setting: uniform. The number of estimators for the DEMVB-I bagging classifiers was set to 20, the number of features (sampling parameter) was set to 0.5 and the number of samples (sampling parameter) was set to 0.37.

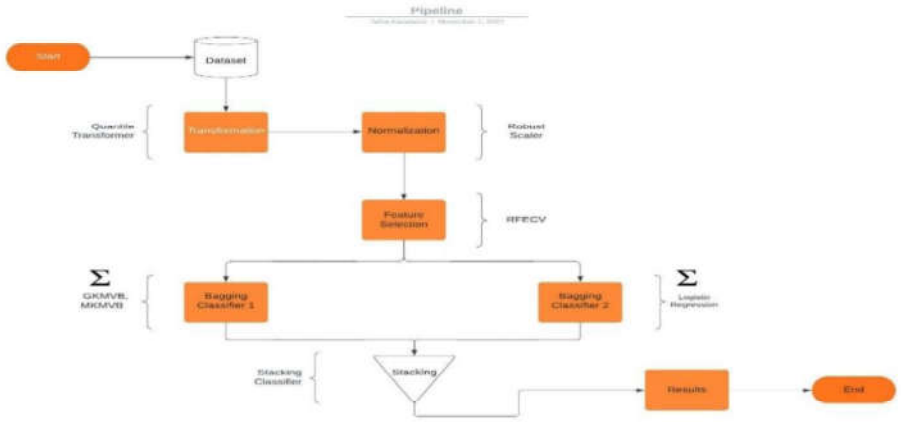

Figure 4.13: GKMVB, MKMVB and DEMVB pipeline

The parameters of `the most robust setting' (the setting for which the average accuracy, sensitivity, specificity and optimised precision were reported) are given below:

- For DEMVB-I on Spectf, $c\_1 = 1.5$, $c\_0 = 1.0$ and the polynomial degree was 3. The quantile transformer output distribution was uniform. The number of quantiles was set to the half of the length of the dataset (half of the number of observations). The base estimator logistic regression regularisation parameter was 0.37. The number of estimators for each bagging classifier was 100.
- For DEMVB-II on Statlog, $c\_1 = c\_0 = 1.0$ and the order of the derivative approximation was 3. The logistic regression regularisation parameter was 0.37. The number of estimators for bagging classifiers was 100.
- For DEMVB-II on Eric, $c\_1 = 1.5$ and $c\_0 = 1.0$. The number of estimators was 50. The number of features for SelectKBest was $k = 6$.
- For IWRF, the number of estimators was 511. SelectKBest was executed while k was 6, and the wavelet transformer feature length was $k = 37$.

The pipeline associated with XGBFN is as follows:

- One-hot encoding of categorical variables;
- Imputation via `median' and `most frequent' strategies;
- Minmax scaling;
- Neighbourhood components analysis feature reduction;
- RFECV feature selection, where the estimator is XGBoost;

- FRPS prototype selection;
- Random oversampling;
- XGBoost 10-fold cross-validation.

8.1.9 RESULTS

Here, we describe the datasets and metrics used. Then, we give the results for each dataset and compare the performance of our methods with the state-of-the-art methods.

8.1.9.1 Datasets

To compare our results with [3], we have conducted experiments on the Spectf and Statlog datasets. Spectf has 44 features obtained from single-photon emission computed tomography (SPECT) images [2]. Statlog, on the other hand, has 13 features: age, sex, chest pain type, resting blood pressure, serum cholesterol in mg/dl, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise relative to rest, the slope of the peak exercise ST segment, the number of major vessels (0--3) coloured by fluoroscopy and defect type [2]. One third of each dataset is used for training, and the rest is reserved for testing.

To compare our results with [4], the Eric dataset is analysed. This dataset has 210 observations and 7 features: age, chest pain, resting blood pressure, blood sugar, resting electrocardiographic results, maximum heart rate and exercise angina. The results on the Eric dataset are reported after 10-fold cross-validation, as in [4].

To compare our results with [105], the Cleveland (a popular dataset), Hungarian and Switzerland datasets are analysed. The Cleveland dataset has 303 samples with some missing attributes. The Hungarian dataset has 294 observations, while the Switzerland dataset has 123 observations. Each of these three UCI datasets has 13 features.

8.1.9.2 Evaluation

The first prototype of the system is evaluated using the accuracy, precision and recall, where

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%, \tag{4.8}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{4.9}$$

and

$$Recall = \frac{TP}{TP + FN} \times 100\%. \tag{4.10}$$

The results of our system's first prototype, namely BRVC and BSCC, can be seen in Table 4.1 and Table 4.2.

Table 4.1: Accuracy Comparison

Method	Spectf	Statlog
NB	79.7	85.2
SVM	79.7	81.5
ANN	77.0	81.5
CAFL	87.2	88.3
BRVC, BSCC	**88.7**	**88.8**

Table 4.2: Precision and Recall Measurements

Dataset	Precision	Recall	F-score
BSCC (Statlog)	85.5	89.8	87.6
BRVC (Spectf)	91.4	93.7	91.4

The second and third versions of the system are evaluated using the sensitivity, specificity and optimised precision [92], where

$$Sensitivity = \frac{TP}{TP + FN} \times 100\%, \tag{4.11}$$

$$Specificity = \frac{TN}{TN + FP} \times 100\%, \tag{4.12}$$

and

$$OP = \left(Accuracy/100 - \frac{|Sn - Sp|/100}{(Sn + Sp)/100} \right) \times 100\%. \quad (4.13)$$

Table 4.3: Spectf Results

Method	Accuracy	Sensitivity	Specificity	OP
NB	79.7	100.0	0.0	-20.0
SVM	79.7	100.0	0.0	-20.0
ANN	77.0	89.3	28.9	25.9
CAFL	87.2	94.2	68.9	71.6
GKMVB	88.7	66.0	90.7	**73.4**
MKMVB	83.1	80.0	83.4	**81.0**
DEMVB-I	87.0	80.0	87.7	**82.4**

Table 4.4: Statlog Results

Method	Accuracy	Sensitivity	Specificity	OP
NB	85.2	82.6	87.1	82.5
SVM	81.5	82.6	80.6	80.2
ANN	81.5	82.6	80.6	80.2
CAFL	88.3	84.9	93.3	83.5
GKMVB	88.3	91.7	84.3	**84.1**
MKMVB	86.1	87.6	84.3	**84.2**
DEMVB-I	87.7	89.6	85.5	**85.4**

The results in Table 4.3, Table 4.4 and Table 4.5 represent the maximum values of the 10 runs. A point that must be stated here is that DEMVB is our most robust model since it gave an **average optimised precision** of 75.3 on Spectf, 84.1 on Statlog and 74.4 on Eric, respectively. Here, DEMVB-I (polynomial setting) is applied to the Spectf dataset, and DEMVB-II (centred differencing) is applied to the Statlog and Eric datasets.

Table 4.5: Eric Results

Method	Accuracy	Sensitivity	Specificity	OP
Bagging	73.2	76.9	68.4	67.4
Adaboost	65.1	68.3	60.9	59.2
Stacking	79.4	87.2	69.6	68.1
NNE	77.0	79.5	73.9	73.3
BagMOOV	80.9	86.3	73.9	73.2
RF	81.8	84.5	75.8	**77.0**
IWRF	83.7	86.2	80.3	**80.1**
DEMVB-II	82.5	88.5	78.9	**78.1**
BMVNC	82.7	83.2	82.7	**82.3**

In Tables 4.6, 4.7 and 4.8, AGAFL stands for the state-of-the-art [105] results for the Cleveland, Hungarian and Switzerland dataset combination.

Table 4.6: Cleveland Results

Method	Accuracy	Sensitivity	Specificity	OP
AGAFL	90.0	91.0	90.0	89.0
LPP + RBFL	68.0	79.0	84.0	65.0
RS + FL	73.0	100.0	67.0	53.0
XGBFN	95.0	94.0	96.0	**94.0**

Table 4.7: Hungarian Results

Method	Accuracy	Sensitivity	Specificity	OP
AGAFL	91.0	92.0	88.0	89.0
LPP + RBFL	67.0	87.0	38.0	28.0
RS + FL	70.0	86.0	35.0	28.0
XGBFN	92.0	89.0	94.0	**89.0**

Table 4.8: Switzerland Results

Method	Accuracy	Sensitivity	Specificity	OP
AGAFL	89.0	97.0	75.0	76.0
LPP + RBFL	72.0	76.0	67.0	66.0
RS + FL	63.0	67.0	72.0	59.0
XGBFN	99.0	100.0	97.0	**97.0**

CHAPTER V

CONCLUSION

We have seen that ensemble methods, specifically bagging classifiers with custom base estimators, are effective tools for the prediction of heart disease. The RVC and SCC were written by exploiting the binary sequences associated with distance measures and shrunk covariance estimators, respectively. These methods yielded good accuracy results when they were backed by the proper feature selection. The precision and recall results were also satisfactory, but the specificity and sensitivity scores needed major improvement. We solved this problem by adopting a better pre-processing scheme, namely by applying quantile transformation and recursive feature elimination. Then, several majority voting classifiers were proposed: GKMVB, MKMVB and DEMVB. The first of these was based on the idea of weighting the features using kurtosis and KS-test values. The second was derived by using an implementation of the Maxwell-Boltzmann distribution, and the third involved the estimation of density functions by analysing the empirical cumulative distribution function.

Our claim is that this work is valuable for two reasons: first, we have achieved results that are better than those of the state-of-the-art methods. Second, our results were obtained via new classifiers (except for XGBFN). As we have seen, the heart disease prediction literature mostly consists of the `application' of some well-known method to the problem. Although a focus on application absolutely does not lower the value of these methods, some consideration of novelty is also needed. Of course, on some level, all methods are reducible to the application of some well-known theory, but striving for the development of new classifiers is still an important challenge. We hope that our methods are usable in other application contexts.

REFERENCES

[1] BASHIR Saba, QAMAR Usman, KHAN Farhan Hassan and JAVED M Younus (2014), "MV5: A Clinical Decision Support Framework for Heart Disease Prediction Using Majority Vote Based Classifier Ensemble", *Arabian Journal for Science and Engineering*, vol. 39, no. 11, pp. 7771–7783.

[2] KARADENIZ Talha, TOKDEMIR Gül and MARAŞ Hadi Hakan (2021), "Ensemble Methods for Heart Disease Prediction", *New Generation Computing*, vol. 39, no. 3, pp. 569–581.

[3] LONG Nguyen Cong, MEESAD Phayung and UNGER Herwig (2015), "A Highly Accurate Firefly Based Algorithm for Heart Disease Prediction", *Expert Systems with Applications,* vol. 42, no. 21, pp. 8221–8231.

[4] BASHIR Saba, QAMAR Usman and KHAN Farhan Hassan (2015), "BagMOOV: A Novel Ensemble for Heart Disease Prediction Bootstrap Aggregation with Multi-objective Optimized Voting", *Australasian Physical & Engineering Sciences in Medicine*, vol. 38, no. 2, pp. 305–323.

[5] RE Matteo and VALENTINI Giorgio (2012), "Ensemble methods", In*, Advances in Machine Learning and Data Mining for Astronomy*, Ed. Michael J. Way, pp. 563–593, CRC Press, Boca Raton.

[6] RAYAROTH Rejeesh (2019), "Random Bagging Classifier and Shuffled Frog Leaping Based Optimal Sensor Placement for Leakage Detection in WDS", *Water Resources Management*, vol. 33, no. 9, pp. 3111–3125.

[7] BÜHLMANN Peter and YU Bin (2002), "Analyzing Bagging", *The Annals of Statistics*, vol. 30, no. 4, pp. 927–961

[8] BAUER Eric and KOHAVI Ron (1999), "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants", *Machine Learning*, vol. 36, no. 1, pp. 105–139.

[9] PHAM Hieu and OLAFSSON Sigurdur (2019), "Bagged Ensembles with Tunable Parameters", *Computational Intelligence*, vol. 35, no. 1, pp. 184–203.

[10] LIU Jun, SHANG Wenqian and LIN Weiguo (2018), “Improved Stacking Model Fusion Based on Weak Classifier and Word2vec”, *IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pp. 820–824, Singapore.

[11] SEWELL Martin (2011), *Ensemble Learning*, *http://machine-learning.martinsewell.com/ensembles/,* DoA 13.12.2022.

[12] DEMARIS Alfred (1995), “A Tutorial in Logistic Regression”, *Journal of Marriage and the Family*, vol. 54, no.4, pp. 956–968.

[13] MCKELVEY Richard D. and ZAVOINA William (1975), “A statistical model for the analysis of ordinal level dependent variables”, *Journal of Mathematical Sociology*, vol. 4, no. 1, pp. 103–120.

[14] MOHAN Senthilkumar, THIRUMALAI Chandrasegar and SRIVASTAVA Gautam (2019), “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques”, *IEEE Access*, vol. 7, pp. 81542–81554.

[15] CHEN Austin H., HUANG Shu-Yi, HONG Pei-Shan, CHENG, Chieh-Hao and LIN En-Ju (2011), “HDPS: Heart Disease Prediction System”, *Computing in Cardiology*, pp. 557–560.

[16] PALANIAPPAN Sellappan and AWANG Rafiah (2008), “Intelligent Heart Disease Prediction System Using Data Mining Techniques”, *IEEE/ACS International Conference on Computer Systems and Applications*, pp. 108–115, Qatar.

[17] PARTHIBAN Latha and SUBRAMANIAN R. (2008), “Intelligent Heart Disease Prediction System Using CANFIS and Genetic Algorithm”, *International Journal of Medical and Health Sciences* , vol. 1, no. 5, pp. 278-281.

[18] BHATLA Nidhi and JYOTI Kiran (2012), “An Analysis of Heart Disease Prediction Using Different Data Mining Techniques”, *International Journal of Engineering*, vol. 1, no. 8, pp. 1–4.

[19] REPAKA Anjan Nikhil, RAVIKANTI Sai Deepak and FRANKLIN Ramya G. (2019), “Design and Implementing Heart Disease Prediction Using Naives Bayesian”, *IEEE 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 292–297, India.

[20] BOUCKAERT Remco R. (2004), “Bayesian network classifiers in WEKA”, https://researchcommons.waikato.ac.nz/bitstream/handle/10289/85/content.pdf?sequence=1, DoA 06.12.2022.

[21] SAXENA Kanak and SHARMA Richa (2016), “Efficient Heart Disease Prediction System”, *Procedia Computer Science*, vol. 85, pp. 962–969.

[22] MITTAL Pooja (2012), "Knowledge Extraction Based on Evolutionary Learning (KEEL): Analysis of Development Method, Genetic Fuzzy System", *Int. J. Comput. Appl. Inf. Technol.*, vol. 1, pp. 22–25.

[23] SHAH Devansh, PATEL Samir and BHARTI Santosh Kumar (2020), "Heart Disease Prediction Using Machine Learning Techniques", SN Computer Science, vol. 1, no. 6, pp. 1–6.

[24] OTOOM Ahmed Fawzi, ABDALLAH Emad E., KILANI Yousef, KEFAYE Ahmed and ASHOUR Mohammad (2015), "Effective Diagnosis and Monitoring of Heart Disease", *International Journal of Software Engineering and Its Applications*, vol. 9, no. 1, pp. 143–156.

[25] DANGARE Chaitrali S., APTE Sulabha S. (2012), "Improved Study of Heart Disease Prediction System Using Data Mining Classification Techniques", *International Journal of Computer Applications*, vol. 47, no. 10, pp. 44–48.

[26] SONI Jyoti, ANSARI Ujma, SHARMA Dipesh and SONI Sunita (2011), "Predictive Data Mining for Medical Diagnosis: An Overview of Heart Disease Prediction", *International Journal of Computer Applications*, vol. 17, no. 8, pp. 43–48.

[27] KHAN Mohammad Ayoub (2020), "An IoT Framework for Heart Disease Prediction Based on MDCNN classifier", *IEEE Access*, vol. 8, pp. 34717–34727.

[28] BORISOV Vadim, LEEMANN Tobias, SESSLER Kathrin, HAUG Johannes, PAWELCZYK Martin and KASNECI Gjergji (2021), "Deep neural networks and tabular data: a Survey", *arXiv preprint*, No. arXiv:2110.01889, pp. 5-6.

[29] RAMALINGAM VV., DANDAPATH Ayantan and RAJA M. Karthik (2018), "Heart Disease Prediction Using Machine Learning Techniques: A survey", *International Journal of Engineering & Technology*, vol. 7, no. 2.8, pp. 684–687.

[30] D'AGOSTINO Ralph B., GRUNDY Scott, SULLIVAN Lisa M., WILSON Peter and GROUP CHD Risk Prediction (2001), "Validation of the Framingham Coronary Heart Disease Prediction Scores: Results of a Multiple Ethnic Groups Investigation", *Jama*, vol. 286, no. 2, pp. 180–187.

[31] RAJDHAN Apurb, AGARWAL Avi, SAI Milan, RAVI Dundigalla and GHULI Poonam (2020), "Heart Disease Prediction Using Machine Learning", *International Journal of Research and Technology*, vol. 9, no. 04, pp. 659–662.

[32] FITRIYANI Norma Latif, SYAFRUDIN Muhammad, ALFIAN Ganjar and RHEE Jongtae (2020), "HDPM: An Effective Heart Disease Prediction Model for a Clinical Decision Support System", *IEEE Access*, vol. 8, pp. 133034–133050.

[33] THOMAS J. and PRINCY R. Theresa (2016), "Human Heart Disease Prediction System Using Data Mining Techniques", *IEEE International Conference on Circuit, Power and Computing Technologies (ICCPCT),* pp. 1–5, India.

[34] SULTANA Marjia, HAIDER Afrin and UDDIN Mohammad Shorif (2016), "Analysis of Data Mining Techniques for Heart Disease Prediction", *IEEE 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, pp. 1–5, Bangladesh.

[35] TANEJA Abhishek (2013), "Heart Disease Prediction System Using Data Mining Techniques", *Oriental Journal of Computer Science and Technology*, vol. 6, no. 4, pp. 457–466.

[36] SINGH Poornima, SINGH Sanjay and PANDI-JAIN Gayatri S. (2018), "Effective Heart Disease Prediction System Using Data Mining Techniques", *International Journal of Nanomedicine*, vol. 13, T-NANO 2014 Abstracts, pp. 121-122.

[37] WANG Xiaoyue, MUEEN Abdullah, DING Hui, TRAJCEVSKI Goce, SCHEUERMANN, Peter and KEOGH Eamonn (2013), "Experimental Comparison of Representation Methods and Distance Measures for Time Series Data", *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309.

[38] PATEL Jaymin, TEJALUPADHYAY Dr., PATEL Samir (2015), "Heart Disease Prediction Using Machine Learning and Data Mining Technique", *Heart Disease*, vol. 7, no. 1, pp. 129–137.

[39] SINGH Archana and KUMAR Rakesh (2020), "Heart Disease Prediction Using Machine Learning Algorithms", *IEEE International Conference on Electrical and Electronics Engineering* (ICE3), pp. 452–457, India.

[40] AYON Safial Islam, ISLAM Md. Milon and HOSSAIN Md. Rahat (2020), "Coronary Artery Heart Disease Prediction: A Comparative Study of Computational Intelligence Techniques", *IETE Journal of Research*, vol. 68, no. 4, pp. 1–20.

[41] METHAILA Aditya, KANSAL Prince, ARYA Himanshu and KUMAR Pankaj (2014), "Early Heart Disease Prediction Using Data Mining Techniques", *Computer Science & Information Technology Journal*, vol. 24, pp. 53–59.

[42] MEDHEKAR Dhanashree S., BOTE Mayur P. and DESHMUKH Shruti D. (2013), "Heart Disease Prediction System Using Naive Bayes", *Int. J. Enhanced Res. Sci. Technol. Eng.*, vol. 2, no.3, pp. 120 – 125.

[43] SUBBALAKSHMI G., RAMESH K. and RAO M. Chinna (2011), "Decision Support in Heart Disease Prediction System Using Naive Bayes", *Indian Journal of Computer Science and Engineering (IJCSE),* vol. 2, no. 2, pp. 170–176.

[44] BASHIR Saba, KHAN Zain Sikander, KHAN Farhan Hassan, ANJUM Aitzaz and BASHIR Khurram (2019), "Improving Heart Disease Prediction Using Feature Selection Approaches", *IEEE 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST),* pp. 619–623, Pakistan.

[45] KAUR Beant and SINGH Williamjeet (2014), "Review on Heart Disease Prediction System Using Data Mining Techniques", *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 2, no. 10, pp. 3003–3008.

[46] GAVHANE Aditi, KOKKULA Gouthami, PANDYA Isha and DEVADKAR Kailas (2018), "Prediction of Heart Disease Using Machine Learning", *IEEE Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 1275–1278, India.

[47] MASETHE Hlaudi Daniel and MASETHE Mosima Anna (2014), "Prediction of Heart Disease Using Classification Algorithms", *Proceedings of the World Congress on Engineering and Computer Science*, vol. 2, No. 1, pp. 25–29, USA.

[48] BASHIR Saba, QAMAR Usman and KHAN Farhan Hassan (2016), "A Multicriteria Weighted Vote-based Classifier Ensemble for Heart Disease Prediction", *Computational Intelligence*, vol. 32, no. 4, pp. 615–645.

[49] DURAIRAJ, M. and REVATHI V. (2015), "Prediction of Heart Disease Using Back Propagation MLP Algorithm", *International Journal of Scientific & Technology Research*, vol. 4, no. 8, pp. 235–239.

[50] ASADI Shahrokh, ROSHAN SeyedEhsan and KATTAN Michael W. (2021), "Random Forest Swarm Optimization-based for Heart Diseases Diagnosis", *Journal of Biomedical Informatics*, vol. 115, pp. 103690.

[51] CABRAL George Gomes and OLIVEIRA Adriano Lorena Inácio de (2014), "One-class Classification for Heart Disease Diagnosis", *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2551–2556, USA.

[52] KHAN Younas, QAMAR Usman, YOUSAF Nazish and KHAN Aimal (2019), "Machine Learning Techniques for Heart Disease Datasets: A Survey", *Proceedings of the 2019 11th International Conference on Machine Learning and Computing*, pp. 27–35, China.

[53] LECUN Yann and RANZATO M. (2013), "Deep Learning Tutorial", *Tutorials in International Conference on Machine Learning (ICML'13)*, pp. 1–29, USA.

[54] POPOV Sergei, MOROZOV Stanislav, BABENKO Artem (2019), "Neural oblivious decision ensembles for deep learning on tabular data", *arXiv preprint,* No. arXiv:1909.06312, pp. 2-5.

[55] EL-SHAFIEY Mohamed G., HAGAG Ahmed, EL-DAHSHAN El-Sayed A. and ISMAIL Manal A. (2021), "A Hybrid Bidirectional LSTM and 1D CNN for Heart Disease Prediction", *IJCSNS*, vol. 21, no. 10, pp. 135.

[56] MOHAMMADI Mahnaz, PAWAR Rupali V. and DHABE Priyadarshan S. (2010), "Heart Diseases Detection Using Fuzzy Hyper Sphere Neural Network Classifier", *CiiT International Journal of Artificial Intelligent Systems and Machine Learning,* vol.2, no.7, pp. 102-107.

[57] HEARST Marti A., DUMAIS Susan T., OSUNA Edgar, PLATT John and SCHOLKOPF Bernhard (1998), "Support Vector Machines", *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28.

[58] SAQLAIN Syed Muhammad, SHER Muhammad, SHAH Faiz Ali, KHAN Imran, ASHRAF Muhammad Usman, AWAIS Muhammad and GHANI Anwar (2019), "Fisher Score and Matthews Correlation Coefficient-based Feature Subset Selection for Heart Disease Diagnosis Using Support Vector Machines", *Knowledge and Information Systems*, vol. 58, no. 1, pp. 139–167.

[59] SHAH Syed Muhammad Saqlain, SHAH Faiz Ali and HUSSAIN Syed Adnan (2020), "BATOOL, Safeera. "Support Vector Machines-based Heart Disease Diagnosis Using Feature Subset, Wrapping Selection and Extraction Methods", *Computers & Electrical Engineering*, vol. 84, pp. 106628.

[60] SHILASKAR Swati and GHATOL Ashok (2013), "Feature Selection for Medical Diagnosis: Evaluation for Cardiovascular Diseases", *Expert Systems with Applications*, vol. 40, no. 10, pp. 4146–4153.

[61] GERSHENSON Carlos (2003), "Artificial Neural Networks for Beginners", *arXiv preprint*, No. cs/0308031, pp. 1-8.

[62] MCCULLOCH Warren S. and PITTS Walter (1943) “A Logical Calculus of the Ideas Immanent in Nervous Activity”, *The Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133.

[63] HECHT-NIELSEN Robert (1989), “Theory of the Backpropagation Neural Network”, *International 1989 Joint Conference on Neural Networks,* pp. 593-605, USA.

[64] BOSER Bernhard E., GUYON Isabelle M. and VAPNIK Vladimir N. (1992) “A training algorithm for optimal margin classifiers”, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 144–152, USA.

[65] SCHOLKOPF Bernhard (2001), “The Kernel Trick for Distances”, *In*, *Advances in Neural Information Processing Systems 13*, Ed. LEEN Todd K., DIETTERICH Thomas G. and TRESP Volker, pp. 301–307, The MIT Press, USA.

[66] HEARST Marti A. (1998), "Support Vector Machines", *IEEE Intelligent Systems and their applications,* vol. 13, no. 4, pp. 18-28.

[67] CRISTIANINI Nello and SHAWE-TAYLOR John (2000), *An Introduction to Support Vector machines and Other Kernel-based Learning Methods*, pp. 27-32, Cambridge University Press, Cambridge.

[68] GRAUMAN Kristen and DARRELL Trevor (2005), “The pyramid match kernel: discriminative classification with sets of image features”, *Tenth IEEE International Conference on Computer Vision (ICCV’05),* vol. 2, pp. 1458–1465, China.

[69] SHAWE-TAYLOR John and CRISTIANINI, Nello et al (2004), *Kernel Methods for Pattern Analysis*, pp. 304-305, Cambridge University Press, Cambridge.

[70] KEOGH Eamonn (2022), *Naive Bayes Classifier*, https://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf, DoA 07.12.2022.

[71] WEINBERGER Kilian (2018), *Bayes Classifier and Naive Bayes*, https://www.cs.cornell.edu/courses/cs4780/2018fa/lectures/lecturenote05.html, DoA 07.12.2022.

[72] RIBEIRO Maria Isabel (2004), “Gaussian Probability Density Functions: Properties and Error Characterization”, https://welcome.isr.tecnico.ulisboa.pt/wp-content/uploads/2015/05/644_probability.pdf, DoA 13.12.2022.

[73] PATTEKARI Shadab Adam and PARVEEN Asma (2012), “Prediction System for Heart Disease Using Naive Bayes”, *International Journal of Advanced Computer and Mathematical Sciences*, vol. 3, no. 3, pp. 290–294.

[74] SHINDE Rucha, ARJUN Sandhya, PATIL Priyanka and WAGHMARE Jaishree (2015), “An Intelligent Heart Disease Prediction System Using K-means Clustering and Naive Bayes Algorithm”, *International Journal of Computer Science and Information Technologies*. 2015, vol. 6, no. 1, pp. 637–639.

[75] GANDOMI Amir H., YANG X-S, TALATAHARI Siamak and ALAVI Amir Hossein (2013), “Firefly Algorithm with Chaos”, *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 89–98.

[76] PAWLAK Zdzislaw, GRZYMALA-BUSSE Jerzy, SLOWINSKI Roman and ZIARKO Wojciech (1995), “Rough Sets”, *Communications of the ACM*, vol. 38, no. 11, pp. 88–95.

[77] REDDY G. Thippa and KHARE Neelu (2017), “An Efficient System for Heart Disease Prediction Using Hybrid OFBAT with Rule-based Fuzzy Logic Model”, *Journal of Circuits, Systems and Computers*, vol. 26, no. 04, pp. 1750061.

[78] KIM Jaekwon, LEE Jongsik and LEE Youngho (2015), “Data-mining-based Coronary Heart Disease Risk Prediction Model Using Fuzzy Logic and Decision Tree”, *Healthcare Informatics Research*, vol. 21, no. 3, pp. 167–174.

[79] QUINLAN J. Ross (1986), “Induction of Decision Trees”, *Machine Learning*, vol. 1, no. 1, pp. 81–106.

[80] ROKACH Lior and MAIMON Oded (2005), “Decision trees”, *In*, *Data Mining and Knowledge Discovery Handbook*, Ed. MAIMON Oded and ROKACH Lior, pp. 165–192, Springer, Boston.

[81] SANTHANAM T. and EPHZIBAH EP. (2015), “Heart Disease Prediction Using Hybrid Genetic Fuzzy Model”, *Indian Journal of Science and Technology*, vol. 8, no. 9, pp. 797.

[82] SAHA Sriparna and EKBA Asif (2013), “Combining Multiple Classifiers Using Vote Based Classifier Ensemble Technique for Named Entity Recognition”, *Data & Knowledge Engineering*, vol. 85, pp. 15–39.

[83] NASEEM Imran, TOGNERI Roberto and BENNAMOUN Mohammed (2010), “Linear Regression for Face Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112.

[84] THARWAT Alaa (2016), “Linear vs. Quadratic Discriminant Analysis Classifier: A Tutorial*”, International Journal of Applied Pattern Recognition*, vol. 3, no. 2, pp. 145–180.

[85] AHA David W., KIBLER Dennis and ALBERT Marc K. (1991), “Instance-based Learning Algorithms”, *Machine Learning*, vol. 6, no. 1, pp. 37–66.

[86] TU My Chau, SHIN Dongil and SHIN Dongkyoo (2009), “Effective diagnosis of heart disease through bagging approach”, *IEEE 2nd International Conference on Biomedical Engineering and Informatics*, pp. 1–4, China.

[87] YUAN Xiaoming, WANG Xue, HAN Jianchao, LIU Jiemin, CHEN Haiyan, ZHANG Kuan and YE Qiang (2019), “A High Accuracy Integrated Bagging-Fuzzy-GBDT Prediction Algorithm for Heart Disease Diagnosis”, *IEEE/CIC International Conference on Communications (ICCC)*, pp. 467–471, China.

[88] FRIEDMAN Jerome H (2001), “Greedy Function Approximation: A Gradient Boosting Machine”, *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232.

[89] YEKKALA Indu, DIXIT Sunanda and JABBAR MA (2017), “Prediction of Heart Disease Using Ensemble Learning and Particle Swarm Optimization”, *IEEE International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, pp. 691–698, India.

[90] KENNEDY James and EBERHART Russell (1995), “Particle Swarm Optimization”, *IEEE Proceedings of International Conference on Neural Networks (ICNN’95)*, vol. 4, pp. 1942–1948, Australia.

[91] SHOREWALA Vardhan (2021), “Early detection of Coronary Heart Disease Using Ensemble Techniques”, *Informatics in Medicine Unlocked*, vol. 26, pp. 100655.

[92] RANAWANA Romesh and PALADE Vasile (2006), “Optimized Precision-A New Measure for Classifier Performance Evaluation”, *IEEE International Conference on Evolutionary Computation*, pp. 2254–2261, Canada.

[93] ALCOVER Pedro Marıa, GUILLAMÓN Antonio and RUIZ Marıa del Carmen (2013), “A New Randomness Test for Bit Sequences”, *Informatica*, vol. 24, no. 3, pp. 339–356.

[94] PARZEN Emanuel (1963), “On Spectral Analysis with Missing Observations and Amplitude Modulation”, *Sankhya: The Indian Journal of Statistics*, vol. 25, no.4, pp. 383–392.

[95] DISATNIK David J. and BENNINGA Simon (2007), “Shrinking the Covariance Matrix”, *The Journal of Portfolio Management*, vol. 33, no. 4, pp. 55–63.

[96] KARADENIZ Talha and MARAS Hakan Hadi (2018), “Covariance Features for Trajectory Analysis”, *Elektronika ir Elektrotechnika*, vol. 24, no. 3, pp. 78–81.

[97] LEDOIT Olivier and WOLF Michael (2004), “A Well-conditioned Estimator for Large-dimensional Covariance Matrices”, *Journal of Multivariate Analysis*, vol. 88, no. 2, pp. 365–411.

[98] FRIEDMAN Jerome, HASTIE Trevor and TIBSHIRANI Robert (2008), “Sparse Inverse Covariance Estimation with the Graphical Lasso”, *Biostatistics*, vol. 9, no. 3, pp. 432–441.

[99] MCLACHLAN Goeffrey J. (1999), “Mahalanobis Distance”, *Resonance*, vol. 4, no. 6, pp. 20–26.

[100] MASSEY JR Frank J. (1951), “The Kolmogorov-Smirnov Test for Goodness of Fit”, Journal of the American Statistical Association, vol. 46, no. 253, pp. 68–78.

[101] KIM Tae-Hwan and WHITE Halbert (2004), “On More Robust Estimation of Skewness and Kurtosis”, *Finance Research Letters*, vol. 1, no. 1, pp. 56–73.

[102] JOANES,Derrick N. and GILL Christine A. (1998), “Comparing Measures of Sample Skewness and Kurtosis”, *Journal of the Royal Statistical Society: Series D (The Statistician)*, vol. 47, no. 1, pp. 183–189.

[103] KRISHNA Hare and PUNDIR Pramendra Singh (2007), “Discrete Maxwell Distribution”, https://www.researchgate.net/profile/Hare-Krishna/publication/228874278_Discrete_Maxwell_Distribution/links/0f31753ab15dee7c69000000/Discrete-Maxwell-Distribution.pdf, DoA 14.12.2022.

[104] CHEN Tianqi, HE Tong, BENESTY Michael, KHOTILOVICH Vadim, TANG Yuan, CHO Hyunsu and CHEN Kailong (2015), “XGBoost: Extreme Gradient Boosting”, *R package version 0.4-2*, vol. 1, no. 4, pp. 1–4.

[105] REDDY G. Thippa, REDDY M., LAKSHMANNA Kuruva, RAJPUT Dharmendra Singh, KALURI Rajesh and SRIVASTAVA Gautam (2020), “Hybrid Genetic Algorithm and A Fuzzy Logic Classifier for Heart Disease Diagnosis”, *Evolutionary Intelligence,* vol. 13, no. 2, pp. 185–196.

[106] GOLDBERGER Jacob, HINTON Geoffrey E., ROWEIS Sam and SALAKHUTDINOV Russ R. (2004), “Neighbourhood Components Analysis”, *In, Advances in Neural Information Processing Systems*, Ed. SAUL L., WEISS Y. and BOTTOU L., pp. 513-520, MIT Press, USA.

[107] VERBIEST Nele (2014), *Fuzzy Rough and Evolutionary Approaches to Instance Selection* (PhD thesis), Ghent University, Ghent.

[108] BOGNER K., PAPPENBERGER F. and CLOKE HL. (2012), “The Normal Quantile Transformation and Its Application in a Flood Forecasting System”, *Hydrology and Earth System Sciences*, vol. 16, no. 4, pp. 1085–1094.

[109] PIRES Ivan Miguel, HUSSAIN Faisal, M GARCIA Nuno, LAMESKI Petre and ZDRAVEVSKI Eftim (2020), “Homogeneous Data Normalization and Deep Learning: A case Study in Human Activity Classification”, *Future Internet*, vol. 12, no. 11, p. 194.

[110] BROWN Gavin, POCOCK Adam, ZHAO Ming-Jie and LUJÁN Mikel (2012), “Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection”, *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 27–66.

[111] LIN Dahua and TANG Xiaoou (2006), “Conditional Infomax Learning: An Integrated Framework for Feature Extraction and Fusion”, *European Conference on Computer Vision*, pp. 68–82, Germany.

[112] BISONG Ekaba (2019), “More Supervised Machine Learning Techniques with scikit-learn”, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 287–308, Apress, Berkeley.