Comparison of machine learning methods for the classification of cardiovascular disease

Hagan, R., Gillan, C. J., & Mallet, F. (2021). Comparison of machine learning methods for the classification of cardiovascular disease. Informatics in Medicine Unlocked, 24, [100606]. https://doi.org/10.1016/j.imu.2021.100606

Published in:
Informatics in Medicine Unlocked

Document Version:
Publisher's PDF, also known as Version of record

Queen's University Belfast - Research Portal:
Link to publication record in Queen's University Belfast Research Portal

Publisher rights
Copyright 2021 the authors. This is an open access article published under a Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution and reproduction in any medium, provided the author and source are cited.

General rights
Copyright for the publications made accessible via the Queen's University Belfast Research Portal is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The Research Portal is Queen's institutional repository that provides access to Queen's research output. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact openaccess@qub.ac.uk.

Download date: 15. Jul. 2021
Comparison of machine learning methods for the classification of cardiovascular disease

Rachael Hagan *, Charles J. Gillan, Fiona Mallett

School of Electrical and Electronic Engineering and Computer Science, Queen’s University Belfast, Queen’s Road, Queen’s Island, Belfast, Northern Ireland, BT9 3DT, United Kingdom

ARTICLE INFO

Keywords:
- Supervised learning
- Classification
- Heart disease
- SVM
- Decision trees
- Neural network
- Ensemble learning
- Uncertainty quantification

ABSTRACT

Background: Researchers are devoting significant effort to use machine learning algorithms, a subset of the wider field of artificial intelligence, to detect disease in a single patient. There exists extensive research in the application of machine learning methods in health care, and more specifically, cardiovascular disease. We have chosen to focus this initial investigation on the case of cardiac disease in order to focus our efforts on as much detail of the methods as possible.

Methods: In this paper we explore the uncertainty that exists across applying machine learning methods, namely: Support Vector Machines (SVM), Multi-Layer Perceptron Neural Networks (MLP) and ensemble methods, for the classification of cardiovascular disease. Our work uses two public datasets with significantly different characteristics in order to assess the potential differences in the uncertainty of the methods. The cardiac arrhythmia dataset from the University California Irvine (UCI) Machine Learning repository has almost three hundred specific physiological data points per patient gathered from analysis of electrocardiogram (ECG) signals on several hundred patients although the distribution of cases is severely imbalanced. Contrast this with one dataset, reporting on cardiovascular disease from the Kaggle collection where there are nearly seventy thousand patient records. However, this Kaggle dataset reports only a small number of parameters per patient record, values such as serum cholesterol level, diastolic and systolic blood pressure, relative blood glucose levels and presence or absence of angina.

Results: Models built for the UCI dataset have an order of magnitude more dimensions or alternatively have much larger numbers of input nodes for neural network models compared to the models developed the Kaggle dataset. On the other hand, the Kaggle dataset has an order of magnitude more records for training and validation than the UCI dataset. Our results compare and contrast the uncertainty in models built using support vector machine, multilayer perceptron neural networks and decision trees for these two datasets. The work suggests that it will be instructive to extend our analysis to datasets of other patho-physiologies.

1. Introduction

There is a general trend toward digital health care, a feature that is being driven in part by the Internet of Things (IoT) and enhanced sensing technologies both coupled to the use of artificial intelligence and machine learning methods. Market analysts [1] and medical professionals [2] recognize that this offers cost efficiency and opens new frontiers for improved care in the health system. Recent work [5] from Brookhaven National Laboratory in the US suggests that accurate prediction of future incidence of Alzheimer’s disease is possible using machine learning techniques with patient electronic health care records (EHRs). The use of computers to automate the detection of cardiac arrhythmia from ECG signals began several decades ago [3], initially by embracing expert systems [4], but continues today as an active field of research using machine learning and artificial intelligence.

Machine learning (ML) is a set of mathematical algorithms, a subset of the wider field of artificial intelligence (AI) algorithms, which offer the potential to provide innovative decision support solutions to many problems in a big data environment therefore offering advances beyond the rule based engines that proliferate in many fields of science. Explainable AI is therefore a hot topic and significant in all heavily regulated industries. The US DARPA has invested significantly in research in this field [6] while other research teams have begun to define the concept of explainable AI with respect to several problem domains [7]. Current research identifies three broad classifications: opaque systems that offer no insight; interpretable systems where mathematical analysis of the algorithm is viable;

* Corresponding author.
E-mail address: rhagan09@qub.ac.uk (R. Hagan).

https://doi.org/10.1016/j.imu.2021.100606
Received 28 January 2021; Received in revised form 11 May 2021; Accepted 16 May 2021
Available online 20 May 2021
2352-9148/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).
and comprehensible systems that emit symbols enabling user-driven explanations of how a conclusion is reached. Our work seeks to use extensive numerical simulation to understand the mathematical kernels that we use to classify the presence or absence of cardiac disease.

A central focus of our work is to try to quantify the uncertainty in our classifiers. Epistemic uncertainty arises from the experimental error in measuring the data points presented in each of our datasets. Aleatoric uncertainty arises in our mathematical kernels, specifically in the use of pseudo-random numbers. Our aim in this work is to quantify uncertainties in our output classifications, using several distinct models, propagated from variability in the parameters used to train the mathematical kernels. We do this by performing multiple numerical experiments including grid searches of the space of hyperparameters for the models, where appropriate. Further, for decision tree methods we explore the entropy of prediction probabilities of each classification to determine the certainty of the models.

Our paper is composed of several sections which have the following structure. We discuss related work by other authors in Section 2 and Section 3 explains the different classification methods that we have investigated in our work flow. Section 4 briefly defines the public data sets that lie at the core of this work. In Section 5, we present and analyze the results that we obtained when executing our software and we finish the paper by summarizing conclusions drawn from our present work in Section 6.

2. Related work

The Kaggle dataset, discussed further in Section 4, has been used in a number of research papers by other offers. Maiga et al. [15] compared the predictive ability of random forest (RF), naïve Bayes, k-nearest neighbour (KNN) and logistic regressions classifiers trained on this dataset, reporting that the random forest method achieves classification accuracy of 73%, specificity of 65% and sensitivity of 80%. However other authors report different success rates applying similar techniques to related datasets in the Kaggle collection. Chauhan [16] applied logistic regression, KNN, support vector machine (SVM), decision trees and RF methods to the Framingham study dataset. This is an on-going cardiovascular study, started in 1948, based in Framingham, Massachusetts. The study aims to predict whether the patient has a 10-year risk of future heart diseases or not. The dataset in Ref. [16] consists of 4238 records of patient data each with 14 independent attributes. Several of these attributes are in common with the attributes in Table 3.

Table 1
Details of sixteen types of classification of cardiac arrhythmia defined and used in the UCI dataset [10] which is discussed in the text.

| Integer code | Description                                           | Instances in data set | Remapped category |
|--------------|-------------------------------------------------------|-----------------------|-------------------|
| 01           | Normal heart                                          | 245                   | Normal            |
| 02           | Ischemic changes (Coronary Artery Disease)            | 44                    | CAD               |
| 03           | Old Anterior Myocardial Infarction                    | 15                    | Basket            |
| 04           | Old Inferior Myocardial Infarction                    | 15                    | Basket            |
| 05           | Sinus tachycardy                                      | 13                    | Basket            |
| 06           | Sinus bradycardy                                      | 25                    | Basket            |
| 07           | Ventricular Premature                                 | 3                     | Basket            |
| 08           | Contraction (PVC)                                     | 2                     | Basket            |
| 09           | Left bundle branch block                              | 9                     | Basket            |
| 10           | Right bundle branch block                             | 50                    | Right BB          |
| 11           | 1st degree Atrioventricular block                     | 0                     | Basket            |
| 12           | 2nd degree AV block                                   | 0                     | Basket            |
| 13           | 3rd degree AV block                                   | 0                     | Basket            |
| 14           | Left ventricular hypertrophy                          | 4                     | Basket            |
| 15           | Atrial Fibrillation or Flutter                         | 5                     | Basket            |
| 16           | Others                                                | 22                    | Others            |

The author found that the logistic regression method gave the best predictive accuracy at 89%.

Kajan et al. [32] studied the suitability of ANNs for medical diagnosis using datasets available from the UCI repository for breast cancer and Parkinson’s disease in addition to the dataset on cardiac arrhythmia that we have used in this work. Their paper reported successful diagnoses across these disease states arguing that ANNs are suitable in cases where traditional classification methods fail due to noisy or incomplete data. Our work, which has been performed in Python rather than with MATLAB, confirms the results of these authors and compares and contrasts it with the use of SVM technique. Aliferis et al. [33] compared the operation of several classifiers on the UCI arrhythmia dataset. These authors used K-Nearest neighbours, feed-forward neural networks, decision trees and a naïve Bayes classifier, which are comparable against the use of ANNs and SVMs in this project scope.

McGregor [34] outlines the potential of big data within neonatal intensive care units (NICU) to assist in the early detection and prevention of a wide range of health conditions. The paper argues that real-time analysis of high-frequency data could be beneficial to healthcare providers and patients. McGregor and her team developed Artemis and Artemis Cloud, which takes bedside physiological measurements (ECG, blood pressure, respiration rate, chest impedance and blood oxygen saturation) from hospitalised female patients and infants. Artemis also compares existing medical observations and treatments with analytical solutions to determine new patterns within real-time physiological data. This is beneficial because it could anticipate the occurrence of various health conditions thus increasing the possibilities for earlier and more targeted medical interventions. Given that this work by

The authors who contributed the UCI arrhythmia dataset developed a supervised and inductive machine learning algorithm which they called voting feature intervals (VFI) [10]. Testing this with their dataset gave accuracy levels of up to 62%, which the authors claim exceeded the use of a naïve Bayes classifier and a nearest neighbour classifier on the same data. Our use of machine learning models trained on the UCI dataset has produced higher accuracy predictions with an ensemble voting feature when applied to the independent MIMIC-III dataset.

Rahman et al. [38] designed and utilised a machine learning
ECG-based heartbeat classifier for early detection of the cardiovascular disease, hypertrophic cardiomyopathy (HCM). This disease causes the heart muscle, ventricular septum, to thicken which potentially fatally obstructs blood flow. Data is accumulated from HCM patients using standard 10-s, 12 lead ECG signals. Heartbeats from non-HCM cardiovascular patients are the controls. Classification performance is considered by testing a random forest classifier and a support vector machine classifier using 5-fold cross-validation. The results are compared against a logistic regression classifier, which performed worse. When implementing the SVM in this project, experiments for optimizing k-fold cross-validation were also conducted.

This work compares a range of machine learning methodologies for the classification of heart disease across two datasets with different characteristics. We perform hyperparameter searches to analyze the optimal parameters for each of the models. Although extensive research has been carried out applying different machine learning methods for optimizing k-fold cross-validation were also conducted.

Classification performance is considered by testing a random forest classifier and a support vector machine classifier using 5-fold cross-validation. The results are compared against a logistic regression classifier, which performed worse. When implementing the SVM in this project, experiments for optimizing k-fold cross-validation were also conducted.

We report in the next section. First we present the salient details for each machine learning kernels that we have used to produce the results that we discuss further in section 2.

3. Methodology

This section of the paper presents the mathematical details of the machine learning kernels that we have used to produce the results that we report in the next section. First we present the salient details for each of the classification algorithms used focusing in particular on the definition of the tunable hyper-parameters which are available for each. The selection of values for these parameters ultimately distinguishes our use of these kernels from that of others, activities which are discussed further in section 2.

3.1. Classification algorithms

Each set of features in one record represents one observation. We define this to be a vector \( x \in \mathbb{R}^d \), such that \( x = (x_1, x_2, x_3, \ldots, x_d) \) is composed of real numbers, \( x_i \). Overall we have \( n \) complete observations in any data set, \( X \), such that which we express in matrix notation as

\[
X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}
\]  (1)

Each observation, \( x_i \), is associated to one and only value, a class label, and where we have a set of \( Y = [y_1, y_2, \ldots, y_d] \) class labels covering all of the data. \( Y \) is defined by Table 1. We use this information to train the ANN and SVM models in our work.

A classification model is a mathematical function, \( f \) that takes an input vector, \( x \) and maps it to one value in the set of classes, \( \mathcal{Y} \), which we express mathematically as

\[
f(x) \rightarrow [0, \ldots, M]
\]  (2)

In this paper we investigate different choices for the classifier function, \( f \). The following subsections explain each of the choices.

3.2. Support vector machine

Following equations (1) and (2), in a support vector machine model [24] approach to build the function \( f \) in equation (2), we take each observation vector, \( x_i \), and augment it with its classification value, \( y_i \) associated with it, thereby forming a new vector \( v_i = (x_i, y_i) \). The SVM approach first transforms the set of observation vectors, \( X \), into new space \( \psi \) using the transform \( \psi \).

\[
\psi : X \rightarrow \psi
\]  (3)

Consider first of all the case of a classification with only two values in the set, \( \mathcal{Y} \). Then in the training phase, the SVM approach uses the method of Lagrange multipliers to solve the following constrained minimization problem

\[
\min_{w, b} \frac{1}{2}w^T w + C \sum_l \zeta_l \quad \tag{4}
\]

subject to \( y_l (w^T \psi(x_l) + b) \geq 1 - \zeta_l, \zeta_l \geq 0 \) \tag{5}

in the transformed space, meaning that it finds a hyper-plane separating the data. In equations (4) and (5) \( \zeta_l \) is an error term for each component and \( C \), where \( C > 0 \), is a constant which defines the magnitude of the sum of error terms. The value \( C \) is predefined by the user when launching the training phase and therefore represents a variability point distinguishing different SVM models. The elements of the weight vector, \( w \), are computed and refined during each iteration.

An interesting aspect of the SVM approach is that the transformation function \( \psi \) does not need to be explicitly calculated. Instead, a kernel function \( K(x_i, x_j) \), which defines the inner product in the space \( \psi \)

\[
K(x_i, x_j) = \phi(x_i)^T \phi(x_j)
\]  (7)

is required for the computation. Any function satisfying Mercer’s theorem [25] can be used for the kernel and in this work we have investigated three commonly used kernel functions:

- Linear: \( K(x_i, x_j) = x_i^T x_j \).
- RBF: \( K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \).
- Polynomial: \( K(x_i, x_j) = (y_i x_j + r)^\gamma, \gamma > 0 \).

The above equations show that having chosen one kernel form over another there are further variability points within the choice. For example in the case of the polynomial kernel, \( \gamma, r \) and \( d \) can be set for each model. Once these choices are fixed, the training phase finds the separating hyper-plane.

In the inference phase, new observations are mapped using the same kernel into the space and their distance is computed from the separating hyper-plane obtained from the training phase. The cluster which lies closest to the transformed observation can be assigned to it.

In our work we implemented our SVM modelling code using the well-known libSVM library [28] to perform our numerical experiments. The library provides the option to modify the various parameters discussed in the equations above. Most important is the fact that libSVM goes beyond the basic binary classification presented briefly above by allowing classification into one of several classes. This operation is performed in a pair-wise fashion, following the equations given above, a technique known as one-against-one classification [29].

3.3. Artificial neural networks

The artificial neural networks (ANN) in this work are non-linear mathematical functions, \( f \), mapping an input vector of observations to a unique value in a set of possible outputs. Each ANN is represented as a set of interconnected nodes. The nodes, which are arranged in layers, are numeric values and the connections are multiply accumulate operations (MAC) which are executed in sequence, corresponding to linking the nodes together. Each MAC operation has several weights which are computed during the learning phase. The result of the MAC operation at each node is input to an activation function which decides whether the result is fed forward to the next stage, or not. Each layer of the network can use a different activation function. In line with common practice by other authors, we use the rectified linear unit (relu) in all but the final layer. For binary classification, that is with the Kaggle cardiovascular dataset, we use the sigmoid function at the final layer whereas for multi-class classification used with the UCI arrhythmia dataset, we use the softmax function [26].

The weights for the network are computed by mini-
mizing a cost function for the network. In the case of binary classification, we use the binary cross entropy loss function, $J$, where

$$J = -y \log(y) + (1 - y) \log(1 - y)$$  

(8)

In equation (8) $y$ are the ground truth labels (taking the binary values 0 or 1) and $\hat{y} \in [0, 1]$ is the predicted value. The minimization process requires a gradient descent and backpropagation though the network to fix the weights. A similar process if followed for the case of multi class classification but using a categorical entropy loss function.

3.4. Ensembles of decision trees

Following Friedman [18], we may state the problem mathematically in the following way. We have an output variable $Y_i$, dependent in general on a vector of $n$ input variables $x = x_1, x_2, x_3, ..., x_n$ through some function, $F(x)$, the form of which is unknown. However, we have a set of $m$ observations, each of which associates one input vector $x_i$ with one output,

$$F(x) \rightarrow Y_i, \forall i \in \{1, 2, 3, ..., m\}$$  

(9)

This set of observations forms our training set from which we seek to find an approximation, $F^*(x)$ to the true function $F(x)$. The criterion for deriving $F^*(x)$ is that we minimize the loss function $L(y, F(x_i))$ over the set of $m$ observations defined in equation (10),

$$F^*(x) = \arg\min_{F(x)} L(y, F(x_i))$$  

(10)

Different techniques have been developed in the literature of machine learning to solve equation (10). Traditional approaches to the problem introduce the approximation that the functional form of $F^*(x)$ is fixed but dependent on a number of coefficients, any set of which we can express as the vector $\theta$ so that the aim is to find $F^*(x, \theta)$ satisfying equation (10).

A further approximation is that $F^*(x)$ can be expanded in a basis set of simpler functions, $h_k(x, \theta_k)$:

$$F^*(x, \theta) = \sum_{k=1}^{F(x)} \beta_k h_k(x, \theta_k)$$  

(11)

where $\beta_k$ are coefficients which are determined during the solution process. Iterative numerical methods are used to solve equation (10), leading to sets of values for the vectors $\theta_k$ and the coefficients $\beta_k$. Neural networks are one such traditional approach, where the solution produces the weights associated with the multiply accumulate operations on each neuron.

An alternative representation for the functions $h_k(x, \theta_k)$ in equation (11) is to use decision trees rather, for example, polynomial forms. This means that equation (11) is solved by searching in a function space. In that approximation, the $\theta_k$ are no longer coefficients but are parameters defining the properties of the individual trees. Specifically these parameters include the number of nodes and the depth of the tree used.

Equation (11) is an ensemble of trees and these trees can be as simple or as complex as required. We can further distinguish several approximations in this case depending on the way that we build the summation (the ensemble) over $h_k()$ defined in equation (11). In the following paragraphs we explain the four different classification tree approximations that we have used.

The principal types of approach which we have used are bagging and boosting. In the bootstrap aggregation approach, known as bagging, we create several subsets of the feature vectors and then use each subset to train a classification tree, that is one $h_k()$ in the summation of equation (11). The subsets are selected randomly and using replacement. The Bagging algorithm and the ExtraTrees algorithm are two similar implementations of the approach.

The random forest algorithm extends the concept of bagging because in addition to using random subsets of data, it adds random selection of features. Thus in our work, each random forest trees does not use all features extracted from the time series at time point $t$ but only a subset of them. Furthermore random forest trees are built to have as many layers as possible.

The alternative boosting approach builds the summation in equation (11) sequentially. Simple trees are used at first and these remain fixed for the rest of the analysis. At each step a new classification tree is added, that is another term in 11. As terms are added the goal is to reduce the error. The GradientBoosting version of the algorithm allows for the optimization of arbitrary differentiable loss functions where in each stage a tree is fitted based on the negative gradient of the loss function. In this sense it combines boosting with gradient descent.

The decision trees created by each algorithm arrive at a decision, known as the leaf node, which is a class label prediction. Each sample in the data set is passed through the series of trees which utilizes the features and parameters set to make a decision on what class the algorithm thinks the sample belongs in. We can calculate the accuracy from here taking the amount of correct predictions against all predictions made.

While each of the decision trees tries to minimize the loss function at each step there still exists a level of error within the models and thus uncertainty in the predictions between different methods. To avoid over fitting we look at setting the max depth i.e. the number of layers in each of the trees, meaning that the leaf node is not pure and therefore the probability of this classification being correct is not 100%. We take the probabilities of each classification and calculate the entropy over each class in order to quantify the uncertainty in each of the ensemble methods.

3.4.1. Uncertainty quantification for ensemble methods

Methods for supervised learning, therefore including the methods used in this paper, can only be expected to handle data that is similar to that on which the model has been trained. Epistemic uncertainty, or model uncertainty, arises when the trained system is asked to make classifications using data that lies outside the subset of data on which it has been trained. This may happen either when data is gathered from a different situation than used for the preparation of the training set, or alternatively the training set lacked data covering certain domains of the input variables. Put simply, trained models can only deal with the things they have seen before and the training data generally only covers a very small subset of the entire input space. This means that models produce arbitrary output values for the vast majority of possible input values. Clearly, this is not an issue for operational deployment as long as we know that future inputs will be within the subspace used for training. If this is not the case, however, we would like to quantify the lack of accuracy due to missing training data.

Noise in the training data is a different kind of uncertainty, known as aleatory uncertainty, and arises in two ways. The homoscedastic case occurs when the noise in the training data follows the same distribution regardless of the actual input values while the heteroscedastic case occurs when the noise is a function of the actual values of the training data. In this paper we consider only the above epistemic uncertainty deferring the treatment of aleatory uncertainty to future work.

We have used the method of k-fold cross-validation, also known as out-of-sample testing, as a baseline to test the epistemic uncertainty of our models. The approach has the additional advantage that it mitigates against over-fitting in the model. Following other authors we adopt the value $k = 10$. This means that the original dataset is randomly partitioned into 10 equal sized sub-samples. Of the 10 sub-samples, a single sub-sample is used as the validation data for testing the model, and the remaining 9 sub-samples are used as the training set. The whole process is repeated 10 times. Each sub-sample is used once and only once as the validation set. The whole process creates 10 results from which we can compute a means ans standard deviation.

This approach is however only a baseline in the analysis of uncertainty for our models. This because some of the data in the validation sample may still lie inside the boundaries of the subset that was used for
training [27]. The literature reports an increasing number of more complex methods for epistemic uncertainty quantification but we defer investigation of these to future work.

4. Public datasets used in this work

In our work we have used to two openly available datasets with significantly different content. We describe these in following subsections. Both datasets have been used by several other authors in their research and we summarise that work in section 2 of this paper. We also explain here the significant pre-processing of both datasets in order to render them suitable for use with the machine learning kernels which we describe in Section 3.

4.1. An arrhythmia dataset from the UCI collection

We first look at the arrhythmia dataset [10] from the repository of machine learning databases provided by the University of California Irvine [11]. Each record corresponds to one multi-lead ECG recording from one patient and most of the features in the dataset are the measured time intervals for the various segments of the PQRS'T wave forms in each ECG. This dataset has 452 records with each record containing almost 300 features. Each record has been classified into one of sixteen categories where each category is represented by an integer. The value 1 indicates that the record is classified as a normal ECG, while values [2,…,15] correspond to different types of arrhythmia as shown in Table 1.

As the table shows, one of the weaknesses of this dataset is that there are relatively few records, and even none for the various pathophyslogenies associated with AV block, meaning that the UCI dataset exhibits severe imbalance [14].

As an initial step to ameliorate the situation, we reduced the number of categories to five. Our mapping is shown on the right hand column of Table 1. We retained the normal, coronary artery disease(CAD), right branch bundle block (BB) and others categories from the original dataset and then merged all other categories into one, which we designate as Basket. We removed all columns where every value was the same and with the exception of gender, all other columns which reported binary values. Finally, we removed the features of type object. This leaves 196 features per record.

We then applied the random oversampling technique [12] to reduce the bias between categories in order to produce a dataset that we can use effectively with machine learning methods.

Random oversampling can increase the likelihood of over-fitting because, for high over sampling rates, it may add substantially more instances of the less common classes that were present in the original distribution (i.e. Table 2 in our case). We found that creating a sampled dataset with 245 records for each of the five categories in Table 2 was sufficient to avoid the over-fitting problem.

4.2. The Kaggle cardiovascular disease dataset

The Kaggle cardiovascular disease dataset [13] presents a binary classification problem recording whether cardiovascular disease is present or not. This dataset has 70,000 records but only 11 features per record. Furthermore these features can be subdivided into three subsets:

Objective: factual information such as gender,
Examination: values measured by clinical examination such as weight,
Subjective: patient self reported data, for example whether a smoker or not

For the sake of clarity, we point out that there are in fact a number of datasets related to heart disease in the Kaggle repository and that we have made a specific choice of one data set with many thousands of readings. Table 3 describes the fields in each record, indicating the type of each.

The three categories for the cholesterol and blood glucose values were replaced by numbers in range [0, 1] and one hot encoding applied to the binary values, including the classification itself. All remaining numerical variables were scaled into the range [0, 1].

5. Results

In the first part of this section we discuss the results obtained with the SVM kernel and then move on to report on the results obtained from the multi-layer perceptron neural network models. After that, we report on the application of the tree based ensemble methods. At the end of the section we discuss on our analysis of uncertainty in the ensemble models.

5.1. The SVM model

We performed multiple numerical experiments with different SVM kernels searching over a grid of hyper parameters to find the values that gave optimum accuracy. Our work identified that changing the C parameter in equation (4) gave the greatest variation in accuracy for each individual SVM model. Tables 4 and 5 display results of using 10-fold cross-validation on the four kernels whilst changing the C and γ parameters.

Table 4 shows that a C value of 0.005 produces low accuracy for all SVM kernels and gamma settings with the UCI dataset. We can achieve the highest accuracy of 0.92 using the RBF kernel and scale gamma function.

Table 5 shows the results of repeating the experiment with the Kaggle dataset. We notice that the highest accuracy obtained is 0.72, significantly lower than with the UCI dataset.

In the case of the sigmoid kernel, the accuracy was relatively independent of the C parameter in our experiments. With values ranging from 0.51 to 0.57 however it is also the least optimal machine learning

| Gamma | C-value | Linear Mean Accuracy | RBF Mean Accuracy | Polynomial Mean Accuracy | Sigmoid Mean Accuracy |
|-------|---------|----------------------|------------------|-------------------------|----------------------|
| scale | 0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |
| 5.000 | 0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |
| 50.000| 0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |
| auto  | 0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |
| 5.000 | 0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |
| 50.000| 0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |
| 500.000|0.005   | 0.26                 | 0.21             | 0.24                    | 0.21                 |

| Gamma | C-value | Linear Mean Accuracy | RBF Mean Accuracy | Polynomial Mean Accuracy | Sigmoid Mean Accuracy |
|-------|---------|----------------------|------------------|-------------------------|----------------------|
| scale | 0.005   | 0.63                 | 0.61             | 0.64                    | 0.54                 |
| 5.000 | 0.005   | 0.66                 | 0.64             | 0.65                    | 0.54                 |
| 50.000| 0.005   | 0.70                 | 0.66             | 0.70                    | 0.54                 |
| auto  | 0.005   | 0.63                 | 0.59             | 0.50                    | 0.57                 |
| 5.000 | 0.005   | 0.66                 | 0.65             | 0.64                    | 0.57                 |
| 50.000| 0.005   | 0.70                 | 0.65             | 0.64                    | 0.56                 |
| 500.000|0.005   | 0.72                 | 0.69             | 0.65                    | 0.57                 |
model in our experiments with the UCI dataset. In contrast, the linear and polynomial kernels showed similarities in accuracy, achieving the optimum 0.72 with \(C = 500.000\) and the RBF kernel achieving a similar accuracy of 0.71. Further, we notice that C values of 0.005 perform comparably with the Kaggle data set across the four SVM kernels and higher C values.

5.2. The MLP models

5.2.1. The Kaggle cardiovascular dataset

We trained a multilayer perceptron on the Kaggle cardiovascular dataset described in section 4 carrying out a hyper parameter search with tenfold cross validation. Back propagation and gradient descent are used to adjust the weights and predict a gradient descent. Calculations for each set of hyperparameters for one k-fold took on average 50 s per epoch, leading to run times for the search across the space of hyper parameters search in the region of one hundred hours elapsed time. Each record of the Kaggle dataset was preprocessed to be suitable for use in the Tensorflow software resulting in a total of fifteen parameters per input record. The categorical data columns - gender, smoke, alcohol and active - were one hot encoded while the columns for glucose and cholesterol were converted to floating point numbers in the range \([0.3, 1.0]\). All remaining columns were scaled to lie on the range \([0, 1]\). The output column, denoting whether the record corresponded to a patient with a diagnosis of cardiovascular disease, or not, was also one hot encoded as required by Tensorflow.

Fig. 1 shows the structure of the optimal model found in our search. In addition to the input and output layers, there are three hidden layers.

Two are dense layers, with 50 and 25 nodes respectively and in the middle there is a dropout layer to avoid over-fitting.

Fig. 2 compares the accuracy and loss values obtained when training the above network. These graphs are the best overall results in our cross validation experiments up to 50 epochs. We found that using extra epochs resulted in only a small improvement in the accuracy and loss functions of the model.

5.2.2. The UCI arrythmia dataset

In a similar way to the previous section, we trained an MLP on the UCI arrhythmia dataset and carried out a search of the hyper parameter space. In general, we found that using a batch size of 32 and training for 400 epochs with the Nadam optimizer function gave the best accuracy in the ten fold cross validation tests. The processed UCI dataset has significantly more columns than the Kaggle dataset used in the previous section, specifically 196 as opposed to 15.

We monitored the best accuracy of the model as a function of the number of columns used in the model that is the number of nodes in the input layer of the neural network. Fig. 3 shows the result obtained. We can see from shape of the graph in the figure that while using 8 features results in an accuracy of 0.22, increasing to 50 features produces an accuracy of 0.74. Clearly, when including more than fifty columns of the dataset, the accuracy is essentially constant. This implies that columns beyond number fifty in the dataset carry redundant information for the neural network. The first fifty columns are more descriptive of the true patient state; with columns 1 to 4 describing demographic features age, gender, height and weight, and columns 5 to 27 reporting the ECG PQRST existence, interval durations, angles, lengths and heart rate. Any columns after this are repeats of features taken from different ECG measurement channels.

5.3. Ensemble decision trees

We then fit a series of ensemble classification techniques to the training data, and performed a hyperparameter search for each of the methods. For each record in the test set we predict the classification using the four different ensemble methods. Tables 6 and 7 highlights these results, reporting a range of accuracy from \([0.91, 0.96]\) across the 4 models for the UCI dataset and a lower range of \([0.72, 0.74]\) with the Kaggle dataset.

We proceed with the optimum parameters for each ensemble method and for each record we take a majority vote across the four methods resulting in an overall accuracy of 0.96 and 0.74 with the UCI and Kaggle datasets respectively.

5.4. Uncertainty quantification of ensemble models

Shannon Entropy is widely used in the literature as a metric for uncertainty quantification. It is used intrinsically in decision tree methods, where the overall goal of each splitting step is to minimize the sum of the entropy over all newly generated branches. For each ensemble model we obtain the probability that each sample is predicted into each class. From here we apply entropy to the probabilities returned by the model in each class, quantify the variance in performance when predicting across the classes. The more uncertain the model, the higher the entropy. Table 8, Figs. 4 and 5 compares the uncertainty across each of the ensemble methods and the mean probability of predictions in each class.

GradientBoosting reports the highest entropy values ranging from \([3.03, 4.00]\) equating to the greatest level of uncertainty in this model predicting across all classes, while having an accuracy of 0.94 as reported in Table 6. Similarly, Bagging has an accuracy of 0.91, however the uncertainty of this model is slightly lower, with a range of \([3.02, 3.82]\), which would therefore lead to Bagging being interpreted as the more certain classifier between the two. Across the different classes we notice that class 4, the remainder arrhythmia, has the lowest mean

Fig. 1. Illustration of the optimal multi-layer perceptron model found in a search of hyper-parameters with the Kaggle cardiovascular data set discussed in the text.
We further compare these results with applying the same methods, using the Kaggle cardiovascular disease dataset described earlier in this paper. We show our results in Table 9. We notice that the Kaggle dataset results in both lower accuracy and higher uncertainties leading us to conclude that greater classification is achieved when the models are trained with a higher number of features, rather than a greater number of records.

In contrast to decision trees, neural networks seek to minimize the probability of classification and further high entropy in all models leading us to believe that this would be harder to detect.

We further compare these results with applying the same methods, using the Kaggle cardiovascular disease dataset described earlier in this paper. We show our results in Table 9. We notice that the Kaggle dataset results in both lower accuracy’s and higher uncertainties leading us to conclude that greater classification is achieved when the models are trained with a higher number of features, rather than a greater number of records.

In contrast to decision trees, neural networks seek to minimize the

---

**Table 6**

| Ensemble Model    | Accuracy | Hyperparameter criteria                  |
|-------------------|----------|------------------------------------------|
| RandomForest      | 0.95     | gini criterion, 100 estimators and max depth 20 |
| ExtraTrees        | 0.96     | gini criterion, 100 estimators and max depth 20 |
| GradientBoosting  | 0.94     | learning rate 0.1, using deviance loss and 60 estimators |
| Bagging           | 0.91     | 5 max features, 100 max samples, 100 estimators bootstrap false |
Cross entropy while traditional SVM models are instead based on variational methods and aim to find an optimal separating hyperplane to enable classification. We note in our work that the classification accuracy of our SVM models is similar to that of the decision tree methods. Newer types of SVM kernels which take account of entropy are available, such as the entropic divergence kernel which is based on Kullback-Liebler divergence [31]. We have not used these types of SVM kernels in our work reserving that to future publications.

6. Discussion and conclusion

In this paper we investigated the application of three machine learning methods to predict cardiovascular disease, searching the space of hyper parameters for each method. The methods used were SVM, multi-layer perceptron neural networks and decision trees. We highlight the importance of applying ten fold cross validation to compare the uncertainty of the trained models generated from two datasets with different characteristics, e.g. Table 4 showing a range of accuracy from 0.21 to 0.92 depending on parameters set for the SVM model. One dataset, from the UCI repository, consists of timing data extracted from the PQRST features in ECG waveforms. This has a relatively large number of features per record but a small number (approximately 400) of records and is also substantially imbalanced between the classification categories. The second dataset is from the Kaggle repository and has a small number of features per record but nearly seventy thousand records. This dataset records features such as the presence or absence of hypertension, cholesterol levels etc. Both datasets record age and gender among their features.

The optimal parameters found for the SVM method applied to the UCI dataset produced an accuracy of 92% but the optimal set of parameters found for the Kaggle dataset reach only 72% accuracy. We found that a three layer MLP model trained on the UCI dataset achieved an accuracy of 74%, significantly lower than the optimal SVM model for this dataset. In the case of the Kaggle dataset, the optimal MLP reached 71% accuracy similar to the performance of the optimal SVM model for this dataset. Application of ensembles of decision trees, produced similar results to the optimal SVM model for each dataset. In the case of the UCI dataset, all ensemble methods achieved over 90% accuracy, we observed 96% accuracy with the Extra Trees model.

Our work reflected the well known characteristic that significant amounts of CPU time are needed to train MLP models. We observed that SVM and decision tree models required, on average, an order of magnitude less CPU time. Within the models that we used, decision trees have proved to deliver the best accuracy for the least training cost.

Other authors have applied a similar methodology to ours for deployment of portable ECG equipment [42] in the community and in relation to other medical challenges. Kajan et al. [43] considered the use of neural networks for diagnosis of several medical conditions. An encephalogram (EEG) measures the electrical activity of the brain. Relative to the ECG, less is known about the wave forms associated with various disease states. Traditionally, neurologists employ direct visual inspection of the wave forms to recognize abnormal functionality. Acharya et al. [45] building on their previous work use of neural networks for diagnosis of ECG features [44] have recently applied a convolutional deep neural network to the diagnosis of seizures from EEG wave forms. Using a 13-layer network, the authors report an average accuracy of 88.7%. The methodology in this study could be extended to embrace a study of various disease states and we plan to report on this in future papers.

---

**Table 7**

| Ensemble Model   | Accuracy | Hyperparameter criteria |
|------------------|----------|-------------------------|
| RandomForest     | 0.74     | Trees built with gini criterion, 100 estimators and max depth set to 10. |
| ExtraTrees       | 0.72     | Trees built with entropy criterion, 60 estimators and max depth set to 60. |
| GradientBoosting | 0.74     | Trees built using a learning rate of 0.1, using deviance loss and 80 estimators |
| Bagging          | 0.73     | Trees built with 10 max features, 100 max samples, 90 estimators and with bootstrap set to false |

---

**Fig. 4.** Calculated probability for each class in the analysis of the UCI dataset.
Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

RH acknowledges funding for a PhD studentship from the Department for the Economy Northern Ireland. The authors are grateful to the UCI Repository of Machine Learning Databases [11] and the owners of the Kaggle site where we obtained the data sets used in this work. Computations were performed at the EPSRC funded Tier 2 Northern Ireland HPC centre.

Table 8
Uncertainty Quantification across all classes for each ensemble method with UCI dataset. Prob. being the mean classification probability for each predicted class and Entropy being the Shannon Entropy.

| Class | RandomForest | ExtraTrees | GradientBoosting | Bagging |
|-------|--------------|------------|------------------|---------|
|       | Prob.  | Entropy  | Prob.  | Entropy  | Prob.  | Entropy  | Prob.  | Entropy  |
| 1     | 0.60  | 3.90     | 0.63  | 3.97     | 0.77  | 4.00     | 0.32  | 3.82     |
| 2     | 0.93  | 2.22     | 0.99  | 1.18     | 0.95  | 3.58     | 0.54  | 3.63     |
| 3     | 0.92  | 2.57     | 0.99  | 1.09     | 0.94  | 3.63     | 0.48  | 3.73     |
| 4     | 0.83  | 3.42     | 0.97  | 1.54     | 0.83  | 3.85     | 0.40  | 3.65     |
| 5     | 1.00  | 1.14     | 1.00  | 1.03     | 0.92  | 3.03     | 0.63  | 3.02     |

Table 9
Uncertainty Quantification across the two classes for each ensemble method using the Kaggle dataset. Prob. being the mean classification probability for each predicted class and Entropy being the Shannon Entropy.

| Class | RandomForest | ExtraTrees | GradientBoosting | Bagging |
|-------|--------------|------------|------------------|---------|
|       | Prob.  | Entropy  | Prob.  | Entropy  | Prob.  | Entropy  | Prob.  | Entropy  |
| 0     | 0.70  | 9.30     | 0.65  | 9.33     | 0.72  | 8.37     | 0.68  | 6.37     |
| 1     | 0.74  | 9.13     | 0.67  | 9.14     | 0.75  | 8.90     | 0.69  | 4.86     |

References

[1] Frost A, White Paper Sullivan. Finding Top-line Opportunities in a Bottom-line Healthcare Market: revenue Generation Opportunities Created by an Innovative Contact-free Continuous Monitoring Solution. 2016. available on-line at, https://www.earlysense.com/medical-services/wp-content/uploads/2013/08/Frost-Sullivan-White-Paper-FINAL.pdf.
[2] E Topal, Deep medicine: how artificial intelligence can make healthcare human again (first ed.), Pub: Basic Books, ISBN-13: 978-1541644632.
[3] Feldman CL. Computer detection of cardiac arrhythmia: historical review. Am. Rev. Diagn. 1983:2:138–45.
[4] Sharma T, Sharma KK. QRS complex detection in ECG signals using locally adaptive weighted total variation denoising. Comput Biol Med 2017;87:187–99.
[5] Ji Hwan Park, Han Eol Cho, Jong Hun Kim, Melanie Wall, Yaakov Stern, Hyunsoo Lim, Shinjae Yoo, Hyesung-Soop Kim and Jiook Cha, Electronic health records based prediction of future incidence of alzheimers disease using machine learning. 10.1101/625582.
[6] DARPA Project Explainable Artificial Intelligence (XAI). Project web site: https://www.darpa.mil/program/explainable-artificial-intelligence.
[7] Doran Derek, Schulz Sarah, Besold Tarek R. What does explainable AI really mean? A new conceptualization of perspectives. In: Proceedings of the first International Workshop on comprehensibility and explanation in AI and ML; 2017. https://openaccess.city.ac.uk/id/eprint/18660.

[10] Guvenir HA, Acar B, Demiroz G, Gekin A. Supervised machine learning algorithm for arrhythmia analysis. In: Proceedings of the 24th annual meeting on computers in cardiology; September 1997. p. 433-6. Lund Sweden.

[11] UCI machine learning repository; Irvine California, USA. University of California, School of Information and Computer Science, available on the web at: http://archive.ics.uci.edu/ml.

[12] Branco Paula, Torgo Luís, Ribeiro Rita P. A survey of predictive modeling on imbalanced domains. ACM Comput Surv 2016;49(2):1-50. https://doi.org/10.1145/2907070.

[13] Uljanova, The cardiovascular disease dataset. available on the web at: https://www.kaggle.com/sulianova/cardiovascular-disease-dataset.

[22] Bell Jason. Machine Learning: hands-on for developers and technical professionals. Indianapolis): Wiley; 2015, ISBN 978-1-118-88906-0.

[25] J. Mercer, Functions of positive and negative type and their connection with the theory of integral equations, Philosophical Transactions of the Royal Society A, 209, 415-446.

[26] Sharma Siddharth, Sharma Simone, Athaiya Anidhya. Activation functions in neural networks. Int. J. Eng. Appl. Sci. Technol. 2020;4(12):310-6. 2455-2143.