Internet of Things 17 (2022) 100485

Contents lists available at ScienceDirect

Internet of Things

journal homepage: www.elsevier.com/locate/iot

Research article

HealthCloud: A system for monitoring health status of heart patients using machine learning and cloud computing

Forum Desai [a], Deepraj Chowdhury [b], Rupinder Kaur [c], Marloes Peeters [d], Rajesh Chand Arya [e], Gurpreet Singh Wander [e], Sukhpal Singh Gill [a,*], Rajkumar Buyya [f]

[a] School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom
[b] Department of Electronics & Communication Engineering, International Institute of Information Technology (IIIT), Naya Raipur, India
[c] School of Science, Dulwich College, United Kingdom
[d] School of Engineering, Newcastle University, United Kingdom
[e] Department of Cardiology, Hero DMC Heart Institute, Dayanand Medical College and Hospital, Ludhiana, Punjab, India
[f] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, School of Computing and Information Systems, The University of Melbourne, Australia

ARTICLE INFO

Keywords:
Machine learning
Smart healthcare
Heart disease prediction
Cloud computing

ABSTRACT

In the context of the global health crisis of 2020, the tendency of many people to self-diagnose at home virtually, prior to any physical interaction with medical professionals, has been increased. Existing self-diagnosis systems include those accessible via the Internet, which involve entering one's symptoms. Several other methods do exist, for example, people read medical blogs or notes, which are often wrongly interpreted by them and they arrive at a completely different assumption regarding the cause of their symptoms. In this paper, a system called HealthCloud is proposed, for monitoring health status of heart patients using machine learning and cloud computing. This study aims to offer the 'best of both worlds', by combining the information required for the person to understand the disease in sufficient detail, with an accurate prediction as to whether they may have (in this case) heart disease or not. The presence of heart disease is predicted using machine learning algorithms such as Support Vector Machine, K-Nearest Neighbours, Neural Networks, Logistic Regression and Gradient Boosting Trees. This paper evaluates these machine learning algorithms to obtain the most accurate model, in compliance with Quality of Service (QoS) parameters. The performance of these machine learning models is measured and compared using the metrics such as Accuracy, Sensitivity (Recall), Specificity, AUC scores, Execution Time, Latency, and Memory Usage. For better establishment of the results, these machine learning algorithms have been cross validated with 5-fold cross validation technique. With an accuracy rate of 85.96%, it has been found that Logistic Regression is the most responsive and accurate model amongst those models assessed. The Precision, Recall, Cross Validation mean and AUC Score for this model were 95.83%, 76.67%, 81.68% and 96% respectively. The algorithm and the mobile application were tested on Google Cloud Firebase with existing user inputs from the dataset, as well as with unseen new data. The use of this system can assist patients, both in reaching self-diagnosis decisions and in monitoring their health.

* Correspondence to: School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, United Kingdom.
E-mail addresses: f.m.desai@se20.qmul.ac.uk (F. Desai), deepraj19101@iiitnr.edu.in (D. Chowdhury), rupinderchem@gmail.com (R. Kaur), marloes.peeters@newcastle.ac.uk (M. Peeters), drrajesharya@yahoo.com (R.C. Arya), drgswander@yahoo.com (G.S. Wander), s.s.gill@qmul.ac.uk (S.S. Gill), rbuyya@unimelb.edu.au (R. Buyya).

https://doi.org/10.1016/j.iot.2021.100485
Received 24 October 2021; Received in revised form 25 November 2021; Accepted 25 November 2021
Available online 11 December 2021
2542-6605/

1. Introduction

The heart muscle continuously sends blood around the body through a group of blood vessels, which together make up the circulatory system [1]. This blood delivers oxygen and nutrients required for organs and muscles to function, and it also removes unwanted carbon dioxide and waste materials [2]. Nonetheless, abnormalities in this flow of blood, caused by high blood pressure, high cholesterol or poor dietary decisions, can trigger several circulatory diseases, known as cardiovascular diseases [3]. Cardiovascular diseases (CVD) are a prominent cause of death worldwide, and over 80% of this mortality rate occurs in developing countries [4]. The World Health Organisation emphasises the importance of detecting CVD's symptoms as early as possible, to be followed by the rapid diagnosis and the quick delivery of treatment [5]. Such symptoms include chest pain and an irregular heart rate [6]. This study mainly focuses on coronary heart disease which is the most common CVD. In coronary heart diseases, the blood vessels get narrow and eventually get blocked, minimising the supply of oxygen-rich blood to the heart muscle.

With the advancement of Artificial Intelligence (AI), the impact of AI in healthcare has become more effective [7]. Diagnosis of the symptoms using AI, has became easier for the user. So the detection of the disease can be done in its early stage [5]. Machine learning based models detect the disease by finding the patterns in data of the previous patients [8].

Moreover, they can help healthcare professionals to understand the common causes of the disease, and they may aid the patient into changing their habits for disease prevention [5]. The large of amount data related to heart patients is generated and stored on public databases [8]. So, there is a need to develop an innovative solution by leveraging cloud computing and machine learning together for the accurate prediction of heart diseases [7]. The earlier prediction can help patients to change their habits for disease prevention and can alert patients if there will be any serious concern [1].

In this paper, machine learning algorithms are explored and assessed based on several classification performance metrics, in order to find ways so that detection of heart disease symptoms can be done early. The most accurate and precise machine learning algorithm can be regarded as the best algorithm, and that can be used for the system to provide the best user experience. It prevent both false hope and unnecessary stress to the patient or user's health [6]. Quality of Service (QoS) is a method to measure the performance of the system in terms of various parameters such as network performance and to check the durability of a system [9]. To provide scalability and make the proposed system suitable for the low performance capable edge device, the application was connected to a cloud-based platform and majority of the execution occurred on that platform, lowering work load of the edge device [10]. Performance would be impacted if there were an increase in user numbers, or in the quantity of data being generated by the system. Machine learning models were tested, and one model was chosen from the tested models based on its latency and execution Time. It has been done to find the most optimal algorithm that was least impacted by the volume of requests. Memory Usage measures the computational power required by the machine learning algorithm to predict one result. The model consuming the least memory was chosen, since the usage impacts (a) the user's device if the application is used locally offline, and (b) the cloud storage if the application's data is stored in a cloud database. If the service that an application should provide is not delivered, or not delivered as expected, the user experience for the application is adversely affected. So, to ensure the user receives the best possible service, developers should focus on user experience by measuring QoS parameters [11]. In this paper, an accurate and responsive predictive system called HealthCloud is proposed which leverages various machine learning models to acquire reliable results. This system aims to provide an accurate and helpful alternative to those currently available, which requires the user to access a mobile application for on-demand results offline. Further, various machine learning are applied, discussed, demonstrated and contrasted in the paper. The main contributions of this work are:

- Proposing a system called HealthCloud for monitoring health status of heart patients using machine learning and cloud computing.
- Finding the most accurate and responsive machine learning algorithm to predict heart disease.
- Building a prototype iOS application to provide the user with a visual display of their data and results.
- Implementing the iOS Mobile Application on Google Cloud Firebase for real-time data analysis.
- Highlighting the promising future directions.

The rest of the paper is structured as follows: Section 2 discusses the related work of diagnostic health applications and quality of service metrics. Section 3 discusses the proposed system. Section 4 discusses the methodology including data source, data exploration and machine learning models. Section 5 presents the performance evaluation. Section 6 concludes the paper and highlights the promising future directions.

2. Related work

This section presents the related work of diagnostic health applications and QoS parameters.

2.1. Diagnostic health applications

In [12], authors analysed two UCI Cleveland Heart Disease datasets and proposed a Cloud-based Android application capable of predicting heart disease. They evaluated Naïve Bayes, Support Vector Machines (SVM), Random Forest, Artificial Neural Network (ANN) and Simple Logistic to conclude SVM to be the preferred model with 97.53% accuracy. The models were analysed by their precision, recall, F1-score, sensitivity and specificity. Waikato Environment for Knowledge Analysis (WEKA), an open-source Java platform, was used for the machine learning experiments and data analysis. A continuous health monitoring system was suggested using an Arduino micro-controller and a mobile application connected to a doctor's and a patient's API to enable a connection between the user and their doctor.

In [13], the authors has designed a diagnostic model for predicting Chronic Kidney Disease (CKD) with an application of Internet of Things (IoT). This was a model to diagnose early CKD symptoms, in the hope to assist physicians and medical teams in developing countries. They managed to achieve 97% accuracy, 99% sensitivity and 95% specificity throughout three different datasets using Decision Trees, SVM, Neural Networks and Naïve Bayes, with Decision Trees producing the better results. So, while, a high accuracy was met, there was a low execution time as data collection and analysis were proposed on a cloud-based system. They prove that data mining and learning approaches are vital to predict disease outcomes, although providing the model with excessive data on the cloud results in slowing down the network and increasing execution time. This involves over-demanding the hardware with heavy computation. Theoretically, this would require dimensionality reduction algorithms, such as Principal Component Analysis (PCA) to extract relevant data. No IoT device was physically implemented to complement their proposed model so it is difficult to determine how the model would perform in real-time when load is applied to the system.

Machine learning can be applied to other diagnostic applications, other than for heart disease as demonstrated by the authors [14]. They used several machine learning methods for diabetes prediction; another chronic disease, affecting many people around the world and it should be detected as soon as possible. The Pima Indian Diabetes Database was trained using the following machine learning models: Decision Trees, Logistic Regression, Discriminant Analysis, SVM and K-Nearest Neighbour (KNN). After analysing these models based on their cross-validation scores and their accuracy, the best performing method was found to be Logistic Regression with an accuracy rate of 77.7%, whilst the worst model seemed to be a Gaussian SVM. Ensemble learning techniques were also analysed, all generating a reasonable score of around 70%. Building on their work, this study aims to improve the accuracy score of Logistic Regression, based on the evaluation of cross-validation and other informative performance metrics.

2.2. Quality of service

In [15], the authors has performed a study of the computation times taken on several Python libraries by the following machine learning algorithms: Support Vector Classifier (SVC), least-angle regression method (LARS), Elastic Net, KNN, PCA and k-Means, on the Madelon dataset. They considered and compared the times taken for training each model. They concluded that Scikit-learn was the fastest library to use on 4 out of 6 algorithms. Their results also showed the faster models seemed to be PCA, Elastic Net and KNN, while the slowest models were k-Means and SVC. In [16], the authors proposed a HealthFog system, using edge and fog technology to provide accurate predictions related to patients suffering from heart diseases and to fulfil accuracy, latency, network bandwidth and time Quality of Service parameters. Edge computing bought the system closer to the user, minimising latency and energy consumption, while fog computing took the heart patient's data from a programmed Raspberry Pi to respond back with an accurate prediction. They ended up using an ensemble Machine Learning model of a deep neural network, achieving an average testing accuracy of 87% using a range of one to five worker Edge nodes and a latency of 29.8 ms. However, the accuracy and latency of this model could still be improved, which is particularly important in medical applications.

2.3. Critical analysis

Table 1 compares the HealthCloud with existing systems. The bodies of work analysed confirmed the decision to use classification models, such as SVC, KNN and Neural Networks to predict whether a person has heart disease or not. They introduce the extent of development that can be done when AI is applied to some data. Quality of Service (QoS) parameters have briefly been mentioned and evaluated in similar applications to heart disease prediction by machine learning techniques. So, proposed system, HealthCloud aims to improve upon the current technologies by allowing for fast responses to be provided by the algorithm and to the user using the mobile application.

3. HealthCloud system

In this section, we discuss application design of HealthCloud system including architecture and legal issues.

3.1. User interface

A prototype iOS mobile application was developed using the CoreML package, provided by Apple. The user interface, as shown in Fig. 1, allows the user to input their values in for each feature. Many of the values that the user must enter, may be unfamiliar to them. So, summarised descriptions of blood tests, fluoroscopy and ECG are given to provide the user with some background knowledge of the test they must undergo. Before they receive their result, they need to manually input values into the text fields provided. If more information is required by the user regarding any of the tests, they can be directed to the website of the British Heart Foundation (2021) (https://www.bhf.org.uk/informationsupport/how-a-healthy-heart-works). by pressing the 'For More Information' link on the screen. If any value is missing in the text fields, the application reminds the user to enter that value via an alert display. Once the user finishes entering all their data, the outcome is shown as a pop-up alert, predicting whether the user may or may not have heart disease with some advice given accordingly.

Table 1
Comparison of HealthCloud with existing systems[a].

Works	ML algorithms	Ensemble learning	Computing	Heart disease	ML metrics	QoS parameters
Nashif et al. [12]	NB, SVM, ANN, RF, ANN, LR	No	Cloud/IoT	Yes	Ar, P, R, F, Sn , Sf	NA
Hosseinzadeh et al. [13]	DT, SVM, MLP, NB	No	Cloud	No	Ar, Sf, R	NA
Pedregosa et al. [15]	SVC, K-means, PCA, LARS, Elastic Net, KNN	No	No	No	No	Computation time
Tuli et al. [16]	Ensemble of Neural Network: MLP	Yes	Fog/ Edge/ IoT	Yes	Ar	Power consumption, Network bandwidth, Latency, Jitter, Execution time
Al-Zebari et al. [14]	DT, LR, Discriminant Analysis, SVM, KNN	Yes	No	No	Ar	NA
HealthCloud (this work)	SVC, KNN, NN, LR, GBT	Yes	Cloud	Yes	Ar, P, R Sp, Sn, CV, ROC Curves (AUC)	Execution time, Latency, Memory usage.

[a]***Abbreviations used in*** Table 1 ***are*** - ML: Machine Learning, LR: Logistic Regression, NB: Naive Bayes, DT: Decision Tree, RF: Random Forest, Ar: Accuracy, F: F-score, P: Precision, R: Recall, Sn: Sensitivity, Sf: Specificity, GBT: Gradient Boosting Tree, CV: Cross validation, SVC: Support Vector Classifier, NN: Neural Network, ANN: Artificial Neural Network and MLP: Multilayer Perceptron.

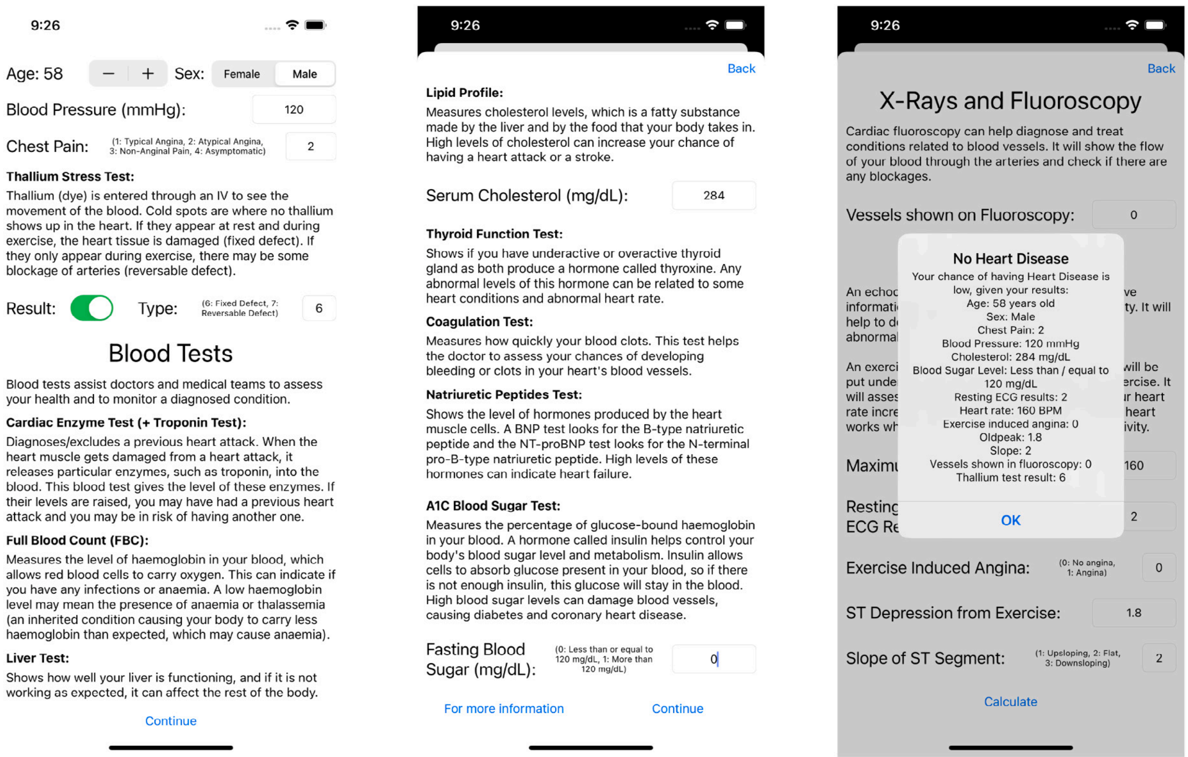

Fig. 1. User interface of HealthCloud to predict the likelihood of having heart disease.

1. **Architecture:** The Model-View-Controller (MVC) architectural pattern was used to develop the mobile application's user interface. The View comprises of code, both to address user interaction and to provide the user interface. It alerts the controller of events that the user initiates. The controller updates and/or requests the model for particular data, depending on the event triggered. The model accommodates both the data in question and the logic of dealing with them. Specifically, the Heart Calculator machine learning model is responsible for calculating the target variable. Data are processed, and the formatted data are sent back to the controller, which then modifies the user interface to produce a different output on the screen. The benefit of the MVC design pattern is that components are separated so code can be modified, without the chance of generating bugs and errors in the architecture.

Table 2
Sample of the pre-processed data from the UCI machine learning heart disease repository.

Sr. No.	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldspeak	slope	ca	thal	target
1	63	1	1	145	233	1	2	150	0	2.3	3	0	6	0
2	67	1	4	160	286	0	2	108	1	1.5	2	3	3	1
3	67	1	4	120	229	0	2	129	1	2.6	2	2	7	1
4	37	1	3	130	250	0	0	187	0	3.5	3	0	3	0
5	41	0	2	130	204	0	2	172	0	1.4	1	0	3	0
6	56	1	2	120	236	0	0	178	0	0.8	1	0	3	0
7	62	0	4	140	268	0	2	160	0	3.6	3	2	3	1
8	57	0	4	120	354	0	0	163	1	0.6	1	0	3	0
9	63	1	4	130	254	0	2	147	0	1.4	2	1	7	1
10	53	1	4	140	203	1	2	155	1	3.1	3	0	7	1

2. **Legal Issues:** According to the General Data Protection Regulation (GDPR), the personal data of a person, that is any information that can identify him or her, must be protected and not collected without his or her knowledge [17]. These data can include age and gender, which are two features that have been included in this dataset and application. However, none of the data entered are being stored in a database elsewhere. This means that all information is erased as soon as the application provides the user's results on screen and they press "OK", assuring the user of confidentiality and security of their data. A privacy policy document would be vital for such an application to make sure users are aware of the application's intentions of handling the data collected. So before any data is entered, the user should read through this document and decide whether they wish to accept it or not.

4. Methodology

Machine Learning approaches were applied and evaluated to determine whether an individual was experiencing heart disease. This was done by using a commonly used heart dataset: the Cleveland Heart Disease dataset from UCI [18].

4.1. Data source

The Cleveland Heart Disease dataset was obtained from the UCI Machine Learning Repository's Heart Disease directory. In [18], the authors have carried out data collection at the Cleveland Clinic Foundation. Since then, this dataset has been available online and intensively studied by many Machine Learning researchers. During the deployment of these data, patients' medical numbers, names and other sensitive information were treated with strict confidentiality. Meanwhile, when classifying patients, it is vital to use machine learning methods that incorporate a combination of features, as there is no one specific measure for identifying a patient's cardiac health.

The shape of this dataset is 303 rows × 14 columns, 13 of which are numerical input attributes. There is also the 'target' output (num), referring to the presence or absence of heart disease in a patient. Within this target variable column, each patient is labelled as either having heart disease (with 1) or as not having heart disease, as shown by a value of 0. As well as the target variable, the dataset contains the following features for each patient: (1) age: Age (in years); (2) sex: Sex (0: Female, 1: Male); (3) cp: Chest pain level (1: Typical angina, 2: Atypical angina, 3: Non-anginal pain, 4: Asymptomatic); (4) trestbps: Resting blood pressure (in mmHg); (5) chol: Serum cholesterol (in mg/dL); (6) fbs: If fasting blood sugar level >120 mg/dL (0: False, 1: True); (7) restecg: Resting ECG results (0: Normal, 1: ST-T wave abnormalities, 2: Left ventricular hypertrophy by Estes' criteria); (8) thalach: Maximum heart rate; (9) exang: Exercise induced angina (0: No, 1: Yes); (10) oldpeak: ST depression from exercise compared to rest; (11) slope: Slope of exercise induced ST segment (1: Upsloping, 2: Flat, 3: Downsloping); (12) ca: Number of major vessels coloured by fluoroscopy (ranges from 0 to 3); (13) thal: Thallium stress test result (3: Normal, 6: Fixed defect, 7: Reversable defect). A sample of this dataset is shown in Table 2. Each of the variables listed above is regarded as an indicator of risk in the subject of heart disease.

Data from this dataset were pre-processed and stored in a more convenient format for data exploration and analysis. Jupyter Notebook was used for these operations.

4.2. Data exploration

The pre-processed data, consisting of 284 instances, were split into 80% training and 20% test data sets, whereby the training data were used to train each algorithm, while the test dataset was used to examine how well the model performs on unseen data. Cleansing and preparation of the data involved managing outliers, and rectifying missing or incorrect values. The values within the 'ca', 'thal' and 'target' columns were readjusted to correspond to the documentation of the dataset.

To gain some insight into the influence of the 13 attributes and to view any trends, the dataset was explored in detail. Since all the features contained numerical values, it was simple enough to plot a histogram distribution of each numerical feature, whereby the x-axis displayed the different values for that feature and the y-axis displayed the number of instances containing each value. Generally, the plots, as shown in Fig. 3, imply that those people with heart disease tend to be males in their 60 s, who experience high blood pressure and angina whilst exercising, etc. The correlation matrix in Fig. 2, indicates that no feature truly has a strong

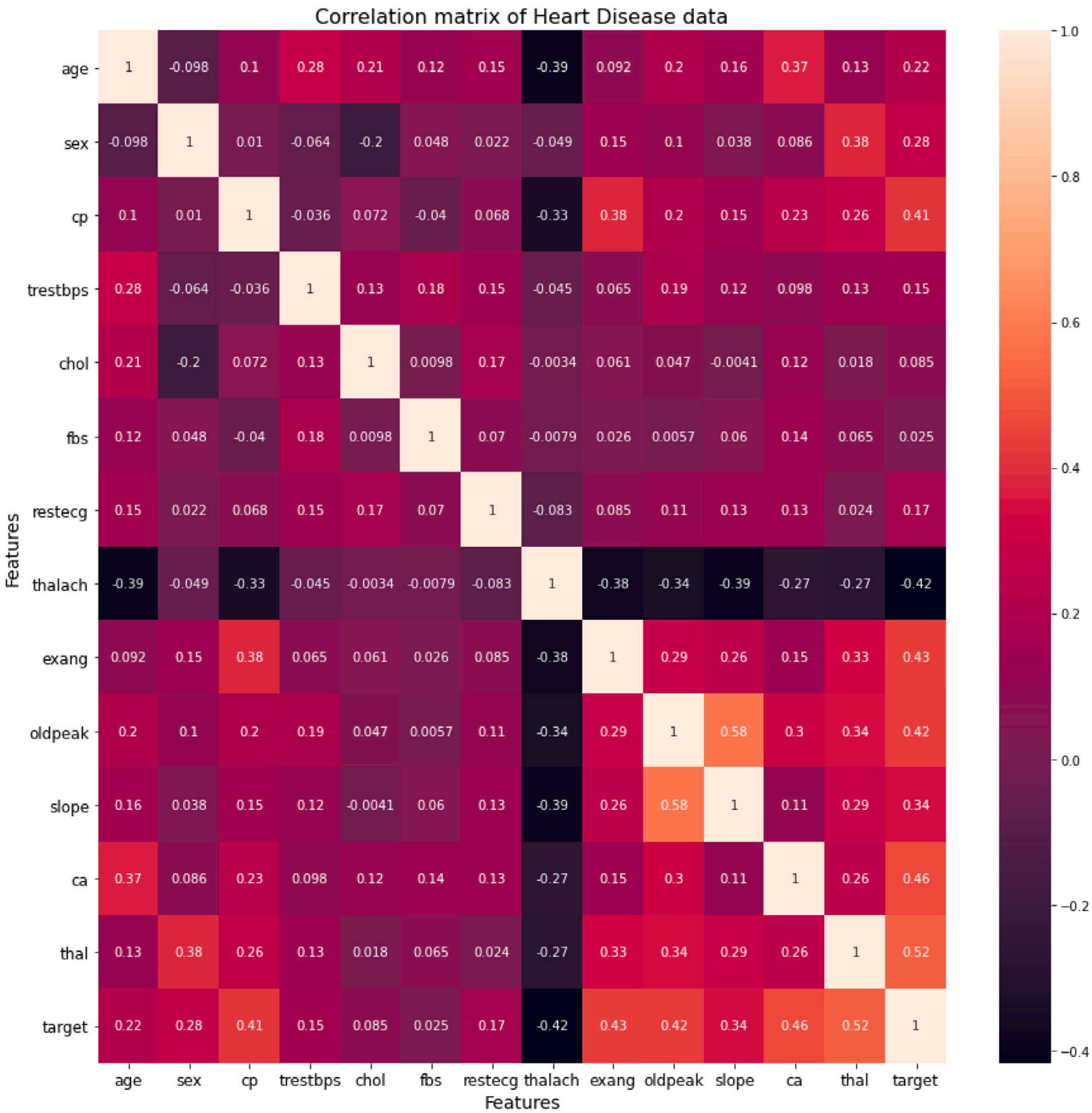

Fig. 2. Correlation matrix of heart disease data.

correlation with the 'target' variable. Hence, when the models are trained, all of these independent features should be included. The correlation between 'thalach' and 'target' is −0.42, suggesting they are weakly and negatively correlated. A higher heart rate is associated with 'no heart disease' since the target is 0. The most positively correlated feature is 'thal', with a value of 0.52, suggesting a defect found via a thallium stress test is related to heart disease, since heart conditions would cause a shortage of blood travelling to the heart. This correlation matrix confirms the common trends associated with heart disease as revealed by the histograms in Fig. 3.

4.3. Machine learning algorithms

The process of selecting the machine learning algorithms to be used was crucial. Many of the algorithms were chosen based on the success of existing work, while others were chosen as a potentially better alternative to algorithms used in current studies. The following five models were used to predict heart disease:

- **Support Vector Classifier (SVC)**: Sklearn's Support Vector Machines (SVM) are sets of supervised learning methods, useful for classification problems. One implementation of SVM is the Support Vector Classifier (SVC), capable of classifying binary problems with a linear kernel. It accepts two arrays and it returns a hyperplane that maximises the margin between the 0 and 1 classes. So, the wider the margin, the clearer the distinction between the two classes. It has proven to be highly effective with unseen and unbalanced data, as well as its use being particularly popular with studies with a healthcare or related focus.
- **K-Nearest Neighbours (KNN)**: The KNN classifier determines the target outcome by identifying closest neighbours. The n-neighbours (K) parameter requires a value that represents the number of neighbours to be used when predicting the target. By default, the Minkowski distance function is used to measure the distance between the data point of interest and a neighbour. The shorter the distance calculated, the more similar these two data points are, indicating that the target outcome would be the same for both points.
- **Neural Networks (NN):** MLPClassifier() relies on a Neural Network to classify by applying a multi-layer perceptron (MLP) algorithm using backpropagation. This algorithm maps input values (features) to output values (target outcome). A MLP consists of multiple connecting layers; the nodes of the layers are known as neurons [19]. Between the input and output layers, there can be one or more non-linear layers or hidden layers. Apart from the input layer, each neuron in the hidden and output layers transforms the values from the previous layer with a non-linear activation function [20]. The appraisal of related works indicated that the use of Neural Networks has appeared within multiple studies, proving to achieve high accuracy and quick speed.
- **Logistic Regression (LR):** LR is a simple classification algorithm, in which the target variable is binary. It relates the independent features to the dependent target outcome, with the latter having two classes, 0 or 1. Unlike linear regression, LR uses a sigmoid function to calculate the probability between 0 and 1, and it predicts the class value in terms of its probability. For instance, if the probability, P(target = 0), is greater than 0.5, the instance is said to belong to class 0 (or 'has no heart disease'), otherwise, it would belong to class 1.
- **Gradient Boosting Trees (GB):** GB is a weak learners-based method, which combines a group of decision trees. Within the existing literature, decision trees were often used to predict disease, and their accuracy levels were high [20]. GB is an ensemble method, which builds one decision tree at a time, each learner improving on the last tree produced, with results from each tree being combined, producing a low variance and a high bias. Conversely, random forests, which demonstrate high variance and low bias, are based on the discrete and independent construction of trees. GB aims to reduce bias, potentially resulting in a better performance than random forests with unbalanced data, such as the dataset used in this study [21].

5. Performance evaluation

In this section, the performance of the five machine learning models is evaluated based on evaluation metrics for classification algorithms and Quality of Service parameters. Additionally, the implementation of the best machine learning model within the application, and on Google Cloud Firebase, is discussed. A confusion matrix is a visual diagram, in which the total numbers of correct and incorrect predictions made by the model are summarised for each target class, as shown in Fig. 7. The confusion matrix helps to identify the errors made by the classifier by the values for TP, TN, FP and FN. These results can be combined in different equations to generate metrics, such as accuracy, precision, sensitivity (recall), specificity and ROC curve [22].

5.1. Configuration settings

To mimic the work implemented on Jupyter and Xcode, the following configurations were used locally:- Anaconda: 4.9.2; Python: 3.8.5; Numpy: 1.21.0; Matplotlib: 3.3.2; Seaborn: 0.11.1; Scikit-learn: 0.24.2; Scipy: 1.7.0; Pandas: 1.2.1; Memory-Profiler: 0.58.0; Xcode: 12.5; Swift: 5.

5.2. Experimental results: Machine learning algorithms

In this section, we discuss evaluation metrics and experimental results of various machine learning algorithms.

5.2.1. Accuracy and precision

Accuracy is a proportion of correct predictions in all of the predictions (Eq. (1)).

$$Accuracy = \frac{TP + TN}{TN + FP + FN + TP} \tag{1}$$

Ideally, a high accuracy should be achieved to ensure there are no incorrect classifications. Failing to diagnose a patient with the disease is more dangerous than diagnosing a healthy patient with the disease. The most accurate model was found to be Logistic Regression with an accuracy of 85.96% (Fig. 4).

However, it is not the best idea to evaluate these models on an imbalanced test dataset like the one used in this study. 39% of test instances had the disease, while 61% did not have the disease. The test accuracy would be 61% if the model predicts every

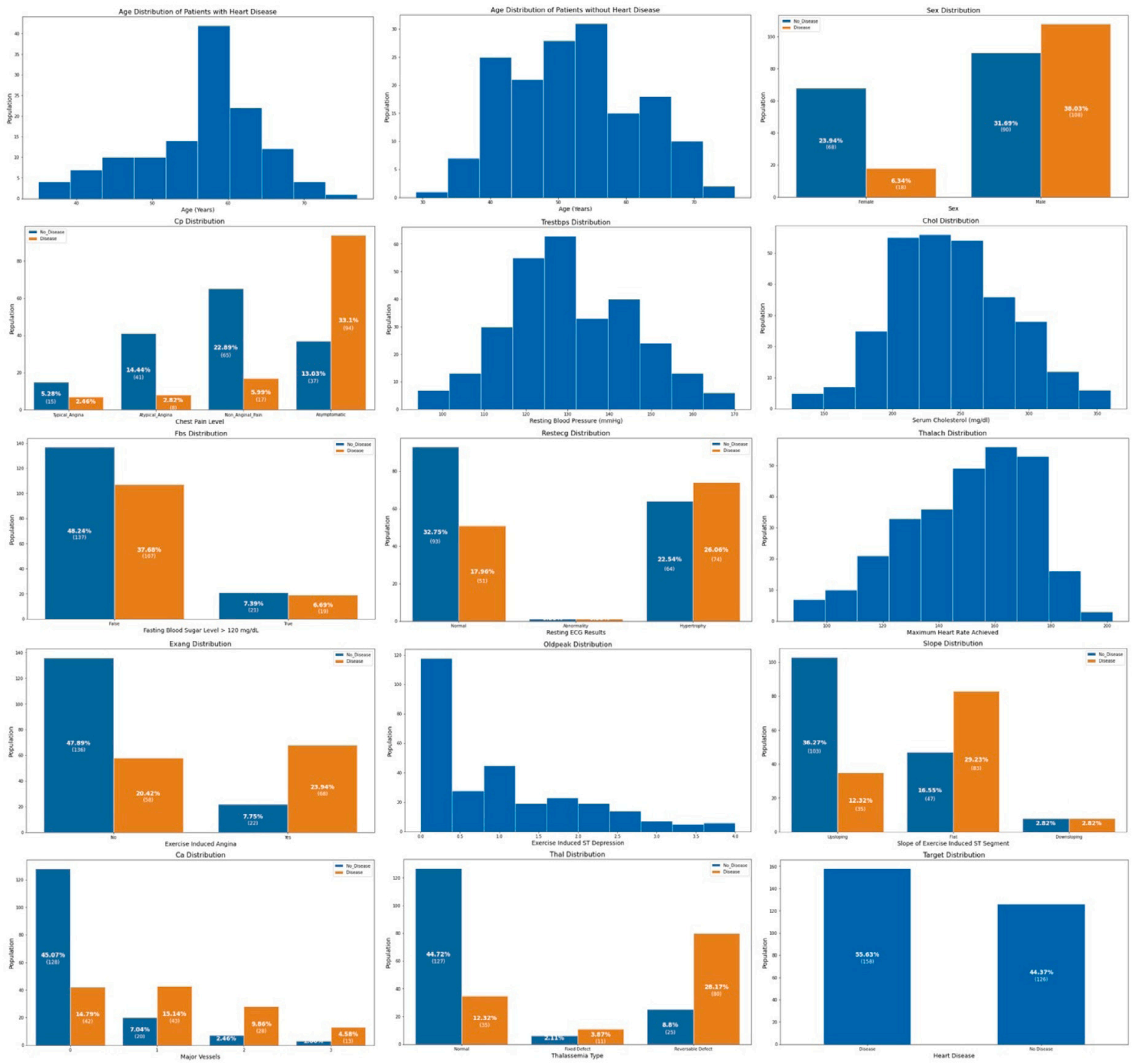

Fig. 3. Histograms of all the features necessary to improve understanding of the distribution of the data in relation to the target variable.

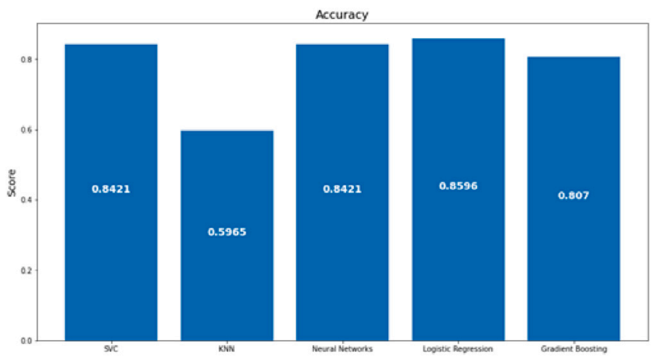

Fig. 4. Test accuracies of machine learning models.

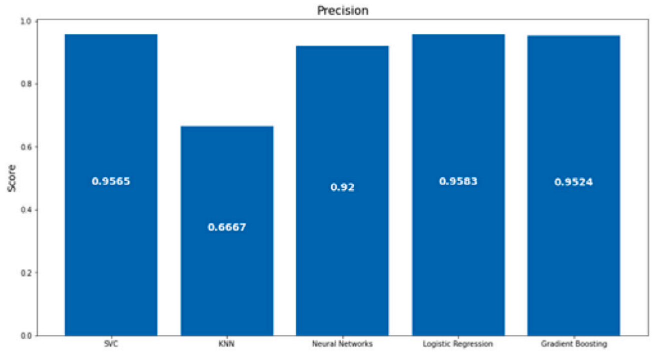

Fig. 5. Test precision of machine learning models.

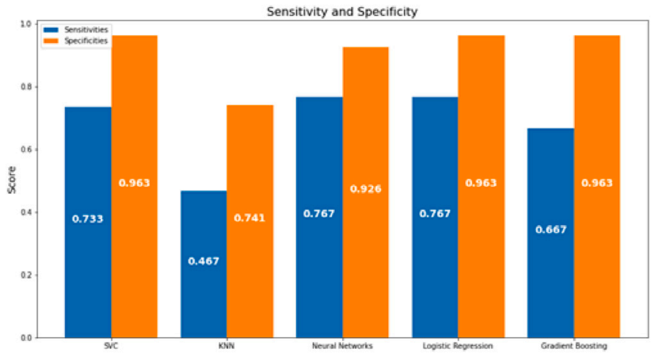

Fig. 6. Sensitivities and specificities of machine learning models.

sample as belonging class 0, but 39% if class 1 is predicted. So, to avoid getting false sense of high accuracy, accuracy needs to be interpreted carefully [23].

$$Precision = \frac{TP}{FP + TP} \tag{2}$$

The level of precision depends on the scenario. Since this is a healthcare related research work, it is necessary that users get a few false results but all patients with the disease are informed. In this case, recall should be maximised over precision. Although, despite the precision/recall trade-off, SVC, Neural Networks and LR all provided high precision (Fig. 5) and high recall (Fig. 6). High precision and high recall ensure patients with the disease are really predicted to have the disease to prevent false hope, whilst also ensuring all patients who are predicted to not have the disease, really do not have the disease to prevent false diagnosis of the disease [22].

5.2.2. *Sensitivity (Recall)*

It is the proportion of correctly identified positive results over all real positive results and shows how well the model can detect heart disease (Eq. (3)). Specificity is the proportion of correctly identified negative results over all real negative results and shows how well the model can detect no heart disease (Eq. (4)).

$$Recall = \frac{TP}{FN + TP} \tag{3}$$

$$Specificity = \frac{TN}{FP + TN} \tag{4}$$

Fig. 6 shows the most specific models are SVC, LR and GB, while the most sensitive models are LR and NN. It is aimed to have both metrics as high as possible to limit the number of false positives and negatives [23], provided by the Logistic Regression model, when compared to the other four models.

False positives are those targets that are really 0, being predicted as 1, diagnosing patients without the disease as having the disease. False negatives are those instances predicted as 0 but do really have the disease, so cases of the disease are being missed by the model.

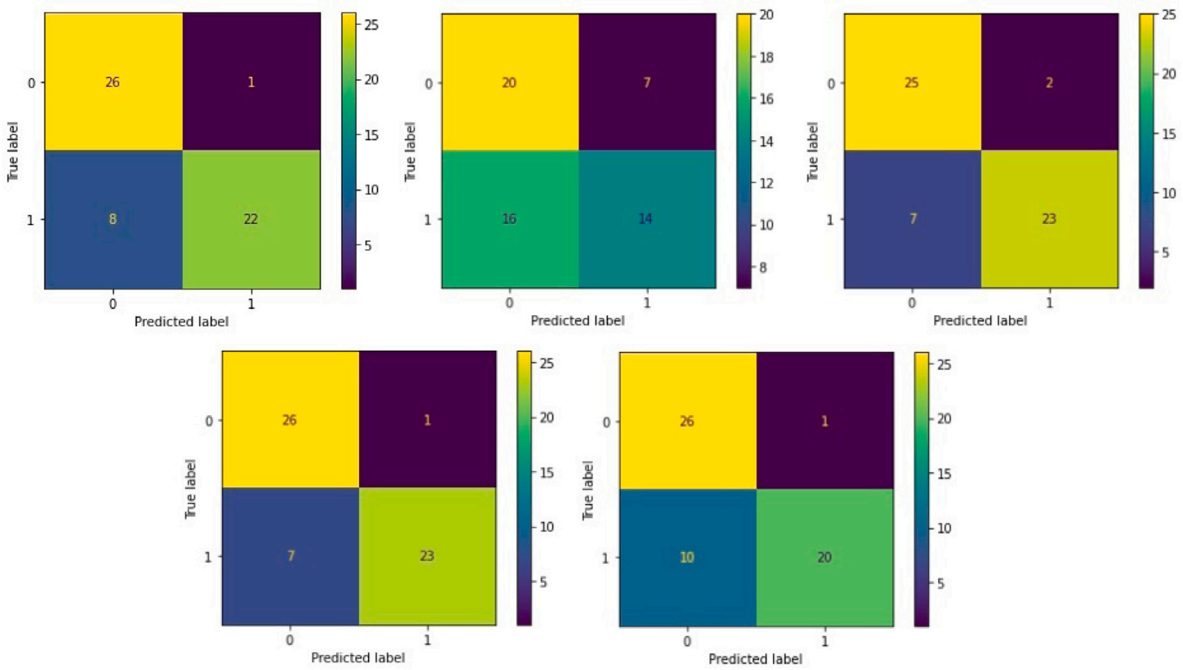

Fig. 7. Confusion matrices to show the performance of the classification algorithms on the test dataset: (a) Support Vector Classifier, (b) K-Nearest Neighbours Classifier, (c) Neural Networks, (d) Logistic Regression, (e) Gradient Boosting Trees.

5.2.3. ROC curve (AUC scores)

Another way of visualising and analysing the numbers generated within the confusion matrix (Fig. 7), is by plotting the ROC curve. The ROC curve plots the True Positive Rate (Sensitivity) against the False Positive Rate (1 – Specificity) at different threshold values. The area under the curve (AUC) is a way of measuring the accuracy of the model. The greater the area under the curve, the better the model is at distinguishing between the two classes. So, the closer the curve is to the diagonal, the less accurate the model is [23]. As shown in Fig. 8, Logistic Regression was the model most capable of distinguishing between classes with an AUC of 0.96, KNN was the worst. In general, while KNN is not a great classifier compared to the rest of these algorithms, according to the ROC curve, while the other four models prove to be excellent classifiers.

5.2.4. Cross-validation

Cross-Validation estimates the performance of a machine learning model when predicting outcomes of test data, which uses a resampling technique to avoid bias of training data. The cross _val _score (cv = 5) function performs K-fold cross-validation, where for five folds, the data is split into different sets of 80% training and 20% test data [24]. It returns a list of accuracy values from the five folds, while accuracy score() calculates the accuracy based on fixed training and test sets. Using all the data, cross-validation ensures the model does not overfit the training data and performs well with unseen data as well as training data. The average of the five cross-validation scores are plotted for each model in Fig. 9 with their respective standard deviations. Cross-validation gives a better approximation for the accuracy calculated using the accuracy _score() function and Fig. 7 confirms the SVC model performed the best at predicting.

5.3. Experimental results: Ensemble learning algorithms

In this section, we discuss experimental results of ensemble learning algorithms. BaggingClassifier() is an ensemble classification method, also used by [16], and it is created through the use of various estimators. It is trained using several sampling techniques, such as (a) random patches, whereby samples and features are selected at random, (b) bagging, whereby samples were drawn with replacement, and (c) random subspaces, whereby random features are drawn. The bagging classifier helps to reduce the variance of individual estimators as it samples the dataset and combines predictions. In this case, the random samples of the dataset are drawn with replacement; hence the method in question is that of bagging. In the case of unstable models, bagging has been shown to have a substantial beneficial effect on accuracy. The 'base _estimator' was formed from the five previously mentioned machine learning algorithms, upon which the random subsets of the dataset were fitted. Ideally, the bagging classifier is used for unstable classifiers, which are those that are less sensitive to changes to the training data [25]. Changing the inputs will not influence the target outcome. For instance, decision trees result in an unstable classifier (high variance and low bias).

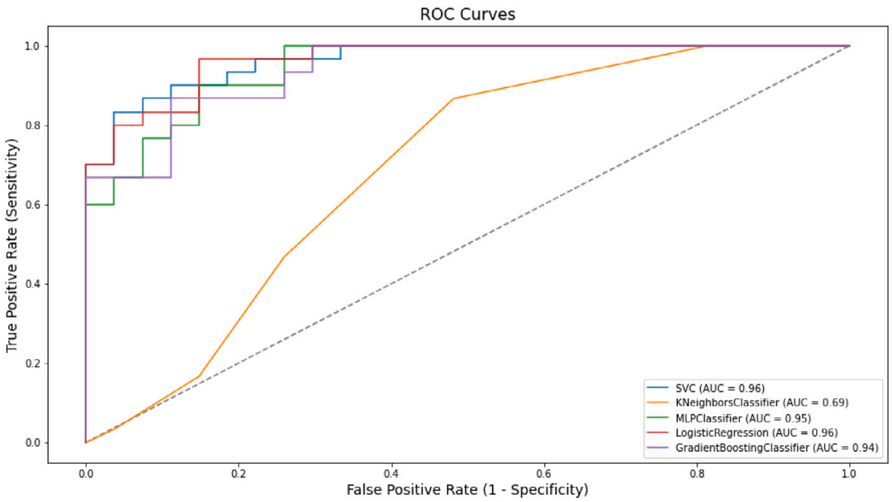

Fig. 8. ROC curves of machine learning models, where AUC is represented by the area under each curve.

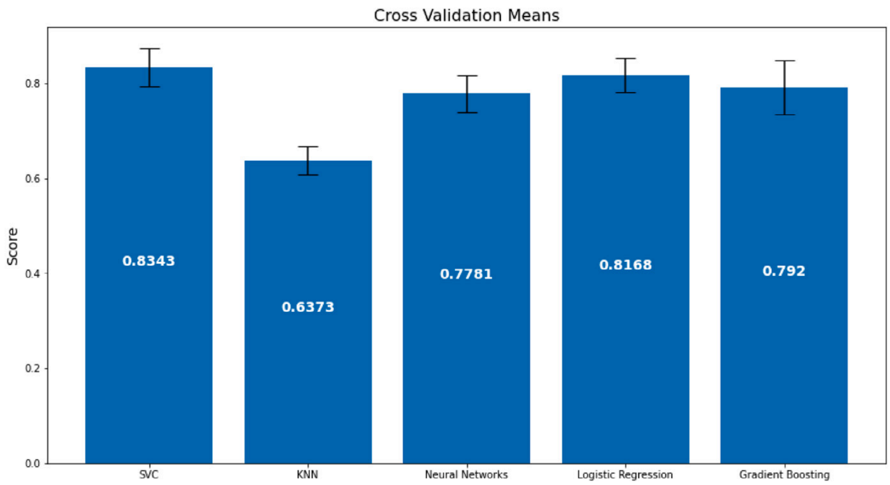

Fig. 9. Cross-validation means of machine learning models with their standard deviations.

As shown in Figs. 10 and 11, in contrast to the work of [16], the use of ensemble learning seemed to make models more latent and less (or equally, in the case of KNN) accurate, as compared to machine learning models without ensemble learning. Since bagging only improves unstable models, KNN accuracy remains unchanged, since KNN (which is based on neighbouring data points) is a stable model [25].

5.4. Experimental results: Computing parameters (QoS)

Typically, here, QoS parameters are being considered for machine learning model evaluation rather than for a network evaluation. Generally, system developers deploy QoS metrics to determine the likely performance of an application in terms of capacity and connection quality, once it is offered to the public via a network, be that the Cloud or a local network [26]. In this work, the cost of machine learning models includes time, memory, reliability, usability, etc. so some of these QoS parameters have been considered to evaluate the performance of machine learning models [26]. It will provide a better user experience if the algorithm does not cause any delays in producing a prediction, and it does not cause their mobile device to slow down due to limited capacity.

5.4.1. Execution time and latency

Two approaches were taken to measure the time required for each model to predict. Both methods utilised the time Python package before and after the predict functions for each model. The first approach measured the total execution time for each model

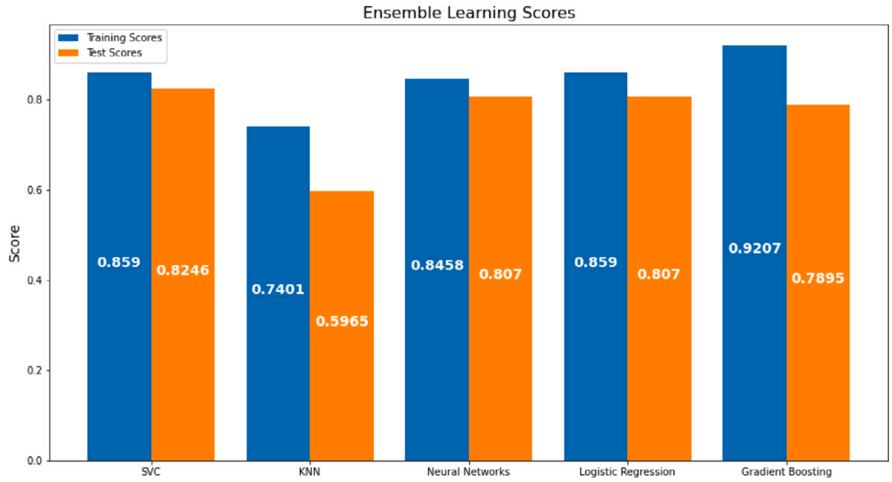

Fig. 10. Training and test accuracy of machine learning models, with ensemble learning applied.

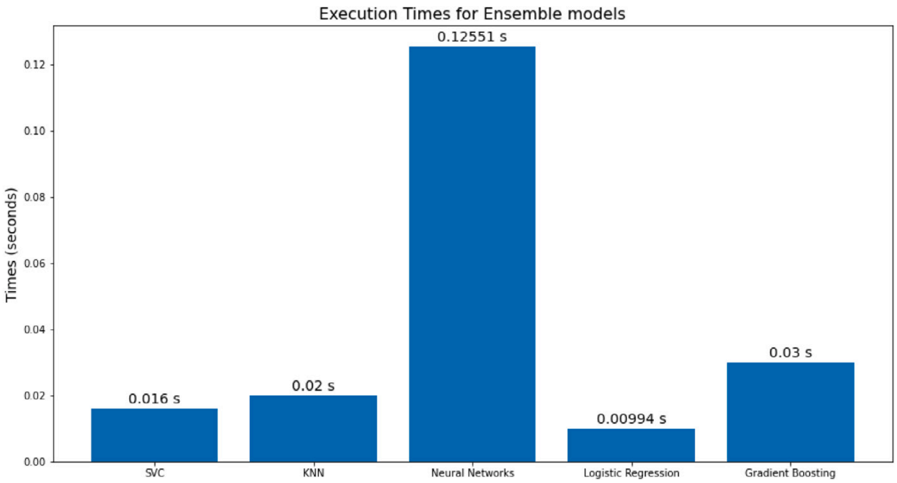

Fig. 11. Time required for the ensemble learning models to predict the target outcomes for all test data instances.

to predict the target outcome for all 57 test instances. Fig. 12 shows KNN took the longest to predict since it has to go through and find the shortest distance between the neighbours, while GB was the fastest. This takes into account the time taken for predictions and not the time taken for training the models. This is because the model would be trained and fitted only once and model prediction would be required for every new instance that is entered into the application.

The second approach measured the latency for each model. Latency changes when load changes, so this approach provided each model with increasing number of test samples, 1, 10, 20, 30, 40 and 50. This measurement aimed to test how the models would be affected as it is provided with more samples to predict at any one time. Initially, it was expected that as load increases, latency would increase proportionally since the more processing is required by the algorithm. However, Fig. 13 illustrates various trends for each model. KNN was the least latent as it took the longest time to respond for every sample size. While the other four models were very similar, LR and SVC seemed to stay the most linear with one spike in their graphs.

Both approaches suggest SVC, NN, LR and GB would not be greatly affected by load and will perform to the same standard despite the increased amount of data being entered into the application for prediction. SVC and LR would be the preferred models in terms of the latency parameter to ensure users get a quick response even if many users are using the application simultaneously.

5.4.2. Memory usage

To measure the consumption of CPU memory of each algorithm, the memory _profiler Python package was used, which depends on another profiling Python package called psutil [27]. The memory _profiler module is capable of simply printing the memory usage of each line in a Python script, in order to visualise where the most memory is being consumed. A function was written, in which each model had to predict the target for one randomly chosen instance from the dataset, so the memory consumed for

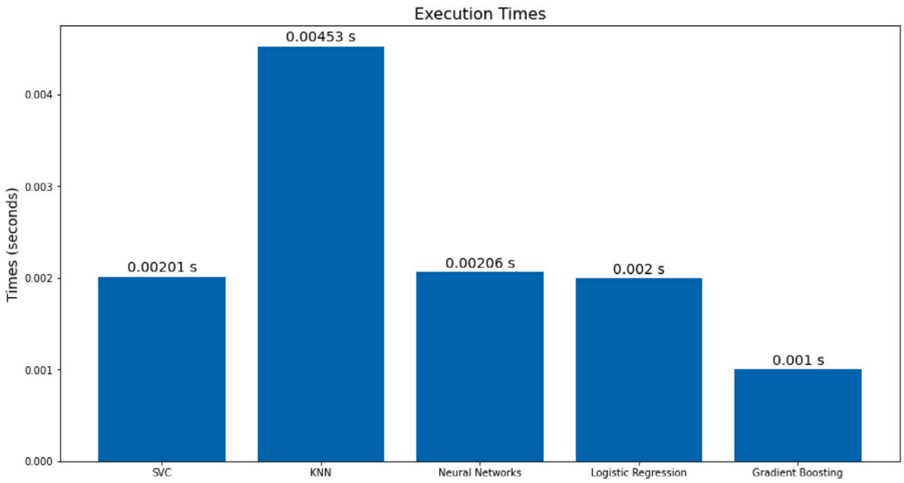

Fig. 12. Times taken for the machine learning models to predict the target outcomes for all test data instances.

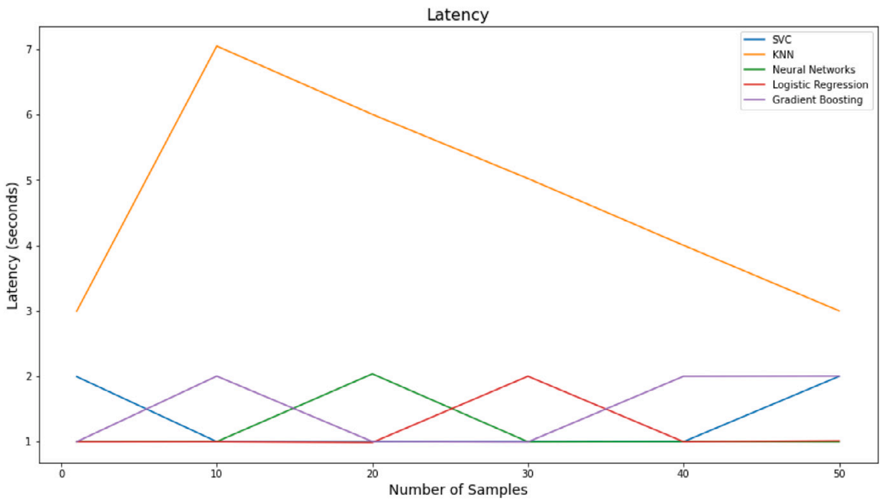

Fig. 13. Comparisons of the time taken for machine learning models to predict the target outcome, as the number of test data samples increases.

training the models is not considered here. The outputs shown in the command line were plotted as shown in Fig. 14. The graph shows LR consumed 0 MiB of memory to predict one target, while KNN consumed the most CPU memory. This is because the entire training dataset needs to be stored for KNN to evaluate the nearest neighbours, while LR stores the one instance in question and calculates the probability for it.

According to the results obtained from this analysis, ensemble learning was not further considered in this study as it did not improve the results from the machine learning models alone. Overall, Logistic Regression provided the highest accuracy, sensitivity and specificity. It complied with the requirements for QoS by consuming the least time and memory. Based on above discussed experimental results, we have chosen Logistic Regression algorithm to implement on Google Cloud Firebase using iOS Mobile Application.

5.5. Implementation of an iOS mobile application on google cloud firebase

Once it was established that Logistic Regression was the most suitable algorithm for the application, the model was accordingly implemented in the application. The classifier had to be trained and tested once again on Apple's CreateML application with the 'target' column separated from the rest of the features. When this was done, an accuracy score of 84%, a precision score of 86% and a recall score of 84% associated with the test data set were provided, once the classification algorithm has been applied. Based on the findings obtained during data analysis, precision decreased by about 10%, while accuracy and recall increased by 2% and 8%, respectively. So, whilst the training and test datasets were kept constant across both approaches, the discrepancies were assumed to

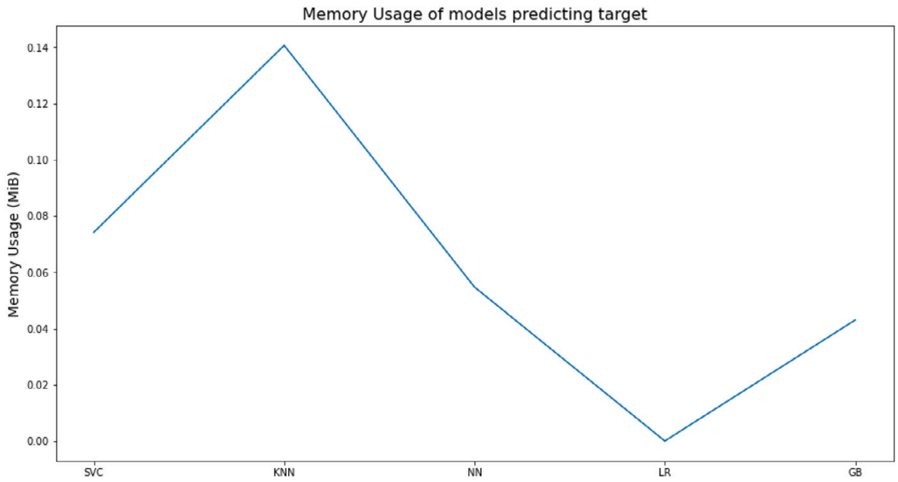

Fig. 14. Memory consumed by each machine learning model in predicting the target outcome for one randomly chosen instance in the dataset.

be due to the different training methods that Python and CreateML utilise. Due to the precision/recall tradeoff [22], it was expected that precision would decrease if recall increased, and vice versa.

CreateML was able to provide an .mlmodel file, containing the details of the trained model required for applications to be developed in Apple's Xcode. The model was ready for use as soon as it had been integrated into the application. This model with the application was tested with about thirty, randomly selected user inputs from the complete dataset as well as several new data values, both to ensure that the model was predicting the expected target, and to make sure that there were no errors related to user-provided inputs.

Additionally, for testing purposes, the application was then linked to Firebase, which is a Google platform for developing and testing mobile and web applications. Specifically, Firebase is a mobile backend-as-a-service (BaaS) provider, used to establish a link between applications and the Cloud. It was found that Firebase is easy to use for real-time user inputs from one local system and that it provides many services. With this application, the main purpose of connecting to the Cloud was to have access to backend services that are not available on a local system. For instance, Analytics provides graphs and data regarding user-based events, which may be employed to help understand users' behaviour. It is useful for the app developer to see the number of users on the application, the range of user inputs being entered into each field, and the proportion of users who may click on external hyperlinks as shown in Fig. 16. On 11th August, 13 sessions went ahead as shown in Fig. 15, out of which the link was only clicked 4 times (Fig. 16). This assists the developer in understanding what type of user is most likely to be drawn to the application for self-diagnosis and/or monitoring. It also indicates the limitations of the application, which may cause the user to misunderstand the inputs that they are required to enter, in the event that they do not receive the response they expect.

In this case, due to time constrains, insufficient data were provided to allow Analytics to establish the types of data that were entered in the text fields. Nonetheless, merely by briefly testing this feature on Firebase, the study found that Analytics could serve as an add-on whilst the application is further developed. Meanwhile, another useful service provided by Firebase, is the user authentication functionality. This can be added on within the application so that the user can log in to the application securely and create a simple user profile. Their data can be viewed in their profile dashboard for monitoring purposes. Still, Firebase may not be the best cloud service provider for achieving the goals of this work, as it is oriented towards Android mobile applications; it provides only basic features and devices for iOS. Moreover, in terms of data storage, since the application has only been used locally for testing purposes, data have not been stored. Since this application would be used for monitoring purposes, conversely and in the event of distribution, users' data should be stored solely for their own personal reference. Unfortunately, Firebase uses a NoSQL database, meaning the integrity of data can be difficult to guarantee; this could lead to data being misplaced. So other Cloud vendors, particularly those involved with Apple, such as Cloudkit, would be implemented as an alternative (see Fig. 15).

6. Conclusions and future work

In this study, an effective diagnostic system called HealthCloud has been developed to predict the presence of heart disease via the use of machine learning. Various machine learning models, such as Support Vector Machine, K-Nearest Neighbours, Neural Networks, Logistic Regression and Gradient Boosting Trees were trained, and their performances were evaluated based on Accuracy, Precision, Cross-Validation results, Sensitivity, Specificity and AUC scores. It was found that Logistic Regression was the fastest and most accurate model, and it was also the one that consumed the least energy when predicting the target for one instance of the dataset. Finally, we built a prototype iOS application and implemented on Google Cloud Firebase for real-time data analysis. The research carried out may be useful for healthcare organisations, such as the National Health Service and the British Heart Foundation. Being able to provide accurate predictions and appropriate advice, is particularly important in health-related scenarios. Providing

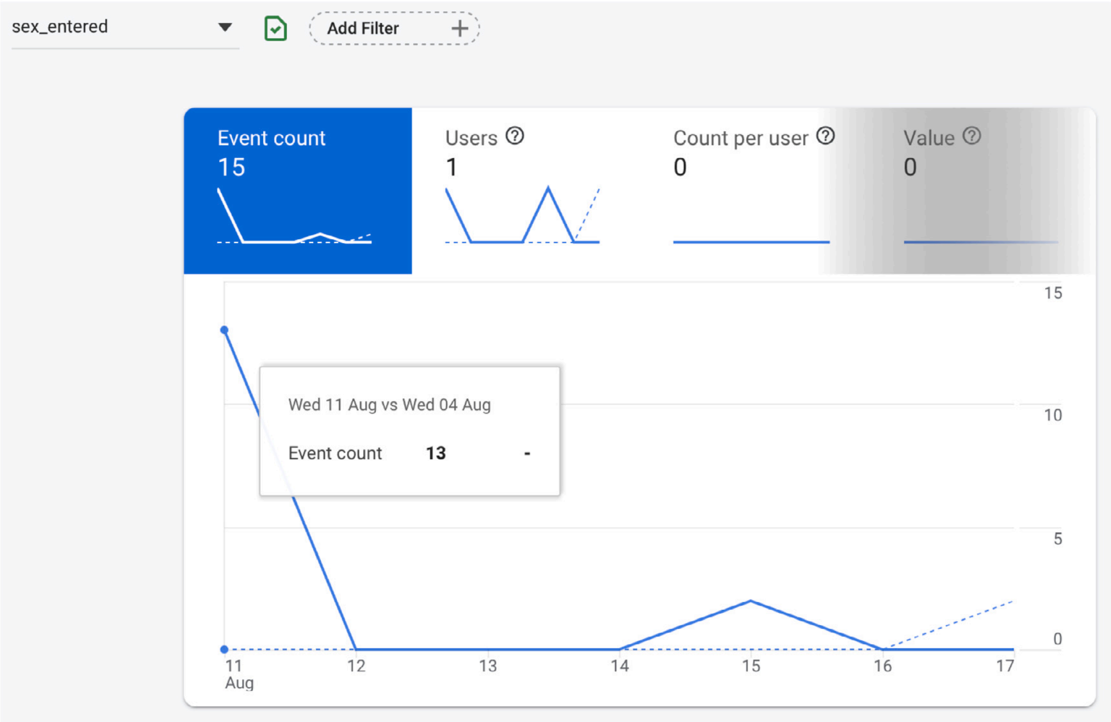

Fig. 15. An example of the event count graphs Google Firebase provides to help developers to understand user behaviour based on Sex feature: sex-entered event.

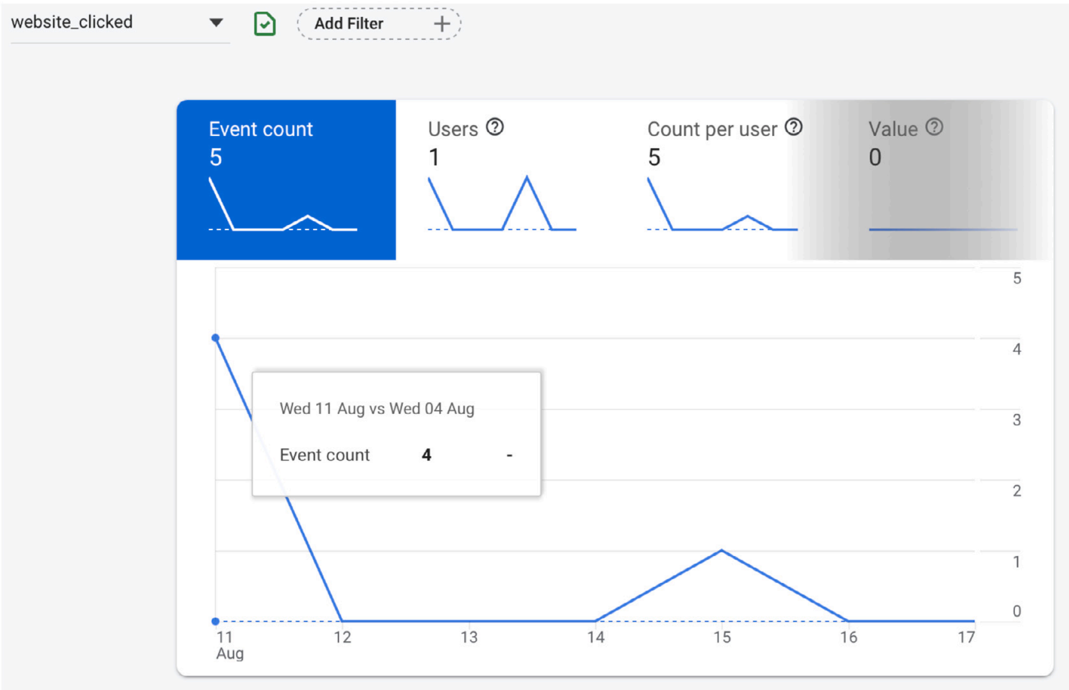

Fig. 16. An example of the event count graphs Google Firebase provides to help developers to understand user behaviour based on Hyperlink: website-clicked event.

an incorrect result can cause unnecessary stress to the user in the case of a false positive. By contrast, a false negative will produce ‘false hope’ and, even more seriously, a potential delay in the treatment of the disease. Relying on machines to diagnose or detect disease can be controversial, so of course, access to medical advice is beneficial, and this system does not aim to replace traditional, physical healthcare, Rather, it has been designed to help patients communicate with their doctors about their symptoms, symptoms, whilst knowing more about the disease that may be causing those symptoms, how to monitor the disease and how it may be treated.

6.1. Future directions

HealthCloud can be extended in the following ways:

- To deploy the application on the Cloud, (in order to test an Apple application), it would be preferable to use one of Apple’s Cloud service providers, rather than Google’s Firebase, as used in this study. Xcode Cloud and Cloudkit are examples of potential Cloud providers, in which users can use their iCloud account to be automatically authenticated; their data would be stored and protected in the Cloud database.
- Cloud computing does have the disadvantage of not being able to acquire real-time data. So, while the Cloud may be used for testing purposes and data storage, an IoT architecture would be developed, utilising sensors connected to an Arduino. This would in turn be integrated with the same machine learning technique to design a real-time automated system. This would allow the mobile application to collect data from patients, such as heart rate, in real-time via the use of wearable sensors.
- To improve the usability of the mobile application, the user interface can be enhanced via additions of colour and graphics, to illustrate what the different values of each feature represent. For instance, the range of chest pain values 1, 2, 3 and 4, could be represented by the colours red, orange, yellow and green, respectively, in order of severity. Graphics can be introduced to represent an upsloping ST segment on an ECG diagram.
- HealthCloud should incorporate dynamic scalability to fulfil the changing demand of user applications without the violation of Service Level Agreement (SLA), which helps to improve performance of cloud computing applications in a cost-effective way and quality of cloud services during peak load.

Such AI applications can also be extended to other domains of healthcare, or indeed, to other industries requiring a useful predictive system. Such uses might include, for example, the diagnosis or monitoring of patients in the contexts of diabetes or of the recent COVID-19 disease. The application might also be used to predict seasonal weather and climate within the agricultural sector thereby aiding farmers, or to predict the financial capacities of home buyers applying for mortgages.

Software availability

We released HealthCloud as an open source software. The implementation code with experiment scripts and results can be found at the GitHub repository: https://github.com/iamssgill/HealthCloud

CRediT authorship contribution statement

Forum Desai: Conceptualization, Data curation, Investigation, Methodology, Software, Visualization, Validation, Formal analysis, Writing – original draft. **Deepraj Chowdhury:** Conceptualization, Data curation, Investigation, Methodology, Software, Visualization, Validation, Formal analysis, Writing – original draft. **Rupinder Kaur:** Investigation, Methodology, Writing – original draft. **Marloes Peeters:** Investigation, Methodology, Writing – original draft. **Rajesh Chand Arya:** Investigation, Methodology, Writing – original draft. **Gurpreet Singh Wander:** Investigation, Data Curation, Methodology, Writing – original draft. **Sukhpal Singh Gill:** Conceptualization, Data curation, Investigation, Methodology, Software, Visualization, Validation, Formal analysis, Writing - original draft, Supervision. **Rajkumar Buyya:** Conceptualization, Investigation, Methodology, Software, Writing - original draft, Supervision.

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.

References

[1] S. Shrestha, P.P. Sengupta, Imaging heart failure with artificial intelligence: improving the realism of synthetic wisdom, 2018.
[2] C. Venkatesan, P. Karthigaikumar, S. Satheeskumaran, Mobile cloud computing for ecg telemonitoring and real-time coronary heart disease risk detection, Biomed. Signal Process. Control 44 (2018) 138–145.
[3] D. Prabhakaran, K. Singh, G.A. Roth, A. Banerjee, N.J. Pagidipati, M.D. Huffman, Cardiovascular diseases in india compared with the united states, J. Am. Coll. Cardiol. 72 (1) (2018) 79–95.
[4] D. Prabhakaran, P. Jeemon, A. Roy, Cardiovascular diseases in india: current epidemiology and future directions, Circulation 133 (16) (2016) 1605–1620.
[5] A. Kishor, C. Chakraborty, Artificial intelligence and internet of things based healthcare 4.0 monitoring system, Wirel. Pers. Commun. (2021) 1–17.
[6] S.S. Gill, R.C. Arya, G.S. Wander, R. Buyya, Fog-based smart healthcare as a big data and cloud service for heart patients using iot, in: International Conference on Intelligent Data Communication Technologies and Internet of Things, Springer, 2018, pp. 1376–1383.

[7] Y. Yan, J.-W. Zhang, G.-Y. Zang, J. Pu, The primary use of artificial intelligence in cardiovascular diseases: what kind of potential role does artificial intelligence play in future medicine? J. Geriatr. Cardiol.: JGC 16 (8) (2019) 585.
[8] A. Mincholé, B. Rodriguez, Artificial intelligence for the electrocardiogram, Nat. Med. 25 (1) (2019) 22–23.
[9] Y. Liu, A.H. Ngu, L.Z. Zeng, QoS computation and policing in dynamic web service selection, in: Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, 2004, pp. 66–73.
[10] S.S. Gill, S. Tuli, M. Xu, I. Singh, K.V. Singh, D. Lindsay, S. Tuli, D. Smirnova, M. Singh, U. Jain, et al., Transformative effects of iot, blockchain and artificial intelligence on cloud computing: Evolution, vision, trends and open challenges, Internet Things 8 (2019) 100118.
[11] M. Varela, L. Skorin-Kapov, T. Ebrahimi, Quality of service versus quality of experience, in: Quality of Experience, Springer, 2014, pp. 85–96.
[12] S. Nashif, M.R. Raihan, M.R. Islam, M.H. Imam, Heart disease detection by using machine learning algorithms and a real-time cardiovascular health monitoring system, World J. Eng. Technol. 6 (4) (2018) 854–873.
[13] M. Hosseinzadeh, J. Koohpayehzadeh, A.O. Bali, P. Asghari, A. Souri, A. Mazaherinezhad, M. Bohlouli, R. Rawassizadeh, A diagnostic prediction model for chronic kidney disease in internet of things platform, Multimedia Tools Appl. 80 (11) (2021) 16933–16950.
[14] A. Al-Zebari, A. Sengur, Performance comparison of machine learning techniques on diabetes disease detection, in: 2019 1st International Informatics and Software Engineering Conference (UBMYK), IEEE, 2019, pp. 1–4.
[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[16] S. Tuli, N. Basumatary, S.S. Gill, M. Kahani, R.C. Arya, G.S. Wander, R. Buyya, Healthfog: An ensemble deep learning based smart healthcare system for automatic diagnosis of heart diseases in integrated iot and fog computing environments, Future Gener. Comput. Syst. 104 (2020) 187–200.
[17] https://www.gov.uk/data-protection.
[18] J. Nahar, T. Imam, K.S. Tickle, Y.-P.P. Chen, Association rule mining to detect factors which contribute to heart disease in males and females, Expert Syst. Appl. 40 (4) (2013) 1086–1093.
[19] R. Saravanan, P. Sujatha, A state of art techniques on machine learning algorithms: a perspective of supervised learning approaches in data classification, in: 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, 2018, pp. 945–949.
[20] B. Akkaya, N. Çolakoğlu, Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases, 0000.
[21] S. Glen, Decision tree vs random forest vs gradient boosting machines: Explained simply, Data Sci. Cent. (2019).
[22] A. Botchkarev, Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology, 2018, arXiv preprint arXiv:1809.03006.
[23] W. Zhu, N. Zeng, N. Wang, et al., Sensitivity, specificity, accuracy, associated confidence interval and roc analysis with practical sas implementations, in: NESUG Proceedings: Health Care and Life Sciences, Vol. 19, Baltimore, Maryland, 2010, p. 67.
[24] P. Refaeilzadeh, L. Tang, H. Liu, Cross-validation, Encyclopedia Database Syst. 5 (2009) 532–538.
[25] Y. Ganjisaffar, R. Caruana, C.V. Lopes, Bagging gradient-boosted trees for high precision, low variance ranking models, in: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2011, pp. 85–94.
[26] A.K. Bardsiri, S.M. Hashemi, Qos metrics for cloud computing services evaluation, Int. J. Intell. Syst. Appl. 6 (12) (2014) 27.
[27] F. Pedregosa, P. Gervais, Memory profiler, 2020, URL https://github.com/pythonprofilers/memory_profiler.

Forum Desai is a Computer Science M.Sc. student at Queen Mary, University of London, UK. Currently, she is pursing training as a Full Stack Web Developer. Forum has a B.Sc. in Chemistry with Biomedicine. Her research interests include IoT, Machine Learning, Smart Healthcare and Cloud Computing.

Deepraj Chowdhury is with department of electronics and communication, IIIT- Naya Raipur. He has Co-authored 4 research papers in different conferences like ICACCP 2021, INDICON 2021. He also has 3 indian copyright registered, and applied for 2 Indian Patent. He is also serving as a reviewer in Wiley Transaction on emerging Telecommunication Technologies.

Rupinder Kaur is an employer at School of Science, Dulwich College, United Kingdom. Prior to this, Dr. Kaur has completed her Ph.D. in Chemistry from Guru Nanak Dev University, Amritsar, India. She received her M.Sc. degree in Chemistry from the Punjabi University, India and BSc Degree from Panjab University, India. Her area of research is to study the micellization behaviour of surfactants in the presence of various additives. Dr. Kaur has published her research in top ranked venues.

Marloes Peeters graduated from Eindhoven University of Technology (The Netherlands) with a degree in Chemical Engineering. For her Ph.D., she moved to research institute IMO/IMOMEC in Belgium where she was part of the BIOSensors group of Prof Wagner. Her postdoctoral research fellowship at the BIOS group focused on aptamers as recognition elements. In 2014, she started working in the group of Prof. Kilburn at Queen Mary University of London. Her independent research started at Manchester Metropolitan University in 2015. Since March 2019, she is a senior lecturer (associate professor) at Newcastle University leading the Bioinspired Materials research group.

Rajesh Chand Arya is a Consultant Cardiac Anesthesiologist at Department of Cardiology, Hero Heart Institute Dayanand Medical College and Hospital, Ludhiana, Punjab, India. He has 12 years of Cardiac Anesthesia experience after post-graduation. Special interest in Trans Esophageal Echo (TEE) & Pediatric, Cardiac Anesthesia. He has delivered more than 20 guest lectures at national & international level.

Gurpreet Singh Wander is a Chief Cardiologist at Department of Cardiology, Hero Heart Institute Dayanand Medical College and Hospital (DMC & H), Ludhiana, Punjab, India. He joined DMC & H in 1988. He has been awarded BC Roy Award in 2007 and K. Sharan Award by national IMA in 2005. He is a Member of Govern Board of API for 6 years. Dr. Wander has 200+ research publications in top ranked venues which include 5 publications in Nature journals (Nature genetics, Journal of Human Genetics, Scientific Reports and Journal of Human Hypertension).

Sukhpal Singh Gill is a Lecturer (Assistant Professor) in Cloud Computing at School of Electronic Engineering and Computer Science, Queen Mary University of London, UK. Prior to this, Dr. Gill has held positions as a Research Associate at the School of Computing and Communications, Lancaster University, UK and also as a Postdoctoral Research Fellow at CLOUDS Laboratory, The University of Melbourne, Australia. Dr. Gill is serving as an Associate Editor in Wiley ETT and IET Networks Journal. His research interests include Cloud Computing, Fog Computing, Software Engineering, Internet of Things and Healthcare. For further information, please visit http://www.ssgill.me.

Rajkumar Buyya is a Redmond Barry Distinguished Professor and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercialising its innovations in Cloud Computing. He has authored over 850 publications and seven text books including "Mastering Cloud Computing" published by McGraw Hill, China Machine Press, and Morgan Kaufmann for Indian, Chinese and international markets respectively. He also edited several books including "Cloud Computing: Principles and Paradigms" (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index=151, g-index=329, 119200+ citations). "A Scientometric Analysis of Cloud Computing Literature" by German scientists ranked Dr. Buyya as the World's Top-Cited (#1) Author and the World's Most-Productive (#1) Author in Cloud Computing. Dr. Buyya is recognised as a "Web of Science Highly Cited Researcher" for six consecutive years since 2016, IEEE Fellow, Scopus Researcher of the Year 2017 with Excellence in Innovative Research Award by Elsevier, and the "Best of the World", in Computing Systems field, by The Australian 2019 Research Review. Software technologies for Grid and Cloud computing developed under Dr. Buyya's leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 50+ countries around the world. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and five IEEE/ACM conferences. These contributions and international research leadership of Dr. Buyya are recognised through the award of "2009 IEEE Medal for Excellence in Scalable Computing" from the IEEE Computer Society TCSC. Manjrasoft's Aneka Cloud technology developed under his leadership has received "2010 Frost & Sullivan New Product Innovation Award". He served as the founding Editor-in-Chief of the IEEE Transactions on Cloud Computing. He is currently serving as Editor-in-Chief of Journal of Software: Practice and Experience, which was established 50+ years ago. For further information, please visit his cyberhome: www.buyya.com.