Multimedia Tools and Applications (2023) 82:30405–30434
https://doi.org/10.1007/s11042-023-14562-3

A scalable and real-time system for disease prediction using big data processing

Abderrahmane Ed-daoudy[1] · Khalil Maalmi[1] · Aziza El Ouaazizi[1]

Received: 26 March 2021 / Revised: 30 June 2022 / Accepted: 31 January 2023 /
Published online: 22 February 2023

Abstract

The growing chronic diseases patients and the centralization of medical resources cause significant economic impact resulting in hospital visits, hospital readmission, and other healthcare costs. This paper proposes a scalable and real-time system for disease prediction from medical data streams. This is carried out by integrating Twitter, Apache Kafka, Apache Spark and Apache Cassandra. Thus, Twitter users tweet attributes related to health, Kafka streaming receives all desired tweets attributes and ingest them to Spark streaming. Here, a machine learning algorithm is applied to predict health status and send back a response message through Kafka. The heart disease dataset, obtained from the UCI repository, was used for experiments. In order to enhance prediction accuracy, Relief algorithm is used for features selection. We compared sex types of relevant machine learning algorithms implemented by Spark MLlib such as Random Forest (RF), Naive Bayes, Support Vector Machine, Multilayer Perceptron, Decision Tree and Logistic Regression with the full features as well as selected features. The highest classification accuracy of 92.05% was reported using RF with selected features. The scalability of RF using Spark MLlib and WEKA framework for both training and application stages was measured. The results show significantly better performances of Spark in terms of scalability and computing times.

Keywords Real-time · Streaming processing · Machine learning · MLlib · Apache Spark · Tweet processing

✉ Abderrahmane Ed-daoudy
a.eddaoudy@gmail.com

Khalil Maalmi
khalil.maalmi@usmba.ac.ma

Aziza El Ouaazizi
aziza.elouaazizi@usmba.ac.ma

[1] Artificial Intelligence, Data Sciences and Emerging Systems Laboratory (LIASSE), National School of Applied Sciences (ENSA), Sidi Mohamed Ben Abdellah University, Fez, Morocco

1 Introduction

Soft computing techniques are widely used for classification of disease in medical field especially data mining which is the computational process of discovering patterns and useful knowledge from databases [3, 22]. Indeed, digital data is becoming increasingly important in many domains like healthcare, technology and society. The amount of data generated in real-time is becoming very important, which involves a number of problems, the main one being the processing and prediction of streaming data event coming with rapid rate. Solving these problems using traditional technologies require hardware resources and time-consuming for the analysis especially machine learning. To deal with these challenges, powerful distributed computing platforms are widely used.

The importance and effectiveness of big data tools in healthcare field is explained in [38]. The authors demonstrate that the effective way on healthcare delivery costing and achieve good healthcare outcomes is by integrating big data tools with data mining, big data analysis and medical informatics, while effective systems support is still lacking for already established machine learning use cases in the big data context [16].

Hadoop MapReduce [17], it constitute the first generation processing engine for big data and the powerful solution for one-pass computations, it require one Map phase and one Reduce phase for any data processing. The main drawbacks of Hadoop MapReduce is does not supports real-time stream processing and in-memory computation. Also, it is not always easy to implement the MapReduce paradigm for all use cases.

Apache Spark [9], the second generation processing framework designed for fast computation, real-time analytic, data processing workloads, ease of use and optimized to run in memory. With its in-memory computation, the performance can be several times faster than other big data frameworks, especially in problems involving iterative machine learning [27]. Spark provides a distributed machine learning framework called MLlib, it consists of a library which implements a set of commonly used machine learning and statistical algorithms. Spark streaming is built on top of core Spark API, it focuses on processing data streams in real-time from various sources like Twitter and Kafka.

Data analytics has been made a revolution in healthcare by transforming the data into valuable information in order to predict epidemics, avoid preventable deaths and improve quality of life [14]. But the growth of volume, complexity and speed in data drives the need for scalable big data analytic algorithms and systems. Due to fast growth of social networks and their role in the daily life of millions of people around the world, social media and mobile applications have opened up new path-ways for healthcare delivery [1].

Based on the challenges facing the healthcare systems, we have proposed and developed a solution in healthcare with a real-time health status prediction use case. This solution is based on Twitter streaming, Kafka streaming, Spark streaming, Spark MLlib, NoSQL Cassandra and Twitter for real-time data transfer. Based on this, the system first preprocesses the available healthcare data and analyzes it to create an offline model for learning system, the model then deployed on system and use it in real-time to predict health status. To balance the incoming load, multiple streams of user data related to health are tweeted from Twitter in a predefined format, ingested and filtered on Kafka streaming. The health attributes are extracted and processed at Spark streaming on which machine learning model is applied. The health status result is sent back to the user, and are then stored in NoSQL based distributed data storage for data visualization and analytics. Efficient processing of data in healthcare increases the quality of patient monitoring.

The proposed system for disease prediction is explained as follows. Section 2 gives the related works. The design and architecture of our system is explained in Section 3. Section 4 will give the implementation of each module, experimental results are presented and discussed. Finally, there is a conclusion part in Section 5.

2 Related works

In the past few years, many researches were centered on the use of machine learning in predicting outcomes especially in healthcare field. Machine learning with data mining tools show its power in predicting diseases, extract patterns and make decisions [32]. But in the big data context, machine learning is handled only in a few works.

In [45] a predictive model related to the risk of diabetes is performed using a scalable RF classification algorithm. A Hadoop based intelligent care system is proposed in [46] that illustrates IoTs based big data contextual sharing across all devices in a health system. Using Hadoop, a novel method based on k-Nearest Neighbors algorithm (KNN) to efficiently detect the outliers in large-scale healthcare data has been proposed in [56]. This method outperformed the KNN and Local Outlier Factor (LOF) in terms of accuracy and processing efficiency. Authors in [59] proposed an automated method that is able to detect abnormal patterns for the elderly living alone entering, exiting behaviors collected from simple sensors equipped in home-based setting. Usage of convolutional neural network based multimodal disease risk prediction algorithm for disease prediction by machine learning over big data from healthcare communities is performed in [15]. In [12] a hybrid fuzzy-based decision tree algorithm for early detection of heart disease using a continuous and remote patient monitoring system was proposed. The proposed system use some parameters related to heart disease obtained from wearable sensor attached to human body. An alert message is sent to the respective physician and the care taker when the obtained value exceeds the threshold value.

Usage of Hadoop technology, authors in [4] have built an application platform in order to load and visualize the data collected from raw sensors, which is a wireless network of wearable computing devices for monitoring the condition of a human body. A new scalable IoTs based architecture has been proposed in [35]. This approach is designed to handle big data from sensor and identify the most significant parameters of heart disease. There are three major components in this architecture, the first level consists of data collection from medical devices. The second level is set to store the huge amount of data in cloud computing by using Hbase. Linear regression is used as a prediction model for the prediction of heart disease using Apache Mahout based machine learning libraries. A new framework for a lambda based cardiovascular disease prediction system is proposed in [24]. The framework uses big data technology notably Hadoop MapReduce to solve the problems associated with real-time analysis of big data. This system can be used to assist, predict and diagnose diseases such as cardiovascular disease. Due to the growing digitalization, it is necessary to move from paper-based medical records to digital by managing the large volume of health data for analytical purposes and using them for effective treatment will be a crucial issue. To deal with this situation, an approach based on the Hadoop MapReduce framework was proposed by [48]. This method uses the big data predictive analysis algorithm to predict the complexities of diabetes mellitus and the type of treatment to adopt. Usage of Spark framework, authors in [53] have proposed a Naive Bayes approach for constructing classifier based on big data predictive analytics model to predict the future health condition of heart disease

data taken from UCI machine learning repository. In [40], a new architecture has been proposed that can support the implementation, storage and processing of scalable sensor data for healthcare applications. The proposed architecture is divided into two sub-architectures : Meta Fog-Redirection (MF-R) and Grouping and Choosing Architecture (GC). The first uses Apache Pig and Apache HBase to collect and store generated big data from different devices. The second architecture is used for securing integration of fog computing with cloud computing. MapReduce based prediction model is used to predict the heart disease.

On the other hand, stream computing over big data is handled only in a few works. In [18] a real-time health status prediction system is proposed, this work focuses on applying machine learning especially decision tree on data streams received from socket streams with breast cancer use case using Spark streaming framework. An overview of big data architectures and machine learning algorithms for processing big data in healthcare and other applications is discussed in [39]. Furthermore, a generic architecture for healthcare analytics has been proposed in [50]. A real-time heart disease prediction system based on big data framework is proposed in [19]. The proposed work is based on Apache Spark which stand as a strong large scale distributed computing platform that can be used successfully for streaming data event against machine learning through in-memory computations.

The combination of streaming big data and machine learning is a revolutionary technology that can have a significant impact in the field of healthcare, especially the real-time detection of heart diseases. In [21], authors proposed a novel heart disease monitoring system based on a new classification approach that combines distributed machine learning and real-time that uses the real-time predictive analytics algorithm in Spark environment to predict heart disease. First, the traditional decision tree algorithm was transformed into a parallel, distributed, scalable, and fast decision tree. Then, this model is applied to real-time data coming from distributed sources to predict heart disease in real time. The result as well as data streams were stored in a distributed database for real-time reporting and monitoring. This system is limited in terms of user interaction which becomes a necessity for prediction and monitoring systems. On the other hand, model classification accuracy rate is quite low, often, a single tree is not sufficient for producing effective results. In addition, decision tree are highly prone to being affected by outliers and often overfit training data. More that, there is no comparison with other type of machine learning algorithm in order to achieve high accuracy. With the increase of data sources and users, to balance the incoming load streams in this system is still challenging as Spark itself is not designed to data management. For this, integrating another framework designed specifically for data stream management will be more efficient in real-time data processing.

There are studies showing the power of social media in particular Twitter data in monitoring and transfer data, such as real-time flu and cancer surveillance system by mining Twitter [37], finding patterns related to the health events [58], earthquake reporting system using Twitter as a social sensor for detecting an event in real-time. In [54], a workflow for data ingestion and data management of Twitter streaming data is developed where they retrieved space-time activities from geotagged tweets and stored them in a single cluster of MongoDB. Finding trending topics is discussed in [23]. In [51], a system is developed for collecting data which uses Twitter to transfer data to a clinician to provide follow up for cardiovascular patients. This work involves healthcare professionals to analyze the data and to send appropriate messages. Author in [49] offers the design of a healthcare system for monitoring patients in real-time. In this system, different parameters influence heart disease are captured from the patient through sensors that are then sent to the patient's Android mobile phone application. RF algorithm is used to predict heart disease. This new system will also

assist the physician visualize the records of other patients with the same medical report of a selected patient using the KNN classification algorithm. In [20], authors proposed a new and general architecture for real-time health status prediction and analytics system using big data technologies. The system focus on applying distributed machine learning model on streaming health data events ingested to Spark streaming through Kafka topics.

Spark is faster than Hadoop and has a better performance especially in problems involving iterative machine learning [26]. Furthermore, authors in [34] found that the proposed methods in the literature were limited to batch processing while big data streams computing are not been widely adopted and remained open to future research. They concluded their work with a recommendation to direct future research towards big data streams processing, it was recommended that research efforts should focus on the development of scalable frameworks and algorithms adapted to real-time data analysis. On the other hand there is a gap of capability of performing more complicated analytical tasks in real-time like managing and analyzing the events streams, machine learning algorithms, data storage, data visualization and transforming healthcare data into valuable information.

Reviewing related work in this field showed that the healthcare analytics solution involving big data are mainly focused on Hadoop. It can process a large volume and diverse data sources in case of batch oriented computing which is not sufficient when it comes to analysing real-time application scenarios, it would be limited for real-time computing [42]. On the other hand, these works consider a specific healthcare data sources or focuses only on batch computing. However, healthcare data sources are continuously generate huge data at a high rate. In addition, they are either considering powerful tools for data analysis such as machine learning and data mining, or they are focusing only on data storage and visualization. Therefore, real-time healthcare analytics, which includes continuous data collection, real-time processing and powerful tools for distributed machine learning, distributed data storage and real-time analytics is necessary to build an effective system for handling distributed healthcare data streams. Others studies in the field of heart disease prediction have focused only on predicting heart disease based on traditional machine learning algorithms with full features. These studies cannot predict heart disease with real-time streaming data. In addition, data from social media platforms as a source is not considered to solve the diseases prediction problems. Other studies use single algorithm to build the model, without using any other type of machine learning algorithm in order to achieve high accuracy.

3 System architecture

The purpose of this study is to develop a real-time data processing, monitoring system combining Twitter streaming, Kafka streaming and Spark streaming. It consists of a four-tier architecture. Tier-1 focuses on collecting data from Twitter status. Tier-2 uses the Spark MLlib to develop the Random Forest model for heart disease prediction. Tier-3 uses Apache Cassandra to store the huge volume of Twitter status data. Figure 1 shows the architecture of the proposed system. Firstly, the user tweets the health attributes, they are sent to the Spark streaming application through Kafka streaming, where the real-time processing is performed. Spark streaming receives health attributes from Kafka streams with Twitter user's name and apply the machine learning model to predict health status. After that, an appropriate message is send back to the user based on Twitter username. The results as well as data streams will be stored in distributed database for historical data analysis and real-time monitoring.

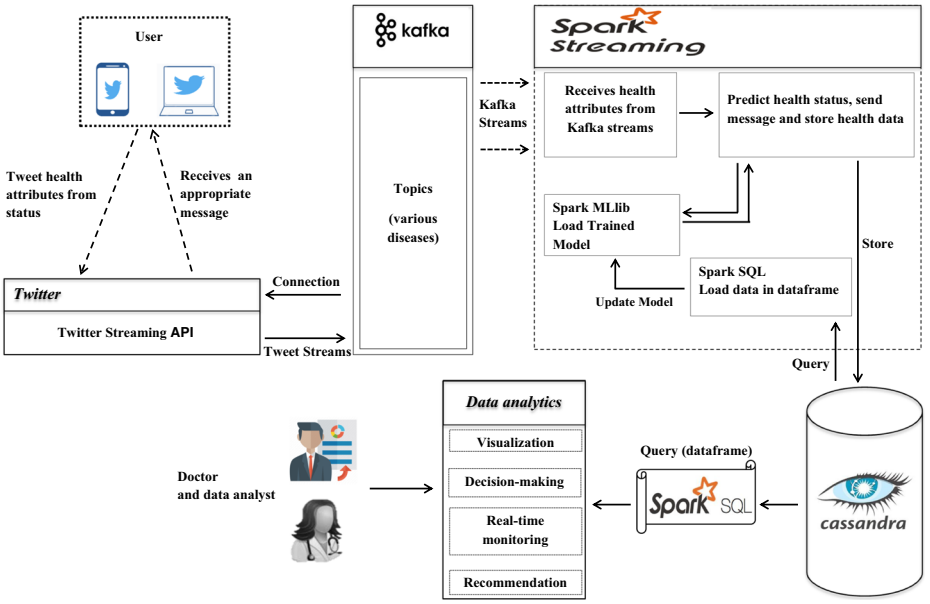

Fig. 1 Architecture of the proposed system

This system has some characteristics which distinguish it from others traditional data analytics approaches. The main idea here is that there is a need for methods to manage and analyze thousands of messages coming from different sources each second in a short amount of time. Also, the system should be independent of imported data volume. We notice that most of state of the art approaches are useful for predicting the health status but the real-time data stream management, data storage, users interactions and data visualization is not covered. With the high populations, to cover all patients by the available doctors is a challengin [31]. On the other hand one of the most important technological challenges of big data analytics is exploring ways to effectively obtain valuable information for different types of users and knowledge discovery in big data generated by all users.

3.1 Data ingestion

Improvement in technology has increased the availability and use of smartphones, personal computer and many others. Due to the rapid growth of social networks and their role in the daily life of millions of people around the world, the number of users and the amount of data generated in these media is increasing exponentially. Twitter is one of the popular social network site, it's a micro blogging site and updates in 140 characters through ultra short messages called tweets in which the peoples can share their opinions, sending and receiving messages [47]. Twitter becomes an inseparable part of human life and fastest way to get real-time information from around the word, it provides free and public access to stream of tweets. Therefore, it is considered as rich source of real-time data in an inexpensive way. As it is supported by Spark streaming and smartphones environments where memory, bandwidth and display size are limited. Twitter can be used as an effective and free real-time communication channel tool. Therefore, it has been integrated to our system. Instead of using Twitter, streaming big data can be generated from other data sources such

as medical IoT technology and wearable sensor devices, from SMS or mobile application to send and receive data to the system, but it requires an additional infrastructure to establish the communication, resulting in an additional investment of time and money.

In this module, we capture tweets streams using a specific keyword related to health disease followed by attributes in the same format as the feature vector in the testing set separated by space:
#rtbigdhdsparkt 1 1 2 0 2.3 3 0 6
This is the first module in our system. For authentication request to the Twitter platform, a Twitter account and Twitter application are required to be created and set the access level to read and write. The same we need applicant Consumer Key and Secrete key. We also require Access token and Access Token Secret to make streaming API requests on your own accounts without sharing password. Then the Twitter streams are ingested to Spark through Kafka [8]. To balance the incoming load streams in our system, Kafka is used, which is more suitable in dealing with real-time streaming data routes. In this module, real-time data is streamed from Twitter through Kafka producer.

3.2 Distributed computing

MapReduce is a programming model introduced in 2004 by Google for large scale processing across clusters [17]. It is the core component of the Apache Hadoop framework, which enables the resilient and distributed processing of massive and unstructured data across clusters where each node has its own storage space. Internally, the framework offers two main features. It distributes the tasks to the individual nodes in the cluster (Map), then organizes them and reduces the results provided by each node into a single consistent response to a query (Reduce). This is made possible by its distributed file system (HDFS).

Apache Spark adopts the MapReduce model and has several advantages over other big data and MapReduce technologies such as Hadoop and Storm. First of all, Spark offers a complete and unified framework to meet big data processing needs for various datasets, diverse in nature (text, graph, etc.) as well as source type (batch or real-time stream). Spark uses the concept of Resilient Distributed Datasets (RDDs) which is the immutable distributed collection of objects. Each dataset in RDD is divided into logical partitions, which may be computed on different nodes of the cluster. Then, Spark allows applications on Hadoop clusters to be executed up to 100 times faster in memory, 10 times faster on disk [52]. It follows the master-worker architecture, for every Spark application it will create one master process and multiple workers (Fig. 2). Spark streaming is a module supports scalable, fault-tolerant processing of live data streams, it has much higher latency, while it provides higher throughput against the most popular frameworks such as Storm and Flink [43]. The ingestion of data supported by Spark streaming can be from many sources like Twitter, Apache Kafka, Tcp Sockets. Incoming data stream is grouped into batches of interval less than a second and processed by the batch processing Spark engine, it can be processed using machine algorithms with high level function such as Map and Reduce. Finally, processed data may be pushed out to databases, file systems and live dashboards for visualization and historical data analysis. Spark streaming provides a high-level abstraction called discretized stream or DStream [57], which represents a continuous stream of data. Internally, a DStream is represented as a sequence of RDDs where each RDD in the sequence is considered as micro batch of input data.

In addition to other Spark API libraries, Spark provides another major library called Spark MLlib [41], which is a toolkit of distributed machine learning and data mining models

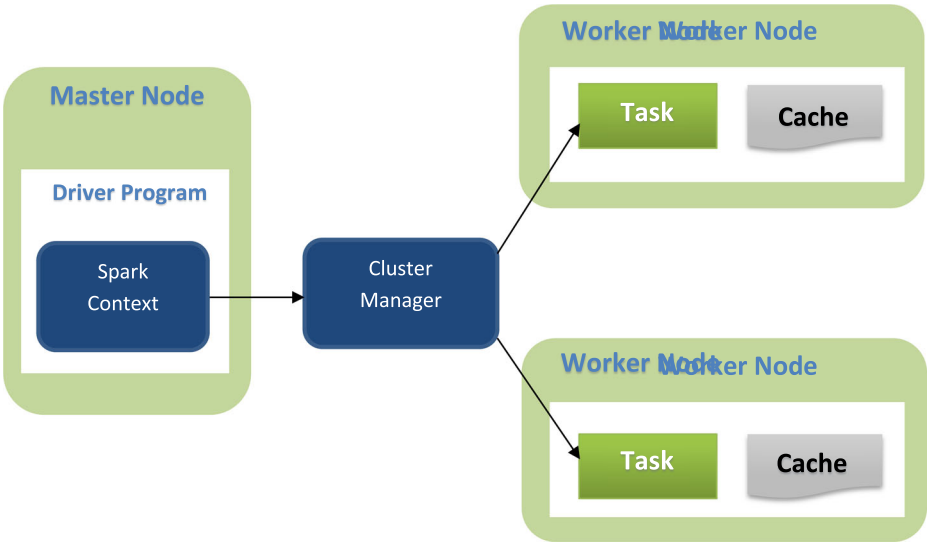

Fig. 2 Spark architecture

that is under heavy development and already contains scalable, high-quality and efficient algorithms for many common machine learning tasks.

Spark SQL [11] is the most popular and prominent feature of Apache Spark. It provides support for interacting and manipulating data with Spark via SQL statements within a Spark program. It represents database tables as Spark RDDs and translates SQL queries into Spark operations to perform parallel queries.

Reasons to choose Spark There are several reasons for choosing Spark, but three are key :

- Spark uses the concept of RDD, which allows us to store data in memory and store it as required. This greatly increases the performance of batch jobs up to 10 to 100 times faster than traditional MapReduce.

- Spark also enables us to cache the data in memory, which is more beneficial in case of iterative algorithms, in which the computation requires multiple passes over the data such as those used in machine learning.

- Particularly, MapReduce is ineffective in case of multi-pass applications that require real-time computing, and low-latency processing with massively parallel processing architectures.

3.3 Use case dataset

In this study, the processed.cleveland.data of heart disease database [28] was used for training and testing the machine learning algorithm which predicts heart disease. It was used in many machine learning research works. For each disease observation, we have constructed a labelled dataset with attributes, where class label attribute labelled with two classes, present presence or absence of heart disease. The class label attribute values modified to just 0 and 1,

where value 1 indicates presence of heart disease replacing values 1, 2, 3 and 4 while value 0 indicates absence of heart disease, turning it to a binary class dataset. The remaining 13 features are described in Table 1 while Table 2 shows the sample heart disease dataset. The databases have 76 raw attributes, only 14 of them are actually used, 139 (45.87%) records present presence of heart disease while 164 (54.13%) present absence of heart disease. Rows (6 rows) with missing values are removed.

3.4 Parallel and scalable random forest based on Spark

The prediction of health status coming from Twitter streams needs to build a classification model, which is capable to classify the attributes of each stream in presence or absence of heart disease. There are many methods for classification in multivariate approach, including discriminate analysis, artificial networks, and regression models, especially logistic regression and fuzzy logistic regression [29, 44].

Decision Tree (DT) [13] is one of the very important tools in data mining, it can even process a large set of data. It is a powerful method for pattern categorizations that is widely used in the fields of medicine, science and technology. They are relatively easy to understand and interpret, it can handle categorical and numerical features and do not require input data to be scaled or standardized. One of the important advantages of DT which separates it from other algorithms is the structural information, reliable and simple variable selection tool for clinical practice. The DT algorithm is a top-down approach that begins at a root node, and then selects a feature at each step that gives the best split of the dataset based on information gain of this split computed from the node impurity. For classification tasks, there is a measure that can be used to select the best split such as Gini and Entropy impurity given by formula (1) and (2) respectively:

$$Gini = \sum_{i=1}^{C} f_i(1 - f_i) \tag{1}$$

Table 1 Heart disease dataset attributes description

No	Attributes	Description
1	Age	Age in years
2	Sex	Sex(1 = male, 0 = female)
3	Cp	Chest pain type
4	Restbpss	Resting blood pressure
5	Chol	Serum Cholestoral
6	Fbs	Fasting blood sugar
7	Restecg	Resting electrocardiographic results
8	Thalach	Maximum heart rate
9	Exang	Exercice induced angina
10	Oldpeak	ST depression induced by exercise relative to rest
11	Slope	Slope of peak exercise ST segment
12	Ca	Number of major vessels colored with fluoroscopy
13	Thal	3(normal), 6(fixed defect), 7(reversable defect)
14	Num	Class(1 = presence of heart disease, 0 = absence of heart disease)

Table 2 Sample heart disease dataset (14 attributes)

Attributes													
Age	Sex	Cp	Restbpss	chol	Fbs	Restecg	Thalach	Exang	Oldpeak	Slope	Ca	Thal	Num
63.0	1.0	1.0	145.0	233.0	1.0	2.0	150.0	0.0	2.3	3.0	0.0	6.0	0
67.0	1.0	4.0	160.0	286.0	0.0	2.0	108.0	1.0	1.5	2.0	3.0	3.0	1
67.0	1.0	4.0	120.0	229.0	0.0	2.0	129.0	1.0	2.6	2.0	2.0	7.0	1
37.0	1.0	3.0	130.0	250.0	0.0	0.0	187.0	0.0	3.5	3.0	0.0	3.0	0
41.0	0.0	2.0	130.0	204.0	0.0	2.0	172.0	0.0	1.4	1.0	0.0	3.0	0

$$Entropy = -\sum_{j=1}^{C} p(s, j) * \log p(s, j) \tag{2}$$

where f_i is the frequency of label i at a node and c is the number of unique labels and $p(s, j)$ is the proportion of instances in s that are assigned to j-th class [6].

RF [30] is an ensemble learning method and one of the most successful and powerful supervised machine learning algorithm that is capable of performing both classification and regression tasks. As the name suggests, RF builds the forest from several DTs. Each tree gives a vote to classify a new element, the class that receives more votes is chosen by RF.

Let S a dataset of samples formalized as:

$$S = \{(x_i, y_j), i = 1, 2, ..., N; j = 1, 2, ..., M\}$$

, where x is a sample and y is a feature variable of S. Namely, the original training dataset contains N samples, and there are M feature variables in each sample. The steps involved in the construction of the RF algorithm are as follows:

Step 1. Create a bootstrapped dataset
In this step, to create a bootstrapped dataset, k subsets are randomly selected from the original dataset S. Finally, k training sub-sets are constructed as a collection of training subsets.

Step 2. DT construction
In this step, each DT is created by using bootstrap dataset, but only uses a random subset of variables (or columns) at each step based on gain ratio discussed previously. The main process of the RF algorithm construction is presented in Fig. 3.

In order to improve the performance of the simple RF algorithm and reduce the data handling cost of large-scale data in a parallel and distributed environment, we propose a Spark Parallel Random Forest (SPRF) algorithm. The SPRF algorithm is optimized using a hybrid parallel approach that combines data parallel optimization and task parallelization. From the point of view of data parallel optimization, a vertical data partitioning method is performed. This method reduce the amount of data and the number of data transmission operations in the distributed environment without reducing the accuracy of the algorithm. From the point of view of tasks parallel optimization, a parallel approach is performed in the learning process of the SPRF algorithm and the Directed Acyclic Graph (DAG) is performed on data based on the dependency of the RDD objects. Then, different task planners are invoked to perform the tasks of the DAG. The parallel training approach maximizes the

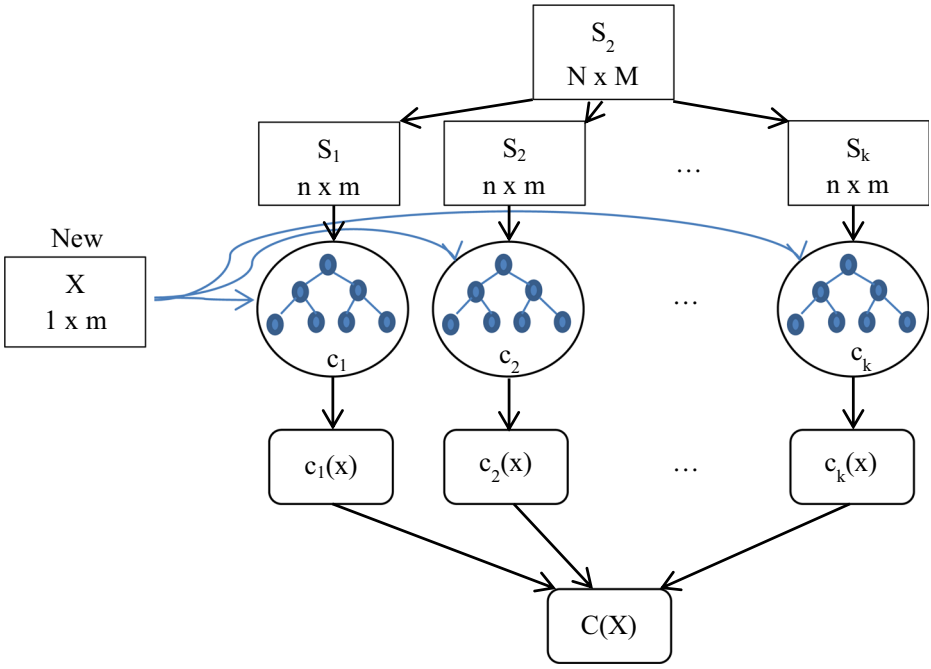

Fig. 3 Process of the RF construction

parallelization of SPRF and improves the performance of SPRF. Then, the task planners further minimize the cost of data communication between Spark cluster and achieve a better balance of workload and fast computation. Hence, an adequate and parallel model for predicting health status in big data context using Spark is needed. Based on this, a RF model adaptation is more important. In this work the Spark streaming handles the Kafka topic data streams using Spark streaming library while the parallelization of RF is performed using Spark MLlib.

Algorithm 1 represents the steps to train and test the RF on Spark based distributed environment. Figure 4 shows the flowchart of the RF model based Spark.

In this part, there are two phase real-time health status prediction, first involves analysis on healthcare dataset to build the machine learning model. The second uses the model in production to make predictions on live health data streams. The working process is given in Fig. 5.

3.4.1 Features selection

When developing a machine learning model, only a few variables in the dataset are useful for building the model, and the other features are either redundant or irrelevant. If we introduce all these redundant and irrelevant features into the dataset, it can have a negative impact and reduce the overall performance and model accuracy. Therefore, it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less important features, which is done using feature selection in machine learning. There are many feature selection techniques, for that purpose, a comparative study

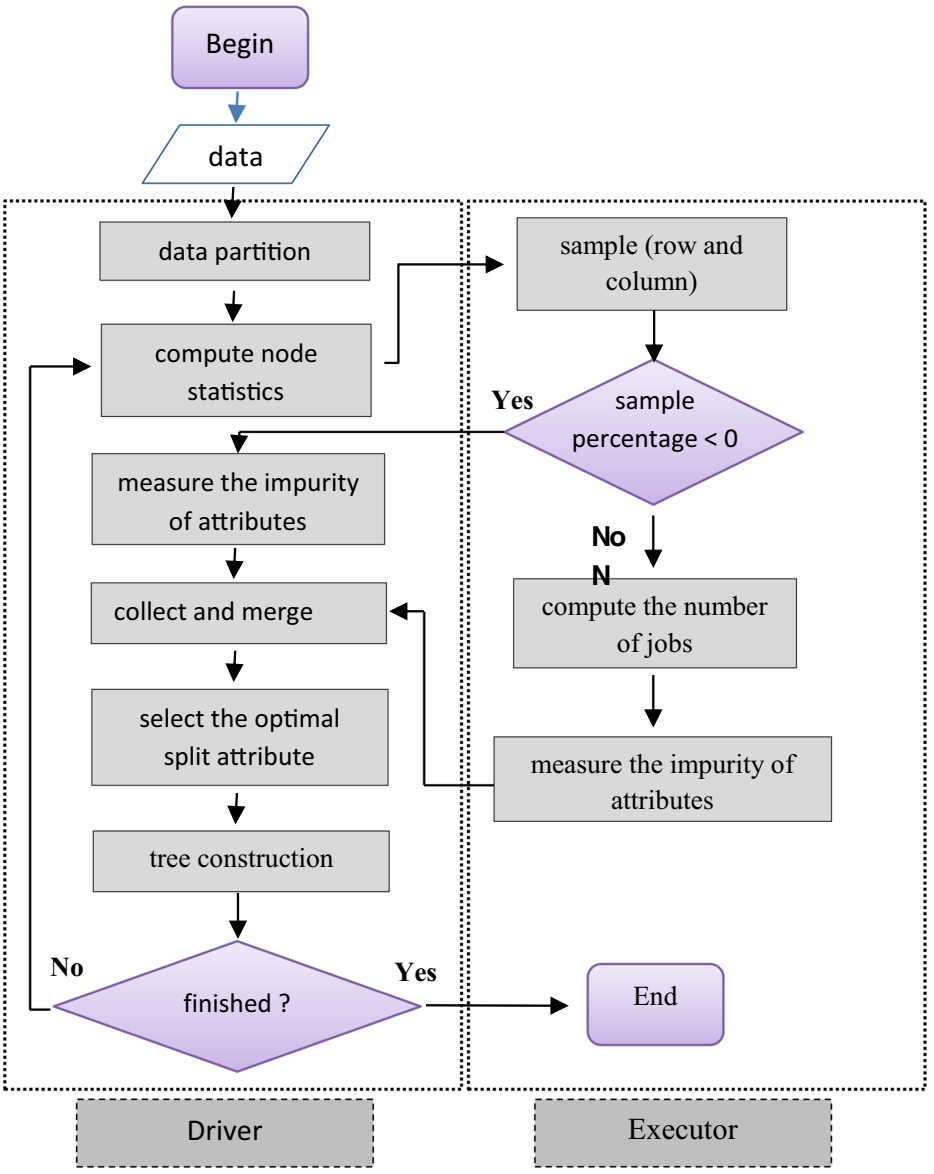

Fig. 4 Flowchart of RF based on Spark

in terms of accuracy has been made for well know techniques such as Relief, Correlation-based feature selection, Chi squared, Filtered subset, Info gain, Gain ratio, One attribute based, Consistency subset, Filtered attribute, Genetic Algorithm (GA) and GA with SVM. We find that feature selection using Relief algorithm achieved the highest accuracy on heart disease dataset.

The Relief algorithm is an algorithm invented in 1992 by researchers Kenji Kira and Larry A. Rendell [33] whose goal is to distinguish between essential and non-essential

Step1: Start new SparkContext
Laoding required package and APIs
sparkContext(master,appName, sparkHome)
step2: Load and parse the dataset into an RDD
rowData(RDD) : sc.textFile(path)
Data(RDD) : Map(pareseFunction(rowData))
parse each input line in parallel
Step3: Split the data into training and test sets
Set related parameters
trainData(RDD), testData(RDD):
randomSplit(Data)
trainData.cache(): cache the trainData in
memory
testData.cache(): cache the testData in memory
train the model
step4: Test the model
LabelAndPredict(RDD) : Map(predictFunction(testData)) parse and
predict each input line in parallel
Save model : save(sc, path)

Algorithm 1 MLlib RF model on Spark.

features, which better takes into account these notions of interactions between different information. This algorithm relies in particular on the measurement of similarities and dissimilarities between input values and tested values, and thus makes it possible to estimate the relevance (or irrelevance) of the various features using a global score. To evaluate the relevance of a value, the algorithm will try to know how a feature evolves (or not) within a class and outside this class.

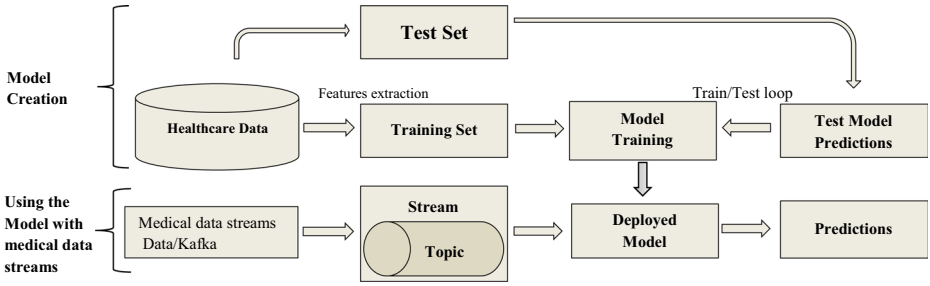

Fig. 5 Machine learning process

Table 3 Selected features

Feature	Ranking
Sex	0,103
Restecg	0,094
Ca	0,093
Thal	0,082
Exang	0,079
Cp	0,079
Slope	0,046
Oldpeak	0,027

3.4.2 Selected features

Relief method assigns for each feature a ranking score. All features are ranked in descending order based on their score. In this part, the important 8 features which have high rank were chosen. Table 3 presents the 8 important features and their ranking. We can notice that Sex has the highest rank at 0.103. The second is Restecg at 0.094 followed by Ca at 0.093, Thal at 0.082. Both Exang and Cp have similar score at 0.179 followed by Slope at 0.046 and Oldpeak at 0.027. At the bottom we find Thalach at 0.023, Age at 0.019, Restbps at 0.017, Fbs at 0.011 and finally Chol at 0.002. Figure 6 represents the ranking of all features. In order to choose the attributes that give the best accuracy, we test all the attributes with the different machine learning methods, and then we remove the attributes one by one starting with the attributes that have a low score. The best accuracy is reached in the case of 8 attributes (Table 3) with high ranking score.

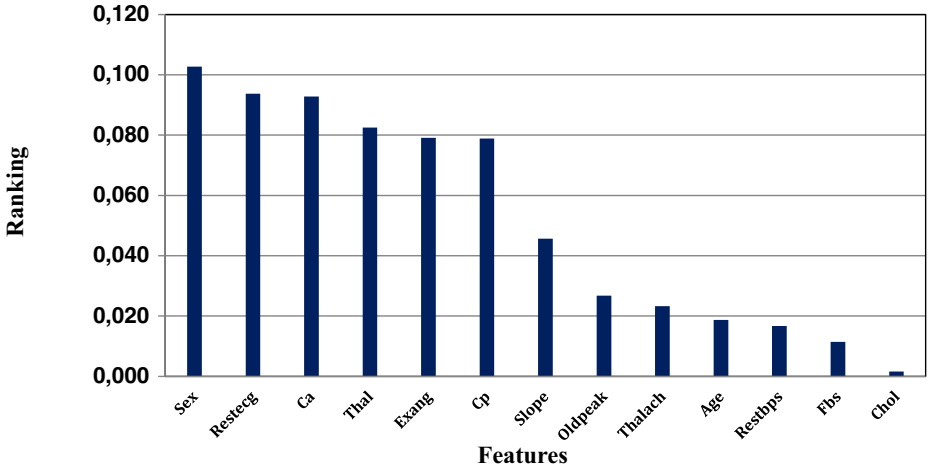

Fig. 6 The ranking of all features

3.5 Data storage and visualization

3.5.1 Distributed database

To get meaningful patterns [5] from big data generated by all users such as patient diagnostic information is also an essential problem, so the predicted results must be stored in a distributed way to ensure the data availability with no single point of failure.

Apache Cassandra [7] is a very powerful distributed database system, it is particularly effective at supporting large volumes of records across multiple servers. This database can be easily scaled to support a sudden increase in demand. It is sufficient to deploy a multinode clusters database with Cassandra. In addition, Cassandra is highly available and has the advantage of not having a single point of failure [36]. Also Cassandra provides extremely fast write and read speeds with Spark [25].

3.5.2 Visualization

Data visualization has become an integral part discipline, necessary to visualize the multitude of data to which we have access and to communicate the most relevant information. Secondly, data visualization seems to mainly willingness to respond in the professional environment to quickly access information. This visualization will allow among other to explain and highlight key information that will be exploited from databases [5]. Data visualization is the graphical representation of information, it lies at the intersection of the fields of communication, information science and design. The main advantage of data visualization is not makes data more beautiful but it provides insight complex data sets by communicating their key aspects, it allows decision makers to see analytics presented visually which plays a key role in decision-making.

Storage and visualization of electronic health records can help in identifying the patterns for disease prediction system, assist healthcare providers to find an accurate and responsive to the patient needs, make better financial and healthcare decisions based on result predictions made by the system [2]. Also, with the availability of data records we can extract the relationship between each attribute (dependent variable) and the outcome (desired attribute). Furthermore, clustering of similar patients records can help doctors to save time when diagnosing diseases. In fact, each group of patients can undergo the same type of treatment. Finally, we can get useful information about diseases based on data statistics.

Here the data visualization is performed using Zeppelin [10] which is a web based and multipurpose notebook that enables interactive data analytics, it is an open source data analysis environment that runs on top of Apache Spark. Using concept of dataframe with Spark SQL, the database can be queried by different queries like, number of instance, number of different cases and making more sophisticated and high-level data analytics. The description for data querying using Spark SQL is as follows :

- Spark SQL, SQL Context is imported.
- Load the data from Cassandra table to construct RDD, and execute the action commands.
- Call Map transformation and Map the RDD into heart disease case class.
- Convert the case class to dataframe and create temporal table.
- Define the executed using Spark SQL query.

In this step we studied the retrieval time of following queries (Table 4) with different size of records and multinode cluster.

Table 4 Liste of queries

No	Queries
1	Total number of patients(records)
2	Count total number of patient which have disease
3	Groupe patient by age
4	Get the last record

4 Results and discussion

4.1 Implementation

The proposed system was written using Scala programming language and Zeppelin as a development platform which support many interpreters like Scala, Spark and Cassandra. Algorithm 2 describes the main steps to implement our system.

Step 1 : Twitter authentification
We set up the authentication credentials of Twitter using the Twitter4J library: consumerKey, consumerSecret, accessToken and accessTokenSecret
Step 2 : Spark context
Create an instance of SparkContext(sc by default in Zeppelin notebook) and StreamingContext to use all Spark streaming features
Step 3 : Get Twitter streams
createStream method of TwitterUtils was used to get DStreams, TwitterUtils uses Twitter4J to get Twitter status streams
Step 4: Kafka producer
Create a Kafka producer which will receives Twitter data streams
filter tweets based on certain keyword
Gets the tweets username and publishes to a specified Kafka topic using send method
Step 5: Get Kafka streams
Create the direct stream with the Kafka parameters and topic using createDirectStream method of KafkaUtils
Step 6: Data processing
Extract username and attributes from each tweet status using foreachRDD method
apply the machine learning model to predict health status
Send appropriate message based on Twitter username using sendDirectMessage method
Save all attributes and predicted label to Cassandra keyspace and table using saveToCassandra method
Step 7: Start the computation
Start Spark streaming context using start method

Algorithm 2 Processing steps of the proposed framework.

Firstly, the proposed system is carried out on single node cluster created on standalone machine with core i7 processors and 8GB RAM, having Ubuntu 16.04 operating system

through Spark platform which integrates MLlib especially RF model with Kafka streaming data handling.

Table 5 shows the characteristics of our master and worker nodes.

The application after establishing connection to the Twitter streaming through Kafka streaming as detailed in Fig. 1, is continuously receiving the data streams from multiple Kafka producers and once it encounters the health status check stream, it extracts the attribute values from the data events sent by Kafka streaming and apply RF model to predict health status. On the other hand, the details of each predicted instance were persisted in Cassandra database table for querying them later. Once the proposed system was successfully tested on single node cluster, a multinode cluster with one master and two workers is created.

4.2 Performance evaluation of machine learning models

The heart disease dataset has been randomly split into a training data set and a test data set with a split seed, 70% of the data is used to train the model, and 30% will be used for testing. The models have been trained using the training set with hyperparameter tuning, different models have been tested, the classification accuracy values are calculated in each case.

Accuracy, sensitivity, specificity, positive likelihood ratio (PLR), negative likelihood ratio (NLR), disease prevalence (DP), positive predicted value (PPV) and negative predicted value (NPV) are calculated for models evaluation. The validations metric are defined by:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP: True Positive, TN: True Negative, FP: False Positive, and FN: False Negative.

$$Sensitivity = \frac{TP}{TP + FN} \tag{4}$$

$$Specificity = \frac{TN}{FP + TN} \tag{5}$$

$$PositiveLiklihoodRation = \frac{Sensitivity}{100 - Specificity} \tag{6}$$

$$NegativeLiklihoodRation = \frac{100 - Sensitivity}{Specificity} \tag{7}$$

$$PositivePredictedValue = \frac{TP}{TP + FP} \tag{8}$$

$$NegativePredictedValue = \frac{TN}{TN + FN} \tag{9}$$

$$DiseasePrevalence = \frac{TP + FN}{TP + TN + FP + FN} \tag{10}$$

Table 5 Cluster nodes characteristics

Parameters	Master	Worker
Processor	Core i7	Core i3
Cores	4	4
Memory	8 GB	4 GB
Operating system	Ubuntu 16.04	Ubuntu 16.04

4.2.1 Accuracy using all features

Using all feature, the diagnosis accuracy is maintained at 87.50% in case of RF. Table 6 presents a comparative analysis of RF algorithm with relevant machine learning algorithms implemented by Spark MLlib such as Naive Bayes (NB), Support Vector Machine (SVM), Multilayer Perceptron (MLP), DT and Logistic Regression (LG).

Figure 7 represents a graphical visualization of accuracies which shows an important advantage for RF classifier among other well-known classifiers in terms of detection accuracy.

4.2.2 Accuracy using selected features

In the second test, we first used Relief-based feature selection to select the best features, and then, we used the same machine learning models as in the previous section. Experimental results showed that highest classification performances are achieved when RF is used as classifier. Table 7 shows the best parameters contributing to best accuracy.

At the best accuracy of 92.05% which minimizes maxBins, maxDepth, numTrees parameters with Gini impurity, the total test simple is equal to 88 and TP = 37 , TN = 44 , FP = 2 , FN = 5.

Table 8 shows a comparison analysis between sex classifier: NB, SVM, DT, MLP, LG and RF. RF achieved the highest accuracy at 92.05%.

Figure 8 shows a graphical visualization of accuracies which shows an important advantage for RF classifier among other well-known classifiers in terms of detection accuracy.

Figure 9 presents a graphical visualization of accuracies using and without using features selection. It can be observed that the accuracy has been improved especially with RF classifier.

Based on the results obtained, the proposed approach to classify heart diseases not only provides the best accuracy compared to previous experimental results with a minimal set of features, but also shows the feature reduction in a simple way leading to time savings during the training phase, especially in the context of big data which has become a great

Table 6 Performance evaluation metrics with other models (MLlib) using all features

Models	NB	SVM	MLP	DT	LG	RF
TP	41	40	36	41	40	39
TN	33	35	35	32	36	38
FP	5	5	10	5	6	5
FN	9	8	7	10	6	6
Sensitivity(%)	82,00	83,33	83,72	80,39	86,96	86,67
Specificity(%)	86,84	87,50	77,78	86,49	85,71	88,37
PLR	6,23	6,67	3,77	5,95	6,09	7,45
NLR	0,21	0,19	0,21	0,23	0,15	0,15
PPV(%)	89,13	88,89	78,26	89,13	86,96	88,64
NPV(%)	78,57	81,40	83,33	76,19	85,71	86,36
DP(%)	56,82	54,55	48,86	57,95	52,27	51,14
Accuracy(%)	84.09	85.23	80.68	82.95	86.36	**87.50**

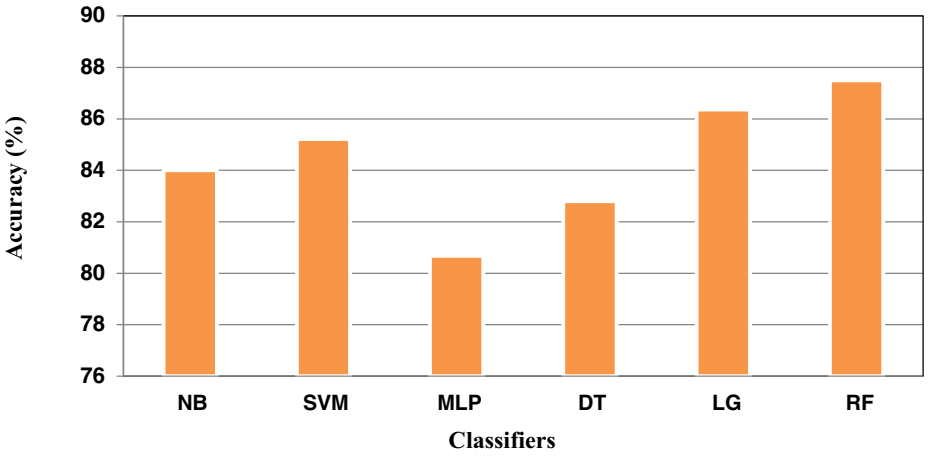

Fig. 7 Accuracies comparison using all features

Table 7 The best parameters contributing to best accuracy

Model	Parameters
NB	smoothing: 2 , modelType: multinomial
SVM	regParam: 0.05 maxIter: 25
MLP	maxIter: 100, layers: [8,6, 8,6, 8, 2] , blockSize: 128
DT	maxDepth: 4 , maxBins: 30 , impurity: gini
LG	regParam: 0.02 , maxIter: 20
RF	maxDepth: 4 , maxBins: 10 , numTrees: 10

Table 8 Performance evaluation metrics with other classifiers (MLlib) using selected features

Models	NB	SVM	MLP	DT	LG	RF
TP	32	37	35	29	37	37
TN	39	41	42	41	41	44
FP	6	4	3	4	4	2
FN	11	6	8	14	6	5
Sensitivity(%)	74,42	86,05	81,40	67,44	86,05	88,10
Specificity(%)	86,67	91,11	93,33	91,11	91,11	95,65
PLR	5,58	9,68	12,21	7,59	9,68	20,26
NLR	0,30	0,15	0,20	0,36	0,15	0,12
PPV(%)	84,21	90,24	92,11	87,88	90,24	94,87
NPV(%)	78,00	87,23	84,00	74,55	87,23	89,80
DP(%)	48,86	48,86	48,86	48,86	48,86	47,73
Accuracy(%)	**80,68**	**88,64**	**87,50**	**79,55**	**88,64**	**92,05**

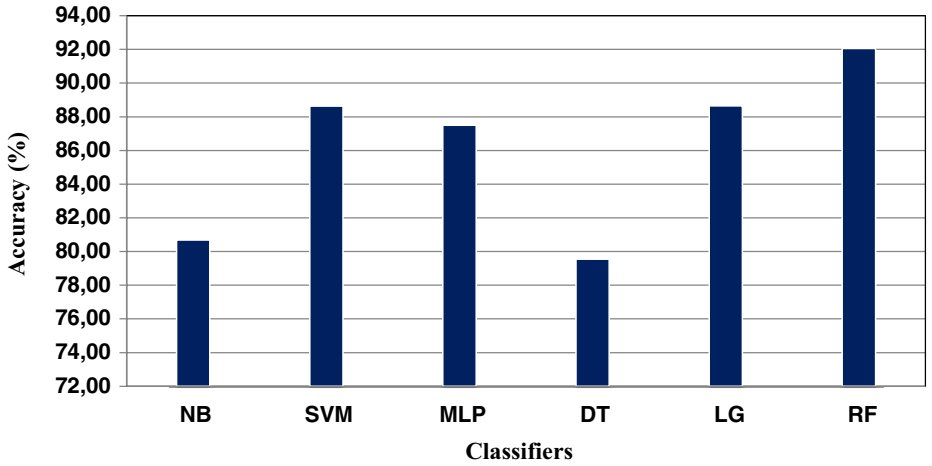

Fig. 8 Accuracies comparison using selected features

challenge. On the other hand, physicians can also benefit from the representative features of heart disease.

4.2.3 Spark based RF scalability

In order to show the scalability of our approach and to have sufficient database size, we generate other data records, the original dataset has been enriched up to 4 millions of rows. The performance of RF was performed in Spark and on the open source machine learning framework Weka [55] (Waikato environment for knowledge analysis) classifier. The result is showing in Figs. 10 and 11. It can be observed that Spark based RF takes less time to build and test the model. Moreover, time taken to build and test the model is less when using 8 selected features.

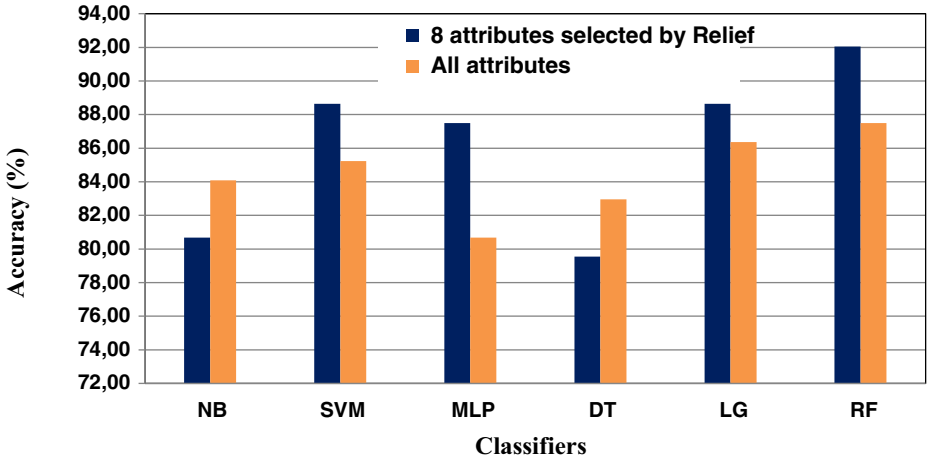

Fig. 9 Accuracies comparison using and without using features selection

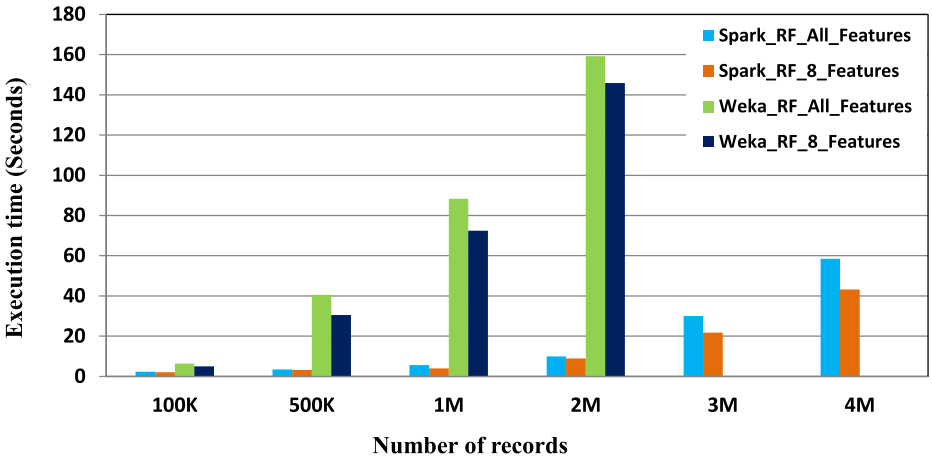

Fig. 10 Execution time of RF using Spark MLlib and Weka, time taken to build model

The parallel RF algorithm of Spark Mllib reaches best scalability owing to the distributed computing on cluster nodes and in-memory computation.

Figures 12, 13, 14 and 15 illustrate the execution time of Spark-based RF algorithm with a different number of nodes.

4.3 Throughput

Initially, the heart disease dataset has been processed in order to construct a labelled dataset. RF model was built and tested separately by varying parameters such as maxDepht, maxBins and numTrees, the minimum model error is taken into account based on the model classification accuracy (Table 8). An offline model with selected features has been created and saved in order to use it in real-time as Fig. 5 shows. In our case study, for testing the purpose, simulator applications acting as Kafka data producers were created, data

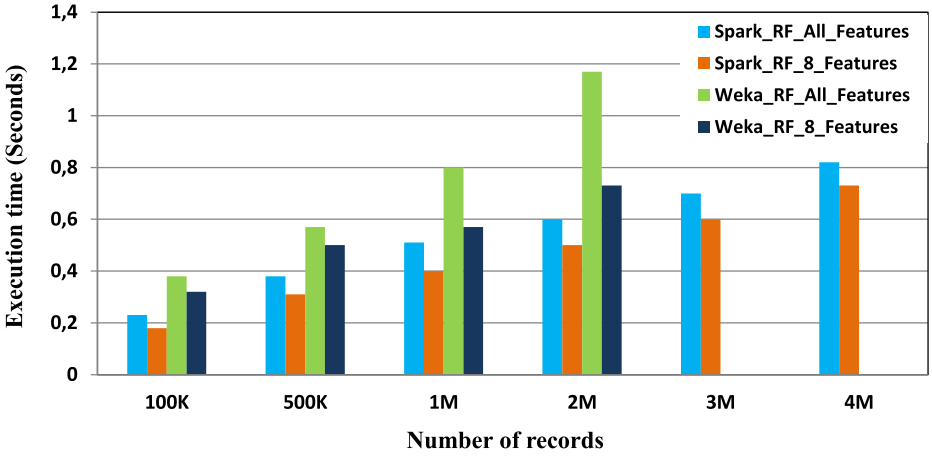

Fig. 11 Execution time of RF using Spark MLlib and Weka, time taken to test model

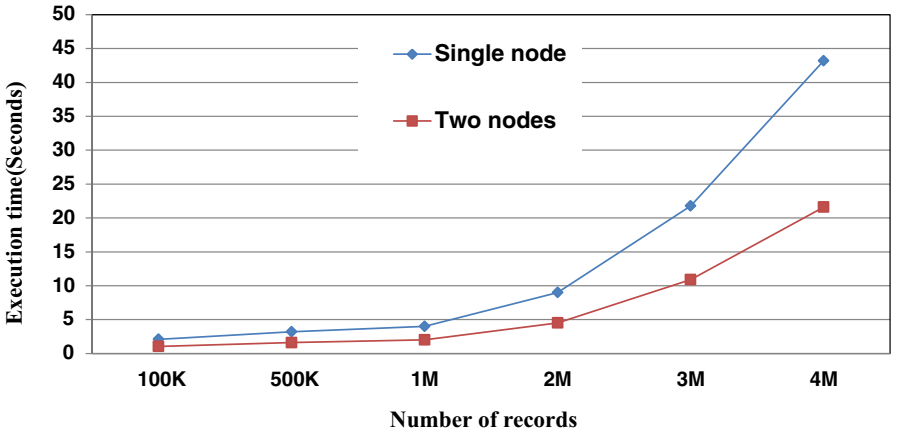

Fig. 12 Execution time of RF using Spark MLlib for different nodes using selected features, time taken to build model

producers consist of two simulator applications for heart disease streams. We are using two producers. Each one sends approximately 280200 (can be more) events per second per node in predefined format which has preceded by a keyword as a header word and which in turn serves them to consumers. We conducted three scenarios with a stream interval of 1, 2 and 3 seconds. Table 9 presents the conducted experiments.

Figure 16 presents the overall performance evaluation of the proposed system.

4.4 Query throughput

The performance of the retrieval time of records is mainly calculated with respect to a specific size of data (3 million of records). Figure 17 shows the query time during each query between traditional relational database management system (RDBMS) like MySQL

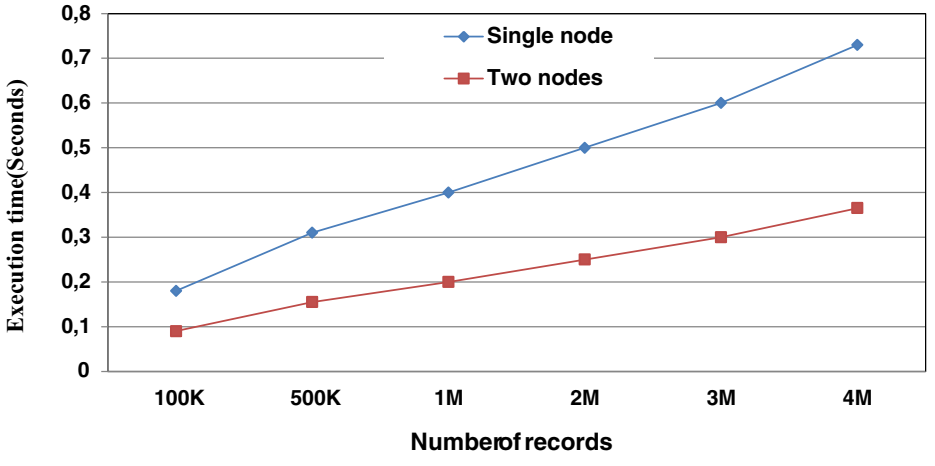

Fig. 13 Execution time of RF using Spark MLlib for different nodes using selected features, time taken to test model

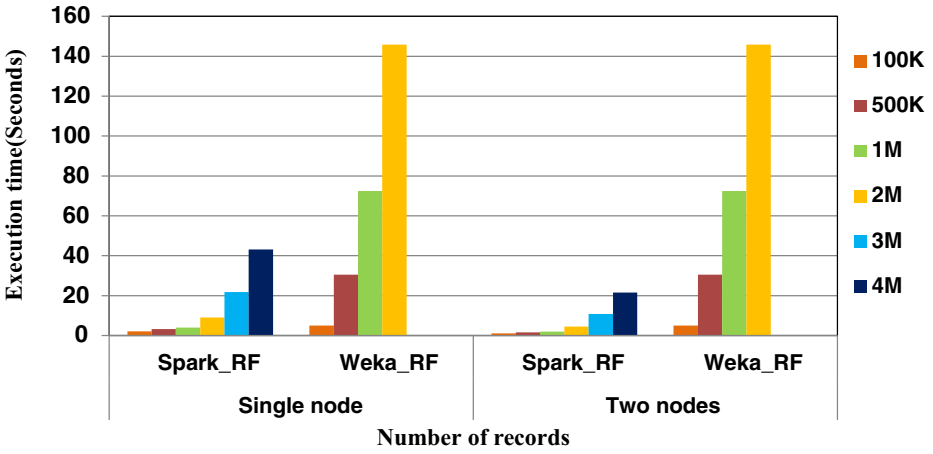

Fig. 14 Execution time of RF using Spark MLlib and Weka for different nodes using selected features, time taken to build model

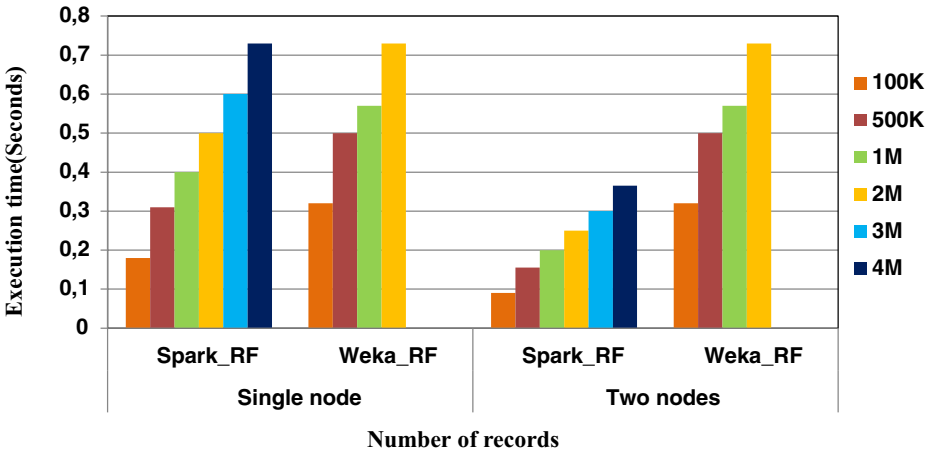

Fig. 15 Execution time of RF using Spark MLlib and Weka for different nodes using selected features, time taken to test model

Table 9 Conducted experiments

No of nodes	Events/second	Kafka	Topic partitions	Spark workers	Cassandra
1	~ 560 500	1	1	1	1
2	~ 1 121 000	2	2	2	2

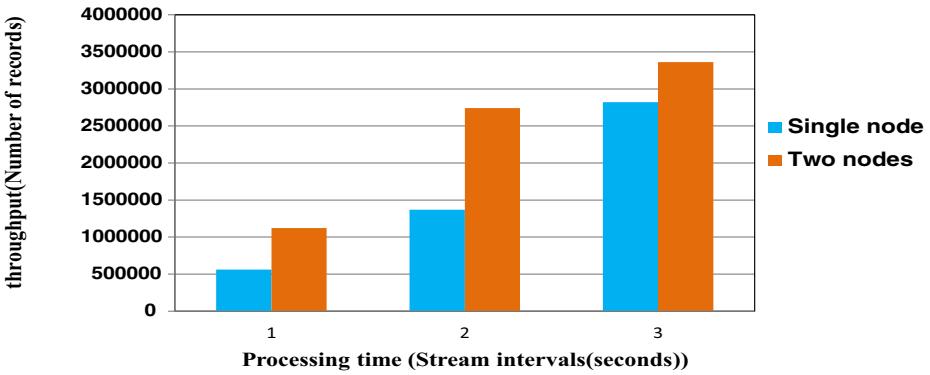

Fig. 16 Processing time with respect to increase in data size

and Spark SQL. For heart disease dataset, during each query, the query time increases significantly for traditional RDBMS rather than Spark SQL.

The figure above shows that Spark speed up the execution time than traditional RDBMS such as MySQL. In this context, Spark can help speed up slow report requests and add more scalability to long running queries.

Using Zeppelin a real-time data dashboard has been created which will retrieve data from Cassandra database and displays it in charts and tables. This application uses Spark SQL and Angularjs to push the data to the web page in fixed intervals, so data will be refreshed automatically.

After testing the system using a simulator application, three Twitter accounts were created. Selected features related to heart disease were tweeted in predefined format preceded by a keyword from Twitter's status. All these tweeted attributes as well as the user name will be captured and filtered by Kafka streaming in real-time, which in turn serves them to Spark streaming as a Kafka producer. At Spark streaming, attribute values and user name were extracted. RF was applied on extracted attributes to predict health status and send appropriate message indicate absence or presence of disease. The result will be saved into

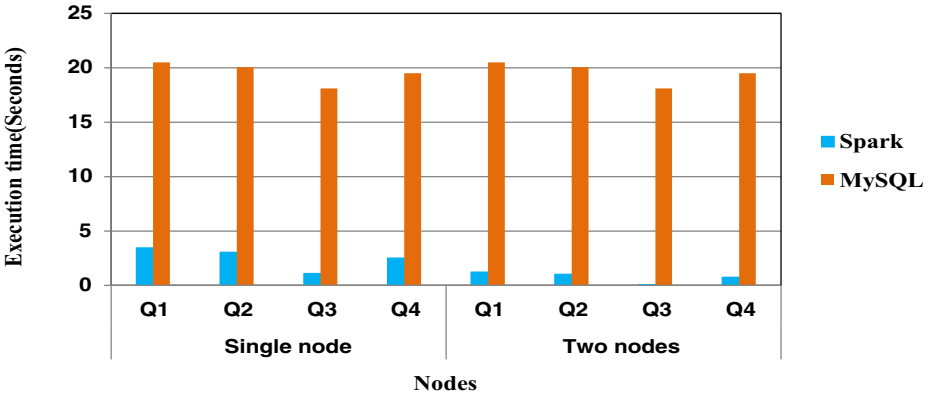

Fig. 17 Query execution times comparison between Spark SQL and MySQL

Table 10 Comparison of different systems with the proposed system

Ref	Features selection	Disease	Big data technology	Real-time analysis	Observations
Sreejith et al. [49]	No	Heart disease	No	No	−Not tested with big data −Big data prediction and data storage is still the main challenge
Basheer et al. [12]	No	Heart disease	No	No	−Not tested with big data −Big data prediction and data storage is still the main challenge
Sampath et al. [48]	No	Diabetic	Yes	No	−Make prediction after the data storage step −Real-time prediction is not performed −Not suitable for real-time
Han et al. [24]	No	Cardiovascular	Yes	No	−Make Prediction after the data storage step −Streaming big data is not performed −Not suitable for real-time
Kumar and Gandhi [35]	No	Heart disease	Yes	No	−Make prediction after the data storage step −Real-time prediction is not performed −Not suitable for real-time
Manogaran et al. [40]	No	Heart disease	Yes	No	−Real-time data analysis is not performed −Make prediction after the data storage step −Not suitable for real-time
Venkatesh et al. [53]	No	Heart disease	Yes	No	−Spark streaming and data storage is not covered −Not suitable for real-time

Table 10 (continued)

Ref	Features selection	Disease	Big data technology	Real-time analysis	Observations
Ed-daoudy and Maalmi [21]	No	Heart disease −Accuracy: **82.40%**	Yes	Yes	−Interaction between system and users is not covered −More general architecture −Classification accuracy is not enough −Stream data management is not covered
Ed-daoudy and Maalmi [20]	No	Heart disease and diabetes −Accuracy: **82.40%**	Yes	Yes	−Interaction between system and users is not covered −More general architecture −Classification accuracy is not enough
Our	Yes	Heart disease −Accuracy: **92.05%**	Yes	Yes	−Real-time big data analysis is the main goal −Big data stream management is covered −Make prediction before the data storage −Real-time machine learning is performed −Data storage and visualization is covered −User interaction is considered and it is the main objective −Suitable for real-time

The proposed system is able to collect, process, analyze and store health status data in real-time. It focuses on applying distributed and real-time machine learning on streaming health data events using big data technologies, namely, Apache Spark instead of traditional frameworks which become limited for real-time computing. Using Apache Kafka as big data stream management, our system can predicts different type of diseases at the same time based on the concept of Kafka topic and multiple machine learning models built on the appropriate dataset. On the other hand, online, distributed and fast prediction model is used to predict health status from user's streaming tweets to make the system scalable. With a simple tweet, Twitter users receive instant information about their health status. The system offers users real-time remote monitoring at no extra cost.

Cassandra database for historical data analysis and visualization. The connection and interaction between the system and the users requires investment of a costly set of infrastructure in term of programming skills, time and money. The use of Twitter as a free channel greatly simplified the communication.

Table 10 presents a comparison of state-of-the-art architectures for health status prediction with the proposed system. Based on the findings, the proposed system can be applied to solve the real-time big data analysis jobs for medical IoT.

5 Conclusion

In this paper, we have presented a real-time system for health status prediction in big data context based on Apache Spark and Apache Kafka. This system is designed to collect, filter, analyze, store and visualize streams of health status data. Firstly, an offline machine learning model has been developed, in this part, to achieve high accuracy, we select important features using Relief algorithm. Using full and selected features, different type of machine learning algorithms implemented by Spark MLlib, namely, RF, DT, SVM, NB, MLP and LG were applied on heart disease dataset from the UCI repository. RF with selected features achieves highest accuracy, it was deployed to our system as an online machine learning model. With Twitter serving as free communication channel, Twitter users tweet they attributes related to heart disease. Kafka streaming receives all desired tweets attributes, serves them to Spark streaming in which RF model is applied on data streams to predict health status and send appropriate message. The streamed data is stored into distributed database Cassandra. The result will be displayed on dashboard and making more sophisticated and high-level data analytics.

Developing a real-time data prediction and analytics system using traditional analytics tools requires a variety of skills, intensive and more expensive programs and considerable amount of time and money. However, using traditional data processing platforms and techniques become difficult to process the enormous generate data. But using open source big data tools especially Spark, significantly improved the performance and the effectiveness of the analytics system, especially in terms of system development time and complexity of programs, execution and the speed it provide. In future works, we aim to integrate other data sources such as mobile application, IoT to our system and other classification models to improve the prediction accuracy.

Data Availability The dataset analysed during the current study is available in https://archive.ics.uci.edu/ml/datasets/heart+disease

Declarations

Ethics approval and consent to participate This article does not contain any studies with human participants or animals performed by any of the authors

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Abbasi A, Adjeroh D, Dredze M, Paul MJ, Zahedi FM, Zhao H, Walia N, Jain H, Sanvanson P, Shaker R et al (2014) Social media analytics for smart health. IEEE Intell Syst 29(2):60–80

2. Acharjya DP, Ahmed K (2016) A survey on big data analytics: challenges, open research issues and tools. Int J Adv Comput Sci Appl 7(2):511–518
3. Agrawal R, Imieliński T, Swami A (1993) Mining association rules between sets of items in large databases. In: Acm Sigmod Record, vol 22. ACM, pp 207–216
4. Al Rasyid MUH, Yuwono W, Al Muharom S, Alasiry AH (2016) Building platform application big sensor data for e-health wireless body area network. In: Electronics symposium (IES), 2016 international. IEEE, pp 409–413
5. Ali SM, Gupta N, Nayak GK, Lenka RK (2016) Big data visualization: tools and challenges. In: 2016 2nd International conference on contemporary computing and informatics (IC3I). IEEE, pp 656–660
6. Apache Spark documentation: official webpage of Apache kafka (2017) http://spark.apache.org//. Online; Accessed 15 Dec 2017
7. Apache cassandra: official webpage of Apache cassandra (2017) http://cassandra.apache.org. Online; Accessed 15 Dec 2017
8. Apache kafka: official webpage of Apache kafka (2017) https://kafka.apache.org/. Online; Accessed 15 Dec 2017
9. Apache spark: official webpage of Apache spark (2017) http://spark.apache.org/ Online; Accessed 15 Dec 2017
10. Apache zeppelin: official webpage of Apache zeppelin (2017) https://zeppelin.apache.org. Online; Accessed 15 Dec 2017
11. Armbrust M, Xin RS, Lian C, Huai Y, Liu D, Bradley JK, Meng X, Kaftan T, Franklin MJ, Ghodsi A et al (2015) Spark sql: relational data processing in spark. In: Proceedings of the 2015 ACM SIGMOD international conference on management of data, pp 1383–1394
12. Basheer S, Alluhaidan AS, Bivi MA (2021) Real-time monitoring system for early prediction of heart disease using internet of things. Soft Comput 25(18):12145–12158
13. Breiman L (2017) Classification and regression trees. Routledge, Evanston
14. Chen H, Chiang RH, Storey VC (2012) Business intelligence and analytics: from big data to big impact. MIS Q:1165–1188
15. Chen M, Hao Y, Hwang K, Wang L, Wang L (2017) Disease prediction by machine learning over big data from healthcare communities. IEEE Access 5:8869–8879
16. Condie T, Mineiro P, Polyzotis N, Weimer M (2013) Machine learning on big data. In: Data engineering (ICDE), 2013 IEEE 29th international conference on. IEEE, pp 1242–1244
17. Dean J, Ghemawat S (2008) Mapreduce: simplified data processing on large clusters. Commun ACM 51(1):107–113
18. Ed-daoudy A, Maalmi K (2018) Application of machine learning model on streaming health data event in real-time to predict health status using spark. In: 2018 international symposium on advanced electrical and communication technologies (ISAECT). IEEE, pp 1–4
19. Ed-Daoudy A, Maalmi K (2019) Real-time machine learning for early detection of heart disease using big data approach. In: 2019 international conference on wireless technologies, embedded and intelligent systems (WITS). IEEE, pp 1–5
20. Ed-daoudy A, Maalmi K (2019) A new internet of things architecture for real-time prediction of various diseases using machine learning on big data environment. J Big Data 6(1):104
21. Ed-daoudy A, Maalmi K (2020) Real-time heart disease detection and monitoring system based on fast machine learning using spark. Health and Technol 10(5):1145–1154
22. Fayyad U, Piatetsky-Shapiro G, Smyth P (1996) From data mining to knowledge discovery: an overview. In: Advances in knowledge discovery and data mining. AAAI. MIT Press, pp 1–34
23. Gao D, Li W, Cai X, Zhang R, Ouyang Y (2014) Sequential summarization: a full view of twitter trending topics. IEEE/ACM Transactions on Audio. Speech Lang Process (TASLP) 22(2):293–302
24. Han S, Kim K, Cha E, Kim K, Shon H (2017) System framework for cardiovascular disease prediction based on big data technology. Symmetry 9(12):293
25. Hassan M, Bansal SK (2018) Semantic data querying over nosql databases with apache spark. In: 2018 IEEE international conference on information reuse and integration (IRI). IEEE, pp 364–371
26. Hazarika AV, Ram GJSR, Jain E (2017) Performance comparision of hadoop and spark engine. In: I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC), 2017 international conference on. IEEE, pp 671–674
27. Hazarika AV, Ram GJSR, Jain E (2017) Performance comparision of hadoop and spark engine. In: I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC), 2017 international conference on. IEEE, pp 671–674
28. Heart disease: UCI (2020) archive.ics.uci.edu/ml/datasets/heart+disease. Online; Accessed 15 Dec 2017
29. Heydari ST, Ayatollahi SMT, Zare N (2012) Comparison of artificial neural networks with logistic regression for detection of obesity. J Med Syst 36(4):2449–2454

30. Ho TK (1995) Random decision forests (rdf). In: Proceedings of the 3rd international conference on document analysis and recognition, pp 278–282
31. Ismail A, Shehab A, El-Henawy I (2019) Healthcare analysis in smart big data analytics: reviews, challenges and recommendations. In: Security in smart cities: models, applications, and challenges. Springer, pp 27–45
32. Kavakiotis I, Tsave O, Salifoglou A, Maglaveras N, Vlahavas I, Chouvarda I (2017) Machine learning and data mining methods in diabetes research. Comput Struct Biotechnol J 15:104–116
33. Kira K, Rendell LA et al (1992) The feature selection problem: traditional methods and a new algorithm. In: Aaai, vol 2 pp 129–134
34. Kolajo T, Daramola O, Adebiyi A (2019) Big data stream analysis: a systematic literature review. J Big Data 6(1):47
35. Kumar PM, Gandhi UD (2018) A novel three-tier internet of things architecture with machine learning algorithm for early detection of heart diseases. Comput Electr Eng 65:222–235
36. Lakshman A, Malik P (2010) Cassandra: a decentralized structured storage system. ACM SIGOPS Oper Syst Rev 44(2):35–40
37. Lee K, Agrawal A, Choudhary A (2013) Real-time disease surveillance using twitter data: demonstration on flu and cancer. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 1474–1477
38. Mallu L, Ezhilarasie R (2015) Live migration of virtual machines in cloud environment: a survey. Indian J Sci Technol 8(S9):326–332
39. Manogaran G, Lopez D (2017) A survey of big data architectures and machine learning algorithms in healthcare. Int J Biomed Eng Technol 25(2-4):182–211
40. Manogaran G, Varatharajan R, Lopez D, Kumar PM, Sundarasekar R, Thota C (2018) A new architecture of internet of things and big data ecosystem for secured smart healthcare monitoring and alerting system. Futur Gener Comput Syst 82:375–387
41. Meng X, Bradley J, Yavuz B, Sparks E, Venkataraman S, Liu D, Freeman J, Tsai D, Amde M, Owen S et al (2016) Mllib: machine learning in apache spark. J Mach Learn Res 17(1):1235–1241
42. Mostafaeipour A, Jahangard Rafsanjani A, Ahmadi M, Arockia Dhanraj J (2021) Investigating the performance of hadoop and spark platforms on machine learning algorithms. J Supercomput 77(2):1273–1300
43. Nasiri H, Nasehi S, Goudarzi M (2019) Evaluation of distributed stream processing frameworks for iot applications in smart cities. J Big Data 6(1):52
44. Pourahmad S, Ayatollahi SMT, Taheri SM, Agahi ZH (2011) Fuzzy logistic regression based on the least squares approach with application in clinical studies. Comput Math Appl 62(9):3353–3365
45. Rallapalli S, Suryakanthi T (2016) Predicting the risk of diabetes in big data electronic health records by using scalable random forest classification algorithm. In: Advances in computing and communication engineering (ICACCE), 2016 international conference on. IEEE, pp 281–284
46. Rathore MM, Paul A, Ahmad A, Anisetti M, Jeon G (2017) Hadoop-based intelligent care system (hics): analytical approach for big data in iot. ACM Trans Internet Technol (TOIT) 18(1):8
47. Rustam F, Ashraf I, Mehmood A, Ullah S, Choi GS (2019) Tweets classification on the base of sentiments for us airline companies. Entropy 21(11):1078
48. Sampath P, Tamilselvi S, Kumar NS, Lavanya S, Eswari T (2017) Diabetic data analysis in healthcare using hadoop architecture over big data. Int J Biomed Eng Technol 23(2-4):137–147
49. Sreejith S, Rahul S, Jisha R (2016) A real time patient monitoring system for heart disease prediction using random forest algorithm. In: Advances in signal processing and intelligent recognition systems. Springer, pp 485–500
50. Ta V-D, Liu C-M, Nkabinde GW (2016) Big data stream computing in healthcare real-time analytics. In: Cloud computing and big data analysis (ICCCBDA), 2016 IEEE international conference on. IEEE, pp 37–42
51. Trigo JD, Eguzkiza A, Martínez-Espronceda M, Serrano L (2013) A cardiovascular patient follow-up system using twitter and hl7. Comput Cardiol 2013:33–36
52. Veiga J, Expósito RR, Pardo XC, Taboada GL, Tourifio J (2016) Performance evaluation of big data frameworks for large-scale data analytics. In: 2016 IEEE international conference on big data (Big Data). IEEE, pp 424–431
53. Venkatesh R, Balasubramanian C, Kaliappan M (2019) Development of big data predictive analytics model for disease prediction using machine learning technique. J Med Syst 43(8):272
54. Wachowicz M, Arteaga MD, Cha S, Bourgeois Y (2016) Developing a streaming data processing workflow for querying space-time activities from geotagged tweets. Comput Environ Urban Syst 59:256–268

55. Weka: Official webpage of Weka (2017) https://www.cs.waikato.ac.nz/ml/weka/. Online; Accessed 15 Dec 2017
56. Yan K, You X, Ji X, Yin G, Yang F (2016) A hybrid outlier detection method for health care big data. In: Big data and cloud computing (BDCloud), social computing and networking (SocialCom), sustainable computing and communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE international conferences on. IEEE, pp 157–162
57. Zaharia M, Das T, Li H, Hunter T, Shenker S, Stoica I (2013) Discretized streams: fault-tolerant streaming computation at scale. In: Proceedings of the 24th ACM symposium on operating systems principles, pp 423–438
58. Zaldumbide J, Sinnott RO (2015) Identification and validation of real-time health events through social media. In: Data science and data intensive systems (DSDIS), 2015 IEEE international conference on. IEEE, pp 9–16
59. Zhao T, Ni H, Zhou X, Qiang L, Zhang D, Yu Z (2014) Detecting abnormal patterns of daily activities for the elderly living alone. In: International conference on health information science. Springer, pp 95–108