Real-Time Healthcare Monitoring System using Online Machine Learning and Spark Streaming

Fawzya Hassan[1], Masoud E. Shaheen[2]
Department of Computer Science,
Faculty of Computers and Information,
Fayoum University, Egypt

Radhya Sahal[3]
Faculty of Computer Science and Engineering
Hodeidah University - Yemen

***Abstract*—The real-time monitoring and tracking systems play a critical role in the healthcare field. Wearable medical devices with sensors, mobile applications, and health cloud have continuously generated an enormous amount of data, often called streaming big data. Due to the higher speed of the streaming data, it is difficult to ingest, process, and analyze such huge data in real-time to make real-time actions in case of emergencies. Using traditional methods that are inadequate and time-consuming. Therefore, there is a significant need for real-time big data stream processing to guarantee an effective and scalable solution. So, we proposed a new system for online prediction to predict health status using Spark streaming framework. The proposed system focuses on applying streaming machine learning models (i.e. streaming linear regression with SGD) on streaming health data events ingested to spark streaming through Kafka topics. The experimental results are done on the historical medical datasets (i.e. diabetes dataset, heart disease dataset, and breast cancer dataset) and generated dataset which is simulated to wearable medical sensors. The historical datasets have shown that the accuracy improvement ratio obtained using the diabetes disease dataset is the highest one with respect to the other two datasets with an accuracy of 81%. For generated datasets, the online prediction system has achieved accuracy with 98% at 5 seconds window size. Beyond this, the experimental results have proofed that the online prediction system can online learn and update the model according to the new data arrival and window size.**

Keywords*—Online machine learning; streaming data; Apache Spark; Apache Kafka; spark streaming machine learning

I. Introduction

Nowadays, the era has been described as the era of big data where all data is digitalized and becomes of great importance in such beautiful fields. An enormous amount of data has been gathered and produced from different sectors, many sources like sensor networks, wireless machines, mobile applications, and from different fields [1]. Especially in the healthcare field, a big data collected in real-time by a remote sensing device, Wearable Medical Devices, and other data gathering tools, which produce new challenges that focus on data size and the fast growth rate of these data [2]. One of the most important challenges in data analytics is dealing with inappropriate technological tools to store, process, visualize, and extract knowledge with large and varied data types. In addition, exploring a new way to obtain valuable information for many users. However, the digital record of the patient's medical history is the primary health care data as it is obtained from various types of health care data sources in both clinical and non-clinical settings. Occasionally, these digital data are not available for research.

In the past, communication between doctors and patients was done by bounded visits, tele, and text communications. Consequently, doctors and hospitals could observe their patients' health constantly, and even more, they couldn't make recommendations accordingly. Thanks to IoT devices represented in wearable medical devices like heart rate monitoring cuffs, blood pressure, glucometer, etc. These devices can determine the space needed for each device and also determine the degree of interaction of people with these devices to provide health care solutions through continuous tracking of health conditions and giving patients access to personalized attention. However, the wearable medical devices continuously generate data; the amount of data stored and processed becomes a vital problem in real life. Furthermore, lately, many citizens and the elderly suffer from chronic diseases, and looking at the disadvantages of traditional health services is a very important thing. Therefore, the medical IoT is used to observe and take the required actions in real time for emergencies, like people with heart disease and diabetes[3]. Consequently, processing these enormous data generated by the sensors and implementing procedures in real time in critical cases is a very important challenge contribution.

Therefore, proposing a system that can handle big data faces three main challenges: First, collecting data from distributed systems is a complicated process due to the enormous amount of data. Second, storing that big and heterogeneous data is a major problem; therefore, the need for a big data storage system with efficient and effective performance is very essential. Third, this challenge relates to data analysis, especially big data processing in real time, including modeling, visualization, prediction, and optimization. Therefore, these challenges require new processing systems to deal with heterogeneous data or big data processing in real time. We will address the challenge related to data analytic in real time in the healthcare domain.

Recently, big data streaming computing has been used as an effective role in big data analytics to investigate the importance of big data in real-time healthcare. For example, a real-time system for flu and cancer monitoring is produced by applying twitter data mining in [4]. A model for real-time medical big data analysis is introduced in [5]. The model is performed by applying Spark Streaming [6] and Apache Kafka [1] using a stream of healthcare data. In [7] a real-time health status prediction system is proposed, this work focuses on applying machine learning especially Desion Tree algorithm on data

[1] https://kafka.apache.org

streams received from socket streams using Apache Spark[2]. However, researchers and developers face problems due to distributed data sources for healthcare (i.e., distributed queuing management technologies) like Kafka, and RabbitMQ[3]. The aggregated health-based streaming data is analyzed using big data platforms for streaming processing such as Apache Spark, Apache Storm[4], Apache Samza[5], Apache Flink[6], relational databases, analytics systems, and other search systems. Most recent research relies on machine learning, but streaming big data that needs to apply machine learning in real time is not handled. In particular, the previous studies just applied traditional machine learning algorithms to analyze and predict health status for patients using historical data. These studies focused mainly on Hadoop, the batch-oriented computing system. For this reason, the important challenge is applying machine learning to streaming data because conventional machine learning systems are not effective in dealing with real-time streaming data.

To the best of our knowledge, no study has been done oriented to online predicting disease in real-time. This motivates us to introduce a new online prediction system that can predict health status in real time, using Kafka data streaming, Spark Streaming, and Spark MLlib. Consequently, the research in this paper works on three important case studies: Pima Indians diabetes, Cleveland heart disease and breast cancer Coimbra because a large percentage of people have been injured with these diseases, and leading to death. So the online prediction for these diseases can decrease the mortality rate. The proposed system is used to achieve high accuracy using streaming data, i.e. Kafka producers produce stream messages of data continuously and then apply an online model to the online prediction in real time by classifying the stream of data as containing disease indication or not. The proposed system has four phases: 1) Data ingestions from the input stream within the data source; 2) Streaming Process Pipeline; 3) Online Prediction; and 4) Output Stream. The paper contributions can be summarized as:

- Developing an online prediction system to the possibility of disease using streaming data from historical medical datasets (i.e., diabetes dataset, heart disease dataset, and breast cancer dataset) and from real-time data in the form of wearable medical devices.
- Generating streaming data from simulated to wearable medical sensors and then capturing a stream of data by Kafka topic provides name to the various diseases.
- Applying StreamingLinearRegressionWithSGD algorithm to classify streaming data.
- Evaluating the result for historical medical dataset and simulated wearable sensor generator to compare the accuracy for different window sizes.

The remainder of this paper is organized as follows: The related work is presented in Section II. The proposed system of online prediction is introduced in Section III. The experimental results are discussed in Section IV. Finally, conclusions are presented in Section V.

II. Related Works

In last years, big data analytics concerning healthcare analytics become an important issue for many research areas like machine learning, data mining with data from healthcare as well as the available information from inside hospitals. The progress of the data collection process is due to the huge development in technology in the field of health care, where data is collected through three main stages of the flow of digital data resulting from the patient's clinical records, health research records, and organization operations records [3].

In [8] discuss an overview about the healthcare data sources. This study analyzes health care data played a very important role in many systems such as disease prediction, methods of prevention, medical guidance, and urgent medical decision-making in order to improve the standard of health care, reduce costs and increase efficiency. Also, Archenaa, J. and Anita [9] explain how can we apply Apache Spark to apply healthcare analytics through in-memory computations that can handle a large amount of structured, unstructured patient data and patients streaming data from their social network activities.

Multiple machine learning models on healthcare data using spark have been introduced in previous studies. For example In [10] Introduced a real time health status prediction system that uses spark machine learning streaming Big Data, The system was tested on the user tweets with his health attributes and the system receives these tweets, extracts the parameters and perform decision tree algorithm for user's health status predict, and finally send a direct message to him/her to take the appropriate action.

Moreover, Alotaibi, Shoayee, et al. [11] proposed a Sehaa which is a big data analytics tool for Arabic healthcare Twitter data in the Kingdom of Saudi Arabia (KSA). The system uses two different ML algorithms, including Naive Bayes, Logistic Regression, and applying multiple feature extraction methods to detect various diseases in the KSA. In [12] the system that can able to predict real-time heart disease based on Apache Spark that applied machine learning on streaming data by using memory computations are explained. The system are developed with two main stages, the first one is streaming processing which use spark MLlib with Spark streaming by applying classification algorithms on data to heart disease prediction. The second stage is data storage and visualization which uses Apache Cassandra to store a huge volume of generated data.

Most of this studies relies on specific healthcare data sources and applying processing on the offline system, but in reality, the sources of health data are various and constantly produce various data of different sizes. In addition, real-time healthcare analytical involves real-time streaming data processing, streaming machine learning algorithms, and analyzing real time to build an online electronic system to deal with the stream of healthcare data. therefore, we developed an online prediction healthcare system for streaming data coming from IoT devices to predict health status for the patient in real time.

[2] https://spark.apache.org
[3] https://www.rabbitmq.com
[4] http://storm.apache.org/
[5] http://samza.apache.org
[6] https://flink.apache.org

III. The Online Prediction System

The online health status prediction system is a data analytic monitoring model that uses Kafka streaming and Spark streaming. The architecture of the proposed system consists of four phases, as shown in Fig. 1. In the first phase, data ingestions from different data sources; Kafka producers continuously generate a stream of data messages that are captured by Kafka streaming from different topics name correlated to various diseases. Second, Streaming Process Pipeline in which Spark streaming receives a stream of medical data with the attributes which related to each disease and then applies a streaming machine learning model to predict health status. Third, Online Prediction receives batches of input data and responsible for online learning and updating our deployed model according to the new data arrival. At the final phase, the Output predicted results to data sinks in which the output stream sends back again to Kafka to be consumed by other data sinks such as web service, alarm systems, dashboard, mobile application, and hospital social networks.

A. Data Ingestions

For analyzing healthcare data, Apache Kafka has been used as the tool for ingestion of the individual's health data from distributed sources of historical data and real-time data. The historical data is ingested to the proposed system using multiple disease datasets; diabetes disease, heart disease, and breast cancer disease, each dataset uses different Kafka's topics named "diabetes_disease_dataste," "heart_disease_dataset," and "breast_cancer_dataset", respectively. The real-time dataset is collected by a simulated wearable sensor generator, which generates diabetes disease data on Kafka's topic named "Diabetes_Generator_Dataset." It is challenging to manage this data with Spark itself; therefore, Kafka is designed specifically for streaming data management[1]. Hence, it has been integrated into our system.

B. Streaming Data Processing over Apache Spark Streaming

The online prediction system is a data analytic model that uses a spark streaming machine learning library (MLlib) i.e. streaming linear regression with the SGD algorithm, which requires training data. Spark streaming receives a stream of records from historical data or real-time data, that is used as training data. Spark streaming data processing uses streaming computation by applying data decomposition (see Fig. 1). In the first, spark streaming transforms the input data stream into a Discretized stream (DS). After that the DSs are converted into Resilient Distributed Datasets (RDD). Therefore, it is urgent to transform and perform the RDD to be able to fulfill streaming processing. Furthermore, spark streaming can process the RDD data based on MLlib; the online prediction system uses the StreamingLinearRegressionWithSGD algorithm, which will be described in the following section.

1) Spark MLlib Streaming linear regression algorithm: Streaming linear regression uses Spark Streaming technology to train or predict a linear regression model based on streaming data. It applies SGD for each new batch of data coming from DStream to update the model. Each batch of data is expected to be an RDD of LabeledPoints. The spark streaming linear regression algorithm takes four parameters: the number of iteration, step size (learning rate), mini-batch fraction time, and initial weights vector. The number of iteration is needed to finish the gradient descent. The learning rate determines how slow or fast the algorithm can update the optimal regression coefficients. The initial weight vector must be provided; the default initial weight is 0.0. The mini-batch fraction time is used to set the batch time. The batch time parameter estimates the window time for spark streaming. Spark Streaming linear regression has a latency of many seconds, because of mini-batch time. Conversely, this mini-batch time efficiently ensures that each stream data will be processed exactly once [13].

C. Online Prediction

The online prediction phase is divided into two main components; the deployed model and the online learning mode. The online learning model takes the batches of input to train the model then send the training queue to the deployed model to learn and predict the result using the StreamingLinearRegressionWithSGD algorithm. The online learning algorithm takes the first batch result model and then picks up one to one observation from the training queue and recalibrates the weights on each input parameter. The deployed model is continuously learning, and it updates parameters for each batch result, which is close to "learning-on-the-fly". It helps to learn variations in distribution as quickly as possible and improve accuracy in many cases. The online prediction is found to be relatively faster than their batch equivalent methods.

D. Output Predicted Results to Data Sinks

In this phase, the output stream is sent back again to Kafka to be consumed by other data sinks. As the proposed system suppose to work in real-time, these data sinks could be any online data consumer within the real-world healthcare application in the industrial setting such as 1) web service for healthcare monitoring [14], [15], 2) alarm systems which connected to the hospital emergency department [16], [17], 3) real-time dashboard [18], 4) hospital medical records which stored in big data cloud storage (e.g., HDFS, MongoDB) [19], [20], and 5) hospital social networks considering patients' privacy.

IV. Experimental Results and Discussion

In this section, the experimental evaluation of the proposed online healthcare monitoring system is presented, and the results are discussed. Experiments were conducted on two different data sources of healthcare 1) historical medical data and real-time data. The historical data is ingested into the proposed system using three medical datasets chosen from UCI machine learning repository; Pima Indians diabetes [21], Cleveland heart disease [22] and breast cancer Coimbra [23]. The real-time dataset is collected by a simulated wearable sensor generator which generates diabetes disease data. Further details will be extensively discussed in the next subsection.

A. Experimental Setup

The proposed system has been implemented on top of Apache Spark using scala. Our experiments are conducted through Apache Kafka and Spark Streaming for data ingestion and data processing, respectively. The online machine learning

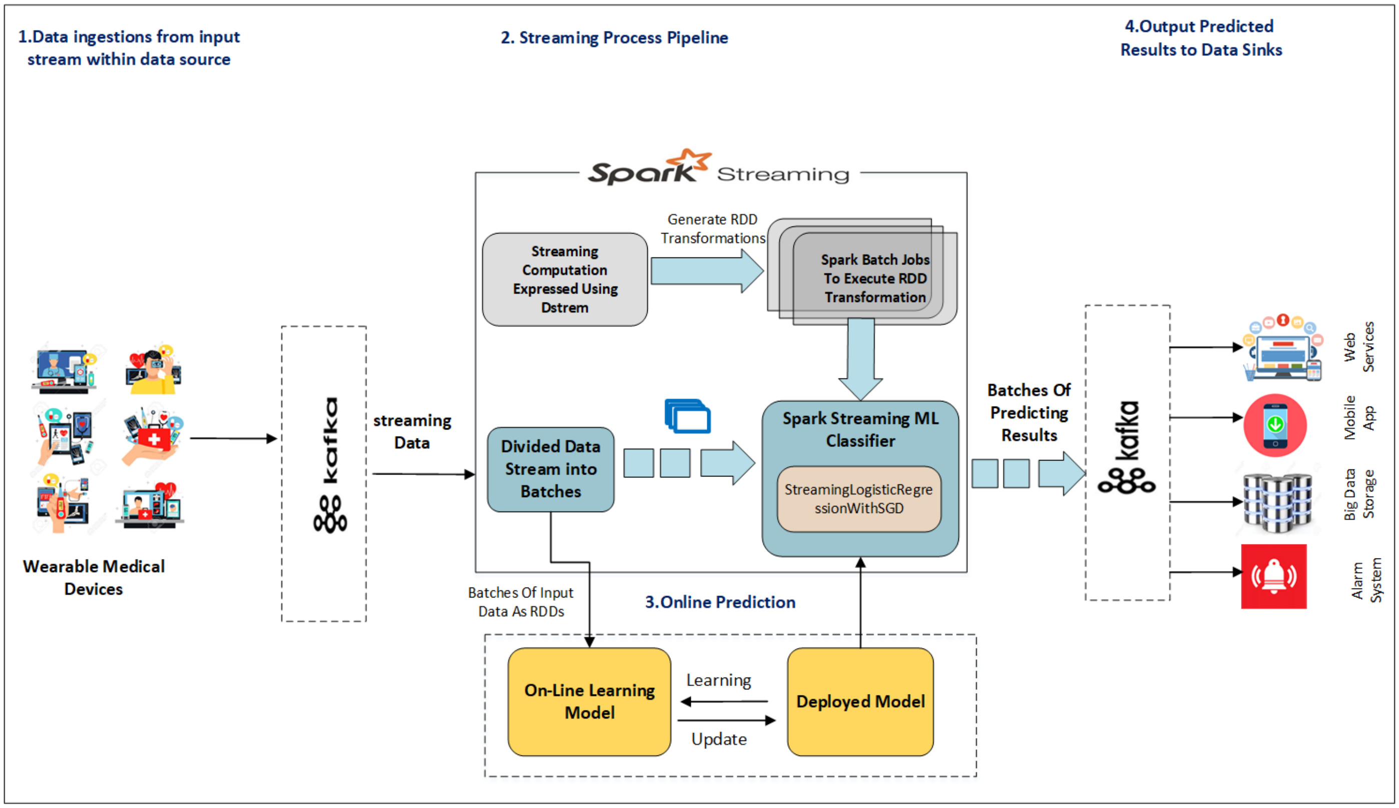

1.Data ingestions from input stream within data source
2. Streaming Process Pipeline
4.Output Predicted Results to Data Sinks
Spark Streaming
Streaming Computation Expressed Using Dstrem
Generate RDD Transformations
Spark Batch Jobs To Execute RDD Transformation
Wearable Medical Devices
kafka
streaming Data
Divided Data Stream into Batches
Spark Streaming ML Classifier
StreamingLogisticRegressionWithSGD
Batches Of Predicting Results
kafka
Web Services
Mobile App
Big Data Storage
Alarm System
Batches Of Input Data As RDDs
3.Online Prediction
On-Line Learning Model
Learning
Update
Deployed Model

Fig. 1. The Online Health Status Prediction System.

classifier, which is the StreamingLinearRegressionWithSGD algorithm, is applied to train on streaming data and then predict patient status. The historical medical data have been read from the chosen datasets and then ingested into Kafka topic while the online data have been generated using the developed simulated generator and then ingested into Kafka topic as well. For hardware specification, the experiments have been performed using Spark cluster version 2.3.1, consisting of one master node and two nodes for workers. Table I illustrates the characteristics of the master node and the worker nodes. For the performance metrics, the performance evaluation of the classification is done through various performance measures such as accuracy, precision, recall, and F1-score. Moreover, the performance comparisons are done according to the number of batches and window size. Accordingly, the performance evaluation of online prediction techniques is done through eight experiments. The first, second and third experiments were conducted using historical medical datasets, and the rest five experiments were conducted using generated datasets and five window sizes such as 1, 2, 3, 4, and 5 seconds.

TABLE I. CLUSTER NODES CHARACTERISTICS.

Parameter	Master	Worker
Precessor	Core i7	Core i7
Cores	4	4
Memory	20 GB	20 GB
Operating System	Ubuntu 18.04.2	Ubuntu 18.04.2

B. *Performance Analysis of the Historical Medical Datasets*

To evaluate the efficiency of the proposed system, various experiments were conducted on different datasets that contain medical records describing the patient's health information in terms of a set of attributes and the corresponding patient health status. For this research work, three medical datasets are used; Pima Indians diabetes dataset [24], Cleveland heart disease dataset [24], and breast cancer Coimbra disease dataset. Table II presents the description of the used datasets in terms of the number of samples, number of attributes, names of attributes, and labels. The reason behind using a real-time streaming method for historical data analysis is to assess the ability of the online prediction for the proposed system using benchmark datasets. Basically, Apache Spark reads data from a file and converts data to an RDD, then the continuous DStreams of data come from Spark Streaming. According to this work, each dataset is read from a CSV file and then sent to Spark streaming. Each RDD in a Dstream splits ingested data according to the window size and the size of the dataset. For the evaluation taken in this work, the datasets have been split into an 80% training set and a 20% testing set.

1) Results of Pima Indians diabetes dataset: In the first experiment, the Pima Indians diabetes dataset is performed. The number of samples of the Pima Indians diabetes dataset is 768, which has been read and sent to Spark Streaming. The window size has been configured into 2 seconds. According to the dataset size (i.e., the number of samples) and the configured window size, Dstream splits ingested data into two batches; the first batch and the second batch denoted by 1st

TABLE II. THE MEDICAL DATASETS' DESCRIPTIONS.

Dataset Name	Number of Samples	Numbers of Attributes	Names of Attributes	Labels
Pima Indians diabetes dataset	768	8	Pregnancies, Glucose, Blood Pressure, Insulin, Skin fold thickness, body mass index, diabetes pedigree function, Age	Classes: 0- absence 1- present
Cleveland heart disease	303	13	Age, sex, chest pain type, resting blood pressure , serum cholestoral, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise induced angina, ST depression induced by exercise relative to rest, number of major vessels (0-3) colored by flourosopy, thal	Classes: 0- absence 1- present
Breast cancer coimbra dataset	116	10	age, BMI, glucose, insulin, HOMA, leptin, adiponectin, resist in and MCP	Classes: 1-healthy 2- patient

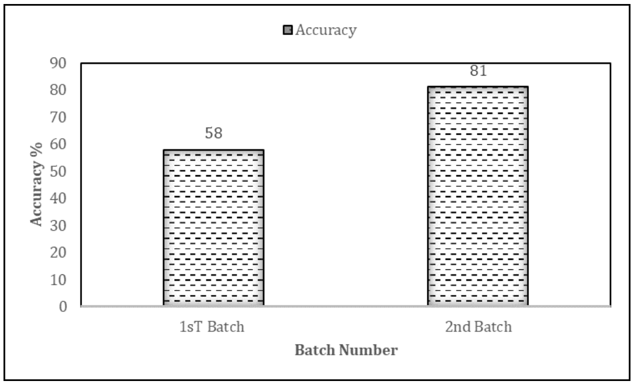


Fig. 2. The Accuracy of Online Prediction using the Pima Indians Diabetes Dataset.

Batch and 2nd Batch respectively. Table III shows the results of the Pima Indians dataset. It is noted that the second batch has obtained higher performances with respect to the first batch. We attribute this behavior to the more data ingested to the model, the performance metrics become higher. In particular, when the number of samples increases, the model learns and updates itself by time. For example, the obtained accuracy in the first batch is 58%, which increases to 81% for the second batch (see Table III and Fig. 2). As can be seen, other performance metrics, including precision, have improved with time as well in the second batch (precision of 83%, recall of 81%, and F1-score of 80%). The improvement ratios for the second batch for the performance metrics with respect to the first batch are 28%, 22%, 28%, and 31% for accuracy, precision, recall, and F1-Score, respectively.

TABLE III. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING THE PIMA INDIANS DIABETES DATASET.

Batch No	Accuracy	Precision	Recall	F1-Score
1st Batch	58	65	58	55
2nd Batch	81	83	81	80

2) Results of Cleveland heart disease dataset: In the second experiment, the Cleveland heart disease dataset is performed. The number of samples of the Cleveland heart disease dataset is 303, which has been read and sent to Spark Streaming. Similar to the first experiment, we have configured the window size to 2 seconds. Consequently, Spark Dstream splits ingested heart disease dataset into two batches; the first batch and the second batch denoted by 1st Batch and 2nd Batch, respectively. Table IV shows the Cleveland heart disease dataset results, as can be seen, that the performances of the second batch are higher with those obtained by the first batch. For instance, the obtained accuracy in the first batch is 58%, which increases to 81% for the second batch (see Table IV and Fig. 3). Also, other performance metrics have improved in the second batch (precision of 82%, recall of 79%, and F1-score of 78%). The reason behind this behavior is that the online prediction model learns and updates its performances by time. Based on this experiment using the Cleveland heart disease dataset, the improvement ratios for the second batch for the performance metrics with respect to the first batch are 18%, 13%, 18%, and 23% for accuracy, precision, recall, and F1-Score, respectively.

TABLE IV. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING THE CLEVELAND HEART DISEASE DATASET.

Batch No	Accuracy	Precision	Recall	F1-Score
1st Batch	65	71	65	60
2nd Batch	79	82	79	78

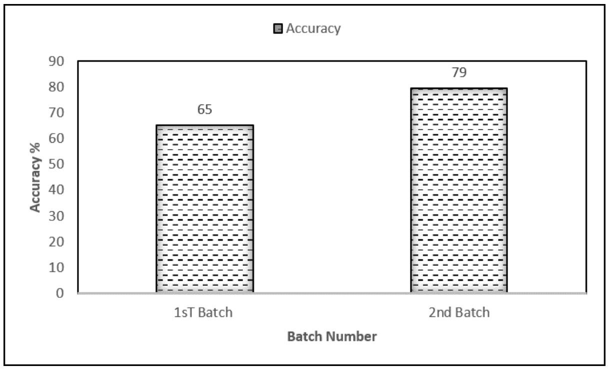


Fig. 3. The Accuracy of Online Prediction using the Cleveland Heart Disease Dataset.

3) Results of breast cancer Coimbra disease dataset: In the third experiment, the breast cancer Coimbra disease dataset is performed. The number of samples of the breast cancer dataset is 116, which has been read and sent to Spark Streaming. Similar to the previously conducted experiments, we have configured the window size to 2 seconds. Consequently, Spark Dstream splits ingested heart disease dataset into two batches; the first batch and the second batch denoted by 1st Batch and 2nd Batch, respectively. Table V shows the results of the breast cancer Coimbra disease dataset. It is noticed that the second batch outperforms the first batch. In particular, the performances of the second batch are higher with those obtained by the first batch. For instance, the obtained accuracy in the first batch is 63%, which increases to 67% for the second batch (see Table V and Fig. 4). Also, other performance metrics have improved in the second batch (precision of 76%, recall of 74%, and F1-score of 73%). The reason behind this behavior is that the online prediction model learns and updates its performances by time. Based on this experiment using the Cleveland heart disease dataset, the improvement ratios for the second batch for the performance metrics with respect to the first batch are 17%, 9%, 15%, and 19% for accuracy, precision, recall, and F1-Score respectively.

TABLE V. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING THE BREAST CANCER COIMBRA DISEASE DATASET.

Batch No	Accuracy	Precision	Recall	F1-Score
1st Batch	63	70	63	60
2nd Batch	76	76	74	73

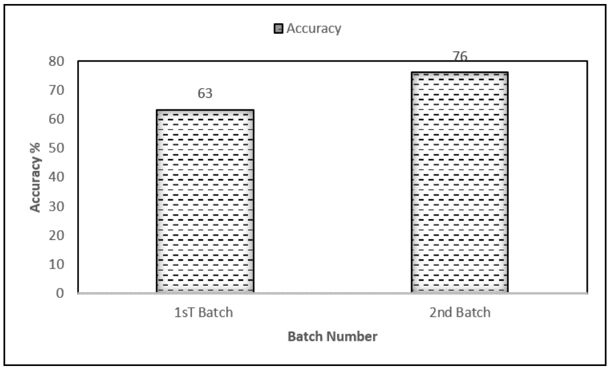


Fig. 4. The Accuracy of Online Prediction using the Breast Cancer Coimbra Disease Dataset.

4) Discussion: In the performance analysis of the historical medical datasets, three datasets have been evaluated using the proposed systems. Fig. 5 shows the accuracies of the second batch for the three medical datasets. As can be seen that the diabetes disease dataset has achieved the highest accuracy at 81% with respect to the heart and breast cancer datasets. The heart disease dataset has recorded the second rank on the average of the accuracy while breast cancer dataset has recorded the third rank (accuracy at 79% and 76% for heart and breast cancer dataset, respectively).

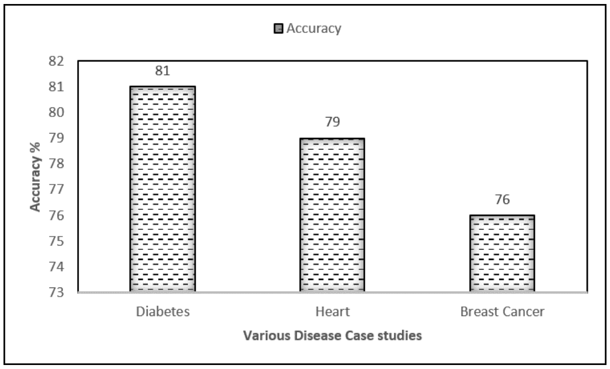


Fig. 5. The Accuracy of Online Prediction for the Second Batch using the Three Medical Datasets.

From the results obtained in our experiments, Fig. 6 depicts deeply the empirical results showing the improvement ratio of the accuracies for the second batch with respect to the first batch for the three medical datasets. It can be noticed that the accuracy improvement ratio, which has been obtained using the diabetes disease dataset, is the highest one with respect to the other two datasets. However, the achieved accuracy of the first batch using the diabetes disease dataset was 58%, which is lower than the accuracies, which have been achieved by the other two datasets in the first batch (accuracy at 65% and 63% for heart and breast cancer dataset, respectively). We attribute this behavior to the online prediction, which is responsible for the batching of input data. The online prediction can online learn and update the model according to the new data arrival. For the diabetes disease dataset, the arrival of real-time training data is larger compared to datasets because of the large number of samples, 768. Based on these results, it can be tentatively concluded that using a larger number of samples for online prediction will improve the accuracy of the proposed system over time. Particularly, the larger number of samples can lead to updating of the model in real-time and further improving the accuracy of real-time prediction. For this reason and due to lack of large datasets, this motivates us to develop a data generator that simulates the medical sensors data to generate a large number of samples, which would improve the online prediction performance.

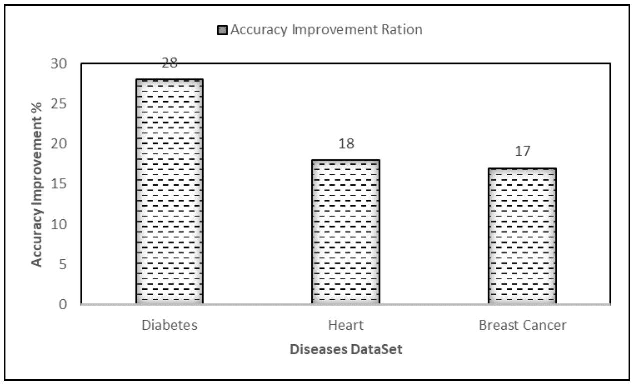


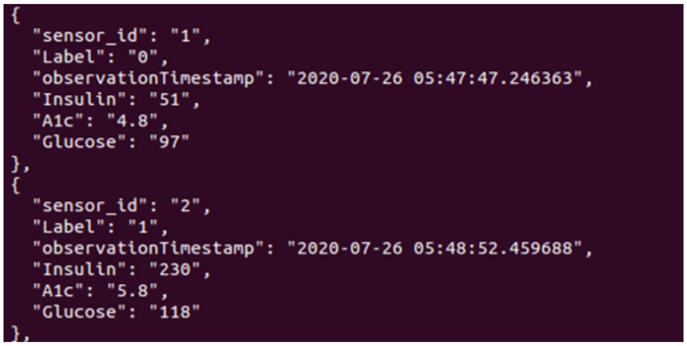
Fig. 6. The Improvement Ratios of the Online Prediction Accuracies for the Second Batch with respect to the First Batch using the Three Medical Datasets.

C. Performance Analysis for Real-time Streaming Dataset

After evaluating the proposed system using the historical medical datasets, we have noticed that the large dataset achieves higher performance for online prediction. Therefore, to assess the efficiency of the online prediction system, we have developed a simulated data generator to generate a large dataset that can be trained for achieving higher performances using multiple DStreams. In particular, the simulated data generator has been developed to generate streaming diabetes samples as JSON format (see Fig. 7). The multiple data streams samples are sent via Apache Kafka and then processed by Spark Streaming. The rule of thumb of generated samples of diabetes dataset is introduced in [25]. The diabetes disease depends on three factors, which represented as attributes; A1c, Fasting Plasma Glucose (FPG), and Oral Glucose Test (OGT) (see Table VI). A1c tests the blood trail of a person for recent months. FPG tests a fasting plasma glucose level to recognize diabetes. OGT describes the oral glucose to analyze diabetes.

TABLE VI. DIABETES CONDITIONS

A1c	Fasting Plasma Glucose(0,199)	Oral Glucose Test(0,846)	Label
A1c >5.7	Glucose >100	Insulin >140	1
A1c <5.7	Glucose <= 99	Insulin <= 139	0



```
{
  "sensor_id": "1",
  "Label": "0",
  "observationTimestamp": "2020-07-26 05:47:47.246363",
  "Insulin": "51",
  "A1c": "4.8",
  "Glucose": "97"
},
{
  "sensor_id": "2",
  "Label": "1",
  "observationTimestamp": "2020-07-26 05:48:52.459688",
  "Insulin": "230",
  "A1c": "5.8",
  "Glucose": "118"
},
```

Fig. 7. JSON-like Generated Streaming Diabetes Data.

1) Comparison using different window sizes: To evaluate the efficiency of the proposed online prediction system, five experiments were conducted by fine-tuning the window sizes such as 1,2, 3, 4, and 5 seconds and using the generated diabetes dataset. The obtained results from diabetes prediction of the first experiment using 1 sec are shown in Table VII. It can be seen that the Spark streaming has split the generated data into 14 batches. Also, it can be noticed that the accuracy of the online prediction has increased linearly and improved by time (see Fig. 8). The highest accuracy obtained with the 1st batch is 59%, while the highest accuracy achieved by the 14th batch is 86%.

Similarly, as we have configured window size in 2 seconds for the second experiment, Spark streaming has split the generated data into 12 batches (see Table VIII). Fig. 9 depicted the accuracy of the online prediction for the 12 batches, where the accuracy of the online prediction has increased linearly and improved by time. The highest accuracy obtained with the 1st batch is 60%, while the highest accuracy achieved by the 12th batch is 88%. For the third experiment, the window size has been configured to 3 seconds, which makes Spark streaming to split the ingested generated data into 9 batches. Consequently, the performances of the online prediction using 3 sec as a window size have been shown in Table IX. Also, Fig. 10 presents the obtained accuracy, which increases linearly from the 1st batch to the 9th batch starting by 64% and ending by 90%.

Table X describes the performance of the online prediction using 4 seconds window size. The performances have been obtained among 6 batches, which are split by Spark streaming. The obtained performances have increased by time similar to the previous experiments. For instance, the accuracy has increased from 71% for the 1st batch and then 78% for the 2nd batch and so on (see Fig. 11). For the fifth experiment, the window size has been set to 5 seconds, which leads to 3 batches. Table XI presents the corresponding performances which have been obtained using a 5 second window size. Also, Fig. 12 depicts the accuracies of the online prediction, which are 85%, 93%, and 98% for the 1st, 2nd, and 3rd batch, respectively. It can be seen that the three obtained accuracies have increased, and the improvement has grown faster by time.

TABLE VII. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING GENERATED DIABETES DATASET AND 1 SEC WINDOW SIZE.

Batch No	Accuracy	Precision	Recall	F1-score
1st Batch	59	66	59	56
2nd Batch	60	68	60	57
3rd Batch	64	70	64	59
4th Batch	66	73	66	62
5th Batch	68	77	68	66
6th Batch	70	78	70	67
7th Batch	71	78	71	68
8th Batch	73	81	73	70
9th Batch	76	76	76	75
10th Batch	77	77	77	77
11th Batch	79	79	79	79
12th Batch	81	82	82	81
13th Batch	83	83	83	83
14th Batch	86	85	85	85

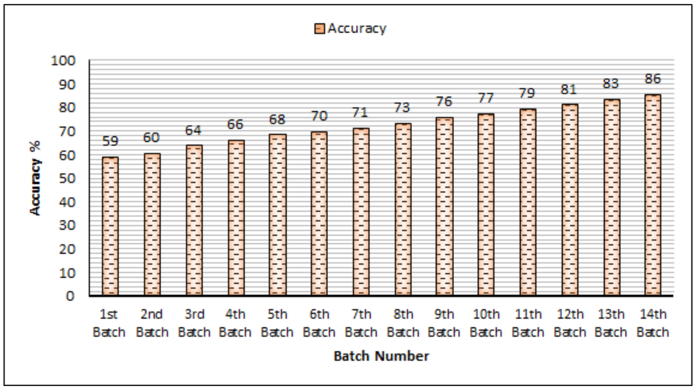


Fig. 8. The Accuracy of Online Prediction using the Generated Diabetes Dataset and 1 sec Window Size.

TABLE VIII. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING GENERATED DIABETES DATASET AND 2-SEC WINDOW SIZE.

Batch No	Accuracy	Precision	Recall	F1-score
1st Batch	60	67	60	57
2nd Batch	62	69	61	59
3rd Batch	66	73	66	61
4th Batch	68	75	68	63
5th Batch	72	81	72	70
6th Batch	75	84	75	73
7th Batch	78	86	78	75
8th Batch	80	80	80	80
9th Batch	83	83	83	82
10th Batch	85	85	85	84
11th Batch	87	86	87	68
12th Batch	88	88	88	88

2) Discussion: We analytically and experimentally summarize the performance gained from different window sizes. Fig. 13 depicts the window size tuning and their superiority over the time. In particular, if the window size is greater, the online prediction performances will be slightly improved. Consequently, a higher window size causes rapid prediction accuracy. For example, the 5 second window size has recorded 98% which is the highest accuracy among the other window sizes because the model learns and updates itself using three batches. Similarly, the 4 second, 3 second, 2 second and 1 have registered 95%,90%,88% and 86% using 6,9,12 and 14 batches, respectively. It can be seen that the larger window sizes allows the online prediction to process large data sizes,

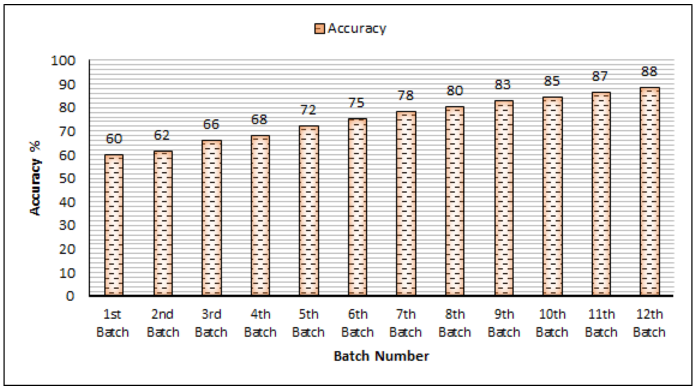


Fig. 9. The Accuracy of Online Prediction using the Generated Diabetes Dataset and 2-sec Window Size.

TABLE IX. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING GENERATED DIABETES DATASET AND 3 SEC WINDOW SIZE.

Batch Number	Accuracy	Precision	Recall	F1-score
1st Batch	64	71	64	60
2nd Batch	70	76	70	67
3rd Batch	74	81	74	71
4th Batch	78	78	78	78
5th Batch	80	80	80	80
6th Batch	84	86	84	84
7th Batch	86	86	86	85
8th Batch	88	88	88	87
9th Batch	90	91	90	90

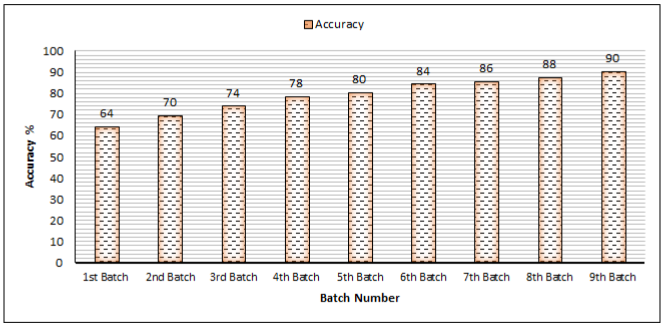


Fig. 10. The Accuracy of Online Prediction using the Generated Diabetes Dataset and 3-sec Window Size.

TABLE X. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING GENERATED DIABETES DATASET AND 4-SEC WINDOW SIZE.

Batch Number	Accuracy	Precision	Recall	F1-score
1st Batch	71	78	71	69
2nd Batch	78	78	78	77
3rd Batch	83	83	83	82
4th Batch	88	89	88	88
5th Batch	92	93	92	92
6th Batch	95	95	95	94

TABLE XI. THE PERFORMANCE RESULTS OF ONLINE PREDICTION USING GENERATED DIABETES DATASET AND 5 SEC WINDOW SIZE.

Batch Number	Accuracy	Precision	Recall	F1-score
1st Batch	85	86	85	84
2nd Batch	93	94	93	93
3rd Batch	98	98	98	97

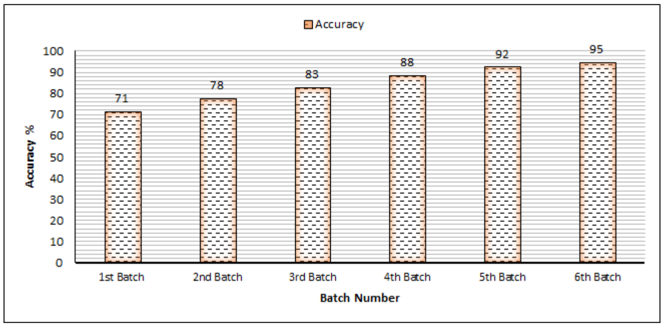


Fig. 11. The Accuracy of Online Prediction using the Generated Diabetes Dataset and 4-sec Window Size.

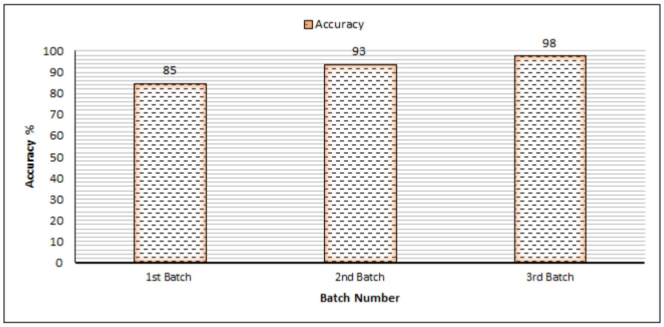


Fig. 12. The Accuracy of Online Prediction using the Generated Diabetes Dataset and 5 sec Window Size.

even less batch number which improves the prediction accuracy. We can conclude that the window size has a significant impact on the processing rate of Spark Streaming in terms of training a larger number of samples.

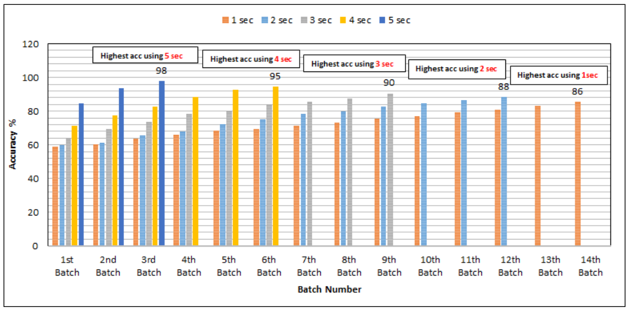


Fig. 13. Comparison of Accuracy of Online Prediction using Different Window Sizes.

V. CONCLUSION

In this paper, we have presented an online prediction system to predict real-time health status. The proposed system has been developed using Spark Streaming, Apache Kafka, Apache Spark, and streaming machine learning algorithm named streaming linear regression with SGD. It has applied to two distributed data sources; historical medical data sources (diabetes, heart and breast cancer) and simulated wearable sensor generator which generates diabetes dataset. The diabetes dataset has achieved the highest accuracy at 81% with respect to the heart and the breast cancer datasets. The generated diabetes dataset has achieved the highest accuracy at 98% using 5-second window size comparing to other window sizes: 1, 2, 3 and 4 seconds. The experimental results have shown that the larger window sizes allow the online prediction to process large amounts of data sizes, even fewer batch numbers, which improve prediction accuracy.

REFERENCES

[1] R. Sahal, J. G. Breslin, and M. I. Ali, "Big data and stream processing platforms for industry 4.0 requirements mapping for a predictive maintenance use case," *Journal of Manufacturing Systems*, vol. 54, pp. 138 – 151, 2020. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0278612519300937

[2] G. Manogaran and D. Lopez, "Health data analytics using scalable logistic regression with stochastic gradient descent," *International Journal of Advanced Intelligence Paradigms*, vol. 10, no. 1-2, pp. 118–132, 2018.

[3] A. Ed-daoudy and K. Maalmi, "A new internet of things architecture for real-time prediction of various diseases using machine learning on big data environment," *Journal of Big Data*, vol. 6, no. 1, p. 104, 2019.

[4] K. Lee, A. Agrawal, and A. Choudhary, "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*, 2013, pp. 1474–1477.

[5] U. Akhtar, A. M. Khattak, and S. Lee, "Challenges in managing real-time data in health information system (his)," in *International Conference on Smart Homes and Health Telematics*. Springer, 2016, pp. 305–313.

[6] A. Spark, "Spark streaming," https://spark.apache.org/docs/2.3.0/streaming-programming-guide.html/, 2020.

[7] A. Ed-daoudy and K. Maalmi, "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, 2018, pp. 1–4.

[8] J. Sun and C. K. Reddy, "Big data analytics for healthcare," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 1525–1525.

[9] J. Archenaa and E. M. Anita, "Interactive big data management in healthcare using spark," in *Proceedings of the 3rd International Symposium on Big Data and Cloud Computing Challenges (ISBCC–16')*. Springer, 2016, pp. 265–272.

[10] L. R. Nair, S. D. Shetty, and S. D. Shetty, "Applying spark based machine learning model on streaming big data for health status prediction," *Computers & Electrical Engineering*, vol. 65, pp. 393–399, 2018.

[11] S. Alotaibi, R. Mehmood, I. Katib, O. Rana, and A. Albeshri, "Sehaa: A big data analytics tool for healthcare symptoms and diseases detection using twitter, apache spark, and machine learning," *Applied Sciences*, vol. 10, no. 4, p. 1398, 2020.

[12] A. Ed-Daoudy and K. Maalmi, "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, 2019, pp. 1–5.

[13] B. Akgün and Ş. G. Öğüdücü, "Streaming linear regression on spark mllib and moa," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, 2015, pp. 1244–1247.

[14] G. B. Laleci, A. Dogac, M. Olduz, I. Tasyurt, M. Yuksel, and A. Okcan, "Saphire: a multi-agent system for remote healthcare monitoring through computerized clinical guidelines," in *Agent technology and e-health*. Springer, 2007, pp. 25–44.

[15] W. N. Robinson, "Monitoring web service requirements," in *Proceedings. 11th IEEE International Requirements Engineering Conference, 2003.* IEEE, 2003, pp. 65–74.

[16] M. Bransby and J. Jenkinson, *The management of alarm systems*. Citeseer, 1998.

[17] R. L. Wears and S. J. Perry, "Human factors and ergonomics in the emergency department," *Annals of emergency medicine*, vol. 40, no. 2, pp. 206–212, 2002.

[18] A. Franklin, S. Gantela, S. Shifarraw, T. R. Johnson, D. J. Robinson, B. R. King, A. M. Mehta, C. L. Maddow, N. R. Hoot, V. Nguyen *et al.*, "Dashboard visualizations: Supporting real-time throughput decision-making," *Journal of biomedical informatics*, vol. 71, pp. 211–221, 2017.

[19] M. Fazio, A. Celesti, A. Puliafito, and M. Villari, "Big data storage in the cloud for smart environment monitoring," 2015.

[20] R. Nachiappan, B. Javadi, R. N. Calheiros, and K. M. Matawie, "Cloud storage reliability for big data applications: A state of the art survey," *Journal of Network and Computer Applications*, vol. 97, pp. 35–47, 2017.

[21] N. I. of Diabetes, Digestive, and K. Diseases, "Pima indians diabetes," https://www.kaggle.com/uciml/pima-indians-diabetes-database, 2020.

[22] "Heart disease uci," https://www.kaggle.com/ronitf/heart-disease-uci, 2020.

[23] "Breast cancer coimbra data set," https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra, 2020.

[24] H. Kahramanli and N. Allahverdi, "Design of a hybrid system for the diabetes and heart diseases," *Expert systems with applications*, vol. 35, no. 1-2, pp. 82–89, 2008.

[25] N. Sneha and T. Gangil, "Analysis of diabetes mellitus for early prediction using optimal features selection," *Journal of Big data*, vol. 6, no. 1, p. 13, 2019.