



## Early Health Prediction System for ICU Patient using Machine Learning and Cloud Computing

---

Asif Ahmed Nelay, Muhammad Shafayat Oshman,  
Md. Monzurul Islam, Md Julhas Hossain and Zunayeed Bin Zahir

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 6, 2019

# Early Health Prediction System for ICU Patient using Machine Learning and Cloud Computing

Asif Ahmed Nelay  
*dept. of electrical and computer  
engineering*  
North South University  
Dhaka, Bangladesh  
asif.nelay@northsouth.edu

Muhammad Shafayat Oshman  
*dept. of electrical and computer  
engineering*  
North South University  
Dhaka, Bangladesh  
shafayat.oshman@northsouth.edu

Md. Monzurul Islam  
*dept. of electrical and computer  
engineering*  
North South University  
Dhaka, Bangladesh  
aurkouchchahs@gmail.com

Md Julhas Hossain  
*dept. of electrical and computer  
engineering*  
North South University  
Dhaka, Bangladesh  
julhas78@gmail.com

Zunayeed Bin Zahir  
*dept. of electrical and computer  
engineering*  
North South University  
Dhaka, Bangladesh  
zunayeed.zahir01@northsouth.edu

u

**Abstract**—Adaptable Intensive Care Unit (ICU) system is a key concern for hospitals in developing countries like Bangladesh. Most of the hospital in Bangladesh lack serving proper health service due to unavailability of appropriate, easy and scalable systems. There also appears communication and information gaps between hospital authority and patient's relative. The aim of this project is to build an adequate system for hospitals to serve the ICU patients with a real-time feedback system. Based on the doctor's suggestions, we have primarily chosen the main factors for our project. In this paper, we propose a generic architecture, associated terminology and a classificatory model for observing ICU patient's health condition with machine learning and cloud computing. Machine learning (ML) health prediction is the key concept of this research. IBM Cloud is the platform for this research to store and maintain our data. For our ml models, we have chosen the following algorithms: Naive Bayes, Logistic Regression. For real-time data and information view, we have developed a Mobile Application named "ICU Patient Management System - IPMS". Our system architecture is designed in such a way that the ml models can train and deploy in a real-time interval by retrieving the data from IBM Cloud and the cloud information can also be accessed through IPMS in a requested time interval. To help the doctors, the ml models will predict the condition of a patient. If the prediction based on the condition gets worse, the IPMS will send an SMS to the duty doctor and nurse for getting immediate attention to the patient. Combining with the cloud storage, distributed database system, ml models and mobile application, the project may serve as a complete medical decision for the doctors.

**Keywords**—*ICU Care, Machine Learning, Cloud Computing, ICU Patient Monitoring System, Naive Bayes.*

## I. INTRODUCTION

Patient Monitoring System is a process where a doctor can continuously monitor more than one patient, for more than one parameter at a time in a remote place and also can have control

over medicine dosage [1]. Whether these systems will prove beneficial for these hospitals depends on the support that will be provided to ensure the effective use of the systems implemented and also on the satisfaction of its users, which is one of the most important determinants of the success of these systems. Development and evaluation of the ICU decision-support systems would be greatly facilitated by these systems. A reliable and efficient real-time remote patient monitoring system that can play a vital role in providing better patient care. Devices such as vital sign monitors, mechanical ventilators and dialysis machines, and some others more are used to support critical patients whose bodies need time to recover and repair [2]. Most of the machines are managed manually by supervising the patient's condition and test reports. In a country like Bangladesh, where there are only 3 doctors for 10000 people [3], it's quite impossible to manage so many patients at a time and also the top experienced doctor has a high demand of the patient. Also, there is a massive 19% rate of producing wrong treatment to the ICU and cabin Patients [4]. So, we thought to reduce the communication and information gap by taking help from modern technology, especially the machine learning and cloud computing. Machine learning models can predict the near future condition of the patients, whether their condition will increase or decrease, whether they need ventilation support or not. If the doctor is not present in the ICU, he/she can send feedback or necessary parameter through IPMS, to the duty doctor or nurse so that immediate action can be taken properly. To generalize our models and data, we have selected IBM Cloud which altogether spans public, private and hybrid environments. As initially we cannot deploy our models directly to the hospital servers or cloud service and install our systems to retrieve data from ICU machines, we had to use IBM Cloud, IBM Watson Studio for storing, testing and deploying our whole system. The machine learning models run within the cloud service and also trains with the live data, the IPMS also can access the Cloud services through Bluemix [5].

## II. MOTIVATION

Health sector seems to be one of the neglected fields in terms of usage of technology in Bangladesh [6]. Although other sectors have adequately taken this advantage, health sector seems to be lagging behind. Government projects to integrate technology into the health sector has mostly failed. Patient monitoring has been one of the most neglected fields in the health sector. Due to inefficient handling of patients during an emergency, most of the cases result in death or permanent physical/mental damage to the patients, the main reason being the attending physician's inability to monitor the patient's vitals immediately [26]. The main method of communication is a mobile phone when the doctor is absent, resulting in communication mismatch. Our research installs the mechanism where the doctor can monitor the patient's vitals remotely, taking full advantage of Machine Learning to prescribe an advanced course and Cloud Computing to access the patient's vitals from any remote location. Cloud Computing's "Always On" nature guarantees an uptime [7], which makes it convenient for the users to access the data whenever needed when a minimal possibility of the server outage. The application that is developed as a result of this project, will benefit doctors and patients equally. Doctors can monitor multiple patients within a short span of time. Patients' relatives can get regular updates without having to visit the hospital every now and then.

## III. FIELD SURVEY

### A. Decision-making assistant for the Doctors

Accommodate a huge number of patients with a limited resource is a challenge for a doctor in Bangladesh [8]. To get a resolvable decision, the doctors will be benefited largely by this application. Some previous works have integrated cloud computing to upload patient data into the cloud to viewed later and machine learning with ICU patient care [9]. We are proposing a system that will integrate both the features and take maximum advantage and in the process, proper patient care to an extended degree of improvement.

### B. Field Survey Report

Before starting the project, we have visited several hospitals in Dhaka including Dhaka Medical College (DMC), Bangladesh Medical College (BMC) and also privately visited doctors to know about the current situation and basic needs for the ICU patients. One of the core intentions was to know more about the complete process that conducts the ICU system. From this survey, we came to know about the parameters that define the medical condition of a patient, the natural health conditions and the critical health conditions and most importantly the crucial time to make adequate decisions. Health conditions and parameters differ from age to age and from patient to patient [10]. Also, a single instance can affect others, like heart disease, can trigger lower pulse or maybe higher pulse based on the high pressure or low pressure. Considering all the survey results, we have created the project plan from step by step building the core components. Based on the doctor's suggestion, we only chose to consider age, sex,

weight, pulse, blood pressure, blood circulation, ventilator, oxygen supply, and internal bleeding as our main factoring features for this project [11].

## IV. SYSTEM DESIGN

The whole system is comprised of six major layouts. Each of them is briefly described below.

### A. Hospital Terminal

Hospital terminal is the main input section for the cloud storage. Through this terminal, the nurse or duty doctor will put the prescribed condition and information or the recent condition of the patient to the IBM Cloud.

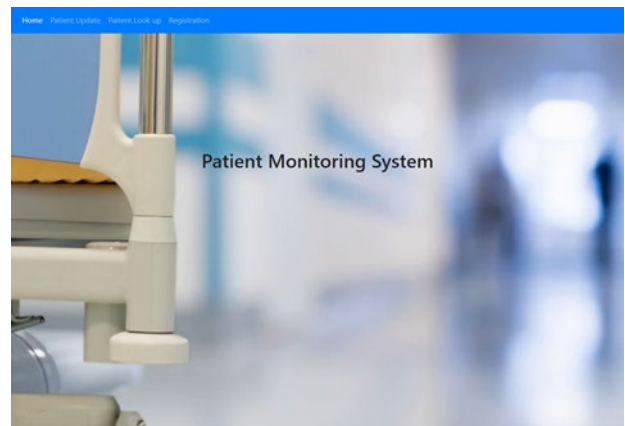


Fig. 1. Hospital Terminal

"Fig. 1", shows the main interface for the hospital. This interface will be an administrative interface for all authorized personals. From the main interface, there will be two distinct option for patients. The register page will be using as the "registration" process for the patients and the patient update page will be the windows for an hourly update of the condition.

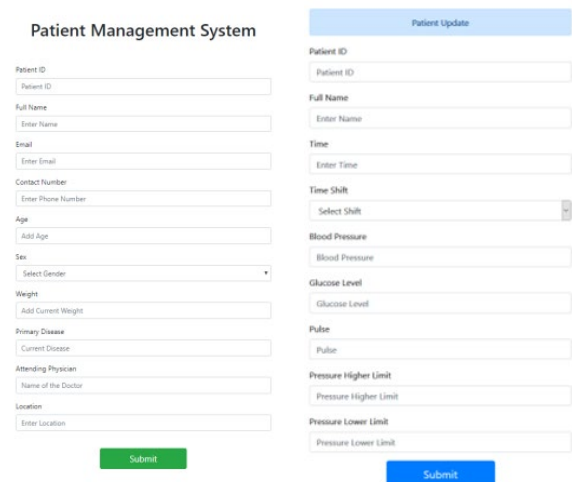


Fig. 2. Registration and Update page for patients

PATIENT_ID	FULL_NAME	CONTACT_NO	AGE	SEX	WEIGHT	PRIMARY_D.	ATTENDING...	ADMITTED_TO	VENTILATOR	INTERNAL...	LC_FEEDBA...	PRESSURE_H	PRESSURE_L	OXYGEN_S...	BLOOD_CR...	GLUCOSE_L...	PULSE	BP	SEND_SMS
INTEGER	VARCHAR(100)	INTEGER	INTEGER	VARCHAR(100)	DOUBLE	VARCHAR(100)	VARCHAR(100)	VARCHAR(100)	VARCHAR(100)	VARCHAR(100)	VARCHAR(100)	INTEGER	INTEGER	INTEGER	VARCHAR(100)	VARCHAR(100)	INTEGER	VARCHAR(100)	VARCHAR(100)
1	1 Mardub Jahan Ay	070290277	19	Male	110	Unknown	Dr. Farbus Anj., Word	No	No	Yes	Normal	120	90	0	Normal	Normal	86	Normal	No
2	2 Blood Bhanan Kam	155633900	37	Male	48.0	Kidney Diseases	Dr. Farbus Anj., ICU	Yes	No	No	Critical	90	85	100	Low	Low	97	Low	Yes
3	3 Md Mardub Islam	074899174	23	Male	210	Kidney Diseases	Dr. Mardub Anwar, Word	No	No	No	Normal	125	90	0	Normal	Normal	68	Normal	No
4	4 Md. Shandji Isam	071003283	33	Male	19.0	Stroke	Dr. Q. Tariku Isam, ICU	Yes	Yes	Yes	Decreasing	145	95	80	Low	Low	78	High	Yes
5	5 Ayesha Jusika	071871009	15	Female	70	Heart Diseases	Dr. Mardub Anwar, ICU	No	No	Yes	Normal	155	100	80	Normal	Normal	86	High	No
6	6 Md Hasem Jari H	071014288	5	Male	74.0	Respiratory Infection	Dr. Mardub Anwar, ICU	Yes	No	No	Decreasing	140	95	100	Low	Low	85	High	Yes
7	7 Md Inam Munira	072297340	23	Female	90	Kidney Diseases	Professor Dr. Fazul Word	No	No	No	Normal	125	90	0	Normal	Normal	71	High	No
8	8 Nazir Ahmed	070023521	65	Male	58.0	Heart Diseases	Professor Dr. Fazul, ICU	Yes	No	No	Critical	140	95	100	Low	Normal	91	High	Yes
9	9 Md Hassem Sarika	078003477	69	Male	38.0	Accident	Dr. Farbus Anj., Word	No	No	No	Increasing	90	70	0	Normal	Low	60	Low	No
10	10 Md. Shafiqul Islam	071651814	32	Male	61.0	Accident	Dr. Farbus Anj., ICU	Yes	Yes	Yes	Decreasing	145	95	100	Low	Low	55	High	Yes

Fig. 3. Sample Dataset

“Fig. 2”, describes the distinct pages for registration and condition update page features. All the information, input from these terminals will be directly stored in the IBM Cloud database.

B. Cloud Architecture

To establish the system proposed by the author’s research, IBM Cloud with IBM Watson was required. In the Bluemix Console of IBM Cloud, an instance of the Db2 Warehouse on Cloud service was created [12]. Db2 Warehouse service supports SQL based database [13], thus it was selected. The NDX9073 schema was selected to build up the database. Tables for Patient Registration, Patient Update, Doctor’s info, ML Result were created, where concerned data were stored.

“Fig. 3”, Illustrates the sample dataset that is stored through Hospital Terminal. Patient Registration contained all the primary data of the patients. Patient Update contained the data of regular updates of the patient’s condition. For daily updates, there will be in total 6 reading per day collected from hospitals terminal.

MLRESULT table stored the data of ml predictions. Main Credentials were created using IBM’s system [14]. These credentials were optimized to integrate the Android Application and Web Application.

C. Machine Learning Models

The ubiquitous monitoring of ICU patients has generated a wealth of data which presents many opportunities but also great challenges.

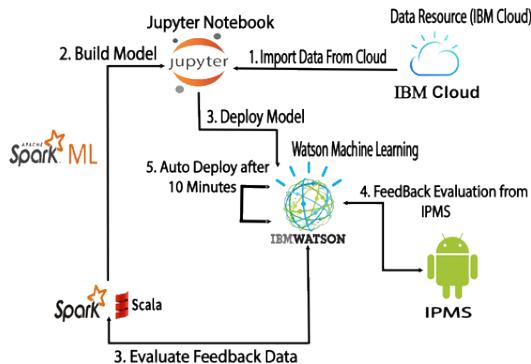


Fig. 4. System Architecture

To turn out the best predictive results by using ml takes carefully selecting the algorithms [15].

a) Import data from IBM Cloud: All the machine ml models will be auto-deployed from the IBM Watson Studio. For connecting the central database, the admin from the master cloud will provide “Username” “password” “Hostname or IP Address” and “Database Type”. To connect any other cloud and database with the master database, these pieces of information are needed to fill out from “Assets” and “Edit Connection” Sections. “Fig. 5”, shows the process to connect the Watson Studio models with IBM Cloud database.

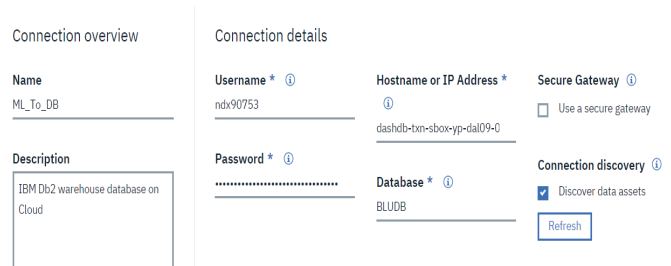


Fig. 5. IBM Watson and IBM Cloud Connection

This process is basically a pipeline between several cloud storage entities [16]. This connection will be auto-refreshed after each data is pushed to the cloud. For manually refresh the data, the “Discover Data Assets” option can be utilized. Also, in case of checking the newly entered data validation this manually, control option can regulate the process.

b) Auto Pre-processing of the dataset: We cannot perform manual pre-processing within our live database [17]. To perform pre-processing from IBM Cloud we have to use “Refine” function from model building. From Refine function, a set of instruction can be set to pre-process automatically. For our dataset, the instructions are - Convert Column Values to Missing, Remove Empty Rows and Replace Substring. For every instruction, we have to set the column values and this will perform automatic cleanse of the live database. After the cleanse, the dataset will be saved in CSV format. “Fig. 6”, shows the core process of pre-processing of the dataset.

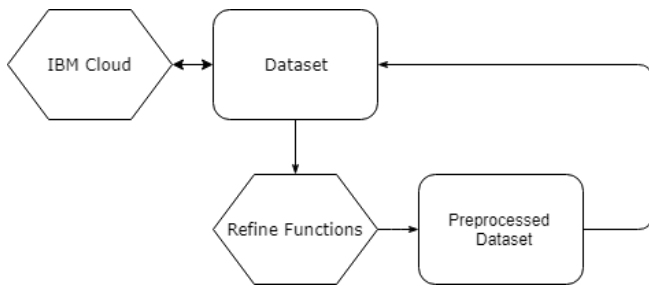


Fig. 6. Auto-Preprocessing of dataset

c) *Watson Machine Learning models:* IBM Watson provides flexible options to create and run ml models [18]. A model can be created from “Model Builder” option. To use any existing trained model from local storage or to import any ml models, the “From File” option is present. For our project, we will use Jupyter Notebook models by setting the training parameter as manual [27]. For importing out dataset into the models we need to select the Pandas DataFrame function to load the ‘registration’ dataset into our models. IBM Boto Client can import the required configuration to the notebook kernel and the dataset onto it. The following code will do the importing process –

```

import sys
import types
import pandas as pd
from ibm_botocore.client import Config
import ibm_boto3
from ibm_dbpy import IdaDataBase, IdaDataFrame

def __iter__(self): return 0

client_xxxx = ibm_boto3.client(service_name='s3',
ibm_api_key_id='xxxxxxx';
ibm_auth_endpoint="https://iam.bluemix.net/oidc/token",
config=Config(signature_version='oauth'),
endpoint_url='https://s3-api.us-geo.objectstorage.service.networklayer.com')

if not hasattr(body, "__iter__"): body.__iter__ =
types.MethodType(__iter__, body)

df = pd.read_csv(body)
  
```

The dataset will be exported as CSV file format.

d) *Select Model Type:* As we are working with model builders, we need to define the parameters for our training models [19]. The targeted output from our dataset is “Send\_SMS” column which carries two class - “Yes” and “No”. For this binary feature, we created “Binary Classifier” for our first model. This model classifies whether to send the SMS to the doctor or not. The estimator and algorithm for the model is - Logistic Regression.

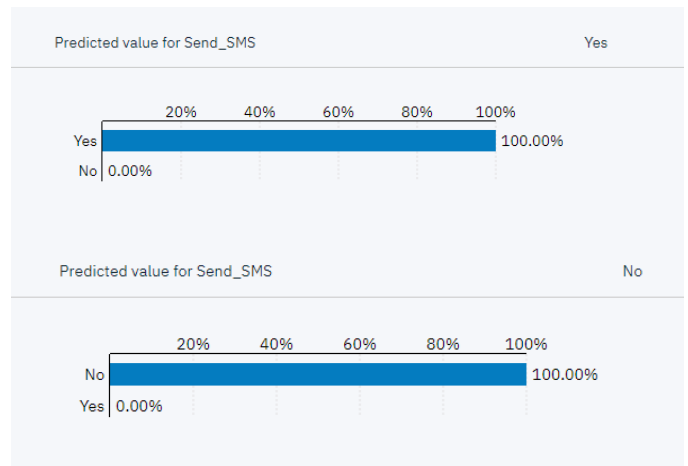


Fig. 7. Logistic Regression prediction for feature “Send\_SMS”

Once we set the estimators and the parameters, the Watson ml engine automatically trained the models according to our actions and provided the training score. Based on that score, the IPMS send SMS to the duty doctor.

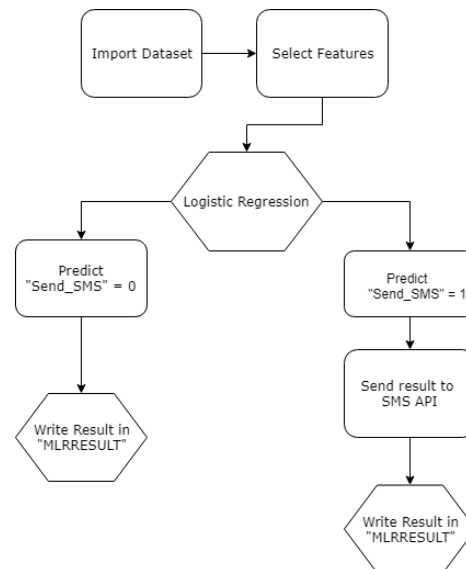


Fig. 8. Logistic Regression integration with SMS API

We will choose another model for the features “Ventilator” and “Condition”. Ventilator score provides the necessity to install the ventilator for the specific patient. Condition feature will give the prediction of the health condition of the patient whether it will increase or decrease or stay stable. If the model predicts more than 80% chance to decrease the condition, the SMS API will send the corresponding results to Doctor and Nurse. “Fig. 9”, illustrates the whole process with SMS API.

For both of these two features, we implemented “Naive Bayes” and the estimators are - Decision Tree Classifier. The estimators basically work as the base predictors for the models [28].

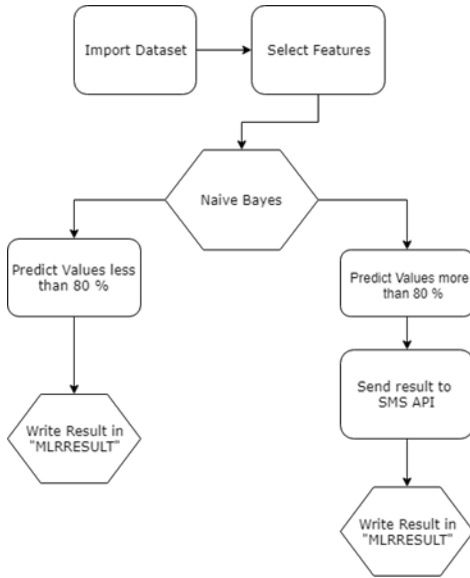


Fig. 9. Naive Bayes integration with SMS API

e) *Deployment of the models:* For ensuring the continuous evolution of our dataset on a large scale and in a selected time scale, we needed to set a deployment of the ml models. For our mobile app and automated dataset series, the “Batch Prediction” performed the best [20]. Batch Prediction deployment is auto-deployed in a selected time interval with all the models that we selected for training. For the deployment, the parameters are following-

TABLE I. Deployment Connection

Input Connection	Output Connection
Type - bluemix cloud object storage	Type - dashdb
{ "firstlineheader": true, "file_name": "REGISTRATION.csv", "infer_schema": "1", "file_format": "csv", "type": "bluemixcloudobjectstorage" }	{ "type": "dashdb", "tablename": "MLRESULT", "schemaname": "NDXXXXX", "writemode": "write" }

Once we selected the parameters for the deployment, the models auto-deployed in selected time series. We can evaluate the deployment from the “Status” section, if the Status is “Successful” then the deployment is running successfully and if not, we need to debug and fix the prominent errors to make it running.

Deployment	
Name	ML_DB
Type	Web Service
Deployment ID	f7fb88dd-14fb-4ffb-a3ff-7f85dd3a0349
Status	DEPLOY_SUCCESS
Asset type	model
Asset name	ML_Model_forDB
Machine learning service	pm-20-cw
Created	26 Nov 2018 12:45am
Last modified	11 Dec 2018 08:46pm

Fig. 10. Deployment Credentials

f) *Create entities for the Models:* To store the deployed ml model’s evaluation, the central database needs to modify afterward. The “MLRESULT” table stores the written results from the ml models. From this table, both our mobile app IPMS and SMS API accessed the predicted results and did their functions accordingly.

TABLE II. Entities of Machine Learning models

Patient ID	Condition (Last Feedback)	Send_SMS	Naive Bayes	Naive Bayes Inbuilt	Logistic Regression
01	Normal	No	72.9	55.6	0
09	Critical	Yes	85.6	80.6	1
155	Normal	No	80.7	56.6	1
147	Decreasing	Yes	98.6	88.6	1

The table allocates each result in a different column. The ml models scores indicate the testing score of the individual models. In logistic regression, “0” indicates “No” and “1” indicates “yes” in column Send\_SMS. The table allocates each result in a different column.

#### D. Connecting the Android Application with IBM:

To make the output visible, more user-friendly and under stable, we introduced our mobile application IPMS [21]. Which is for both doctor and patients relative. It can make a new perspective to monitor patient even away from the hospital. It can also relieve the relative of the patient by monitoring patient condition even from home or from a remote place. The UI is very friendly and easy to access. Our app is only utilized as an output device. It is not for entering new data in the database.

a) *Collect dataset from IBM Cloud:* The full app is based on a database from IBM Cloud. The admin will provide

“Username” “password” “Hostname or IP Address” to connect the app with a database. After connecting the database from login to patient all will be displayed as an output of ml model. Every feature of the ml model database is used here. These connections are auto-refreshed after each data is pushed or ml models results are updated to the cloud. From “Fig. 11”, we can see an overview of the working model.

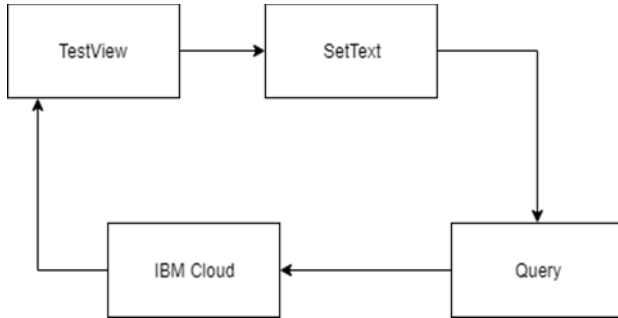


Fig. 11. Android data fetch workflow.

b) *Login UI*: We made the login panel as reliable as possible. We did not divide the UI model for doctor and patient. Both doctor and patient can log in from same Login UI. Username and access code easily identify the doctor and patient. The access code is manually generatable from hospital authority. The first condition is based on ‘username’. If the username matches it looks for the access code. ‘Patient’ and ‘doctor’ cannot have the same access code. Thus, the user code can define who is a doctor and who is a patient. If the doctor access code matches it calls for an intent name ‘doctor’, same goes for the patient. Every patient will receive a different access code when they will be admitted and only the db administrator will have the access to provide these codes [22]. After deciding the patient or doctor, two different UI model will be opened.

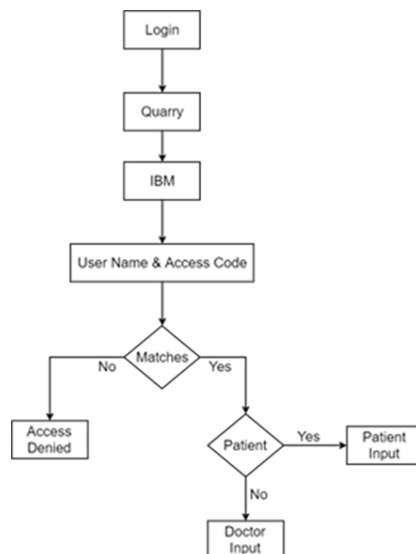


Fig. 12. Login UI principle

c) *UI Doctors*: Doctor has full access to patient data. Can easily monitor all the patient condition at a glance as a short description. All the patient list under the doctor will be shown on a simple UI. We selected patient admitted date, age, sex, primary disease and name for short description. We circulated a linear layout to make the long list visible so that the doctor can scroll down [23] easily. The simple query generated to collect date, age, sex, primary disease etc. All this info is collected from the “Registration” table. All these data are shown by “textview” and “button view” of the android studio. If the query fulfilled the condition “Settext”, then it has been called to place the data on “textview”. To access the full condition of the patient, the doctor needs to select the “Details” button. “details” button can call new “Intent”. As we are representing all the patient list at a glance, shortlist of functions has been introduced. The doctor has no access to enter new data. There is an emergency condition that a doctor can call the nurse, who is allocated for a patient. “Emergency” button can access the nurse number under the patient. The query can identify the number from the “Registration” table through “Fetch” function.

d) *UI for Patients*: Hospital has their own privacy over data and the relative of a patient also have the right to know the condition of a patient [24]. Considering this, we designed the patient UI so that the patients relative can see the details and not violating the privacy of the hospital. So, we filtered the dataset and choose those data that needs to display only to visualize the current condition of a patient. Here also “textview” has been set. Then the “settext” changes the content according to a different patient. This time only “Registration” table has been accessed to identify the patient's details.

The Android UI is specially designed for the doctor. All the info of the patients will be displayed at a glance with details. All the problems like blood pressure, heart rate, current condition etc. depends on patient disease will be shown here. Our ml prediction results also placed here. The doctor can easily monitor a patient from here and can have time to time update. In a short, the whole patient's parameters are shown here. The query can fetch all the info of a single patient via “textview” and “settext” functions.

### E. Integration of SMS Sending System

In order to inform about the patient condition to doctors and patient relatives, we regulated an SMS system. For this, we connected an SMS API named “BD Bulk SMS”. The API can send bulk SMS to any local operators. With the API we can send unlimited numbers of SMS from our system with negligible time difference because of using the IBM Watson SMS API gateway [25]. For the process, the system collected the prediction of the ml model from “MLRESULT” table and also “Full\_Name”, “Patient\_ID”, “Send SMS”, “Contact\_Number”, “Ventilator” and “Last\_Feedback” columns from the dataset. If the Send\_SMS column is “Yes” then the system collects corresponding doctors and the

relative’s mobile phone number in the following 11-digit phone number format “01XXXXXXXXXX”.

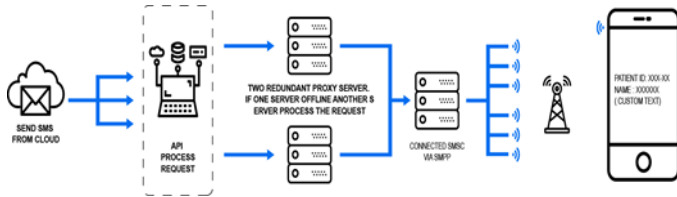


Fig. 13. SMS API working process

The column “condition” carries 4 values - Normal, Critical, Increasing and Decreasing of fallen. Based on the detected column values, the SMS is requiring to create in a single String. The API will send selected text based on patient condition, i.e. “Dear Mr. “Doctor\_Name”, “Parameter” of Patient ID: “Patient\_ID, has “MLRESULT” in the latest report. Please advise a course of action”.

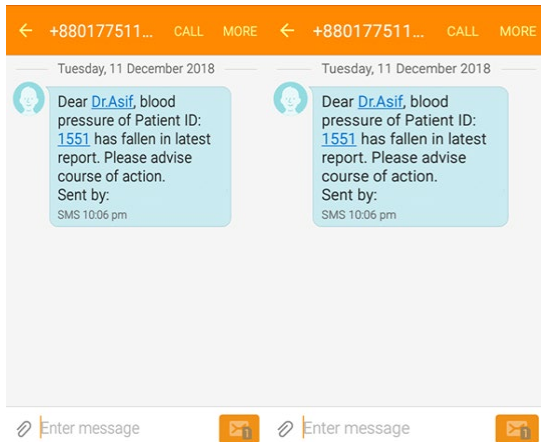


Fig. 14. Sample SMS for critical patients

After sending the SMS the system moves to the first phase and repeat the process again based on the ml model prediction.

#### F. The Central Control Unit (CCU)

The CCU basically comprised of IBM Cloud Database, IBM Watson Studio of ml models, Android Application IPMS and SMS API. CCU is a base store for all the input vector for the Mobile Application and the Machine learning models. There is a web interface for data input for the condition of the patient. This data will be updated by duty doctor for every 6 hours. Furthermore, this section will be using for the “Test Result” store section.

The “PatientUpdate” table will store all these data from daily updates. From the cloud storage, SMS API will access this table and send the output through mobile application IPMS. Accessing the cloud through SMS API is a complex process. Some of the security credentials like password, id, hostname needed to update from cloud and Watson studio. For this process “Configure” table was created. To access

credentials, the SMS API only needed to access this table and save this information in CSV format through SQL query.

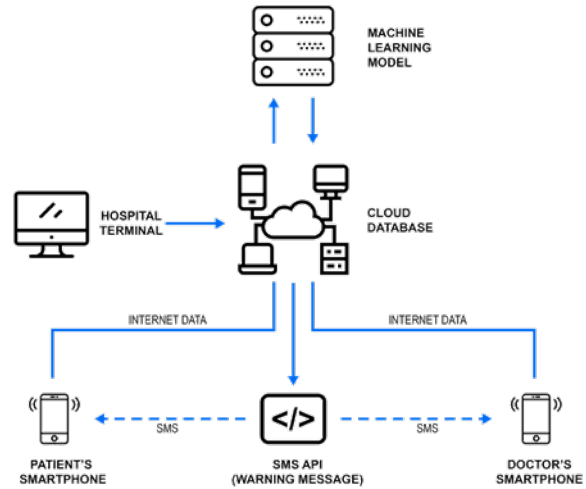


Fig. 15. CCU Architecture

## V. RESULT TESTING AND ANALYSIS

Testing part is the most crucial part of this project. As per as the nature of this project, we had to follow the waterfall model to deliver each feature [28]. Firstly, we tested the IBM Cloud data entry process. In this process, we had to fix several Cloud issues to avoid data loss [29]. We pushed in total almost 700 sample patient info from both Hospital terminal as registration and daily update schema. we found the following statistics from data schema.

TABLE III. Cloud Entry Testing

Service	Schema	Total Rows	Successful	Success Rate
NDX9073	Registration	522	489	93.67
NDX9073	Patient Update	175	165	94.29

IBM Cloud showed more than 90% data successfully converted from the terminal to cloud. From this data, the Watson studio carried out the machine learning model accuracy and validity.

TABLE IV. Machine Learning Models Accuracy

Model	Precision	Recall	f1-score	Support
Naive Bayes	0.95	0.94	0.93	200
Logistic Regression	0.90	0.91	0.92	75



Avg accuracy from the models varied from 93%-96%. For Validation of the models we established the Confusion Matrix and the Receiver operating characteristic (ROC) curve [30] [31]. We observed the following results-

TABLE V. Machine Learning Models Validation

Model	Confusion Matrix
Logistic Regression	$\begin{bmatrix} 50 & 2 \\ 1 & 22 \end{bmatrix}$
Naive Bayes	$\begin{bmatrix} 142 & 12 \\ 10 & 48 \end{bmatrix}$
Naive Bayes Inbuilt	$\begin{bmatrix} 12 & 20 \\ 5 & 13 \end{bmatrix}$

We carefully observed the ROC Curve, whether it could show the accurate areas under prediction. The ROC Curve brought out most of the areas under True Positive Rate.

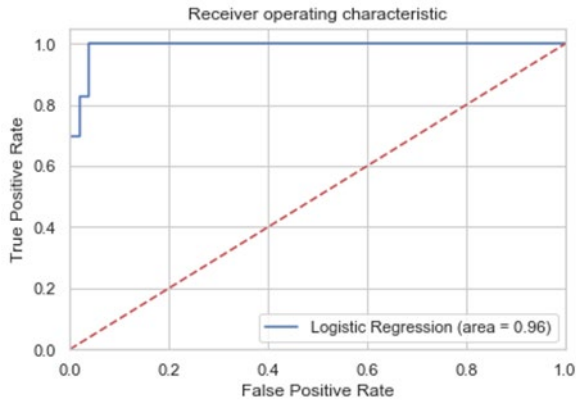


Fig. 16. ROC Curve

Deploying the machine learning models was the last test to satisfy all the Watson Studio process. We chose hourly parameter as deploy condition. All the models will be deployed when the hourly update of the patient will be input from the hospital terminal. For each deploy models, all the accuracy will be validated through the evaluation curve.

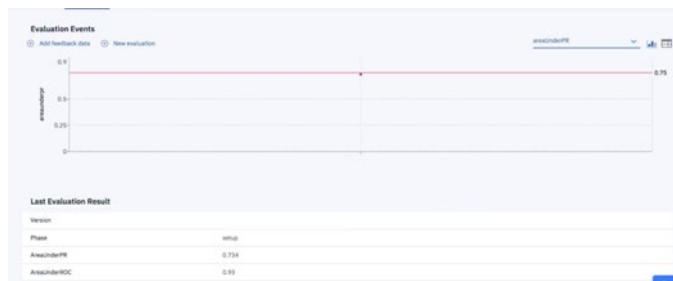


Fig. 17. Auto-deployment of models

After deploying the model, we wanted to see the result in person by examining in ICU. To do so, we visited Bangladesh Medical College Hospital (BMC), Uttara, Dhaka to collect live samples reading and run the machine learning models. After examining 12 hours, we got the following results –

TABLE VI. Hospital testing reports

Patient age	Primary Disease	Model Prediction	Actual Condition
56	Heart Disease	Normal	Normal
63	Stroke	Decreasing	Normal
6	Unknown	Decreasing	Decreased
47	Stroke	Normal	Increased

From the 12 hours survey, we collected in the total condition of 6 patient, among the prediction 4 were correct and 2 were wrong. As we only used 7 parameters to examine the condition, the results were satisfactory for small-scale prediction.

As the predicting results showed satisfactory scenario, we moved on to test this results in predicting health condition and sending SMS task. Logistic regression predicted 560 SMS needed to be sent as per as the condition forecast, among them 485 was the valid situation and 48 times the model predicted wrong and 27 times the SMS sending was failed.

TABLE VII. SMS API testing

Total SMS	Sent	Right Prediction	Wrong Prediction	Failed Attempt
560	533	485	48	27

Nearly 90% accuracy rate also worked for this SMS API Testing. Most of the wrong prediction case occurred for “Normal” and “Increasing” condition. This indicates the correlation between all the health parameters wasn’t distinguished successfully. For failed attempts of sending, the SMS API failed to grab the contact number from central cloud storage. So, the reason couldn’t be identified. This can either occur for network breakdown or also can occur for central server responding issues.

For our Android app verification, we examined each feature separately and it worked perfectly for most of the parameters. Only in some cases, like the “Patient Condition” section took longer to update the reading. The machine learning models deployed in time but the android “Get” function failed to read from the table.

Also, we took a survey among our classmates and faculties in North South University, Dhaka to give feedback about the user interface and features, the table shows the survey results and feedback.

TABLE VIII. Audience Feedback

Total participant	Good Interface	Good Features	Need to improve	Not good at all
32	23	22	11	5

According to their suggestion, we updated and reassembled our android app user interface and features to make it more reliable.

### CONCLUSION

Health Sector is one of the most rapidly growing field in science and engineering. Most of the developed countries in the world catch the latest technological tools to improve the quality of the treatment. In this case, the developing and underdeveloped countries in South Asia like Bangladesh suffers the most. To provide better treatment we require more advanced technologies with very low cost. We started this project to bring out a good result in the hospitals to serve the patient. We used some of the existed techniques and technologies to give a new shape in the hospital and nursing sector. Most of the ml models scored varied from an accuracy of 90% to 96%. The lowest accuracy obtained is 90%. The IBM Cloud showed good promising actions by keeping more than 90% success rate. Altogether the results we obtained from our project and experiments are showing promise to rise this system in large scale for urban and low economical side peoples. With the help of this project, a virtual doctor can be established to serve the people better and monitor patients with appropriate care. This is also a decision-making assistant for the doctor. As we have established this project with very few parameters of the physical segments, we can improve this project more by adding full parameters to measure the human body circulations. In the future, we are planning to install an embedded system to take a live reading from Ventilator, Medicine Pump, Heart Monitor, and other ICU machines. This will also increase the overall working accuracy of this project.

### ACKNOWLEDGMENT

We would like to express my very great appreciation to Bangladesh Medical College and Dhaka Eye Care Hospital authorities. Also, the students and faculties of North South University for helping us in the survey.

### REFERENCES

[1] Gardner R.M., Shabot M.M. (2006) Patient-Monitoring Systems. In: Shortliffe E.H., Cimino J.J. (eds) Biomedical Informatics. Health Informatics. Springer, New York, NY

[2] S. Brett, T. Gould, P. McNaughton, Z. Puthuchery, & V. Nangalia (Eds.). (2015). Handbook of Mechanical Ventilation A User’s Guide. London: The Intensive Care Foundation.

[3] Ahmed, S. (n.d.). BREAST CANCER: PRESENTATION AND LIMITATION OF TREATMENT – BANGLADESH PERSPECTIVE. doi:10.4172/1948-5956.S1.041

[4] Clarke, F & McDonald, Ellen & Griffith, Lauren & Cook, D & Mead, M & Guyatt, G & Rabbat, Christian & Geerts, W & Arnold, D & Warkentin, T & Crowther, Mark. (2004). Thrombocytopenia in medical–surgical ICU patients. Critical Care. 8. 1-1. 10.1186/cc2592.

[5] "Rational Unified Process", URL: [online] Available: [https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251\\_bestpractices\\_TP026B.pdf](https://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251_bestpractices_TP026B.pdf).

[6] Anwar Islam, Tuhin Biswas. Health System in Bangladesh: Challenges and Opportunities. American Journal of Health Research. Vol. 2, No. 6, 2014, pp. 366-374. doi: 10.11648/j.ajhr.20140206.18

[7] Jlelaty M, and Monzer Y, Factors in Cloud Computing adoption, Master’s thesis, Lund University, 2012

[8] Joarder, T. (n.d.). Retaining Doctors in Rural Bangladesh: A Policy Analysis. Retrieved from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6186485/> DOI:10.15171/ijhpm.2018.37

[9] Clear DATA. (2018). How Machine Learning is Helping us Get Smarter at Healthcare [Brochure].

[10] Choi, N. G., DiNitto, D. M., & Kim, J. (2014). Discrepancy Between Chronological Age and Felt Age: Age Group Difference in Objective and Subjective Health as Correlates. Journal of Aging and Health, 26(3), 458–473. <https://doi.org/10.1177/0898264314523449>.

[11] W Haider, Agha & G Larson, Martin & S Franklin, Stanley & Levy, Daniel. (2003). Systolic Blood Pressure, Diastolic Blood Pressure, and Pulse Pressure as Predictors of Risk for Congestive Heart Failure in the Framingham Heart Study. Annals of internal medicine. doi:138. 10-6. 10.7326/0003-4819-138-1-200301070-00006.

[12] Catalog-IBM Cloud. (2016, May 01). Retrieved from <https://console.bluemix.net/catalog/IBM>.

[13] Db2 Warehouse. (n.d.). Retrieved from <https://www.ibm.com/us-en/marketplace/db2-warehouse>.

[14] Generating service credentials. Retrieved from [https://console.bluemix.net/docs/services/ObjectStorage/os\\_credentials.html](https://console.bluemix.net/docs/services/ObjectStorage/os_credentials.html).

[15] Rish, I. An Empirical Study of the Naive Bayes Classifier. In IJCAI-01 Workshop on Empirical Methods in AI. 2001.

[16] Hussam Abu-Libdeh, Lonnie Princehouse, Hakim Weatherspoon, RACS: a case for cloud storage diversity, Proceedings of the 1st ACM symposium on Cloud computing, June 10-11, 2010, Indianapolis, Indiana, USA [doi: 10.1145/1807128.1807165].

[17] Das, Sudipto & Nishimura, Shoji & Agrawal, Divyakant & El Abbadi, Amr. (2010). Live Database Migration for Elasticity in a Multitenant Database for Cloud Platforms.

[18] D. A. Ferrucci, "Introduction to this is watson", IBM Journal of Research and Development, vol. 56, no. 3. 4, pp. 1-1, 2012.

[19] H. Wu, J. M. Mendel, "Classification of battlefield ground vehicles using acoustic features and fuzzy logic rule-based classifiers", IEEE Trans. Fuzzy Syst., vol. 15, no. 1, Feb. 2007.

[20] D. Wang, Robust data-driven modeling approach for real-time final product quality prediction in batch process operation. IEEE Trans. on Industrial Informatics, Vol. 7, No. 2, May. 2011.

[21] Jason. (2017, July 05). Benefits of mobile apps. Retrieved from <https://vidvidus.com.au/insights/benefits-of-mobile-apps/>

[22] Devi N.D, A. (2018, November 02). Database Administrator Roles and Responsibilities in an Organization. Retrieved from <https://www.mercurysolutions.co/blog/roles-and-responsibilities-of-the-database-administrator>.

[23] Linear Layout | Android Developers. (n.d.). Retrieved from <https://developer.android.com/guide/topics/ui/layout/linear>

[24] Patient Rights and Responsibilities. (n.d.). Retrieved from <http://www.uhsystem.com/Conway/patient-rights-and-responsibilities>.

[25] Getting started with SMS Gateway. (n.d.). Retrieved from [https://www.ibm.com/support/knowledgecenter/en/SS4U29/sms\\_getting\\_started.html](https://www.ibm.com/support/knowledgecenter/en/SS4U29/sms_getting_started.html).

[26] P. Griffiths, A. R. Saucedo, P. Schmidt, G. Smith. Vital signs monitoring in hospitals at night. (n.d.). Retrieved from <https://www.nursingtimes.net/clinical-archive/assessment-skills/vital-signs-monitoring-in-hospitals-at-night/5089989.article>.

- [27] What is the Jupyter Notebook? (n.d.). Retrieved from [https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what\\_is\\_jupyter.html](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html).
- [28] Overview: Estimators, transformers and pipelines - spark.ml. (n.d.). Retrieved from <https://spark.apache.org/docs/1.6.0/ml-guide.html>.
- [29] Data Loss Prevention and Encryption. (n.d.). Retrieved from <https://www.ibm.com/security/services/endpoint-data-protection>.
- [30] Simple guide to confusion matrix terminology. (2018, October 31). Retrieved from <https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
- [31] Plotting and Interpreting a ROC Curve. (n.d.). Retrieved from <http://gim.unmc.edu/dxtests/roc2.htm>.