Prediction of Health Problems and Recommendation System Using Machine Learning and IoT

Sanjay J P
School of Electrical and Electronics Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu, India
E-mail: sanjayprabahar@gmail.com

Tummalapalli Naga Deepak
School of Electrical and Electronics Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu, India
E-mail: naga2deepak@gmail.com

Manimozhi M
School of Electrical and Electronics Engineering
Vellore Institute of Technology
Vellore, Tamil Nadu, India
E-mail: mmanimozhi@vit.ac.in

2021 Innovations in Power and Advanced Computing Technologies (i-PACT) | 978-1-6654-2691-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/i-PACT52855.2021.9696622

***Abstract* – Nowadays, people are getting infected with various diseases due to environmental conditions and their improper lifestyles. As a result, more people search online for health-related facts such as illnesses, diagnoses, and remedies. If it is possible to make a recommendation framework for doctors and medical users by using medical data, it will save much time. The most challenging task is the accurate prediction of the illness. In the classical method of diagnosis, the patient needs to see a doctor, and the doctor conducts numerous diagnostic tests and comes to a conclusion. This procedure takes both doctor and patient time. This paper proposes a Graphics user interface-based disease prediction system that utilizes seven classification Machine Learning algorithms to detect and forecast the problem based on symptoms. It also suggests alternate treatments and provides detailed information on the predicted disease. It also uses NodeMCU and ThinkSpeak to monitor the heart and temperature values over the internet, and it records all of the information in a real-time database that a doctor or user can access in the future.**

Keywords: Healthcare, NodeMCU, Real-time database, Graphics user interface, ThinkSpeak, Machine Learning, Python

I. INTRODUCTION

Today, Machine Learning is more needed in the healthcare and medical sectors. The healthcare industry generates a lot of data regarding clinical evaluations, patient reports, cures, follow-up appointments, medicine, and so on. When such Machine Learning techniques are applied correctly, helpful knowledge can be derived from vast databases, allowing medical practitioners to make more informed decisions and enhance health care. The goal is to assist the physician and patients by using the classification techniques.

Classification is a technique for categorizing data into a set of categories. The primary purpose of a classification challenge is to determine which category/class new data will belong to. The new data in the paper are symptoms based on which the disease can be predicted. The class is the common diseases faced by people. Proper classification can be done only if we have created a proper dataset. Based on the dataset, the accuracy and result can be improved. Depending on algorithms cannot improve the accuracy. Good mining and feature extraction should be done to a dataset in order to improve the score.

Machine learning has been used in a number of detailed studies and initiatives. D. Dahiwade, G. Patle, and E. Meshram[1] published a study that examined the accuracy of CNN and KNN predictions. W. Hong, Z. Xiong, N. Zheng, and Y. Weng [2] published a work in which they used prior medical history to forecast approaching disease using a deep-learning-based hybrid recommendation algorithm. As a result, we devised the idea of creating a database to store the user's data, which could then be used for future applications. S. Grampurohit and C. Sagarnal [3] presented and compared decision tree, random forest, and naive Bayes algorithms for multi-class classification problems, which let us learn more about the algorithms and how to apply them. Other research publications and articles [4],[5],[6] were referred to how to build a model framework and how to use data mining. The authors M. Patil, V. B. Lobo, P. Puranik, A. Pawaskar, A. Pai, and R. Mishra [7] have documented the usage of SVM in multi-class classification problems in detail, which we used to transform multi-class classification problems to multi binary classification problems. Also, in order to take care of patients' health and monitor their health, we developed hardware that measures heart rate and temperature. The usage of IoT sensors mixed with algorithms has been proposed and addressed by Arnab Dey, Pramit Brata Chanda, and Subir Kumar Sarkar [8]. We also found a few research papers [9], [10], and [11] that helped us comprehend how IoT monitoring and data analysis could be advantageous. As a result, the user can provide symptoms to help predict disease and monitor his heart rate and temperature, with data being transferred and visualized through IoT.

II. OBJECTIVE

The primary goal is to use symptoms as input to predict disease. Other parameters such as height, weight, and glucose level are used for BMI calculation and to ensure that blood sugar levels are stable. As a result, each of the seven algorithms utilized in this paper predicted the best potential outcome. As we use more algorithms to predict, the most common output from the above algorithms will be the primary disease, and we can conclude primary disease by cross-checking with suggestion information in the Graphics user interface. Machine learning algorithms used are decision tree, Random Forest, Naive Bayes, K-nearest neighbor (KNN), Support Vector Machine (SVM), Multinomial Logistic Regression, and Multilayer Perceptron (MLP).

Implement a real-time database, which stores patient data from the Graphics user interface. Hardware that measures and monitors the heart rate and temperature of the user, which can be visualized in IoT cloud ThinkSpeak. If there is any abnormality, an email will be sent by IFTTT, which is interfaced in Think speak by API key, and a buzzer will be ringed to alert the patient. Graphics user interface-based input/output makes it easier for the user to enter data and see the results. Provides detailed information regarding the problem and also recommends when to see a doctor.

III. EXISTING SYSTEM

A literature survey was conducted to compare different algorithms and ways of the approach used for different classification problems in real-time. In the existing paper, they have proposed based on two to three algorithms. However, it had a flaw in terms of prediction. When the code was executed, each one yielded a different result for most of the symptoms.

The main advantage of our system is that seven algorithms were used in this paper to predict the best possible output. As more algorithms are used to predict, the primary disease will be the most common output from the above algorithms, and the output from the suggestion information will be equally crucial in determining whether the primary disease is true or not. In the Graphics user interface, other necessary details are added for the user as well as the doctor to check in the future for further details.

IV. METHODOLOGY

A. Dataset

	A	B	C	D	E	F	G	H	I	J
1	Diseases	Fever	Chills	Sweating	Headache	Vomiting	Muscle_p	Rapid_bre	Cough	Diarrhea
2	Malaria	1	1	1	1	1	1	0	0	1
3	Malaria	1	1	0	1	1	1	1	1	1
4	Malaria	0	0	0	1	1	1	0	0	1
5	Malaria	1	1	1	1	1	1	0	0	1
6	Malaria	0	0	0	1	1	1	0	0	1
7	Malaria	1	1	1	1	1	0	0	0	0
8	Malaria	1	1	1	1	1	1	0	0	1
9	Malaria	1	0	1	1	1	1	0	0	0
10	Malaria	1	1	0	1	1	1	0	1	1
11	Malaria	1	1	0	0	0	0	1	0	0
12	Jaundice	1	0	0	1	0	0	0	0	0
13	Jaundice	1	0	0	0	1	0	0	0	0
14	Jaundice	1	1	0	1	1	0	0	0	1
15	Jaundice	0	0	0	0	1	0	0	0	0
16	Jaundice	1	1	0	0	0	0	0	1	0
17	Jaundice	1	0	0	0	1	0	0	0	0
18	Jaundice	1	0	0	0	0	0	0	0	1
19	Jaundice	1	1	0	0	0	0	0	0	0

Fig. 1. Dataset in excel sheet

We created a word table of the most common diseases that humans confronted with and narrowed the list down to 29 diseases. Then, we conducted research and data mining for each disease on many medical websites, including the World Health Organization, Mayo Clinic, Healthline, very well health, WebMD, Healthgrades, live healthily, eMedicineHealth, and the National Health Service. Then it was transferred to an excel sheet in the form of ones and zeros, as shown in Fig. 1. We created a dataset of 233 rows for 29 diseases and 107 symptoms, implying that 8 to 10 data were obtained for each disease.

B. System Architecture

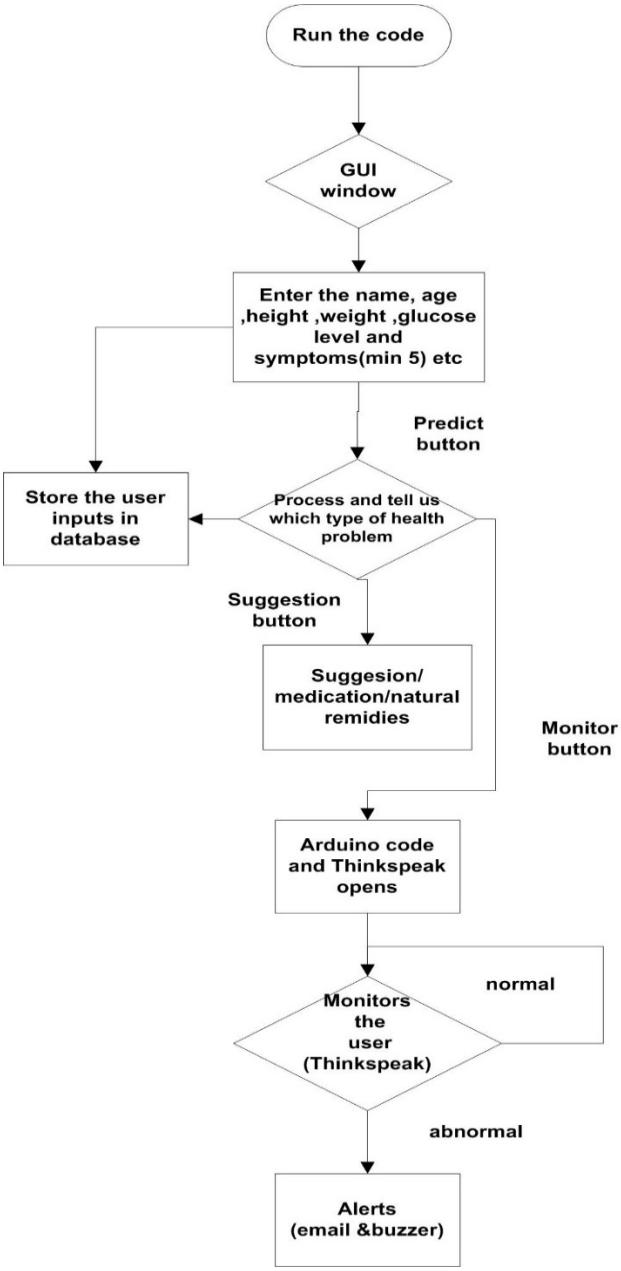


Fig. 2. System block diagram

Fig. 2 depicts a block diagram of the entire system workflow. When we run the code, the Graphics user interface appears where the user must enter their name, age, and all five symptoms in order for the code to run; otherwise, the Graphics user interface pops an error. When the user enters the details and clicks the predict button, the disease is displayed, and all of the information is saved in the database. If the user wants to check or monitor their heart and temperature, click the monitor button to open the hardware part. If users need detailed information to confirm the primary problem, they can click the suggestion button in the Graphics user interface.

C. Algorithm

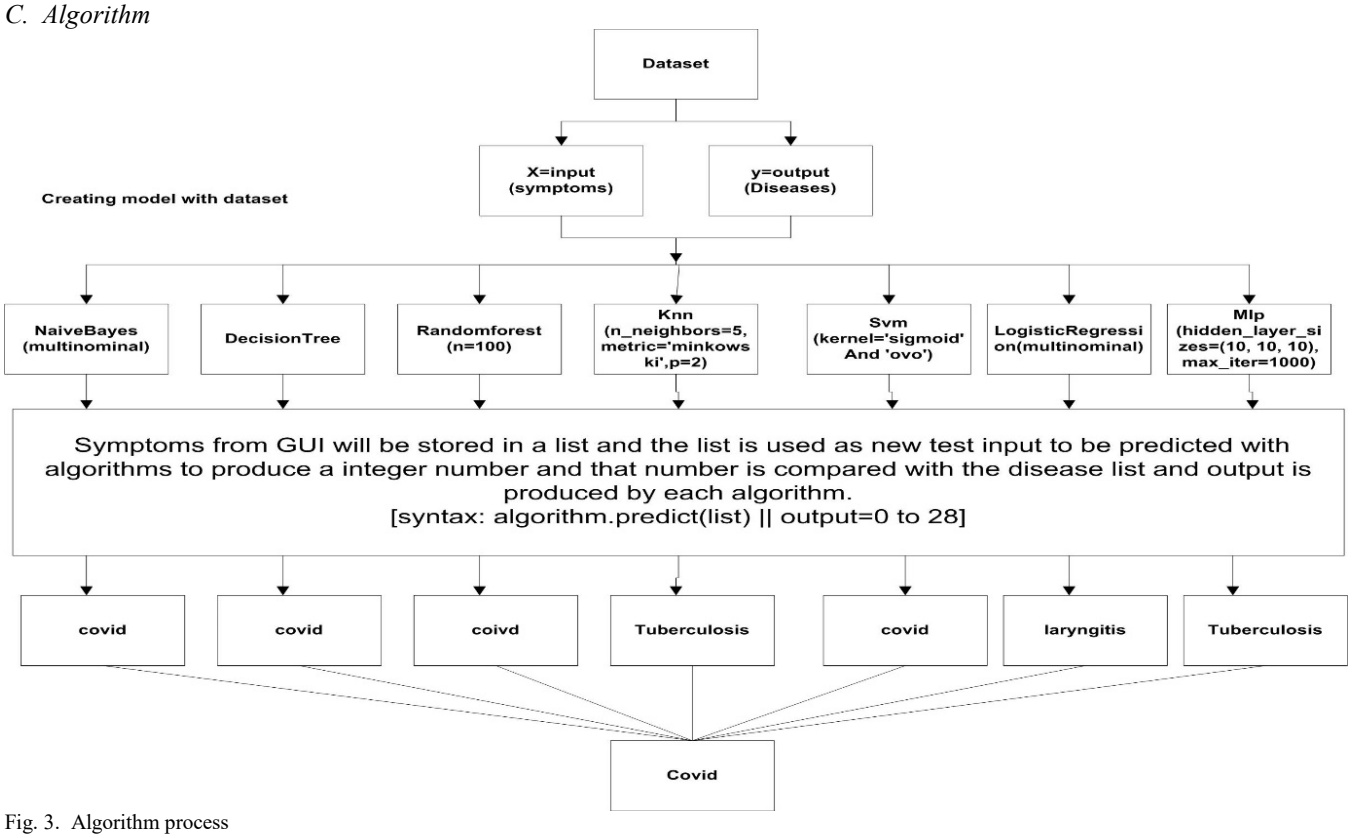


Fig. 3. Algorithm process

All algorithms used in our system are supervised learning classification ML algorithms, namely decision tree, Random Forest, Naive Bayes, K-nearest neighbor, SVM, Multinomial Logistic Regression, and Multilayer Perceptron.

Decision Tree is a traditional binary tree, with a Yes/No decision at each split before the model reaches the result node. Fig. 3 illustrates how a decision tree would solve for covid symptoms. As we enter each symptom, the decision tree examines if it is true or false, then makes a judgment and finishes the diagnosis.

Random decision forest is a model that is made up of multiple decision trees. When new data is entered, all the decision trees anticipate the output, and the average of the output is the output of the random decision tree. As shown in Fig. 3, we chose n equals to 100, which means that when we enter the symptoms, all 100 decision trees will predict the output disease, and the average of all the output diseases will be the output of the random decision forest, and the disease is covid. Random decision forest range is taken between 64 - 128 trees. Here, we have taken 100 because we will have a good balance between ROC, AUC, and processing time which increases the accuracy.

Naive Bayes technique is a supervised machine learning algorithm that uses the Bayes Theorem to classify new data. Bayes theorem is a probability theorem used to find conditional probability. As shown in Fig. 3, when symptoms are entered in the naive Bayes algorithm, it classifies the symptoms and predicts the output to be covid because there is more likelihood for covid to occur. Here, we are using multinomial because we are dealing with a multi-class classification problem.

During the training process, K-nearest neighbor (KNN) simply stores the dataset, and when it receives new data, it classifies it into a very close group to the new data. It locates objects using Euclidean distance. The outcome value predictions are determined by searching the entire data set for k data nodes with identical values and determining the resulting value using the Euclidian number. When a new data point is given, KNN starts to find the most common using Euclidian distance to classify the new data point. As shown in Fig. 3, we used n neighbors to represent the number of data points to compare with, metric to represent Minkowski, and p to represent the technical value for Euclidean.

Support vector machine (SVM) algorithm uses the hyperplane, which is a line that divides data input nodes with different values, and the vectors from these points to the hyperplane may either support or oppose the hyperplane. As illustrated in Fig. 3, we have used sigmoid because our dataset is non-linear; the dataset is composed of 1's and 0's and output are also in 1's and 0's, which in turn converted to integer is shown in Fig. 3. Ovo (one vs one) is an approach where a multi-class classification model is converted to multi binary classification model, and the end, it uses majority voting and distance from hyperplane margin to conclude.

The logistic regression model is to predict categorical classification in or the likelihood of category membership on a dependent variable. The dependent variables are the symptoms entered in the Graphics user interface. Multinomial logistic regression is like a binary logistic regression, which

evaluates the probability of categorical inclusion using maximum likelihood estimation. Fig. 3 depicts the multinominal syntax used for identifying specific type of diseases.

Multilayer Perceptron is a feedforward artificial neural network model. MLP is made up of multiple layers, each of which is ultimately linked to the one before it. Between the input and output layers, there could be one or even more non-linear hidden layers. A perceptron takes in data, then passes it through an activation function to generate an output. Adding layers of perceptron's together, creating a multi-layer perceptron model. As shown in Fig. 3, hidden layer sizes are 10, 10, 10, defining three hidden layers with ten neurons each, and the activation function is ReLU because the dataset is non-linear and small.

D. Graphics User Interface (GUI)

The GUI framework was created using python language. Tkinter module was specifically used as it is simple and has huge geometry managers design and option. Python code was executed using Spyder ide.

The training CSV file was imported into python for training with algorithms. Then we replace the disease names with numbers from 0 to 28, and we split the data into input(X) and output(y) sets.

The main keyword for building the GUI is root. We can adjust the scale, color, name, and label by calling root.

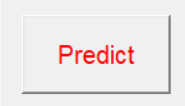


Fig. 4. Button

A button activates a specific function when it is clicked. In our GUI, we have a predict button shown in Fig. 4. Its primary function is to execute all the algorithms. The exact syntax was created for a number of buttons with different functions.

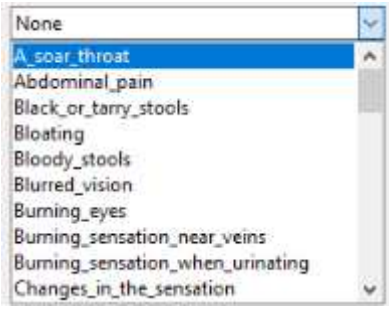


Fig. 5. Symptoms options

A Combobox is a combo of a list box and a text area. It is a Tkinter widget with a down arrow that allows you to choose from a list of choices. We used for symptoms option as shown in Fig. 5; It assists users in making decisions based on the choices presented. A pop-up of the scrolled Listbox is shown down the entry field when the user clicks on the drop-down arrow on the entry field. All symptoms are arranged in alphabetical order.

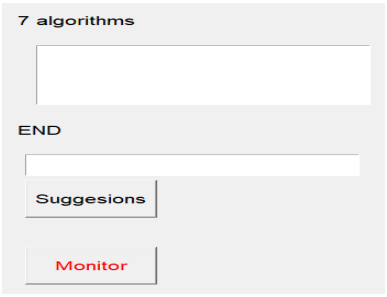


Fig. 6. Result viewer

Fig. 6 shows the blank space, where after the user clicks predict button, all seven algorithms' results will be printed, and the most common disease of all algorithms is printed in the below of END blank space.

Suggestions Button: When the user clicks suggestion button after executing the algorithms, a pop will appear and shows the user the detailed information for the predicted disease.

Monitor button: When the user clicks the monitoring button, it is the bridge that connects to the hardware portion It accesses the Arduino code and the ThinkSpeak website for our hardware section, which will be briefly discussed in the upcoming.

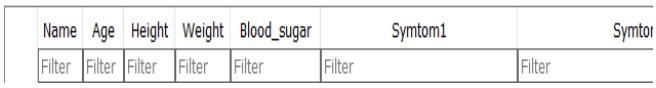


Fig. 7. Database creation

Fig. 7 depicts the construction of our database. We used SQLite database to store our details from GUI. Once we run the code, it produces the table contents as seen in Fig. 7 and database can be viewed by DB browser.

E. Hardware:

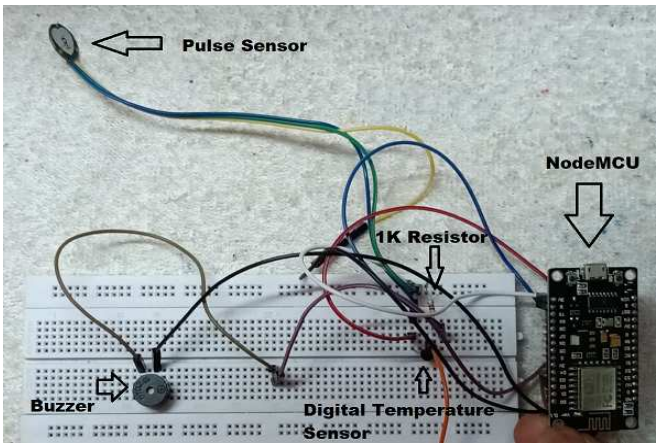


Fig. 8. Complete hardware setup

As shown in Fig. 8, the hardware components included a NodeMCU, a pulse sensor, a digital temperature sensor, a 1k resistor, a buzzer, a breadboard, and wires. We used

hardware to determine and monitor a person's heart rate and temperature. GUI (Graphic User Interface) is connected to the hardware through the monitor button. When we click the monitor button, the Arduino program with the code is launched, as well as the ThinkSpeak website. When we click the upload button in the Arduino software, the code is uploaded to NodeMCU. The code will be validated before being uploaded into the NodeMCU, which is linked to sensors. Sensor data such as heart rate and temperature will be obtained and printed on a serial display, after which the data will be transmitted to the ThingSpeak channel. A pulse rate chart and a temperature chart will be generated from the data in the ThingSpeak channel. Anyone with the ThinkSpeak account's user id and password can access the heart and temperature chart from anywhere. A buzzer is activated, and IFTTT email services are triggered when the heart rate or the temperature exceeds particular threshold set value.

V. RESULTS AND DISCUSSIONS

A. GUI:

Fig. 9. Vertical view of GUI

Fig. 9 show the complete structure of our GUI system. GUI consists of name, age, blood glucose, height, weight, five symptoms with the buttons to predict the disease based on the five symptoms, to check whether blood sugar level is average, BMI calculator, suggestions/remedies for the predicted disease, the monitor to open our hardware part, reset to remove all details in GUI and lastly exit button to quit the GUI.

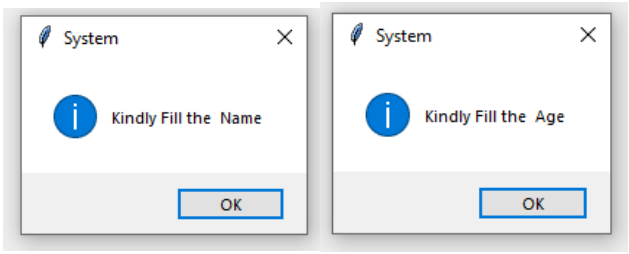


Fig. 10. Pop-up warning

If the user did not enter name or age in the GUI and clicked the predict button on GUI, a pop-up warning will appear as shown in Fig. 10.

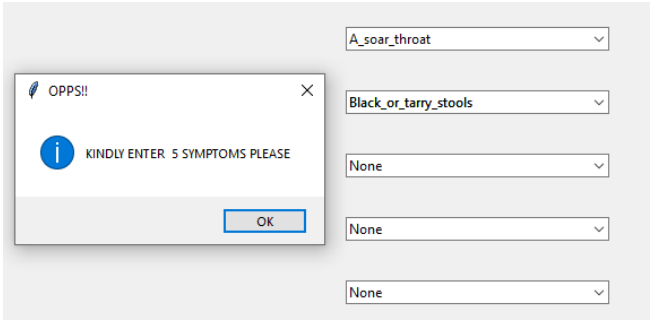


Fig. 11. Symptom warning

Fig. 11 shows that a pop-up alert appears when the user selects the predict button and does not enter all five symptoms. Its compulsory for the user to enter a name, age, and all five symptoms.

B. Blood Glucose:

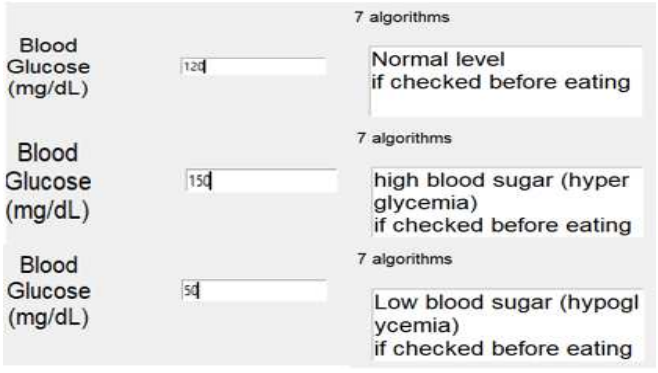


Fig. 12. Blood glucose level

For users to check their blood glucose state, once they enter the value and click the blood sugar button in GUI, it shows under 7 algorithms blank space. It shows normal, if it is between 70 to 140. If it is below 70, its shows low blood

sugar, and higher than 140 prints high blood sugar as shown in Fig. 12.

C. BMI

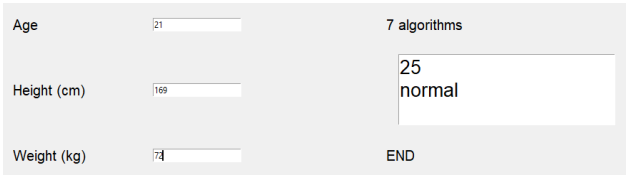


Fig. 13. BMI

To check the BMI of the user, whether it is normal or overweight or thinness, the user has to enter their height, weight and click the BMI button in GUI. Once it is clicked, it shows the BMI under 7 algorithms blank space as shown in Fig. 13.

D. Covid Symptoms Testing:

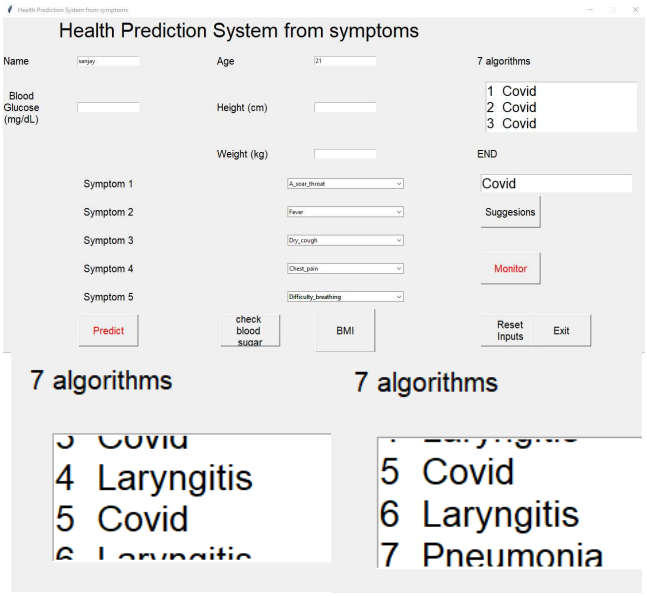


Fig. 14. Covid symptoms

The symptoms we used for testing covid are (fever, dry cough, sore throat, chest pain, and difficulty in breathing) from WHO. It's clear to see why we utilised seven classification algorithms to predict disease. As we can see in Fig. 14, not every algorithm generates covid for the entered symptoms. Only four algorithms are shown as covid, while the others show other diseases which are identical to covid symptoms. We can see that, it printed covid in the END space, which means covid is the most commonly predicted output by the algorithms. As Four algorithms out of seven predicted covid, it was printed in the END blank space.

E. Tuberculosis Symptoms Testing:

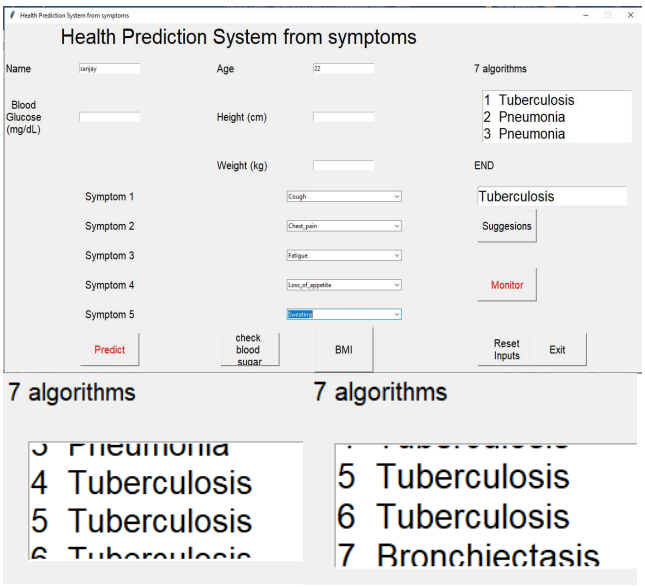


Fig. 15. Tuberculosis symptoms

Another testing example is show for tuberculosis symptoms in Fig. 15. Symptoms for tuberculosis were filled in the GUI option, and when predict button was executed, we got four algorithms to predict tuberculosis, and other algorithms predicted similar diseases.

F. Suggestion Button:

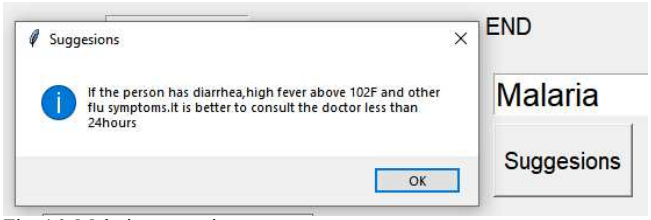


Fig. 16. Malaria suggestion

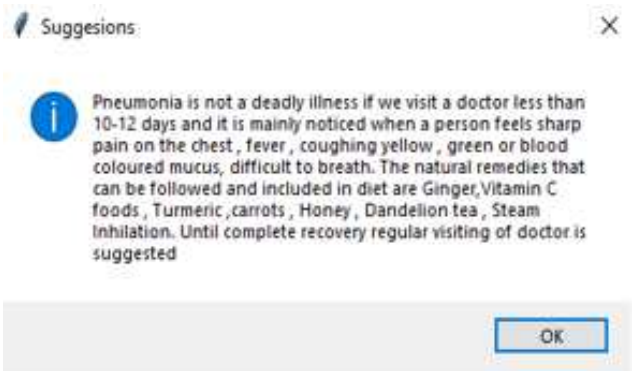


Fig. 17. Pneumonia suggestion

Fig. 16 shows suggestion for malaria, and another example of pneumonia is shown in Fig. 17. Once after predicted the disease, when we click the suggestion button in the GUI, it will suggest remedies and detailed information for the predicted disease. Based on the algorithm output and the suggestion details, we can decide and confirm the primary problem. All the suggestions were researched and collected from top medical websites and articles.

G. Database:

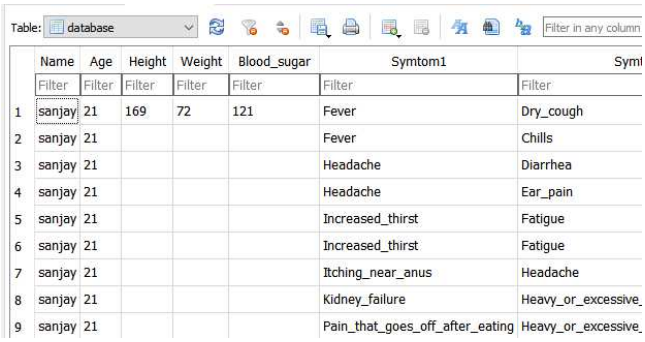


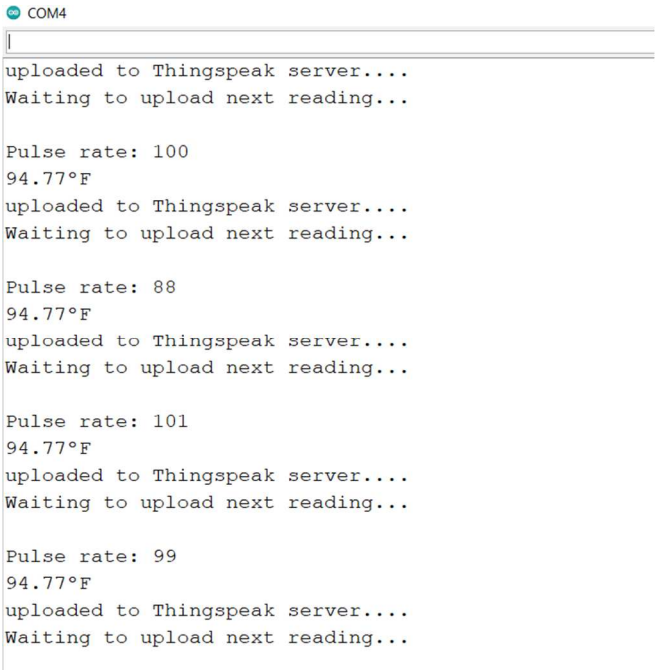
Table: database

	Name	Age	Height	Weight	Blood_sugar	Symtom1	Symt
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	sanjay	21	169	72	121	Fever	Dry_cough
2	sanjay	21				Fever	Chills
3	sanjay	21				Headache	Diarrhea
4	sanjay	21				Headache	Ear_pain
5	sanjay	21				Increased_thirst	Fatigue
6	sanjay	21				Increased_thirst	Fatigue
7	sanjay	21				Itching_near_anus	Headache
8	sanjay	21				Kidney_failure	Heavy_or_excessive_
9	sanjay	21				Pain_that_goes_off_after_eating	Heavy_or_excessive_

Fig. 18. Database

Once the user enters the five symptoms and clicks the predict button in GUI,It automatically takes all GUI details and stores them in the SQLite database. We use DB browser software to view the database to view SQLite tables, as shown in Fig. 18. user details such as the name, age, height, weight, all five symptoms, and the predicted disease are stored in the database. The size of the database file is in KiloBytes, which is easy to transfer and portable.

H. Serial Monitor:



COM4

```
uploaded to Thingspeak server....
Waiting to upload next reading...

Pulse rate: 100
94.77°F
uploaded to Thingspeak server....
Waiting to upload next reading...

Pulse rate: 88
94.77°F
uploaded to Thingspeak server....
Waiting to upload next reading...

Pulse rate: 101
94.77°F
uploaded to Thingspeak server....
Waiting to upload next reading...

Pulse rate: 99
94.77°F
uploaded to Thingspeak server....
Waiting to upload next reading...
```

Fig. 19. Serial monitor

The output of the sensors will be displayed in the serial monitor, where the first one is heart rate and the next is temperature shown in Fig. 19. We can also observe that the uploaded to ThinkSpeak message is displayed, which gets transferred to the ThingSpeak channel. For transfers, we use the API key of the created particular cloud account. In the ThingSpeak channel, it can be visualized in the charts as follows.

I. ThingSpeak

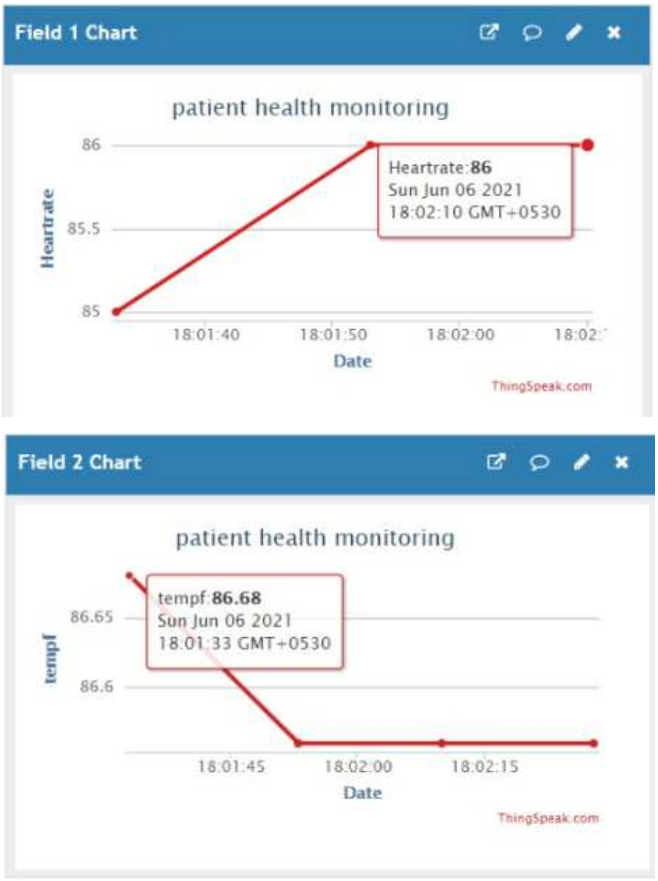


Fig. 20. ThinkSpeak chart

The values will be visualized and analyzed in charts, as shown in Fig. 20. One chart is for pulse rate, and the other chart is for temperature in Fahrenheit degrees. The X-axis is time, Y-axis is the heart rate for the heartrate measuring chart, and the temperature in Fahrenheit for the Temperature measuring chart. Both charts have red dots that display the date and time of the data obtained from the serial monitor. Anyone with the same API key and user ID can see the charts live from any location.

J. Email Alert Through IFTTT and Buzzer Alert:

Fig. 21. IFTTT email alert

When our pulse rate or temperature exceeds particular threshold value, we will be alerted through email. The mail further specifies if and why it was caused. The email was activated as a result of a heart condition, and it also indicates the moment it was triggered is shown in Fig. 21. Same condition applies for buzzer.

K. Accuracy

TABLE I ACCURACY OF DIFFERENT ALGORITHMS

Algorithms	Disease-1 Covid	Disease-2 Malaria	Disease-3 Jaundice	Average
Naïve Bayes	0.970	0.970	0.970	0.970
Decision Tree	0.944	0.901	0.910	0.915
Random Forest	0.983	0.987	0.987	0.985
KNN	0.970	0.970	0.970	0.970
SVM	0.953	0.953	0.953	0.953
Multinominal Logistic	0.983	0.983	0.983	0.983
MLP	0.979	0.957	0.970	0.968

The average Accuracy for three different diseases is shown in Table I. As this paper is about multi-class classification, we tend to use the cross-validated method. On the dataset, we utilized the K fold cross-validation approach (cv=3) to test the performance of all algorithms. We may deduce from these results that all algorithms perform very well on the dataset. When compared to the other five algorithms, however, random forest and logistic may perform somewhat better. KNN, SVM, logistic and naive Bayes has shown the same accuracy for different diseases.

VI. CONCLUSION

It can be seen from the history of machine learning and its applications in the medical field that systems and methodologies have emerged that have allowed sophisticated data analysis through the straightforward application of machine learning algorithms.

This paper, therefore, proposes an automated disease prediction system that relies on user feedback to save the time needed and also money for the initial diagnostic symptom process, may check or monitor their heart and temperature over the internet and store their information in a table format that a doctor or user may refer to in the future. We have used seven machine learning algorithms and a suggestion summary to conclude the primary disease.

Due to the abundance of large data generated and processed by modern technology, Artificial Intelligence will play an even more important role in data analysis in the future and other health related sensors can be further used for gathering and processing data.

REFERENCES

[1] D. Dahiwade, G. Patle and E. Meshram, "Designing Disease Prediction Model Using Machine Learning Approach," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pp. 1211-1215 ,2019.

[2] W. Hong, Z. Xiong, N. Zheng and Y. Weng, "A Medical-History-Based Potential Disease Prediction Algorithm," in IEEE Access, vol. 7, pp. 131094-131101, Sep 2019.

[3] S. Grampurohit and C. Sagarnal, "Disease Prediction using Machine Learning Algorithms," 2020 International Conference for Emerging Technology (INCET), pp. 1-7, Jun 2020.

[4] S. Ambekar and R. Phalnikar, "Disease Risk Prediction by Using Convolutional Neural Network," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 2018, pp. 1-5, April 2019.

[5] P. S. Kohli and S. Arora, "Application of Machine Learning in Disease Prediction," 2018 4th International Conference on Computing Communication and Automation (ICCCA), pp. 1-4, 2018.

[6] S. Mohan, C. Thirumalai and G. Srivastava, "Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques," in IEEE Access, vol. 7, pp. 81542-81554, 2019.

[7] M. Patil, V. B. Lobo, P. Puranik, A. Pawaskar, A. Pai and R. Mishra, "A Proposed Model for Lifestyle Disease Prediction Using Support Vector Machine," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1-6,2018.

[8] A. Dey, P. B. Chanda and S. K. Sarkar, "Patient Health Observation and Analysis with Machine Learning and IoT Based in Realtime Environment," 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), pp. 196-201,2020.

[9] R. Ani, S. Krishna, N. Anju, M. S. Aslam and O. S. Deepa, "Iot based patient monitoring and diagnostic prediction tool using ensemble classifier," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1588-1593, 2017.

[10] R. L. Priya, A. Vaidya, M. Thorat, V. Motwani and C. Shinde, "SAARTHI: Real-Time Monitoring of Patients by Wearable Device," 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), pp. 194-199,2020.

[11] S. Bhoyar, N. Wagholikar, K. Bakshi and S. Chaudhari, "Real-time Heart Disease Prediction System using Multilayer Perceptron," 2021 2nd International Conference for Emerging Technology (INCET), pp. 1-4, 2021.

[12] S. S. Sarmah, "An Efficient IoT-Based Patient Monitoring and Heart Disease Prediction System Using Deep Learning Modified Neural Network," in IEEE Access, vol. 8, pp. 135784-135797, 2020.