2021 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C) | 978-1-6654-2569-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDI3C53598.2021.00051

CAMISA: An AI Solution for COVID-19

Bhanuteja G
Electronics and Communication
MVJ College of Engineering
Bangalore, India
bhanuteja.g@mvjce.edu.in

Kaustubh Anand Kandi
Micro Electronics Design Facility
URSC, ISRO
Bangalore, India
kaustubh@ursc.gov.in

Srinidhi K
Electronics and Communication
MVJ College of Engineering
Bangalore, India
srinidhikrao05@gmail.com

Paulson Premsingh S
Electronics and Communication
MVJ College of Engineering
Bangalore, India
paulsonpremsingh7@gmail.com

Dikshitha R
Electronics and Communication
MVJ College of Engineering
Bangalore, India
dikshitharavisha@gmail.com

Anil Kumar R
Electronics and Communication
MVJ College of Engineering
Bangalore, India
anilkumarr0475@gmail.com

***Abstract*—The COVID-19 pandemic has created an unparalleled need for remote patient monitoring and has primarily impacted the world as the mortality rate has increased rapidly. As long as coronavirus exists, mutations of the virus continue to happen, which also insists on the need for remote monitoring. Healthcare sectors require the help of many new technologies such as IoT, Artificial Intelligence, Neural Networks, and sensor technology which can play an important role. The proposed system predicts the COVID-19 symptoms in a patient with the integration of sensor technology and AI. This system is effective in solving the crisis. It includes a shirt and a mask measuring the heart rate, blood oxygen level, and respiration rate. In addition to this is the predictable AI model, where the symptoms predict whether the patient is COVID-19 positive or not.**

Keywords—COVID-19, IoT, Contactless Health Monitoring, Neural Network, Sensor Technology, Lilypad Arduino, Pulse Oximeter

I. Introduction

Coronavirus has caused much havoc around the world. As on May 20, 2021, there have been 165,565,058 confirmed cases, resulting in 3,434,004 deaths and disturbing life all around the world. The losses are exacerbating day by day. With very little research done globally on the pandemic since SARS COV-2, there is no cure and less vaccination until recent days. Controlling the growth by early testing individuals and quarantining the coronavirus-affected individuals is the only implicit solution against the infectious COVID-19. However the ease to proceed with this strategy in order to control the deadly virus is not viable. There is a high risk of spread of the virus in crowded places amongst people with low immunity is still a major concern. The need for testing this virus in an early stage is still a key differentiator in most of the countries in order to reduce the pandemic curve.

In many countries the vaccination drives are happening at a slow pace, this exposes the majority of the population to COVID-19. The availability of beds has also narrowed down in hospitals, and the health status of COVID-19 infected patients are not remotely monitored as expected. The need for patient monitoring which is both easy to use and also produces accurate results, is a solution that can dissolve the problem. We try to provide a solution that solves this issue which is very inexpensive for measuring the health status of a patient. We aim to produce an AI-based platform for monitoring the patient who has been infected by the coronavirus. The integration of the AI model on wearable devices solves the problem to some extent.

As coronavirus was recognised only in the year 2019, the information available on the symptoms was very less. Considering the above situation, the previous work on this problem is highlighted where algorithms used for detecting COVID-19 such as LSTM in [1] (Long Short Term Memory) and Fuzzy rule-based predictions, are having smaller datasets. In order to overcome this, we provide a solution that evaluates the symptoms experienced by an individual, by using a relatively larger dataset. We perform algorithms like backpropagation which is a simple technique and is fast in execution. This is an important aspect in resolving the COVID-19 pandemic by integrating the neural network model with the hardware that consists of a shirt along with a nebulizer.

When the patient uses our proposed system during the incubation period, the data obtained from the sensors is sent to the user-friendly application for real-time monitoring. This also gives alert notifications whenever the patient's health is at risk to hospitals and guardians.

Section II gives us an overview of the previous work, Section III contains the detailed discussion of our proposed system, and Section IV gives the result and compares the solution with the previous work carried out. Finally, Section V provides the conclusion of this project.

II. Literature Survey

As per the previous research for health monitoring systems, the technology used changes day by day to be more accurate and use advanced methodologies. The IoT has been a helping hand in achieving this goal today with many patients being monitored remotely in real-time. Continuous care is given to the patients as their well-wishers are also aware of their health status.

Maneesh Gupta and Hana Qudsi [2] share an important aspect from their perspective where the thermistor is used to measure the breathing rate as COVID-19 affected patients are targeted with the damage of lungs. This device measures an individual's respiration rate by detecting changes in temperature which is mounted on the base of the face mask. However, a delay was observed as the main drawback and is being solved by advancing the hardware used.

AI 4 COVID-19 [3] by Ali Imran, throws light on the fact that AI can solve the problem of COVID-19 and focus on AI over the healthcare system to predict the contagious disease using various algorithms. The ResNet-100 Convolutional Neural Network, a deep learning technique together with a Logistic Regression classifier, is employed

978-1-6654-2569-8/21/$31.00 ©2021 IEEE
DOI 10.1109/ICDI3C53598.2021.00051

to spot the coronavirus pandemic rapidly. With the help of AI, we can categorize a person's health situation as having few symptoms of COVID-19 will be under continuous monitoring.

Dr.R.Poovendran, in his paper [4], has mentioned cost-effective hardware components that are use to detect heart rate, breathing rate and SpO2, and informs the guardian about the patient's situation through alert or alarm call-out system. The paper mentions that the patient uses a headset with the mask for determining the breathing rate and heart rate using a sensor. It has a user-friendly application to collect the data, design all these wearables with the integration of an application where the alarm or alert system can save patients immediately, even in a remote location.

Vishal Varun [6] in his paper mentions that measuring respiratory rate as one of the most tedious tasks. It requires utmost accuracy in order to prove that the patient is suffering from breathlessness or not. The calculation of barometric pressure and the usage of thermistor provides an accuracy of 93% in this research. However the author faces issues with the data communication from the mask to the device like mobile/tablet etc. The usage of signal processing algorithms is seen here monitoring lung functions.

Muhammad E. H. Chowdhury [7] has mentioned solving the problem of COVID-19 through the datasets of SARS using CNN model and chest X-rays through image. Since the dataset collected was only about 319 CheXNet dataset, it wasn't accurate enough to detect pneumonia in humans. The pre-dataset being trained is not sufficient to provide high accuracy rate as the image dataset available from the period of pneumonia is ineffective in detecting coronavirus.

III. Proposed Architecture

A. Hardware

For the hardware which is the combination of a shirt and a mask the following components are considered and are connected to give us the results of pulse, blood oxygen level, breathing rate, real time location and temperature. The proposed block diagram is illustrated in Fig.1.

- The microcontroller used is Lilypad Arduino, which is based on the ATmega328V. It's designed for e-textiles, thereby it can be sewn on fabric with other components using a conductive thread. The LilyPad Arduino can be programmed with the Arduino IDE.
- The NodeMCU is used to collect the sensory outputs from the microcontrollers and send it to the App over the Internet. It runs on ESP8266 firmware having 128KB RAM and 4MB of flash memory to store the data.
- Here we use the MAX30100 Pulse Oximeter sensing device. It's a measuring device that obtains its readings from received intensity of light which is later converted to electrical signals. The LEDs are used to get the data by placing the sensor on the fingertip. The sensor is embedded on the shirt circuit, connected by an elongated cable that reaches the tip of the finger measuring accurate values.

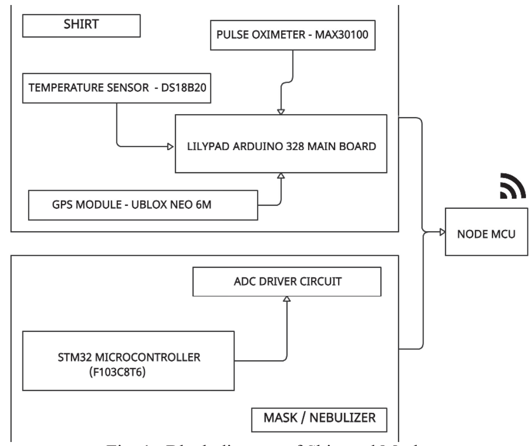

Fig. 1. Block diagram of Shirt and Mask

- The STM32 Blue Pill is an ARM Cortex M3 Microcontroller. The STM board operates on a 3.3V power supply and uses a 32-bit processor. With the thermistor attached, a code was built for measuring the respiration rate.
- The Ublox NEO6M GPS module used in this project determines its location and we obtain the output, that is the latitude and longitude of its position. It exhibits high sensitivity when used indoors. The battery in the GPS module is used for power backup, and the EEPROM present in the GPS module stores the configured settings providing the needful results.
- There is a patch antenna present in the module which has a sensitivity of -161dBm. A U.FL cable is used to connect the antenna and the module. This allows great flexibility when the GPS module is mounted onto our shirt.
- The temperature sensor that is embedded in the shirt is the DS18B20 and has a 1-Wire bus used for communication with the microcontroller. It operates over the temperature range of -55°C to +125°C and accuracy is ±0.5°C for the range from -10°C to +85°C.
- For heart rate and SpO2 (Blood Oxygen level) which is the major parameter for COVID-19, if the SpO2 level drops below 90%, it will create an alert system through WiFi module or by sending an SMS to the patient's guardian and hospital.

B. Software

The software used for programming the components of the shirt and mask is Arduino IDE. The ease with which an Arduino can obtain sensor values is one of the features that makes it very useful. ThingSpeak is used to aggregate, visualize, and analyze live data streams, which are sent to the app over the Internet. PlatformIO is another software used, which is an open-source ecosystem for IoT development.

C. Camisa - Shirt

With the ongoing swath of the COVID-19 pandemic situation, health care monitoring has become an essential part of human lives. A few health monitoring systems including wearable devices like watches, waist belt, shirt, mask, etc. are effective when used for remote applications.

One such confirmation for real-time health monitoring [5] is the use of a shirt that is embedded with sensors. The circuit of shirt is shown in Fig. 2. This gives us accurate results of patients when they are in their quarantine period and get alerts when the sensor values reach a threshold level indicating at-risk condition.

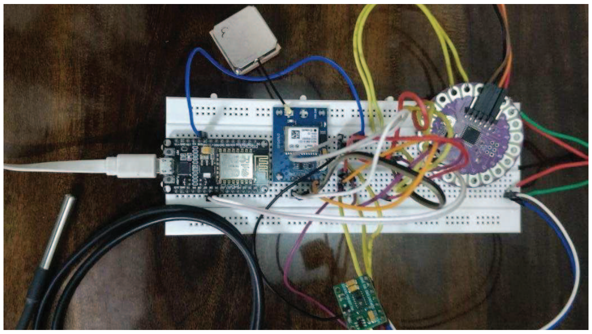

Fig. 2. Shirt Circuit which can be sewn on textile.

The need to measure blood oxygen saturation decreasing during the incubation period gives way to the pulse oximeter. Based on the light signals which are reflected from the blood cells, it converts them to electrical signals. The output is mathematically processed to measure oxygen saturation level and heart rate. The patient's temperature is recorded by the temperature sensor.

If the threshold is exceeded, the real-time location of the patient is shared with the hospital to rush the patient for immediate intensive care. The threshold values of sensors in the shirt is given in Table I.

TABLE I. THRESHOLD VALUES FOR SHIRT

S. No.	Ideal versus Risk Values for Alert Systems		
	Parameters	***Ideal***	***At-Risk***
1.	SpO2 Level (%)	94 -100	< 91
2.	Heart Rate (bpm)	70-100	>110 [a]
3.	Temperature (in °F)	97 - 99	>100

D. Camisa - Mask

The use of face masks and shields has reduced the risk of virus transmission. Adding onto this primary solution, our focus is to develop a smart nebulizer, as illustrated in Fig. 3, which can be worn by the patient during their quarantine period. This is helpful in order to continuously monitor their breathing pattern and temperature using the NTC thermistor that is attached inside the nebulizer [6].

The STM32 is an ARM Cortex M3 Microcontroller that is programmed with a thermistor to collect the breathing pattern of an individual. We try to analyze the number of 'no breaths' conditions, using the parameter shown in Table II. After consecutive 'no breaths', an alert is notified to the person monitoring.

After the data is collected from the shirt and the mask, the data is visualized on the application for remote monitoring. The transfer of data to the application is illustrated in Fig. 4.

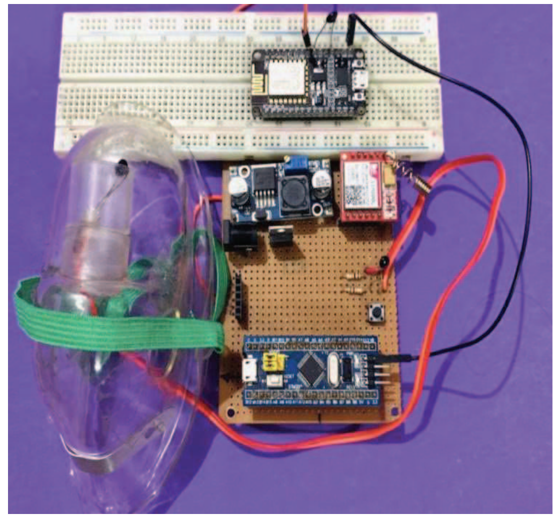

Fig. 3. Circuit for Mask

TABLE II. THRESHOLD VALUES FOR MASK

S. No.	Ideal versus Risk Values for Alert Systems		
	Parameters	***Specified Value***	***At Risk***
1.	Breathing Rate (respirations per min)	12 - 20	> 20 [a]

a. Varies between individuals

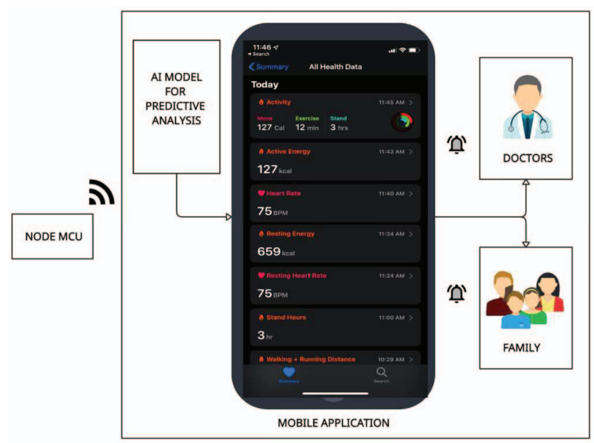

Fig. 4. Transfer of data to the application

E. Neural Network Implementation

An ANN model is used to design and compute mathematical algorithms resembling how the brain reacts to sensory inputs. The brain consists of neurons that are interconnected to form a huge network. Neural Network, in general, give us various advantages in numerous fields, one such advantage is the healthcare system, which uses the neural network in prediction and spread of disease. A neural network is made up of millions of artificial neurons and is interconnected by nodes, therefore called a processing unit.

Initially, a mathematical model of all the neurons and the interconnections between them are created. Then the inputs are provided to the network, and the neurons convolute with the inputs systematically, to generate an output, which is fed into the next level of neurons, as shown in Fig. 5. The process repeats itself until the output is reached, in the predefined hidden layers.

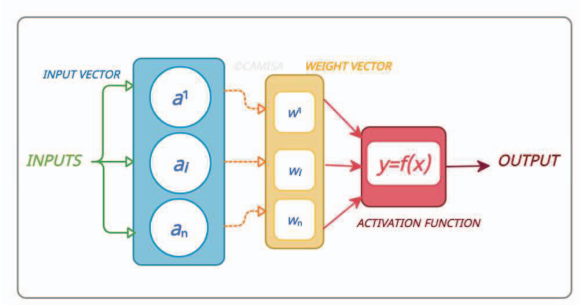

Fig. 5. Neural Network Architecture

The ANN model can also be used to estimate the confirmation of the coronavirus through the available datasets [7]. In order to limit the propagation speed and the propagation power of the coronavirus, it is important to predict and identify the infected patients and isolate them. The flowchart of neural network is explained in Fig. 6. Below are the steps carried out for the Predictive AI model using C language and Python.

a) Collection of data from various sources: Since the virus exhibits different symptoms in different countries due to the mutations, the dataset is obtained from a standardized database. The network is tuned by training it on this dataset built on health conditions of 300,000 people from different geographic locations.

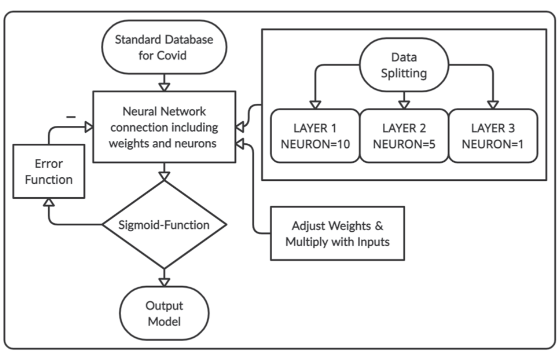

Fig. 6. Flowchart for Neural Network

b) Database formatting and creation of labelled data: The data obtained is formatted for inputs with 20 different parameters as shown in Fig. 7, where each bit in the every input corresponds to a symptom experienced by an individual like cold, fever, cough, breathlessness, diarrhoea, and so on. Thereby giving us maximum of 20 different parameters to predict the condition of a single user. The parameters are set giving the input layer size over 200,000 inputs. We labelled the data as per the requirement and get the final raw data for further tuning.

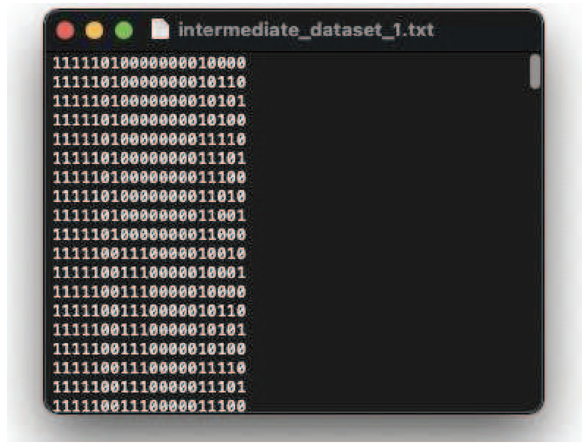

Fig. 7. Formatted Input Dataset File

c) Database formatting to make it compatible with AI: The raw input data is made compatible for AI by preprocessing. The raw data is processed and fed to the 3 layer feed forward network, where forward propagation is performed below as in Fig. 8.

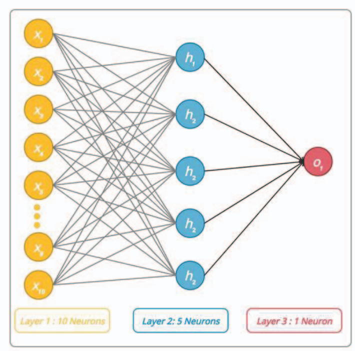

Fig. 8. A 3 Layer Feed Forward Network

d) Theoretical modelling of ANN: For performing the theoretical analysis, the arrays are set up by assigning random weights. This is a 2D array matrix where one column represents a single node, thus obtaining a 10x20 matrix. This final matrix obtained is used to train the model. Equations (1) is the input vector fed into the network. Equations (2) and (4) calculate the weighted sum of inputs of the respective layers which is subsequently used to measure the activation function given by (3) and (5).

$$x_i = a_i^{(1)}, i \in 1,2,3 \quad (1)$$

$$z^{(2)} = W^{(1)}x + b^{(1)} \quad (2)$$

$$a^{(2)} = f(z^{(2)}) \quad (3)$$

$$z^{(3)} = W^{(2)}a^{(2)} + b^{(2)} \quad (4)$$

$$a^{(3)} = f(z^{(3)}) \quad (5)$$

Initially, a loop is started to run through the data used for training in each layer. In each iteration the order in which the training data runs through is randomized, to ensure that local minima do not have a union.

The data is fed through the network in order to calculate the activation function of the hidden layers and output layer's nodes (in our case, sigmoid function as in (6))[9].

$$\sigma(x) = \frac{1}{1+e^{-x}} \qquad (6)$$

Later on updating the associated weights and comparing the output after multiple iterations, on how well the neural network performed as to the given training sample. This is referred to as the cost function, given by (7) and illustrated in Fig. 9.

$$\text{Cost Function (J)} = \frac{1}{n}\Sigma_{i=0}^{n}(y^{(i)} - (mx^{(i)} + b))^2$$

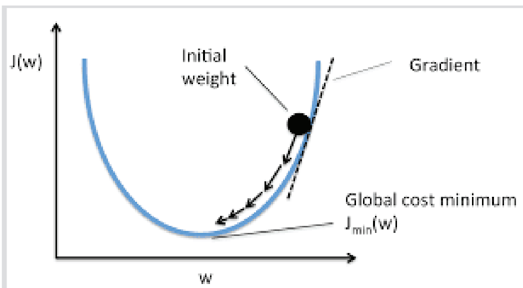

Fig. 9. Cost Function

The backpropagation algorithm checks for the least value of the error function in the weights vector. On backpropagating the error function to the hidden layer, the algorithm tunes the network in accordance with the error rate [10]. This calculates the gradient of the obtained loss function with respect to weights and thus minimizes the error.

$$\text{Gradient Descent} = \frac{\partial J}{\partial w} \qquad (8)$$

The gradient descent for a given algorithm is an optimised algorithm used to find the parameters that reduce the cost function as shown above in (8).

e) Parameter tuning: Using the obtained values, we perform parameter tuning in order to find the error functions. It is observed that the error functions are in a decreasing trend as seen in Fig.10. The obtained outputs of the error function are compared as the values of various parameters such as learning rates, i.e, alpha are varied (9).

$$dx = alpha * |\frac{\partial J}{\partial w}| \qquad (9)$$

f) Training the model and validation: The model and algorithms are trained on 60% of the dataset where the dataset is split. The remaining 40% of the dataset is used for validation, to prove the predictability of the algorithm.

g) Multiple iterations and parameter tuning to obtain the least error: This is the final step of training the model. The output thus obtained is an error function that is obtained after every sample is trained. The Fig. 11. illustrates the output file generated, showing 1-bit binary output for every combination of input parameters. Zeroes and ones represent absence and presence of COVID.

Fig. 10. Error Plot of the trained model

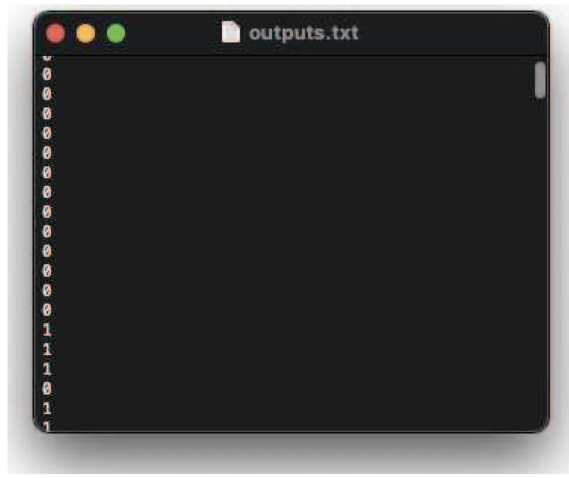

Fig. 11. Output file generated showing the presence of COVID-19

Neural network gives us a total of (20x10) + (5x1) = 205 trainable parameters in the model. This is almost similar to a 205th-degree equation since it is highly impossible to fit in such a high degree equation using mathematical methods or any conventional methods. Neural networks giving a very efficient solution to this problem is a good tool to model such diverse datasets.

To conclude this model we integrate the output obtained from the neural network, which gives us information about whether a person has COVID-19 or not based on the symptoms.

Our work shows that it is possible to acquire a quality model for the prediction of disease using AI, with inputs as symptoms experienced by an individual. The predictions prove that AI models can also solve problems in prediction of any disease. The application of AI methods should also be modeled keeping in mind the present and future spread of diseases and in an attempt to predict and prevent the impact of such infections. Our future plan widely leans towards making the model available worldwide solving the problem of COVID-19.

F. User Friendly Application

The health monitoring app combines the output from sensors and modules, then visualizes the data received. After the predictive model is trained, it is deployed on the application. The app consists of two main section:

a) Self Assessment: It has a questionnaire filled using toggle switches that record users' responses, which correspond to the COVID-19 symptoms experienced by the user. Based on which the app predicts an output of whether the patient is COVID-19 positive or not. Fig. 12 illustrates the questionnaire.

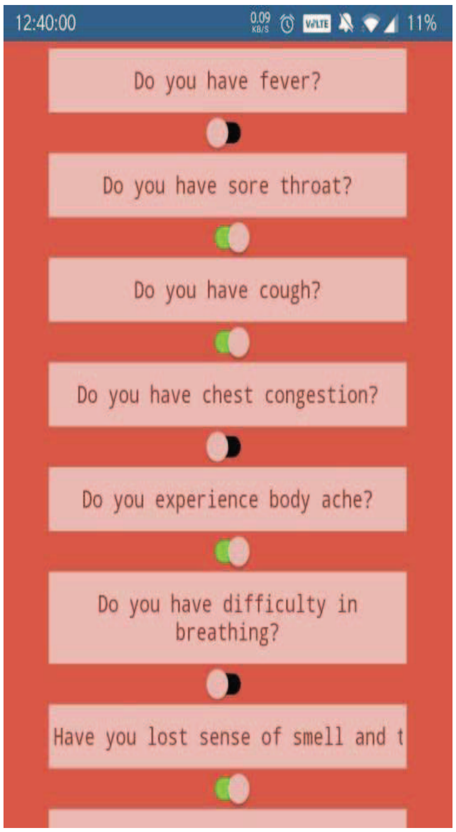

Fig.12. Self Assessment Questionnaire

b) Real-Time Health Monitoring: The health parameters obtained from each of the sensors are visualized on the app using the Thingspeak IoT Platform. The app alerts the guardians and shares the real-time location of the patient when the patient is at risk.

Fig. 13 shows heart rate and blood oxygen saturation in real time. Similarly the other parameters, breath rate, temperature and location of the patient is also visualized on the application.

IV. RESULT COMPARISON

The comparison of key characteristics of prior work and proposed work is shown in Table III. As the parameters considered vary from each other, our work tries to overcome the cons and provide accurate results.

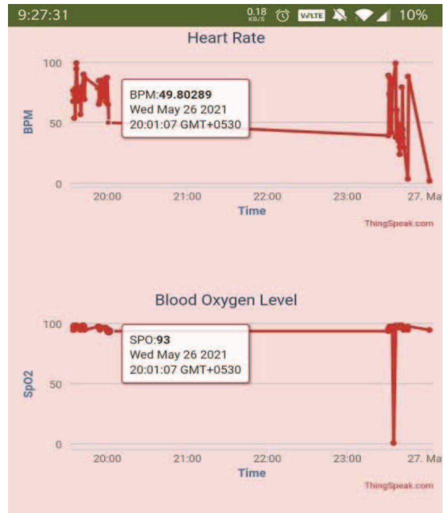

Fig. 13. Health Monitoring on the App

TABLE III. RESULT COMPARISON

Parameters	***Reference [5]***	***Reference [2]***	***Reference [3]***	***Proposed Work***
Application	Heart Rate, Temperature	Breathing Rate	ANN Network	Heart Rate, Temperature, Blood Oxygen level, Breathing Rate
Main Integrant	Heart Rate Sensor, Temperature Sensor	Thermistor	Neural Network	Pulse Oximeter, Thermistor, Temperature Sensor
Working Principle	Sensor Technology	Comparison of Relative Voltages	Prediction by LSTM	Sensor Technology & Predictive Analysis using Neural Network
Microcontroller	Atmega328	TI-MSP430	-	Lilypad Arduino

V. CONCLUSION

In today's world due to rising threats of COVID-19, the need to solve the problem with the help of different technologies is essential. The solution provided above satisfies the need and also can be further considered for advancements.

Artificial intelligence is a very important and promising tool for detecting early COVID-19 infections and monitoring the condition of infected ones remotely. The advent of useful algorithms greatly improves the sequence of processing and decision-making. The participation of the Internet of Things that integrates all the above technologies provides a good result. In this paper, we've introduced a low-power IoT wearable devices for the COVID-19 application called Camisa (Contactless Patient Monitoring System).

We have identified the foremost components of the proposed system and explained its implementation details. Performance ratings indicate that wearable shirts and masks are an inexpensive device. Future scope to this area will include ensuring data access and security to shield the privacy of hospitals and patients, further as enabling voice recognition for the proposed wearable device which can be more advanced.

REFERENCES

[1] Shahid Farah Anleela Zameer and Muhammad Muneeb "Predictions for COVID-19 with Deep Learning Models of LSTM GRU and B1-LSTM" Chaos Solitions &Fractals 140(2020)

[2] Gupta, Maneesh, and Hana Qudsi. "Low-cost, thermistor based respiration monitor." 2013 39th Annual Northeast Bioengineering Conference. IEEE, 2013.

[3] Imran, Ali, et al. "AI 4COVID-19: AI enabled preliminary diagnosis for COVID-19 from cough samples via an app."

[4] Nachiar, Cecil C., et al. "Design of Cost-effective Wearable Sensors with Integrated Health Monitoring System." 2020 Fourth International Conference on I-SMAC

[5] Aziz, Kahtan, et al. "Smart real-time healthcare monitoring and tracking system using GSM/GPS technologies." 2016 3rd MEC International Conference on Big Data and Smart City (ICBC). IEEE

[6] Kalavakonda, Rohan Reddy, et al. "A Smart Mask for Active Defense Against Coronaviruses and Other Airborne Pathogens." IEEE Consumer Electronics Magazine (2020).

[7] Huang, Chiou-Jye, et al. "Multiple-input deep convolutional neural network model for covid-19 forecasting in china." MedRxiv (2020).

[8] Che, Zhen-Guo, Tzu-An Chiang, and Zhen-Hua Che. "Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm." International Journal of Innovative Computing Information and Control 7.10 (2011): 5839-5850.

[9] Tsai, Chang-Hung, et al. "A hardware-efficient sigmoid function with adjustable precision for a neural network system." IEEE Transactions on Circuits and Systems II: Express Briefs 62.11 (2015): 1073-1077

[10] Erb, Randall J. "Introduction to backpropagation neural network computation." Pharmaceutical research 10.2 (1993): 165-170.