International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 05 | May 2022 www.irjet.net p-ISSN: 2395-0072

IoT Health Monitoring with Voice Recognized Prescription

Nathan Noronha[1], Pratil Patil[2], Anchal Thool[3], Prof. Rajesh Patil[4]

[123]B.Tech Student, Dept. of Electronics and Telecommunications Engineering, VJTI, Mumbai, Maharashtra, India
[3]Associate Professor, Dept. of Electrical Engineering, VJTI, Mumbai, Maharashtra, India

---***---

Abstract - *In recent times, healthcare problems persist due to the misinterpretation of handwritten prescription and real time monitoring of the patient in the required time. Existing research monitors health parameters in real time but fails to get the advice from the doctor rapidly. Voice based prescriptions have limited work in the field and hence have a lower accuracy of the Natural Language Processing model(NLP). In order to accomplish the task of real time monitoring and acquiring a voice-based prescription from the doctor instantly, this work is divided into two segments. The first segment is an Internet of Things(IoT) kit that transmits data to the cloud with the help of health sensors interfaced to a microcontroller. The second segment is a website which acts as a user interface for the doctor with a NLP model which converts speech to text and generates a medicinal prescription to the patient. The website also displays the health parameters of the patient by acquiring the data from the cloud. The doctor can view these parameters of multiple patients and can prescribe accordingly. This will save time and reduce the error in interpretation of the prescription for the patient.*

***Key Words*: Voice based prescription, NLP, IoT, Data, Health, Cloud.**

1.INTRODUCTION

Today's world is heavily reliant on a variety of medications and drugs that aid in the smooth operation of our society. Rapid medical advancements enable ordinary people to live longer, better, and stronger lives. However, the majority of the prescriptions given to patients are written by hand. As a result, they are more prone to interpreting errors. Such misunderstandings can result in fatal accidents. Similarity in drug brand names or pharmaceutical names can cause prescription errors due to misinterpretation. As a result, we can conclude that human factors play a significant role. The introduction of automation and technology can help to reduce the number of errors made during the prescription process.

In India, a major issue is that most prescriptions are still written by hand, and the readability of these handwritten prescriptions is poor. Several cases have surfaced in which a chemist's misinterpretation of a prescription resulted in the patient receiving the incorrect medication, causing serious health problems. However, for a doctor in India, using a traditional Electronic Health Record System (EHR) to generate an electronic prescription is time consuming and costly. Setting up such a system would necessitate proper infrastructure. Either the doctor would operate the system on his own, which would take time, or an operator would be hired, which would be costly.

Here, we have implemented a system enabling the doctor and the patient with an environment to easily access each other and save time and energy. This system is divided into two sections, the first section deals with transmitting the data from the patient using an IoT kit. The kit will be made with an ESP32 microcontroller interfaced with health sensors. The IoT kit senses the health parameters of the patient and transmits the data to a database using Wi-Fi. The patient can also send the symptoms faced by him/her to the doctor. In the second section, The doctor will be able to observe these parameters and other symptoms (if mentioned by the patient) and prescribe medicine using our voice based prescription. The patient will receive an alert for this prescription, hence the time consumed will be very less compared to the traditional methods.

2. Related Work

With the help of the Internet of Things, Jayeeta Saha et al. [1] have proposed an easier solution for remote real-time health monitoring of patients from the hospital as well as at home (IoT). Sensors collect data on a variety of parameters related to patients' health, which the Internet of Things stores and displays via a website that allows for remote monitoring. The combination of Raspberry Pi and IoT has solved this problem by introducing a new innovative technology into the healthcare system that allows for remote monitoring of the patient's health. According to the literature evidence base, the system sends data and alerts using healthcare sensors connected to a Raspberry Pi and cloud integration. However, using a Raspberry Pi for such an application is inefficient because it consumes more power, and deploying such systems for every doctor is not practical because traditional sensor-based diagnosis in the medical field requires more sensors and human effort when processed on a large scale.

Vani Yeri et al. [2] designed an IoT-based health care application in their research work to address the problem raised in the previous literature. The proposed system is made up of a web and mobile application that is based on continuous wireless patient monitoring. The goal of the paper was to develop a low-cost system for transmitting patient vital signs in an emergency. Sensors are used to use the wireless network to measure the patient's vital signs. The data from the sensors is collected and sent to the cloud for storage via the controller's Wi-Fi module. The data is processed in the cloud, and feedback is given on the analyzed data, which can then be analyzed further by a doctor remotely. Remote viewing relieves doctors' workload and provides patients' exact health status. A message is sent to the doctor if the patient requires immediate attention. However, because the patient is unable to alert the doctor, when necessary, the doctor is only notified when there is an abnormality in the health parameters. To alert the doctors, each patient must be mapped with the doctor's treatment zone.

As proposed by Sarfraz Fayaz Khan [3], IoT and RFID tags are used to create a comprehensive and effective healthcare monitoring system that aids in the monitoring and weighing of the patient's health. Hospitals have begun to use cell instruments for communication purposes, and internet of things (loT) has been used and fused with wi-fi sensor nodes resembling Radio Frequency Identification (RFID), Near-field communication (NFC) tags, and small sensor nodes for this purpose. RFID can be used with an Android phone over the Internet. The parameter is detected by RFID and sent to a mobile device. Patients must be within RFID range in order to do so; these devices are placed at the entrance or in the desired location of patients. Active RFID systems work in the ultra-high frequency (UHF) band and have a range of up to 100 meters, but they can't be used remotely in an emergency. It does not include medication or precautions based on the patient's health status, which are administered by controlling the appliances that deliver the prescribed medication.

Asif Mehmood et al. [4] proposes a system based on android and a web application in which a doctor can prescribe patients using an android application and a stylus pen, and other users such as patients, receptionists, pharmacists, and administrators can interact with the system using their web accessibility. The proposed system is based on the use of Arduino and E-Health sensors to integrate Internet of Things and Cloud Computing technologies. The general concept is to collect data using an IoT setup with health sensors and send it to the doctor via API (Application Programming Interface) and MQTT (Message Queue Telemetry Transport) protocol on the cloud. A mobile application and a stylus can be used by the doctor to write prescriptions. However, styluses require smartphones or tablets to function, which are generally more expensive than the alternatives. In the case of poor handwriting, the issue of handwritten prescriptions may also arise. This also necessitates practical experience in order to achieve the desired outcomes.

3. Proposed Work

3.1 Remote monitoring of health parameters:

The patient's parameters can be monitored by using various types of health sensors. The data from these sensors will have to be accurate as the data will be seen by the doctor before prescribing medicines. All this data will be parsed by a micro-controller into JSON format. The parsed data will also need to have a timestamp to keep track of the time of the monitored parameter. This will also allow for a real time series data, which can be used to keep track of any anomalies in the patient's routine.

3.2 Portal for patient and doctors:

A user interface for both patients and doctors will be made. The patients will be able to send their health parameters and symptoms to the doctor and ask for the advice. The doctor will have a user interface to observe his/her patient's parameters and symptoms. Using this interface, the doctor will be able to diagnose the patient and give his advice using voice recognition.

3.3 Tagging data using Named Entity Recognition:

The doctor will prescribe the patient using voice recognition. To implement a voice-based prescription, an NLP model will be used to preprocess the text dictated by the doctor by removing the stop words and using tokenization. Once the text is processed, NER tagging will be performed on the text to get annotated data of tagged entities in it. Using the tagged labels, the prescription data will be extracted from the text.

4. Practical Implementation

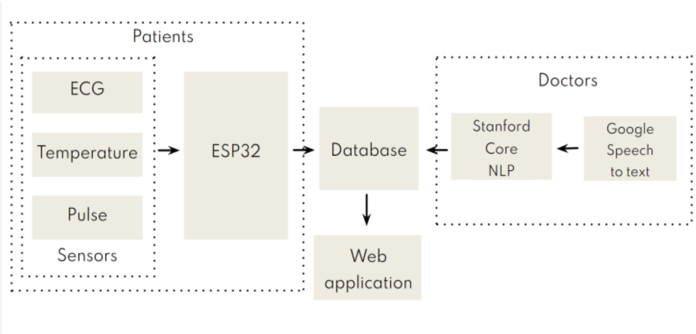

Fig - 1: System Architecture

The architecture diagram shown in the above Fig.1 shows various components and entities in the application.

- **Health parameters of patients:** Retrieval of information on patient's temperature, heart's rhythm and its electrical activity, oxygen level, pulse rate, etc. using the various sensors which are connected to ESP32 micro-controller.

- **Database:** The parameters which are captured using a micro-controller are stored in a MongoDB database. These parameters are referred by doctors for prescribing the patient. The data is stored as a time series data which can be used to observe the changes with respect to the time, to study the direction of the patient's health.

- **Web Application:** A User interface enabling the doctors and patients with an environment to communicate with each other.

- **Doctors' portal:** Doctors can give prescriptions based on voice using the Google Speech-to-Text API and extract data using Stanford NLP package.

4.1 Hardware Setup:

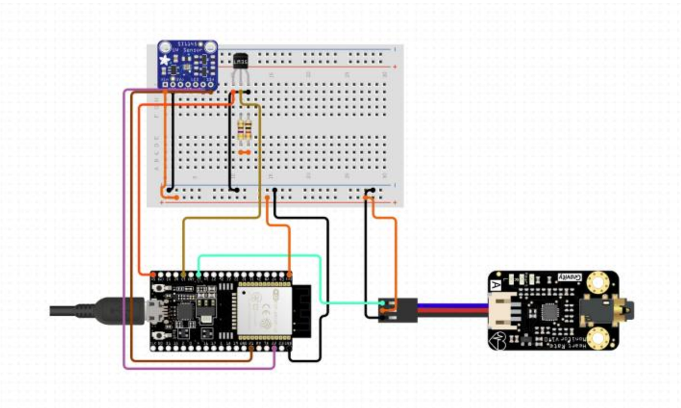

Fig 2: Circuit Diagram

- **ESP32:** It's a feature-packed micro-controller with built-in Wi-Fi and Bluetooth connectivity that can be used in a variety of IoT applications. **Features:**
  - Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz
  - The ESP32 uses a combination of proprietary software to achieve ultra-low power consumption.
  - 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
  - Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
  - 34 Programmable GPIOs.
  - It can provide Wi-Fi and Bluetooth functionality to other systems via its SPI / SDIO or I2C / UART interfaces. (such as the ADC).[6]

- **AD8232 ECG Sensor:** The AD8232 Heart Rate Monitor is a low-cost board that monitors the electrical activity of the heart. An ECG (Electrocardiogram) can be used to visualize this electrical activity and can be read as an analog reading. Because ECGs are notoriously noisy, the AD8232 Heart Rate Monitor acts as an op amp to help obtain a clear signal. It's designed to extract, amplify, and filter small signals in noisy environments, such as those caused by motion or the placement of remote electrodes.
  Features:
  - Operating Voltage - 3.3V
  - Analog Output
  - 3.5mm Jack for Biomedical Pad Connection.[7]

- **LM35 Temperature Sensor:** The LM35 is a temperature sensor that produces a proportional signal to the current temperature. The output voltage can be easily translated into a Celsius temperature reading. The advantage of the LM35 over the thermistor is that it does not require external calibration.

  Features:
  - Operating Voltage - 3.3V - 30V
  - Calibrated Directly in Celsius (Centigrade)
  - Linear + 10-mV/°C Scale Factor
  - 0.5°C Ensured Accuracy (at 25°C)
  - Less than 60-μA Current Drain.[8]

- **MAX30102 Heart Rate Pulse Sensor:** A pulse oximeter and a heart rate monitor are included in the MAX30102 biosensor module. It has a red LED and an infrared LED, as well as a photo detector, optical components, and low-noise electronic circuitry that suppresses ambient light. The MAX30102 has a 1.8V power supply and a separate 5.0V power supply for internal LEDs in wearable devices for heart rate and blood oxygen acquisition that are worn on the fingers, earlobes, and wrists.

  Features:
  - Operating Voltage - 3.3V
  - Ultra-low shutdown current - 0.7μA
  - Low-power heart rate monitor. (<1mW)

- Tiny 5.6mm x 3.3mm x 1.55mm 14-pin optical module.[9]

4.2 User Interface:

The user interface will be a website consisting of portals for both patients and doctors. For the website the following technologies are used.

- **Flask:** Flask is a web framework that is free and open source. It is written in Python and requires no ORM (Object Relational Manager) to use.

  Features:

  - URL Routing.
  - Template Engine.
  - RESTful request dispatching.
  - Support for secure cookies (client-side sessions).[10]

- **MongoDB:** MongoDB is a NoSQL database and an open-source document database. It stores data in databases called collections in the form of documents. It uses JSON format to store the data in the document.

  Features:

  - Ad-hoc queries for optimized, real-time analytics.
  - Indexing appropriately for better query executions.
  - Replication for better data availability and stability.

- **HTML, CSS, JavaScript:** For the frontend, HTML is used to design the basic layout of the website using Jinja2 templating by Flask. Jinja2 parses HTML content and displays it onto the website by working in collaboration with Flask. CSS is used to style the web pages and add padding with a CSS framework called Bootstrap. JavaScript helps in making the web pages interactive in the real time.

4.3 Natural Language Processing:

The recorded voice from the doctor is converted into speech by the Google Speech-to-Text API. The text is extracted by using NLTK to preprocess the text, and finally performing Named Entity Recognition on the preprocessed data. NER is performed by using the Stanford NER package.

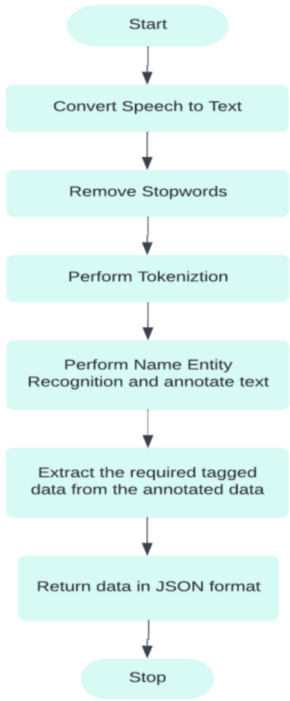

Fig 3: NLP Flowchart

- **Natural Language Toolkit (NLTK):** It includes a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, as well as wrappers for industrial-strength NLP libraries. It also includes easy-to-use interfaces to over 50 corpora and lexical resources. [14] NLTK is used to preprocess the data by removing redundancies and then performing NER tagging on the words to extract the required information in this implementation.

- **Stanford NER Tagger:** It's a Java implementation of a Recognizer for Named Entities. The process of extracting features and labeling entities in a text is known as named entity recognition. It uses a CRF (Conditional Random Fields) Classifier, which is a supervised learning technique. [12] The CRF training features selected for a given application domain can have a significant impact on entity recognition performance [13]. Simple token level training of CRFs, for example, yields poor results, but text features such as word prefix/suffix/shape, word/phrasal clustering, and Part-Of-Speech (POS) tags can be used to create a CRF language model that is very effective at recognizing the intended entities. The model was re-trained using suitable Indian drug names for increasing the accuracy for extracting data from an Indian doctor.

5. Results

The following results were obtained from the implementation of this system.


```
_id: ObjectId('62528a1d8039fe7bc1b398ca')
sensors: Object
  Heartrate: 78
  Oxygen: 98
  ECG: 1500
  Temperature: 28
ts: 2022-05-19T07:41:17.445+00:00
p_id: "d95a0939b0f94a3f81c127b2dc52ee41"
```


Fig 4: Health parameters in MongoDB database.

Fig.4 displays the health parameters of the patient, as received by the IoT kit of the patient. It consists of an array of health parameters with the timestamp and the patient ID which refers to a particular patient in the patient collection of MongoDB.

Ehealth

Dr. Nathan Noronha

Name: Anchal

Age: 21

Date: 2022-05-24

Diagnosis: Viral fever

Advice:

		Breakfast		Lunch		Dinner	
Name	**Days**	**Before**	**After**	**Before**	**After**	**Before**	**After**
Crocin 5 mg tablet	7 days	☐	☑	☐	☑	☐	☑
Robitussin 5 ml syrup	7 days	☐	☐	☑	☐	☐	☑

PDF

Fig 5: Generated prescription.

Fig.5 is the generated prescription that is obtained after performing speech to text conversion and extracting data from the text using Stanford NER tagger. The prescription can also be verified by the doctor and any changes can be made to the prescription, if needed, by using JavaScript. The prescription shows the name of the doctor, patient, with the date, diagnosis and the medicines prescribed by the doctor.

6. Conclusion

The main aim of the project was to solve the problem of illegible handwritten prescriptions and provide an interface for the patients and doctors to get in touch with each other. The implemented system overcomes these problems with two major advantages over the existing healthcare system.

- The prescription is computer generated and in text format which can solve the problem of sloppy handwriting. The doctor's notes are comprehended easily as it performs NER tasks on both normal English medical phrases as well as doctor's abbreviations.

- The patient can get real time feedback from the doctor about the symptoms and the parameters of the patient, thereby saving the time of both the doctor and the patient.

Voice-based E-prescription requires only a subtle influence in a doctor's workflow, but it will have a significant impact on the development of a digital ecosystem for patients in the long run.

7. Future Scope

The monitoring IoT kit can be made patient specific. For instance, if a patient has Diabetes, then a glucometer can be used and interfaced with the kit easily, making the hardware scalable and flexible. The model now works only with the English language. To make models compatible with many more languages to make it user friendly for doctors as well as the patients. Discontinuity in speech (For example, Coughing, sneezing, stammering, etc.) may affect the accuracy of the speech to text conversion. Find ways to eradicate these errors. A graph of the health parameters can be displayed on the website.

REFERENCES

[1] "J. Saha et al., "Advanced IOT based combined remote health monitoring, home automation and alarm system," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, pp. 602-606, doi: 10.1109/CCWC.2018.8301659.

[2] V. Yeri and D. C. Shubhangi, "IoT based Real Time Health Monitoring," 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp. 980-984, doi: 10.1109/ICIRCA48905.2020.9183194.

[3] S. F. Khan, "Health care monitoring system in Internet of Things (IoT) by using RFID," 2017 6th International Conference on Industrial Technology and Management (ICITM), 2017, pp. 198-204, doi: 10.1109/ICITM.2017.7917920

[4] A. Mehmood, F. Mehmood and W. -C. Song, "Cloud based E-Prescription management system for healthcare services using IoT devices," 2019 International Conference on Information and Communication Technology Convergence (ICTC), 2019, pp. 1380-1386, doi: 10.1109/ICTC46691.2019.8939916.

[5] S. Deshmukh, R. Balani, V. Rohane and A. Singh, "Sia: An interactive medical assistant using natural language processing," 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC), 2016, pp. 584-586, doi: 10.1109/ICGTSPICC.2016.7955368.

[6] ESP32 Datasheet: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[7] AD8232 Datasheet: https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf

[8] LM35 Datasheet: https://www.ti.com/lit/ds/symlink/lm35.pdf

[9] MAX30102 Datasheet: https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf

[10] Flask: https://flask.palletsprojects.com/en/2.1.x/

[11] MongoDB: https://www.mongodb.com/docs/

[12] Stanford NER Tagger: https://nlp.stanford.edu/software/CRF-NER.html

[13] M. Tkachenko and A. Simanovsky, “Named entity recognition: Exploring features,” in Proceedings of KONVENS 2012, J. Jancsary, Ed.ÖGAI, September 2012, pp. 118–127, main track: oral presentations.

[14] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O’Reilly Media Inc.