JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

VOICE CONTROLLED PERSONAL HEALTH ASSISTANCE DEVICE USING NATURAL LANGUAGE PROCESSING AND RANDOM FOREST DISEASE CLASSIFIER

CHETHANA SALIGRAM[1], DEEPALI B K[2], GOWRI K S[3], J AJAY KUMAR[4], DR. JYOTI R MUNAVALLI[5]

[1,2,3,4] Student, Dept. of ECE, BNM Institute of Technology, Bangalore, India
[5]Associate Professor, Dept. of ECE, BNM Institute of Technology, Bangalore, India
[5]jyotirmunavalli@bnmit.in

ABSTRACT:

Health assistant devices are a need of the hour. In this paper, we outline one such device that has been developed. It is voice-controlled device that uses Google's speech to text converter API and Natural Language Processing for the communication with the user. The user provided input symptoms are classified into a disease using a Random Forest classifier. The NLP and classification are performed within the Raspberry Pi which acts as a standalone device. Temperature sensor is also interfaced for better user experience. The developed classifier has the accuracy of 95.02% which has further scope of improvement with a larger data set.

KEYWORDS: Health Assistant, Random Forest classifier, Disease classification, Firebase, Natural Language Processing, DS18B20 Temperature Sensor

I. INTRODUCTION

The access to medical care is unacceptably low worldwide in spite of the increasing demand for medical care due to population aging and increasing burden of diseases. Especially in resource limited places, patients do not have access to medical care in a timely manner. Thus, tools to help patients with self-diagnosis and self-triage is urgently needed to mitigate the shortage of medical care resources. Health monitoring devices are the need of the hour in this era where the number and kinds of diseases are on rise. All the diseases have to be treated as soon as possible and a remedy has to be provided to stop it from making more severe damage.

Conventionally, when people fall sick, they try to be the doctor themselves and ignore their issue by swallowing a few tablets to get rid of it for the time being. But these issues might pile up later causing them bigger trouble. To help people get sooner diagnostics and help them take more care about their health, this study has come up with an idea of a health-aid device. The study attempts to effectively provide a potential diagnosis for patients and direct them to the appropriate care setting.

The device serves to facilitate self-diagnosis and to assist with triage. The diagnosis deals with typically educating patients on the range of diagnoses that might fit their symptoms. The triage informs patients whether they should seek care at all and, if so, where and with what urgency (that is, emergently or within a few days).

For patients with a non-emergent problem that does not require a medical visit, our device can reassure and recommend they stay home and prevent them from using unnecessary antibiotics. Reducing the number of visits saves patients' time and money, deters overprescribing of antibiotics. Logging of all the diagnosis and the symptoms that led to it and other relevant information would help in statistical analyzing and interpretation.

We propose to implement a solution to help the user with primary care medical advice connecting symptoms with ailments with complete storing of the diagnosed information in a cloud-based database for any further analytics. Basically, it makes a diagnosis of a disease based on the data on the prevalence of disease and its sensitivity and specificity of symptoms. In the future, such devices would help keeping up the quality of primary care. Remainder of the paper is organized as follows: Section 2 presents the Literature Survey, Section 3 explains Materials and Methods, Results and Discussion in Section 4 and conclusion in Section 5.

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

II. LITERATURE SURVEY

Several articles have been published with different methods to assist patients. This section presents the literature that has been relevant to this study.

A hybrid sensing and wearable device for emergent medical situations with individual modules monitors blood glucose, blood-pressure, heart rate, body temperature, and ECG [1]. It is capable of taking necessary initiatives for emergency medication and treatment in case of unpredictable situations and notifying the patient about routine medication like a medical assistant according to physician prescription. Such healthcare and fitness apps, hands-free voice activated assistants (VAA) are also available [2].

For such applications, Natural Language Processing (NLP) has been used particularly the extraction of symptoms from the input sentence. Emad S. Othman has proposed a Raspberry Pi based personalized voice command system which basically takes the audio input of the sound using a microphone and is further passed through the system to search for keywords. The modules work on the essence of searching for keywords and giving output by matching keywords. Once the symptoms put forward by the user have been processed, a text to speech converter converts text output to speech and this audio output is heard through a speaker [3]. Having researched about the different methods available to aid the patients, some specific domains tend to serve better in building a unique solution to the problem. Retrieval of information through medical data from narrative clinical documents is performed using NLP [7]n retrieval of. The medical text differs from normal text as it contains complex medical terminologies due to the synonym and disease names, the system proposed [7] tries to understand the text relating to a list of symptoms and gives the relevant answer. The system described does not store the information or the medical history anywhere and the extraction techniques can be improved further. Gustav Cederblad [8] focuses on bridging the gap between laymen and medical professionals by extracting the synonyms from a corpus consisting of medical texts and to evaluate whether these words are actually related. That is because sometimes the more specific medical words are described by a more commonly used familiar word. It aimed to find synonyms and related words for different diseases. However, it handles only unigrams, which could decrease the overall performance of the system. Hence, it must be improved to manage n-grams larger than unigrams. The initial symptoms extraction process from the audio voice input is as detailed. The whole process is divided into four steps namely, Speech synthesis, modifying the dataset, interpreting the input and review [11].

Next thing is about classifiers and [9] discusses on what and why decision trees are used in Data Mining and some of the most popular and widely used algorithms used to create decision trees, namely: ID3 (Iterative Dichotomiser), C4.5, CART and Random Forest. Based on the advantages and drawbacks of each of the methods, Random Forest Classifier was shown to be a better match for this study.

From several studies in [6], we can deduce that the Raspberry Pi is an inexpensive, fully customizable, and programmable credit card-sized single board computer, which supports a large number of peripherals and network communication. Therefore, it is suitable for small scale prototype testing and is very efficient and cost inexpensive.

Healthcare database differs from a traditional database. Health care database includes a lot of information and fields that have to be created and managed [10]. M. Sathya, S. Madhan, K. Jayanthi in their paper on IoT [4] focus on addressing the lack in database connectivity between the different cloud environment in monitoring the data in constant time intervals and to analyze data. A cloud based IoT system is proposed that can be implemented in different health monitoring systems. In the paper [5], a Raspberry Pi based IoT model is presented which uses docker container instead of virtual machine for the server. Pi accepts data from sensors, and these are transferred to the cloud using server and users can access the database from the cloud. Since it uses a docker container, it intends only in sending the data to different users and nothing about the analysis of the data.

Having reviewed the literature and analyzing the advantages and disadvantages discussed in the above research papers, the proposed method absorbs the necessary and useful elements from each one of them and it also works on the preferable performance improvements.

III. MATERIALS AND METHODS

This section presents the proposed work in the form of block diagram, NLP, classification and hardware details.

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

3.1. Block diagram

Figure 1 shows the conceptualization of the proposed technique. The complete system is embedded on a Raspberry Pi 3 Model B+ (RPi) board, which acts like a standalone device performing all the necessary computations.

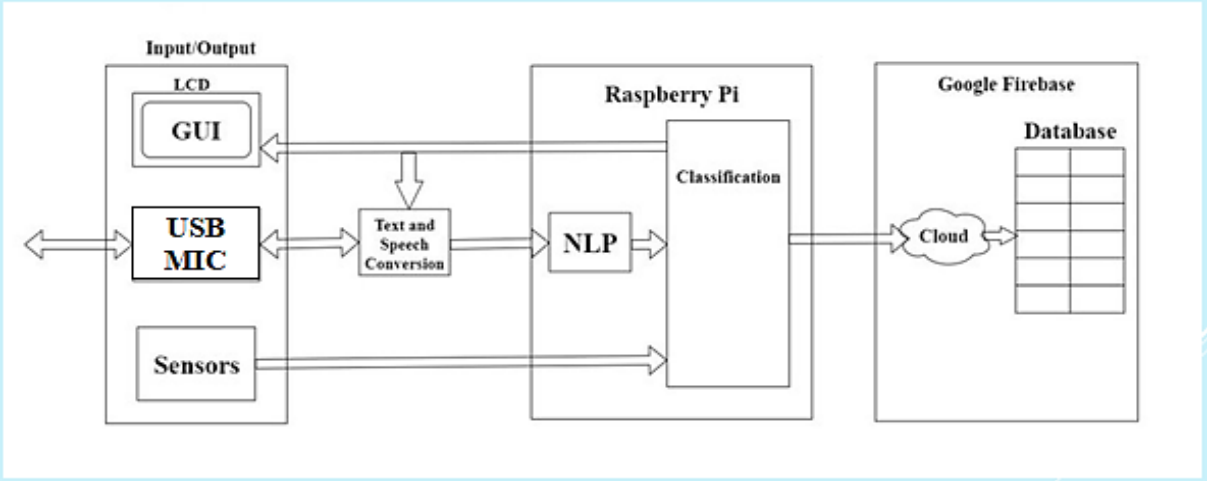

Figure 1. Proposed Block Diagram

The proposed device is interfaced with sensors. The device takes voice commands and processes those commands to provide details about a disease when the symptoms are provided to the device. It poses the users a series of questions about their symptoms or requires users to input details about their symptoms themselves. The symptoms are classified into a disease using a decision tree and an audio and text output is provided to the end user at each and every step until it drills down to the closest disease that is matched. The output includes the disease related information, and any of the common treatment procedures if it is not a serious one. The temperature sensor acts as one of the symptoms measuring interfaces that are present on the device. The implementation includes NLP, classification and hardware details.

3.2. Natural Language Processing (NLP)

The basic idea for the input part is to implement a voice system that can take in the symptoms the user is facing and to incorporate it into the classification part to output the estimated prognosis clearly. The user interacts with the device via speech input which will be captured by an USB microphone mounted on the RPi. Since the RPi Model 3B+ doesn't inherently possess a microphone for capturing audio input, we use an USB microphone for capturing the speech input. The audio is converted into the text format using an API, which is then broken down into segments to understand and analyze. Hence, text normalization is done accordingly with the aid of appropriate natural language processing tools like stemming, synonym extraction, tagging, and so forth. Nonetheless, there could be some symptoms that the speech-to-text API has failed to recognize properly or even, some fallacies in the extraction process leading to any of the symptoms not being caught from the user input. On that account, we make use of some string-matching algorithms to perfect extraction of the symptoms from the input.

We make use of two phonetic algorithms, i.e. Soundex and Metaphone. Soundex is an algorithm that depicts how two words are related depending on their English pronunciation. It converts the input string into a 4 character long alphanumeric value which is compared with the Soundex value computed for the other string. The key values are the same for two similar sounding words. Whereas Metaphone, which is a refinement on the Soundex algorithm, has a wider set of English pronunciation rules allowing for various lengths of keys. For example: The Soundex and Metaphone values of the following two symptoms are:

A = 'muscle wasting' B = 'muscle paining'

Soundex(A) = 'M242' Soundex(B) = 'M241'

Metaphone(A) = 'MSKL WSTNK' Metaphone(B) = 'MSKL PNNK'

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

From the above example, when compared on the basis of how similar sounding two words are, we note that the Soundex values differ by 1 whereas the second part of the Metaphone values of both the words are not very similar, indicating that Metaphone gives more accurate results. Hence, using a combination of these algorithms we capture the symptoms that would otherwise have gone unrecognized.

Accuracy can be improved by other methods; however, the efficiency and the time taken are important parameters. The output of each and every stage in the NLP process is compared with the symptoms dataset where this data is stored in the RPi and the matched symptoms are recorded in an output list. This list is further presented to the classifier.

3.3. Classification

To begin with, a model based on simple binary Decision Tree was built. It was a model with a node matrix for the training and testing of the Data Frame. Each symptom present against its disease was marked as "1" else as "0". Although the model was built successfully, there were issues faced during entering of the symptoms as well as the predictions. Due to the above stated problems, the model was dropped.

The second model that we tried was based on Random Forest Classifications. It used Label Encoder for pre-processing and fitting the transform of the target variable to achieve classification. It took in symptoms one at a time as either "1" or "0" as Yes or No respectively. Although the model was promising in results, the process of entering inputs was tedious and long. Further, it also was not compatible with the input methods used by the NLP section of our project and our proposed hardware. Thus, this version of the classifier was also dropped.

Due to the promising results shown by the second model, it was decided for us to continue using Random Forest Classifier as our classifier. Elements of the previous model such as Label Encoder for pre-processing, fit transform of the target variable and consideration of target variable probability was kept intact. To achieve a model based on our requirements and proposed idea many changes were made.

Bagging (Bootstrap Aggregation) -Decisions trees tend to be sensitive to the data they are trained on, small changes to the training set can result in significantly different trees or tree structures. Random forest takes advantage of this by allowing individual trees to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging.

Random forest takes advantage of this process of randomization of data to each tree, which leads to higher tolerance to variance overall. It means that the variance levels may be varied for different trees but that doesn't act as a tradeoff when taking the forest in its entirety.

1.3.1. Working of the Random Forest Classifier:

Before any classification process, the data has to be prepared and pre-processed as standard procedure.

This is achieved by pre-processing package imported from package sklearn (Python). The method known as Label Encoder from the pre-processing package is used for data prepping. The Label Encoder is used to label encode and convert the data in machine readable form. It assigns a unique number (starting from 0) to each class of data. Here, the encoded data is the target variable or the "Prognosis" and used in fitting and transform fitting the model. For the Random Forest implementation, the Random Forest Classifier package is imported from sklearn. This model makes use of transform fitting by the package to fit the data. The transformed data being the target variable, "Prognosis".

The number of trees is an important factor in forest building. From testing, it was observed that number of trees i.e. n_estimators = 10 gives the better results (as opposed to the other number of trees we tested). Thus, the number of trees in our model is 10 and is given as input attribute to the Random Forest Classifier function used.

From the package sklearn, metrics and classification_report is also imported. In the model, four methods of the package Random Forest Classifier is used namely: fit, fit_transform, predict_proba and classes.

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

1.3.2. Steps in classification:

Using the symptoms given by the user initially, the code creates a list of the symptoms covered and a new data frame. The symptoms list created appends each symptom one at a time after checking if such a symptom exists in our training set of symptoms. The new data frame created such that it contains only columns of the entered symptoms.

1. Further, using the above data frame, probability of the possible diseases is determined. It checks each disease that has one or more symptoms of the entered symptoms and predicts the probability of the new list or class of diseases.
2. On calculation, a new list of the possible diseases is created. As an additional step to narrow down on the prediction, the disease with least or minimum probability is removed. At this stage, to get the disease names in user readable form inverse transform is taken to cancel the transform done at the beginning of the prediction.
3. On the output side, the symptoms covered, the possible diseases, probability of possible diseases and the remaining symptoms are displayed as lists.
4. The probabilities of the diseases are checked, if one of them is above 0.75 then the suspected disease is returned and breaks from the loop.
5. Else, if none of the probabilities are above 0.75, there exists a loop to allow the user to add more symptoms. This step is prompted by remaining symptoms of the possible diseases in the list.
6. There is also a provision to not add any symptoms by giving input as "None".
7. This loop continues for 3 iterations. If within 3 rounds, a disease has probability above 0.75, it is returned as suspected disease and loop is broken.
8. Else if by the end of 3 iterations, a singular suspected disease is not narrowed down and if none of the possible diseases have probability above 0.75, then the loop is broken.
9. In such a case where disease has not been able to be classified even after all the iterations, there exists a special round with our training data specific questions to ultimately point to a particular disease.
10. At the end of classification, the disease name and its remedies are displayed on the output screen.

3.4. Hardware

The block diagram is as shown in Figure 1. The proposed work is implemented using Raspberry Pi 3 Model B+ (RPi) board, which acts as the heart of the device. Starting off with user interface part, the user interacts with the device via speech input which will be captured by a USB microphone that is mounted on the RPi. Since the RPi in general doesn't inherently possess a microphone for capturing audio input, we use USB microphone for capturing the speech input. The audio is converted into the text format using speech to text convertor and is fed to the NLP software. Here the text is parsed, and the required keywords are extracted and matched with the dataset of disease-symptoms that is stored a priori in the RPi. The matched keywords are then passed to the classification algorithm that takes place using a tree which filters out the closely matching disease through further classification as per the user inputs. The classification and the predicted diseases are all displayed on a 5 inches HDMI LCD screen through Graphical User Interface (GUI) and also audio output facility is available on connecting a speaker or a headphone or an earphone to the RPi through the 3.5mm jack present on the RPi. Additionally, sensor DS18B20 Waterproof Temperature sensor is interfaced with the RPi in order to capture vital signs like temperature, which are transferred into the classification block. Having done the prediction of the disease along with the remedial information alongside, all the interaction data that includes the prognosis, the symptoms that led to the prediction and other information are all pushed to Google firebase.

DS18B20 Waterproof temperature sensor: DS1820 waterproof temperature sensor serves to retrieve body temperature data by placing the tip of the sensor on the armpit. The result of body temperature detection is received by Raspberry Pi and processed through conversion from temperature to voltage. The resulting voltage changes will

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

adjust to the condition of the body temperature of the patient at that time. The result of a translation made by Raspberry Pi will be displayed on the HDMI display attached to the Raspberry Pi. All patient data stored on Raspberry Pi can be called or viewed on the portable or personal computer.

The DS18B20 sensor interface with the Raspberry Pi is simple as shown in the figure below, making it a highly portable device. Connection complexity is less and the sensor is easy to handle, thereby ensuring optimum comfort when it is used for measuring the body temperature.

DS18B20 temperature sensor testing is done by placing the end of the sensor or pairing the sensor on the armpit for one or two minutes. Armpit is a body part that can use as a basis in the sensor readings. The readings of the body temperature sensor on the HDMI display in the form of Fahrenheit reading indicate that the sensor works well. Furthermore, to obtain accuracy testing of temperature sensor data, a digital thermometer was used as a comparison in getting the error rate. The success of this test can be seen on the HDMI display or by portable readings. The measurement results displayed on the HDMI display is an indicator that the sensor DS18B20 works appropriately.

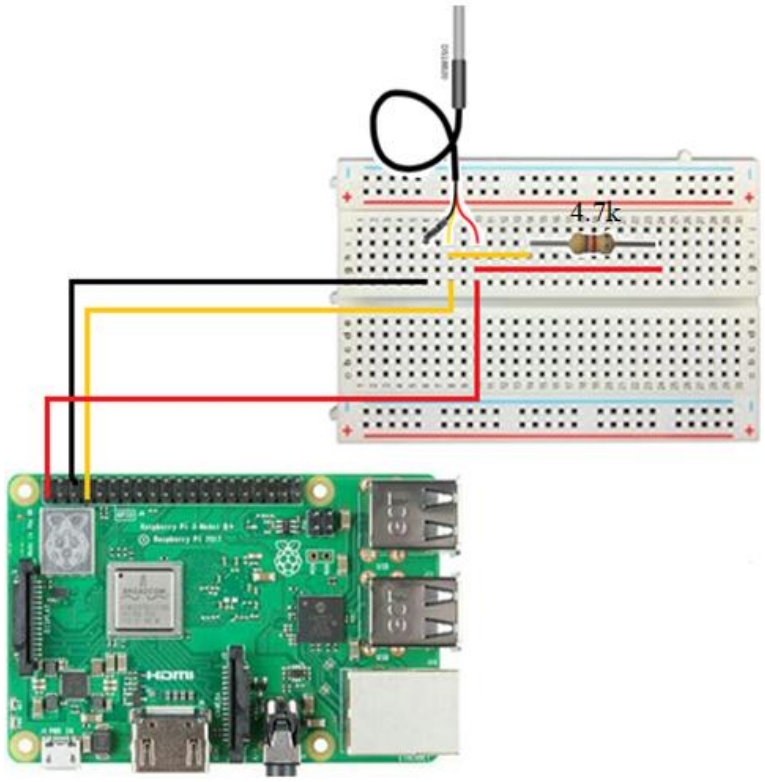

Figure 2. DS18B20 Temperature Sensor Circuitry

IV. RESULTS AND DISCUSSION

The proposed methodology was implemented using Python (version 3.7.3) and the graphical user interface (GUI) was developed using tkinter package and the results obtained are presented in this section. The voice input is taken into the system. The input sentence is processed and the possible symptoms are extracted. The possible symptoms are displayed and asked for confirmation from the user. This step double checks for only the required symptoms and ignores the not necessary symptoms. The confirmed symptoms are displayed. These are the symptoms that will be used further for the classification process [11].

The classification model used here is random forest classifier. The classification accuracy of the model implemented was tested using several test cases. All the test case results can be grouped into three different classes. First one being the case when the initial input itself is sufficient to predict the disease. The second case is the one in which it asks further questions about the symptoms the user is facing and eventually predicts the symptom. In the

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

third case, even after asking several questions about the symptoms, the classifier does not predict one possible suspected disease, instead gives a list of probable diseases.

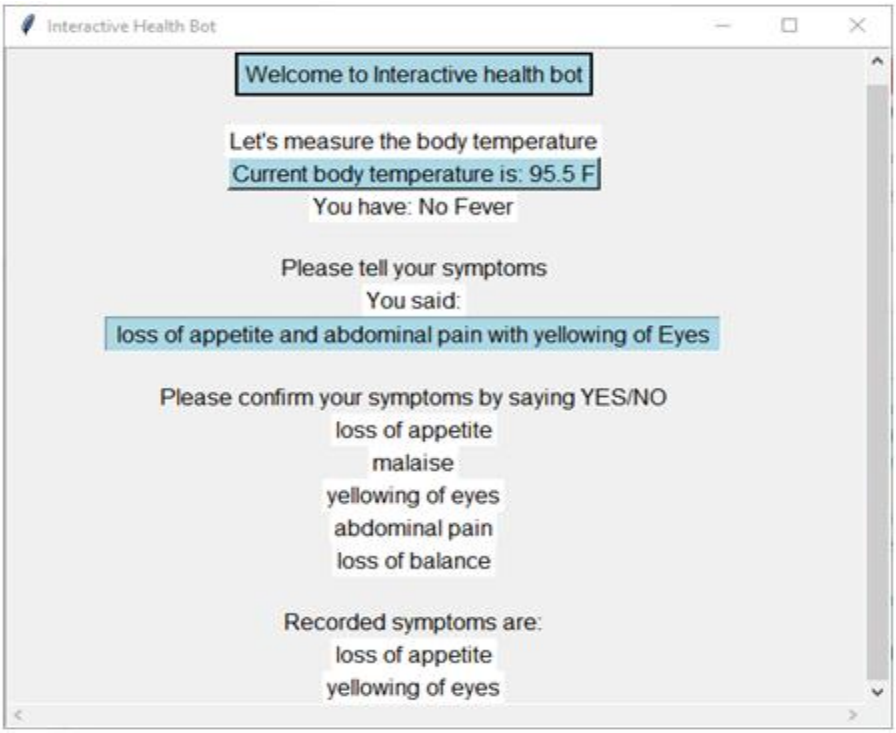

Figure 3. GUI Window Screenshot

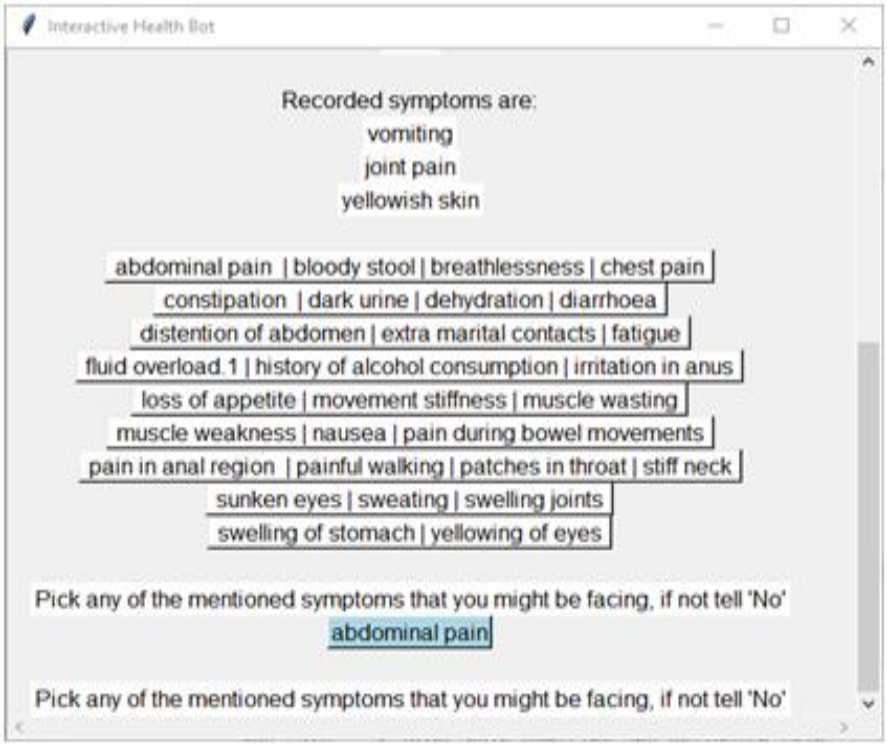

Figure 4. Displaying Remaining Symptoms

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

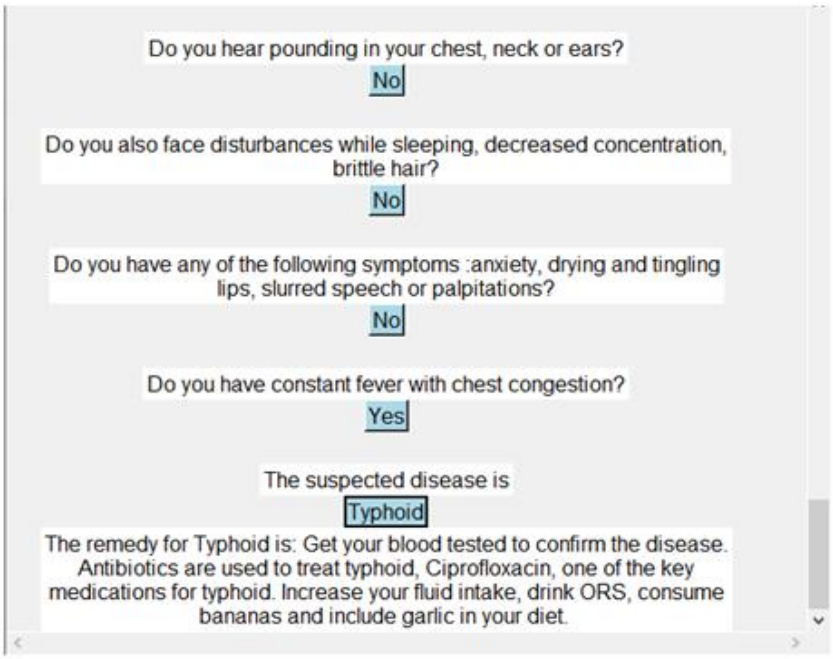

Figure 5. Special Question Implementation and Displaying the Remedies

In the further modification, we have implemented special questions. Special questions were framed regarding the disease. This is based on research and it aims on dealing with the symptoms that are not there in the considered symptoms. This helps in deciding between two or more similar symptom diseases.

For classification, based on the correct prediction of the disease when legitimate symptoms are provided to the classifier, accuracy of it was measured. Initially while the tests were carried out, several problems were encountered which led to wrong prediction or ambiguous prediction of diseases. Later, these errors were corrected and the same set of inputs was tested again to check if the problems were solved. In Figure 6, there is a stacked bar graph that depicts the number of wrong and right predictions before and after the corrections was made.

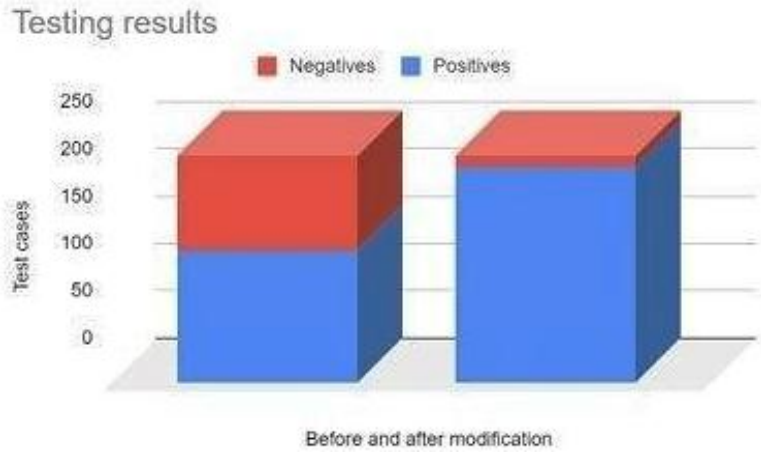

Figure 6. Testing Results

Before the modification, there were 140 positives and 101 negative results among the 241 test cases. It accounted to 58.2% accuracy of the classifier. After the corrections were made in both natural language processing part and classification thresholds, the positives went up to 229 with only 12 negatives. The current accuracy of the classifier is at 95.02%.

All the entries are recorded in a systematic manner in a Firebase console database. It is as shown below. The recorded entities are the diagnosed disease, reading from the temperature sensor, the symptoms entered and the

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

timestamp. The next figure shows the frequency of the occurrence of each disease in the database entry. This tally gives us knowledge of what diseases are being recorded the most.

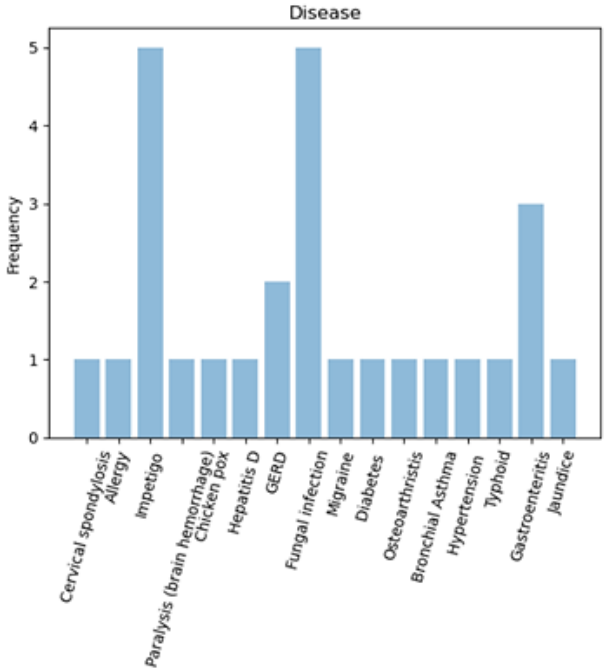

Figure 7. Disease Tally

Figure 8 shows similarly the tally for the symptoms. It gives us an estimate of the most number of symptoms being recorded.

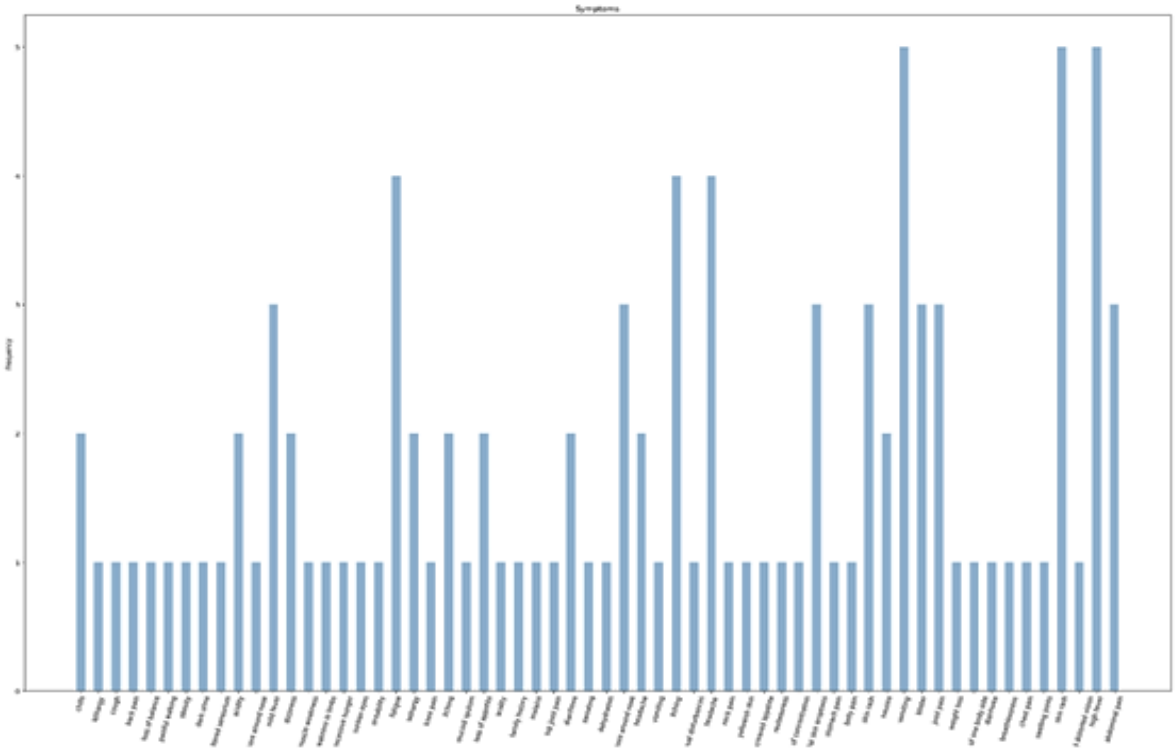

Figure 8. Symptom Tally

ISSN- 2394-5125 VOL 7, ISSUE 15, 2020

To analyze the frequency of symptoms occurring for each disease, the report of a particular disease taken as an example is shown. This information is pretty useful as it tells us what the most occurring symptoms are for any diagnosed disease.

```
Diabetes
['weight loss', ' fatigue', ' lethargy', ' restlessness', ' obesity', ' increased appetite']
count of symptom in disease:  Diabetes   =  [(' fatigue', 4)]
count of symptom in disease:  Diabetes   =  [(' lethargy', 2)]
count of symptom in disease:  Diabetes   =  [('weight loss', 1)]
count of symptom in disease:  Diabetes   =  [(' obesity', 1)]
count of symptom in disease:  Diabetes   =  [(' increased appetite', 1)]
count of symptom in disease:  Diabetes   =  [(' restlessness', 1)]
```

Figure 9. Disease Report

The interactive bot is used for the diagnosis of all sorts of diseases. Most of the diseases consist of fever, be it high or low fever as a symptom. Generally, fever is the precursor to any kind of disease that the humankind can think of. Onset of fever may or may not indicate the presence of a particular disease, but it is very important to detect the presence of fever. Fever is a common symptom that most of us would have experienced in our lifetime, because of myriad factors. Fever is detected by checking the temperature of a human body mainly with the help of a thermometer. Normal human body temperature is around 98.6^{o} F (37^{o} C). Anything above the normal human body temperature gives rise to feverish condition or simply fever. Fever can be mild or high depending on the body temperature. Body temperature up to 102^{o} C will cause mild fever. Temperature between 102^{o} C and 104^{o} C will lead to high fever. Anything above that, gives rise to high-graded fever which ultimately can lead to the death of the person. Hence, it is very much necessary to detect the presence of fever beforehand, which can help in the detection of any possible disease quickly thereby saving time and money for a patient.

Our interactive bot comes equipped with a body temperature measurement sensor known as "DS18B20 Waterproof Temperature Sensor". It is compact in design, flexible and very easy to use. This sensor can be interfaced to Raspberry Pi very easily where the connection complexity is less. DS18B20 sensor provides the accurate body temperature measurements and few tests were conducted physically to verify the accuracy of the temperature readings. The readings were accurate enough which will help in the detection of fever accurately. Below are the test results of the physical test that was conducted in order to verify the sensor reading accuracy:

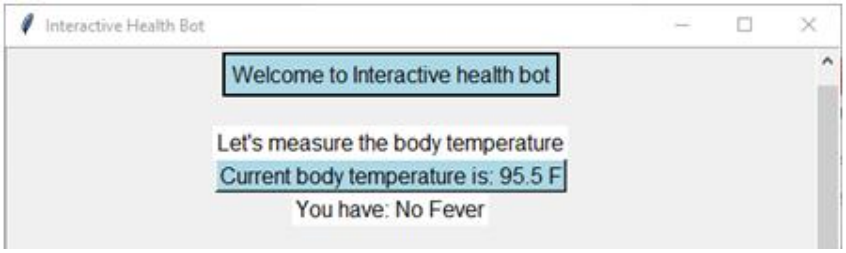

Figure 10. DS18B20 Temperature Sensor Readings

As from the above test results of a normal user where the normal body temperature is about 95.5^{o} F, we can verify the accuracy and consistency in the temperature readings which are the factors influencing the selection of a particular device. Since the temperature readings are consistent and accurate enough, the selection of DS1820 Temperature sensor for our interactive bot is justified. These readings will help the user in detection of fever very easily, thus serving its intended purpose. Presence of high or mild fever will speed up the NLP classification process in the detection of a possible disease.

V. CONCLUSION

An interactive bot for predicting diseases using symptoms is proposed and implemented. The model uses natural language processing and decision trees for classification. The natural language processing acts on the speech input that is taken and classification acts on the keywords that are output of NLP. A temperature sensor is interfaced to the Raspberry Pi which helps in collecting a vital parameter. The collected parameter and all the related

information regarding the predicted disease, is uploaded onto the Google firebase database where analysis on the data is carried out for deriving important information. It is a stand-alone model that works as personal health assistant and this model can be used for other disease management. As a future study, we intend to work on the data analysis (cloud) part of the model.

REFERENCES

1. Mahboob Qaosar, Saleh Ahmed, Chen Li, Yasuhiko Morimoto, "Hybrid Sensing and Wearable Smart Device for Health Monitoring and Medication: Opportunities and Challenges", The 2018 AAAI Spring Symposium Series.
2. Arlene E. Chung, Ashley C. Griffin, DashaSelezneva, David Gotz, "Health and Fitness Apps for Hands-Free Voice-Activated Assistants: Content Analysis", JMIR mHealth and uHealth 2018.
3. Emad S. Othman, "Voice controlled personal assistant using raspberry pi", International Journal of Scientific & Engineering Research Volume 8, Issue 11, November-2017.
4. Rajasekaran Rajkumar, Vasudev Sharma: *Visualization of data mining techniques for the prediction of Breast Cancer with high accuracy rates*. Journal of Computer Science 01/2019; 15(1)., DOI:10.3844/jcssp.2019.118.130
5. M. Sathya, S. Madhan, K. Jayanthi, "Internet of things (IoT) based health monitoring system and challenges", International Journal of Engineering & Technology, February 2018, 7 (1.7) (2018) 175-178.
6. Rajkumar Rajasekaran, Govinda k, Ashrith Reddy, Uday Sai Reddy, Yashwanth Reddy: *Visual Analysis of Temperature Time Series and Rainfall Using Big Data*. DOI:10.36872/LEPI/V50I3/201023
7. Kavitha Jaiswal, Srichandan Sobhanayak, Bhabendu Kumar Mohanta, Debasish Jena, "IoT-Cloud based framework for patient's data collection in smart healthcare system using raspberry-pi", 2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA), 21-23 November 2017.
8. M.Saari, "Survey of Prototyping Solutions Utilizing Raspberry Pi", 2017, 40thInternational Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).
9. Rajkumar Rajasekaran, Govinda k, Ashrith Reddy, Uday Sai Reddy, Yashwanth Reddy: *Visual Analysis of Temperature Time Series and Rainfall Using Big Data*. DOI:10.36872/LEPI/V50I3/201023
10. Rajasekaran Rajkumar, Jolly Masih, K.Govinda: *An analysis of mobile pass-codes in case of criminal investigations through social network data*. International Journal of Computers and Applications 09/2019;, DOI:10.1080/1206212X.2019.1662169
11. Rajasekaran Rajkumar, K.Govinda, Anushka Jindal, Rushil Mehtani, Jolly Masih, Maddineni Charan Sai: *Visualization Effect of Changing Climatic Trends on Natural Calamities:Hurricanes*. DOI:10.36872/LEPI/V50I3/201022
12. Jolly Masih, Rajasekaran Rajkumar: *Internet Addiction Disorder and Mental Health in Adolescents*. DOI:10.4172/2167-1044.S13-002
13. Gunjan Dhole and Nilesh Uke, "Medical information extraction using Natural Language Processing interpretation", Advances in Vision Computing: An International Journal (AVC), Vol.1, No.1, March 2014.
14. Gustav Cederblad, "Finding Synonyms in Medical Texts: Creating a system for automatic synonym extraction from medical texts", Linköping University, Bachelor thesis, 18 ECTS, Spring term 2018.
15. Rajasekaran Rajkumar, Rishabh Jain, Sruthi Mohan: *Patient health monitoring system and detection of Atrial fibrillation, fall and air pollutants using IOT Technologies*. Incorporating the Internet of Things in Healthcare Applications and Wearable Devices, 10/2019: pages 165-183; IGI-Global., ISBN: 2327-9354, DOI:10.4018/978-1-7998-1090-2.ch011
16. Bhumika Gupta, AdityaRawat, "Analysis of Various Decision Tree AlgorithmsforClassification in Data Mining", International Journal of Computer Applications (0975–8887) Volume 163 – No 8, April 2017.
17. Deepak Tripathee, Raffles University, Department of Engineering Management, Ph.D. Scholars, "Hospital Database Management System", IJCSMC, Vol. 5, Issue.4, April 2016, pg.71 – 73.
18. Chethana Saligram, Deepali B K, Gowri K S, J Ajay Kumar, Jyoti R Munavalli, "Symptoms Extraction from a Voice Input using Natural Language Processing", International Journal of Engineering Research & Technology (IJERT), Vol. 9 Issue 04, April-2020.
19. Low cost audio based intelligent guidance system for visually impaired people,Vattumilli Komal Venugopal, Alampally Naveen, Rajkumar.R, K. Govinda, Jolly Masih. International Journal of Psychosocial Rehabilitation, ISSN: 1475-7192.DOI: 10.37200/IJPR/V24I3/PR200809, Pages: 515-520.