International Journal of Research in Engineering, Science and Management
Volume 4, Issue 6, June 2021
https://www.ijresm.com | ISSN (Online): 2581-5792

Virtual Medical Assistant Using Machine Learning

K. Anitha[1*], C. S. Aravind[2], K. Deepikha[3], N. Uma[4]
[1,2,3]*Student, Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, India*
[4]*Assistant Professor, Department of Information Technology, Sri Venkateswara College of Engineering, Sriperumbudur, India*

***Abstract*: In the current scenario, people are very much hesitant to visit hospitals in fear that they might catch an infection. This fear escalates when they are made to wait at the hospital for a long time. Also, doctors find it difficult to write or type prescriptions with gloves and protective clothing on. The aim of this project is to design and develop an application which acts as a virtual assistant to the doctor. The application at the patient's end would ask patients to register themselves for an appointment. They are asked to record their symptoms. The application then extracts the symptom related words from the recording and predicts the possible diseases. The extracted symptoms and diseases assist the doctor in taking decisions. The doctor could listen to the recording if he/she wants to. The doctor could ask the patient to take some tests, if the patient needs any, and also schedule a call with the patient if necessary. The system would also help the doctor record his prescription by voice, and convert it to prescription format and send it to the patient. This application would save the doctor's time and energy and avoid misinterpretation of drug names, thereby simplifying the whole process. The proposed system will be developed in python, applying principles of Natural Language Processing, Keyword Extraction and Machine Learning.**

***Keywords*: keyword extraction, machine learning, natural language processing.**

1. Introduction

In this current Century daily tasks are completed efficiently with the help of virtual assistants. The personal assistant used by the users such as smart environment control, calendars management, appointment booking and many more things are virtual assistants that use artificial intelligence. Many companies have developed their own virtual assistant such as Siri (Apple), Assistant (Google) and Alexa (Amazon). These virtual assistants understand the user requests, provide an interactive user interface (text, speech, or both) and have the ability to handle logical and complex tasks. Using the machine learning model, these assistants analyse, predict and generate a valid response for the given user requests and inputs.

In recent times, the healthcare sector adopted machine learning which facilitates advance patient care, disease diagnosis, guidance in treatment, manages patient information and streamlining of administrative data in hospitals. Healthcare sector is also shifting or advancing towards virtual assistants. It prevents unnecessary need of clinicians, disease spreading and patients visit to the hospitals for minor symptoms which improves patient safety. It reduces excessive manual administration tasks in hospitals, repeated documentation of patient details for every visit, hand-written prescription errors and travel burden. Few healthcare systems automatically capture the data of doctor's hand written prescriptions in the background and allow them to store it. These prescription systems generate the prescription by taking manual input from the doctor and burnout the time of doctors. In some hospitals, the doctor types the prescription into a document along with the medicines, dosage etc. In this system, as the prescription has been typed, there won't be any misinterpretation of the prescription, making it better than hand written prescriptions. But since the doctor has to type all the information, this might be time consuming, considered the doctor attends to multiple patients per day.

The virtual medical assistants help the physicians, patients by providing the proper information and guidance and organizes the patient data for future references. The assistant automates the documentation process, covers administrative tasks, provides flexibility in appointment booking and manages patient-doctor records. Several assistants are used for predicting the diseases based on medical symptoms such as Sensely, Your.MD and MedWhat. These systems analyze the symptoms and generate relevant diseases. Some virtual Healthcare Assistant systems cannot meet the basic demands such as medical assistance from doctors, communication with the clinicians and poor internet bandwidth. The systems web service stores patient health issues, ailments and provides the history. The relevant disease is predicted from user symptoms and prescription is generated. The system does not include the doctor recommendation, advice and cannot ensure the accuracy of disease and health conditions.

2. Related Works

[1] Natural language processing is used to retrieve medical data from clinical documents. How the NLP tools such as relationship extractors, parts of speech taggers, tokenizers and entity recognizers are used for information extraction is explained by this system. The system inputs a text relating to a list of symptoms and tries to provide relevant possible answers.

*Corresponding author: anithaakm47@gmail.com

[2] Raspberry pi is used to develop a personal assistant which performs tasks like reading news, weather prediction and much more. This device uses online speech to text conversion by the AVS. The system uses open source modules and continuous internet connection is required to perform the task. Using medical knowledge base and natural language processing a symptom checker is developed from open source medical resources, medical texts and information extraction engines a knowledge base is created automatically. Unified Medical Language Systems (UMLS) integrates the collected data into a knowledge base.

[3] Automatic suggestion of possible diseases based on patient input symptoms. Naive Bayes classifier is used for prediction. The system reduces the doctor consultation fees but failed to prove the efficiency in mapping the symptoms. [4,5] Various machine learning techniques are used for heart disease prediction such as association rule mining, clustering and classification algorithms. The chosen dataset is pre-processed and split into training and testing sets. The efficiency of heart disease prediction is calculated. These systems fail to explain the detailed working of techniques used. [6,7] Predicts the possible common diseases based on the symptoms chosen by users using data mining techniques. The model consists of various modules to take care of the prediction process. For training the dataset ID3 algorithm is used. Modules perform authentication of doctor and patient login, allows to choose doctor and helps for conversation with doctor.

[8], [9] Extract prescription from the collected medical discharge summaries. The system used hybridized machine learning classifiers and semi-supervised machine learning approaches to extract medication names, drug dosages and its frequencies.

3. Proposed System

In the current scenario, when the patients visit the hospital they have to wait in the reception area to book an appointment and they have to provide all their details which is time-consuming and increase the workload of the receptionist. If any test reports need to be taken by the patients then they have to re-visit, book an appointment and then have to consult the doctor. This is a lengthy and tedious process. This proposed system has been implemented to reduce patient's unnecessary visits to hospitals and to ensure their safety during pandemic situations. The system consists mobile application for the patient's side and a web application for the doctor's side. The patients are allowed to choose the department of the doctor. This also eases the process of storing patient and doctor details.

The mobile application on the patient side allows patients to register themselves for an appointment by entering their details. An appointment is set up with the patient and a doctor in the department which the patient chooses. Then the patient records their symptoms as an audio and sends it. This audio is saved to a cloud database. The application on the doctor's side listens to the voice recorded symptoms and extracts the symptom related words. With the extracted symptoms, the application predicts the possible diseases the patient might suffer from. This is generated in the form of a chart. The doctor can view the chart and make a decision accordingly. The doctor dictates the drugs and dosage verbally and the application converts this speech to text and generates the prescription as a pdf. The doctor might also ask the patient to take some tests, or schedule a virtual meeting with the patient. The patient can choose the doctor's department. Otherwise, the application will set up an appointment with a doctor from a department based on the predicted disease. Once the prescription is generated, the patient is notified through the app. The patient can download the prescription. Similarly, the patient can scan and upload test results through the app. At the end of an appointment, a consultation bill is sent to the patient, which they might pay online.

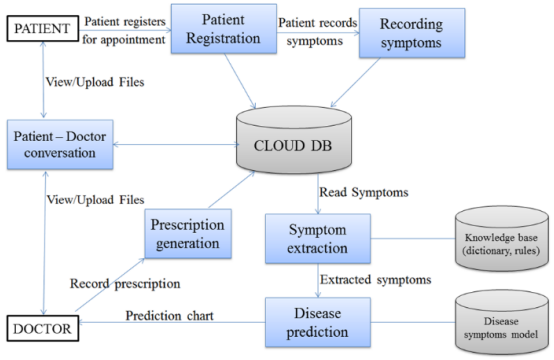

Fig. 1. System architecture

Fig. 1 depicts the overall architecture of the Virtual Medical Assistant.

4. Modules and Analysis

The proposed modules are implemented using java and python. The mobile application is developed in android studio to help patients to utilize the system. This application is used by patients to register, record symptoms, fix appointments with the doctor, view their prescriptions and to view bill details. The patient can choose to upload results for tests that the doctor has asked them to take. The patient can upload it in the form of pdf or images. All the details entered in the mobile app are stored in firebase cloud database.

The doctor's side web application is implemented with the flask framework. This application assists doctors by predicting the possible disease from the patient's recorded symptoms. It uses Google's speech recognition API for the speech to text conversion. The doctor can choose to view patient details, all the active appointments set up with that doctor are displayed along with the patient details and appointment time. The application in doctor's mode provides an interface for webkit Speech Recognition tool in JavaScript to record audio in real time to generate the prescription. The user interface is built using HTML, CSS styles and JavaScript. The prescription, consultation bill and virtual meet link is sent to the patient's mail ID which uses Smtplib to generate mail. NLP and prediction algorithms will be running in the backend while a user interface present in the frontend provides the user a visual experience. All of these methods are implemented and their performance is enhanced.

A. Module 1 Patient Registration

The Patient registers through a mobile app. The patient enters their details and chooses the department (ENT/ Cardio/ Ortho etc.). Later, the patient is asked to record the symptoms in the form of an audio. Once the patient uploads these details, an appointment is set up for the patient with a doctor in the department that the patient chose and the patient is notified of the same.

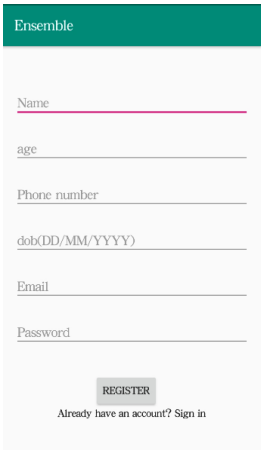

Fig. 2. New Patient Registration

Fig. 2 shows the registration screen for a first time user. A patient can register and then use the email and password for his/her next sign in.

B. Module 2 Patient Dashboard

For an existing patient (already registered), the patient is taken to the patient dashboard once he/she logs in. The patient can view the following details from the patient dashboard:

1) Download and view prescription details
2) View/edit patient's details
3) Upload test results from phone
4) View bill details for completed appointments.

The prescription details are available for a patient only if the doctor has processed the same for that patient. The prescription also contains the disease that the doctor diagnosed and the general advice from the doctor's side. The patient can choose to upload results for tests that the doctor has asked them to take. The patient can upload it in the form of pdf or images.

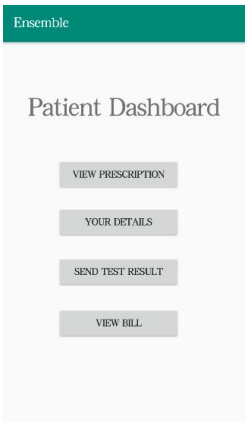

Fig. 3. Patient Dashboard

Fig. 4. Choosing Department

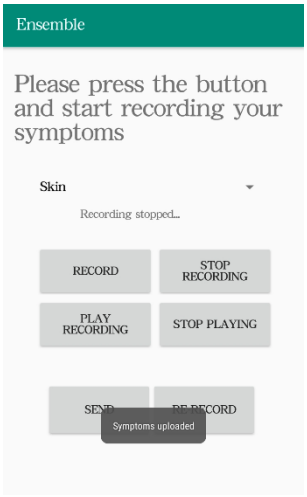

Fig. 5. Recording Symptoms

A registered user is taken to the patient dashboard as shown in Fig. 3. For a user who registers for the first time, he/she is taken to the symptom recording screen as seen in Fig. 4 and Fig. 5. The patient is asked to choose the type of specialist he/she might want to visit. They can also leave this section blank. Then the patient records their symptoms through audio. The patient can listen to the recording and decide whether to send it to the doctor or to re-record. Once the patient uploads these details, an appointment is set up for the patient with a doctor in the department that the patient chose and the patient is notified of the same.

C. Module 3 Doctor Dashboard

The doctor dashboard is a web application implemented with flask framework. Doctors can login with their username and password. Once the doctor logs in, he/she is taken to the dashboard, which has the following options.

1) View patient Details
2) View previous patient history
3) Process bill

When the doctor chooses to view patient details, all the active appointments set up with that doctor are displayed along with the patient details and appointment time. When the doctor

chooses one of the patients, the details of that patient will be displayed. This includes the patient's symptoms (in the form of audio uploaded by the patient), the audio converted to text, symptoms extracted from the text and diseases predicted by the model based on those symptoms. Depending on these details, the doctor can make one of the following decisions.

1) Record Prescription
2) Schedule virtual/in-person meeting
3) Ask the patient to take test

When the doctor chooses to view previous patients' history, details of all completed and closed appointments will be displayed for doctor's reference. When the doctor chooses to process the bill for a patient, the doctor would enter the consultation fees and the patient would be notified of the same.

Fig. 6. Doctor Dashboard

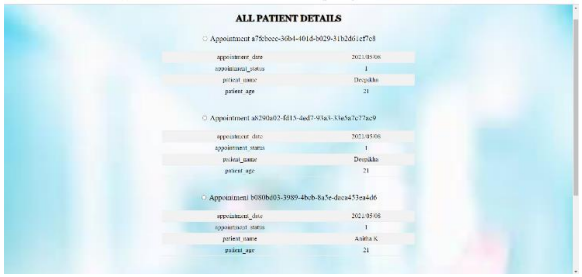

Fig. 7. All Patient's Details

The doctor dashboard is shown in Fig. 6. The doctor can choose to view their patients' details, view previously closed appointment details, or generate the bill for completed appointments. The fig. 7 shows only the patient details whose appointment status is either open or in progress under the particular doctor they have chosen. The doctor can choose to view the particular patient by clicking the view patient detail button.

Fig. 8. Patient's Details

The patient details screen is the screen containing all details of the chosen patient. This is shown in Fig. 8 and Fig. 9. For a selected patient, the doctor can view their basic details. This includes the patient's symptoms (in the form of audio uploaded by the patient), the audio converted to text, symptoms extracted from the text and diseases predicted by the model based on those symptoms. The doctor can also listen to the recording of the patient, update the appointment status and download the test results uploaded by patients. The appointment status can be changed to open, In-progress, bill pending and closed. The doctor can also download the test results if any uploaded by the patients. The doctor can choose to record the prescription or schedule a virtual/in-person meeting with the patient or ask the patient to take tests.

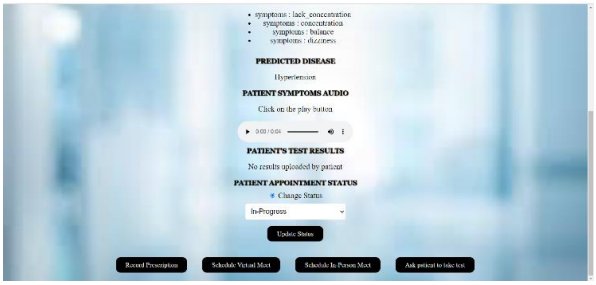

Fig. 9. Patient's Details

D. Module 4 Symptom Extraction

When the doctor chooses a particular patient to view their details, the application on the doctor's side would download the symptom audio uploaded by the patient during registration and listen to this symptom audio of the patient and extract the symptom related words from it. This is done in the steps explained below.

1) Speech to Text Conversion

The symptom audio file recorded by the patient is read from the firebase cloud database and then processed using google speech recognition and is extracted as text.

2) Tokenization

The function word_tokenize() is used to split a sentence into words. The text extracted from the audio file is tokenized. The output of word tokenization can be converted to Data frame for better text understanding. It can also be provided as input for further text cleaning steps such as stop words removal, numeric character removal or pos tagging.

3) Stop Words Removal

Stop words are the most common words in any natural language. For the purpose of analyzing text data and building NLP models, these stop words might not add much value to the meaning of the document. Removing stop words can potentially help improve the performance as there are fewer and only meaningful tokens left. Thus, it could increase classification accuracy.

4) Pos Tagging

A part-of-speech tagger, or POS-tagger, processes a sequence of words and attaches a part of speech tag to each word. Pos tagging is applied on tokenized words and only nouns are extracted as most of the symptoms are in noun form.

E. Module 5 Disease Prediction

Once the symptoms are extracted, the application then

predicts the disease that the patient might be suffering from, based on the symptoms. This is a process which has its own steps as explained below.

1) Dataset Loading

The system uses the symptom dataset that has 4921 rows with 130 columns, 129 of them being symptoms experienced by patients and the final column is the class label that contains the disease name based on the symptoms. The dataset is loaded as 75% for training and 25% for testing. The train data is used to train the machine with patient's symptoms and disease. The test data is used to evaluate the machine and to predict the disease.

2) Decision Tree Classifier

Decision Trees are a non-parametric supervised learning method used for both classification and regression tasks. Decision Tree Classifier is a class capable of performing multi-class classification on a dataset. Decision tree is a controlled method used for the prediction of unconditional as well as numerical values. It represents the data occurrences along with their class label in the form of a tree. A set of rules can be constructed from the tree which can be used to order the unknown data record to its output value. A test on an attribute is performed on the core node. The result of the test is depicted by the branch of the tree and class labels are present at the leaf node. The split is recognized from the parameter or the factor which is dogged to be the best splitter or differentiator. The data is already split into training and test data. The next step is to train the decision tree algorithm on this data and make predictions. Decision Tree Classifier class is used for this prediction. The fit method of this class is called to train the algorithm on the training data, which is passed as parameter to the fit method. Once the classifier has been trained, the prediction is done on the test data. To make predictions, the predict method of the Decision Tree Classifier class is used.

3) Random Forest Classifier

In a random forest a randomly selected set of attributes is used to split each node. Every node is split using the best split among a subset of predictors which are purposely chosen at random at the node. The random forest (RF) is a hierarchical collection of tree structured base classifiers. Text data usually has many numbers of dimensions. The dataset contains a large number of irrelevant attributes. Only a few important attributes are informative for the classifier model.

RF algorithm uses a simple predetermined probability to select the most important relevant attribute. This methodology is different from the ones followed in standard trees, in which each node is split using the best split among all attributes which are available in the dataset considered. Further, new values are predicted by combining the predictions of many constructed decision trees. Random forest represents an ensemble model or an algorithm because it derives its final prediction from numerous individual models. These individual models could be of either similar type or different type.

Random forest pseudocode:

1. Randomly select "n" features from total "k" features. Where n < < k
2. Among the "n" features, calculate the node "n" using the best split point.
3. Categorize the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until the "l" number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of trees.

4) Linear Support Vector Machine

Linear SVM is the newest extremely fast machine learning (data mining) algorithm for solving multiclass classification problems from ultra large data sets that implements an original proprietary version of a cutting plane algorithm for designing a linear support vector machine. Linear SVM is a linearly scalable routine meaning that it creates an SVM model in CPU time which scales linearly with the size of the training data set. Our comparisons with other known SVM models clearly show its superior performance when high accuracy is required.

F. Module 6 Prescription Generation

When the doctor decides to record a prescription for the patient, he/she could do so by recording the drug names, dosage, number of days to take the medicine and BF/AF. The doctor could add additional instructions or general advice for the patient. The web application uses Webkit Speech Recognition tool in JavaScript to record audio in real time as the doctor recites it and converts it to text and displays it on the screen. The doctor can also edit the converted text to change any of the drug names or advice. Once the doctor completes reciting the medicines and advice, he/she would choose to generate the prescription pdf. A pdf is generated from this data that will contain the doctor and patient details along with the disease diagnosed, medicines in table format and the general advice from the doctor. This prescription is mailed to the patient's mail ID.

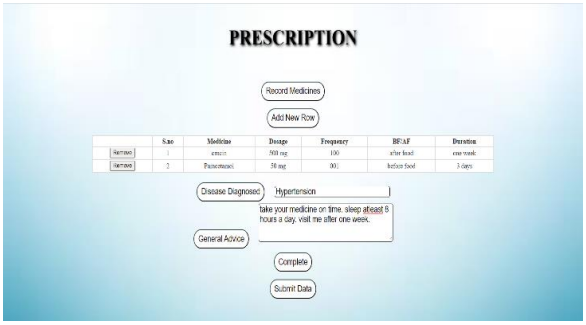

Fig. 10. Prescription Recording

Fig. 10 shows the prescription recording screen of the doctor. The doctor has a choice to either record their prescription via audio or input through a keyboard. Once done, the doctor can submit the data for prescription PDF generation.

When the doctor submits the prescription data, from the recorded data, a prescription pdf is automatically generated. The doctor also gets to view the prescription as shown in Fig. 11.

When the doctor submits the prescription data, from the recorded data, a mail is sent from doctor side to the patient as shown in Fig. 12. The patient can either download the

prescription through mail or through their mobile application.

Fig. 11. Prescription PDF Generation

Fig. 12. Prescription Mail to Patient

G. Module 7 Test to be Taken

After viewing the patient's details, the doctor can ask the patient to take some tests if necessary. The patient can take these tests at nearby labs and upload the test results through the mobile app. The doctor can enter the tests to be taken in the form of a table. They can add and delete entries. Once the doctor enters the tests to be taken, the patient will be notified of the same through mail.

Fig. 13. Recording tests to be taken

The doctor records the list of tests that need to be taken by patient as shown in fig. 13 and submits the data.

When the doctor submits the test data, from the recorded data, list of tests is sent to the patient via Mail as shown in Fig. 14. The patient can take these tests at nearby labs and upload the test results through the mobile app.

H. Module 8 Schedule Virtual/In-Person Meeting

If the doctor decides that he/she needs to interact further with the patient before coming to a conclusion, the doctor can schedule a meeting with the patient. If the doctor wants to examine the patient in-person, he/she can schedule an in-person meeting. Else, the patient can schedule a virtual meeting (using google meet). The doctor would choose the date and time of the appointment. Once the doctor schedules a meet, a google calendar invite will be sent to both the patient and doctor via mail.

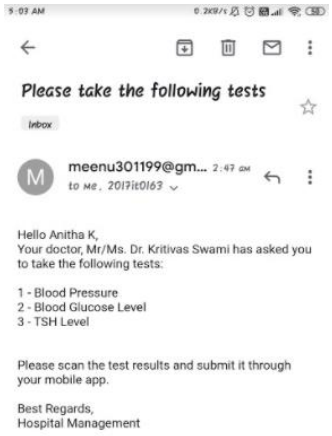

Fig. 14. Test to be taken mail

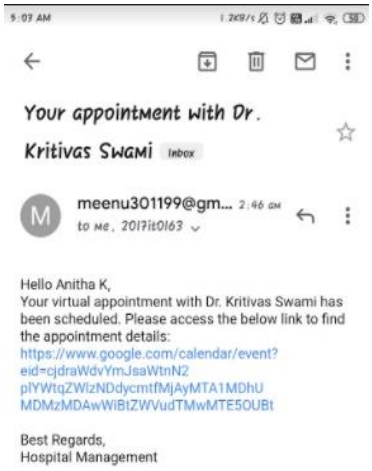

Fig. 15. Scheduled Meet Link Mail

The doctor can choose the date and time and schedule a virtual meeting with the patient. Once the doctor submits the details, the virtual meet link is sent to the patient through mail as shown in Fig. 15 and mobile app.

I. Module 9 Generate Bill

Once an appointment is completed, the doctor can choose to generate a bill for that patient. When the doctor chooses the 'generate bill' option, all the completed appointments will be shown. The doctor can choose which appointment to process. The doctor will enter the consultation fees and the patient would be notified through both email and mobile app. Once the patient completed the bill payment, the status of the appointment will be changed to 'closed'.

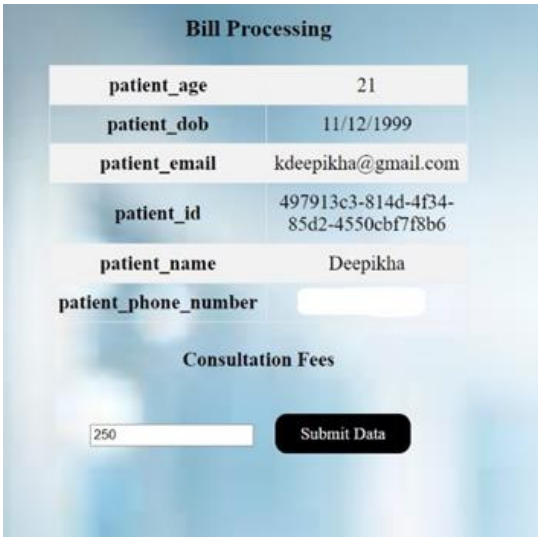

Fig. 16. Bill Generation

Fig. 17. Bill Generation Mail

The doctor will enter the consultation fees and submit the data as shown in Fig. 16. The patient would be notified through both email as shown in Fig. 17 and mobile app. Once the patient completed the bill payment, the status of the appointment will be changed to 'closed'.

5. Conclusion and Future Work

The proposed and implemented system aims to reduce the amount of time consumed in creating and accessing patient records. This project is made available to everyone through their smartphone. Patients can book appointments at home, avoiding fear and inconvenience that comes with visiting a hospital. The project also provides assistance to doctors in real-time. At the time a doctor views a patient's record for the first time, the doctor will have a report at hand which will help him/her in the diagnosis of disease. The project also provides an innovative solution to solve the problem of illegible handwritten prescriptions. It is convenient for doctors to record prescriptions by voice because it's difficult to write or type with gloves and protective clothing on. Voice-based e-prescription needs a minimal change in the workflow of doctors but in the long run, it will create a huge impact in developing a digital ecosystem for patients. Overall, the system minimizes the need for patients to visit hospitals. The system makes everything virtual, assisting both doctors and patients in the process.

Future enhancements will also include a more user friendly interface on both the doctor and patient's side. The application will also avail a chat window for the doctor and patient to have a conversation directly. Future work will also include increased security constraints on the patient data. Patient's data is sensitive and needs to be secured. No unauthorized person can access the patient's information. Thus, in addition to firebase's security, encrypting the patient's data would provide an added layer of security. It is aimed to design the system to be usable in the real hospital ecosystem to test and validate the implementation and to analyze the impact it will create in the healthcare domain.

References

[1] Gunjan Dhole and Nilesh Uke, "Medical information extraction using Natural Language Processing interpretation", Advances in Vision Computing: An International Journal (AVC), March 2014.

[2] Emad S. Othman, "Voice Controlled Personal Assistant Using Raspberry Pi", International Journal of Scientific & Engineering Research, Vol.8, Issue 11, November 2017.

[3] Nikita Kamble, "Smart Health Prediction System Using Data Mining". International Journal of Scientific Research in Computer Science Engineering and Information Technology, Volume 2, Issue 5, 2017.

[4] Sharmila S, "Analysis of Heart Disease Prediction Using Data Mining Techniques," International Journal of Advanced Networking & Applications, Volume 8, Issue 5, 2017.

[5] Vidya Zope, Pooja Ghatge, Aaron Cherian, Piyush Mantri, and Kartik Jadhav, "Smart Health Prediction using Machine Learning", International Journal for Scientific Research & Development, Volume 4, Issue 12, 2017.

[6] Shratik J. Mishra, Albar M. Vasi, Vinay S. Menon, K. Jayamalini, "GDPS - General Disease Prediction System," International Research Journal of Engineering and Technology, Volume 5, Issue 3, March 2018.

[7] Pooja Reddy G, "Smart E-Health Prediction System Using Data Mining," International Journal of Innovative Technology and Exploring Engineering, Volume 8, Issue 6, April 2019.

[8] Patrick J, Li M, "A cascade approach to extracting medication events," in Proceedings of the Australasian Language Technology Association Workshop, 2009.

[9] Carson Tao, Michele Filannino, and Özlem Uzuner, "FABLE: A Semi-Supervised Prescription Information Extraction," 2010.