2019 22nd International Conference of Computer and Information Technology (ICCIT), 18-20 December, 2019

Disha: An Implementation of Machine Learning Based Bangla Healthcare Chatbot

Md. Moshiur Rahman, Ruhul Amin, Md Nazmul Khan Liton and Nahid Hossain
Department of Computer Science and Engineering
United International University, Dhaka, Bangladesh
moshiurrahmanb11@gmail.com, ruhulamin6678@gmail.com, liton079@gmail.com, and nahid@cse.uiu.ac.bd

***Abstract*— While humans as a whole do live longer nowadays than ever before, we now suffer certain illnesses to a degree never seen in the past - including rising rates of Diabetes, Hypertension, Hypotension, Cholesterol imbalance, obesity and, ailments such as fever. People around the world are preemptively seeking medical advices on how to live a healthy lifestyle. They are looking to lower their risk of various diseases. A healthcare chatbot can play a significant role to monitor a person's health status. A healthcare chatbot is a computer program designed to simulate conversation with human users as a virtual medical assistant. Our study shows, there are some healthcare chatbots available in English and other languages but not in Bangla. In this paper, we have demonstrated a machine learning-based closed domain Bangla healthcare chatbot 'Disha (Direction)' which can converse in Bangla with the user with the help of its knowledge base and through learning from interactions with the user. It helps a user to diagnose potential diseases based on inputted symptoms, keep track of a user's health status, and alert a user from potential health hazards. This paper explores the use of six supervised machine learning approaches and showed significant efficiency.**

Keywords—Bangla, Healthcare, Chatbot, Machine Learning

I. Introduction

Over 95% of the world's population has health problems experiencing more than five ailments, according to a major new analysis from the Global Burden of Disease Study (GBD) 2013, published in The Lancet [1]. Nowadays people are prone to several common but dangerous diseases due to lack of awareness and early prevention. A healthcare chatbot installed in any digital device such as a smartphone or laptop can play an important role in this situation. A basic healthcare chatbot can identify potential disease by user-inputted symptoms. Moreover, it may assist a user by providing information on diseases. Bangla (endonym Bengali) is the 6th most widely spoken language around the world with approximately 228 million total speakers worldwide [2]. However, our study shows there is no Bangla healthcare chatbot available although there are many healthcare chatbots available in English and other western languages. One of the major reasons behind this is that according to our study, there is no Bangla dataset available that can be used to diagnose diseases. Therefore, we have designed and implemented a machine learning based Bangla healthcare chatbot. The chatbot can diagnose disease, provide information about diseases, keep track of user's health status, alert the user on potential health hazards, remind the user to take the prescribed medicines and other basic actions entirely in Bangla.

As we have mentioned earlier there is no Bangla healthcare chatbot available, however, our study shows, there are two minor works that have been done related to Bangla chatbot. In 2017, Orin, in her unpublished bachelor thesis proposed a Bangla chatbot named Golpo that can converse basic conversations in Bangla [3]. Then, in the year of 2018, Anirudha et. al. proposed a focused domain contextual AI chatbot framework for resource-poor languages which includes Bangla language also [4]. However, both works are very inadequate and to have elementary conversations only. On the other hand, our study shows there are some significant research works have been done in English healthcare chatbots in recent years. In 2017, Rarhi et al. proposed an automated medical chatbot developed by AIML (Artificial Intelligence Mark-up Language) [5]. The chatbot uses traditional approaches such as Keyword Extraction (KE) to detect possible medical problems. In the same year, Kowatsch proposed a text-based healthcare chatbot especially designed for childhood obesity problems [6]. Rashmi and Neeta projected a medical chatbot in 2018 [7]. The chatbot uses Google API to convert STT and TTS conversation and uses word order similarity between sentences to understand the user command. Oyebode and Orji proposed a medical chatbot named Likita that helps in diagnosing common ailments and improve healthcare delivery in Africa [8]. Finally, in 2019, Fadhil and Gianluca have demonstrated analysis and survey on User Experience (UX) design principles and Conversational User Interfaces (CUIs) for healthcare chatbots [9].

In this paper, we have demonstrated a comprehensive text-based Bangla healthcare chatbot. We have named the system "Disha" which is a Bangla word that means "Direction" in English. The chatbot directs and assists the user to detect diseases and maintain a healthy lifestyle. The chatbot can converse basic conversations just like other regular chatbots. However, we have focused especially on the medical domain. The chatbot is based on huge datasets containing sufficient features to classify almost all the common diseases described in the proposed method section. We have applied six different machine learning algorithms to classify diseases. They are Decision Tree (DT) [10], Random Forest (RF) [11], Multinomial Naive Bayes (MNB) [12], Support Vector Machine (SVM) [13], AdaBoost [14], and K Nearest Neighbor (KNN) [15]. All these algorithms are supervised machine learning algorithms and can be used in both classification and regression. We have used TF-IDF [16] for vectorization of the Bangla text and Cosine Similarity Measure [17] to generate the similarity between texts. All the algorithms have shown significant accuracy and the SVM shows the highest accuracy with 98.39 per cent. Therefore, we have used SVM as our core classifier of the system.

The paper is organized as follows: In section II, we describe the proposed system and step by step demonstration of the system diagram with proper examples and algorithm. The paper demonstrates experimental results and performance analysis in section III. Section IV demonstrates the performance analysis while section V concludes the paper with limitations of our system and future work.

II. Proposed Method

The proposed system is text-based and therefore, the system takes written text commands as input. At the very beginning, the system collects basic information of a new user such as name, age, and blood group for future references. The chatbot interactively asks the following questions to collect the user's basic information.

a. আপনার নাম বলুন? (What is your name?)
b. আপনার বয়স বলুন? (What is your age?)
c. আপনার রক্তের গ্রুপ বলুন? (What is your blood group?)

One of the most challenging tasks of the chatbot is to handle natural language inputs from the user. A user can provide information in any structure and any organization of the sentence. A different user can provide inputs in different ways even the same user can provide different inputs in a different time. Since the user can answer his name with different ways such as:

a. আমার নাম শাকিব। (My name is Shakib)
b. শাকিব। (Shakib)
c. শাকিব আমার নাম। (Shakib is my name)

Thus, we have used a Named Entity Recognition (NER) algorithm [18][19] to extract the name of the user. A modified version of the algorithm is used to extract age and blood group of a user. On the other hand, if the user information is already available to the chatbot, it greets the user with name such as "প্রিয় শাকিব, আপনাকে কিভাবে সাহায্য করতে পারি? (Dear Shakib, how can I help you?)". At this point, our chatbot waits for a command. In our system, there are mainly two types of command s,

1. Disease Classification command
2. General Commands

The disease classification command is the core task of the system. At this point, to trigger the disease classification mood a user can ask for help in any natural Bangla language form. For example,

a. আমি শারীরিক সমস্যায় আছি (I am in physical problem)
b. আমি সমস্যায় আছি (I am in problem)
c. আমার শরীর ভালো নেই (I am not feeling good)
d. আমার রোগ নির্ণয় করুন (Diagnose my disease)

The system will then ask the user to put symptoms. Our system takes input one symptom at a time until the symptom count reaches a threshold. After reaching to the threshold, the system asks user "আরও নতুন উপসর্গ থাকলে লিখুন, না থাকলে 'না' লিখুন। (Write more symptoms, write 'No', if there are no more symptoms.)". The threshold value is 3 by default, less than 3 may generate an inaccurate result. On the other hand, in a chatbot, we cannot expect a user to put too many symptoms as well. In order to classify disease, a user must put at least 3 symptoms. The system encourages users to put symptoms as many as possible. More symptoms mean more accuracy of the actual disease prediction. When a user puts a symptom our system immediately detects the feature in our training set using Cosine similarity Measure and TF-IDF for vectorization between user-inputted symptoms and the features in our train set. After taking all the symptoms, our system classifies the symptoms using the SVM classifier since it gives the highest accuracy in our system based on the train and test set. Then, the system generates appropriate suggestion based on the classification result.

On the other hand, in general commands, we extract keywords and detect whether the command requires just medical information such as:

a. হৃদরোগের লক্ষণগুলি কী কী?(What are the symptoms of heart disease?)
b. হাঁপানি রোগ কি? (What is asthma?)
c. উচ্চ রক্তচাপ কেন হয়? (Causes of Hypertension?)

We have a rich knowledge base which contains an enormous collection of disease information. We have collected this information from several sources mainly from internet sources. Any command that requires information about diseases, the system fetches the information from our knowledge base based on cosine similarity measure. Some commands require a save operation, for example,

a. আজ আমার রক্তচাপ ৮০-১৩০, সংরক্ষণ/সেভ করুন (Today my blood pressure (BP) is 80-130, save it)
b. আমার বয়স পরিবর্তন করুন (Edit my age)
c. আমাকে প্রতিদিন রাতে ঔষধ খাওয়ার কথা মনে করিয়ে দিন। (Remind me to take medication every night)

We save this information in a separate customized knowledge base of the user. For some specific commands, such as the "Remind me to take medication every night", we set a timestamp and prompt an alert on the specific time. The default times for morning, noon and night are at 08:00, 13:00 and 21:00 respectively. However, a user can change these default times using commands. The system keeps track of the user's BP history and alerts if it sees any abnormality. It alerts the user instantly when the user says its BP range is not between 80 and 120. The top number is called systolic pressure and the bottom number is called diastolic pressure. The normal range of systolic pressure is between 90 and 120 and the for diastolic pressure is between 60 and 80 [20]. The system diagram of the proposed system is shown in Figure 1.

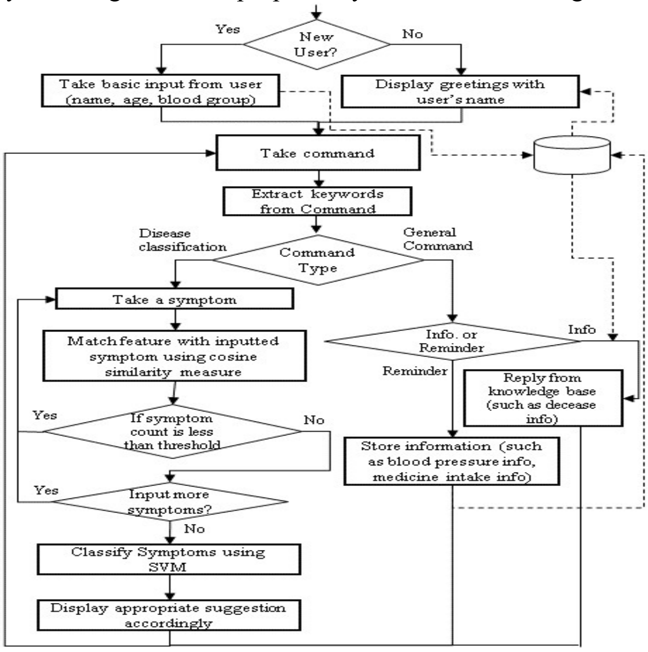

Fig. 1. System diagram of the proposed system

The creation of training and test datasets are among the most important tasks of the system. In order to classify real-life symptoms for real-life diseases, we required a reliable source of information. We have used the most used disease classification train and independent test datasets can be found in [21]. According to the authors, they have built the datasets by collecting information directly from several doctors and hospitals. The summary of the source datasets is given in Table 1.

Table 1. Summary of the source train and test datasets

	Train	Test
Size of samples:	4920	41
Unique Disease:	41	41

We have reduced the complexity of the datasets without changing originality to handle the natural language inputs from the users. As we have said earlier, users usually put fewer symptoms and in different variations. On the other hand, the datasets have a large feature count with '0's indicates negatives and '1's indicates positive features. For example, we have the following abstract data for the 'Heart Attack'

chest pain	vomiting	itching	breathlessness	fever	
1	1	0	1	0	Heart Attack

In step 1, we have removed all the negative features and kept only the positive ones. For example,

chest pain	vomiting	breathlessness	
1	1	1	Heart Attack

Then in step 2, we have placed the feature name in the places of positive features.

chest pain	vomiting	breathlessness	Heart Attack

Finally, in step 3, we translated the entire row to Bangla Unicode text using the Google Translation API [22]. We have proofread the entire datasets to ensure there is no spelling mistake remains.

বুক ব্যথা	বমি	ঊর্ধ্বশ্বাস	হৃদরোগ /হার্ট অ্যাটাক

In this way, we have built our customized Bangla datasets of the disease classification. We then combined the source training and test datasets and split it into two parts for new training and test data. The new training dataset has 3720 instances (75%) and the test dataset has 1241 instances (25%). We have another file that maps the name of the disease with its specialty and type of doctor. Table 2 shows examples of the mapping between prognosis, specialty, and type of doctor.

Table 2. example of mapping between prognosis and specialty

prognosis	specialty	doctor type
হৃদরোগ/হার্ট অ্যাটাক (heart attack)	হৃদ্বিজ্ঞান (cardiology)	হৃদরোগ বিশেষজ্ঞ (cardiologist)
ছত্রাকের সংক্রমণ (fungal infection)	ত্বক-বিজ্ঞান (dermatology)	ত্বক-বিশেষজ্ঞ (dermatologist)

After the classification of the disease, based on the mapping the system displays appropriate suggestions with the type of doctor the user should consult. The proposed system is explained in Algorithm 1.

Algorithm 1 Algorithm of proposed system

1: **if** new user **then**
2: take basic input from user(name, age and blood group)
3: else
4: display greetings with user's name
5: end if
6: *com* ← Take command input
7: *com* ← Filter *com* by removing additional white spaces and punctuation(such as '।'(full stop)
8: *key* ← extract keywords from *com*
9: *typ* ← detect command type from *key*
10: count ← 0
11: **if** Command *typ* is Decease Classification **then**
12: *sym* ← take a symptom input
13: count ← count + 1
14: *fetsym* ← match feature with *sym* using cosine similarity measure
15: **if** *count* is less than the threshold **then**
16: **go to** 12
17: **else**
18: **if** *count* user wants to put more symptoms **then**
19: **go to** 12
20: **else**
21: classify symptoms using SVM based on *fetsym*
22: display appropriate suggestion accordingly
23: **end if**
24: end if
25: else
26: **if** user needs information **then**
27: reply from database accordingly (such as personal, disease information etc.)
28: **else**
29: Store information in customized database (such as reminder, medicine intake etc.)
30: end if
31: end if

III. Experimental Results and Analysis

In this section, we have demonstrated how our system acts with some important commands and a brief

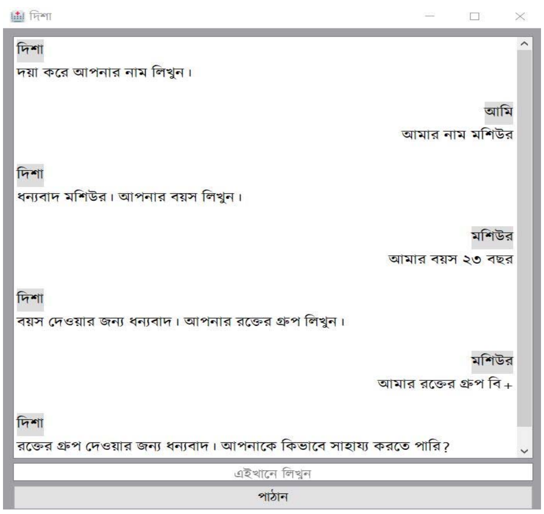

Fig. 2. Basic information collection from user

analysis of the outcome. As we have mentioned earlier, at the very beginning of the operation we take some basic information about the user.

Figure 2 shows the interaction between our system and a user to fetch basic information about the user. Our system saves all this information in a customized database. Figure 3 shows how the system saves other basic medical information of the user such as blood pressure for future analysis and warning if any abnormality observes in the given data. In addition, the figure also shows how a user set a reminder on medicine intake and Figure 4 shows the reminder prompt on time.

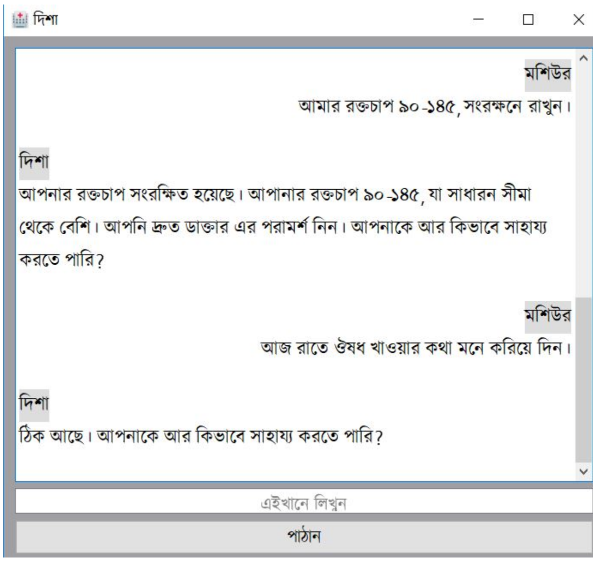

Fig. 3. Saving blood pressure and setting reminder on medicine intake

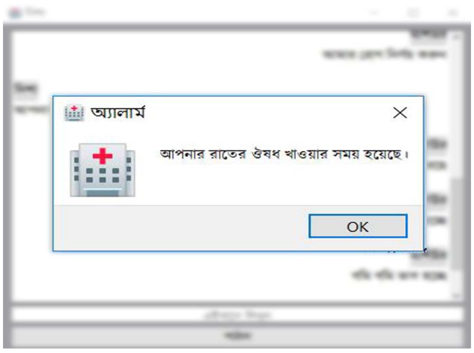

Fig. 4. Medicine intake reminder prompt

A user can always view and modify personal and medical information. A user can also add an emergency mobile number to contact in an emergency situation. A simple help command triggers our system to send an emergency predefined SMS and real-time call to that mobile number. However, this feature depends on several conditions such as device access permission, network, and balance. Figure 5 shows commands to show personal information and adding an emergency number of the user.

One of the core functions of our system is to provide medical information. As we have mentioned in our proposed method section, we have a database of this information. Our system can provide information on the most common diseases. Figure 6 shows a conversation between our system and the user about a disease called Jaundice.

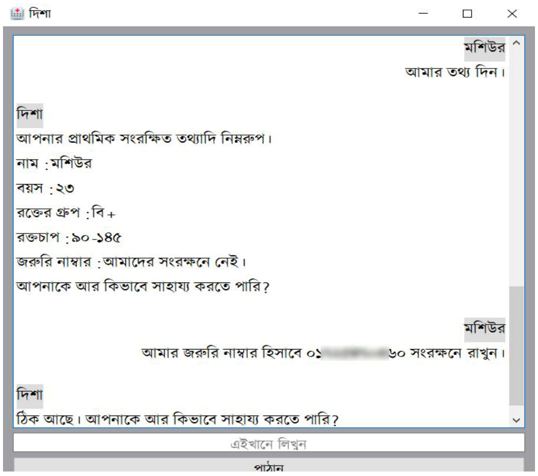

Fig. 5. Viewing personal information and adding emergency number

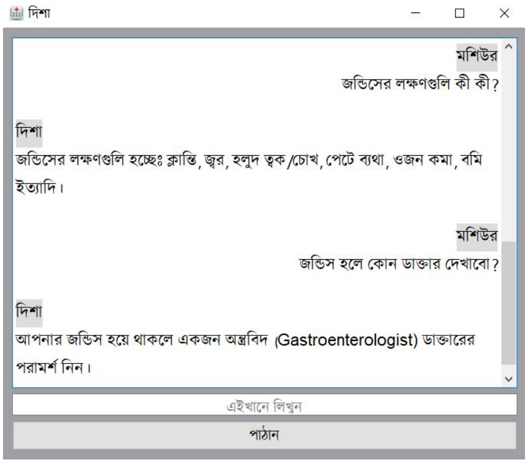

Fig. 6. Providing information about diseases

Finally, Figure 7 shows the disease classification from user inputted symptoms. This triggered with the command "আমার রোগ নির্ণয় করুন (Diagnose my disease)".

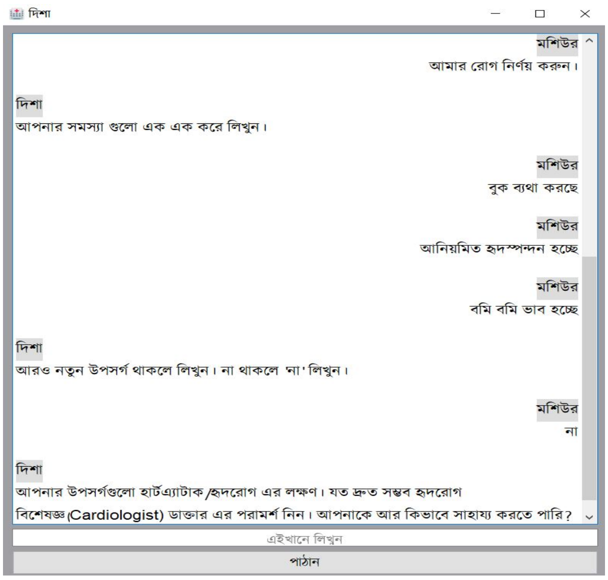

Fig. 7. Diagnose potential disease from inputted symptoms

Although a user can put symptoms as much as a user wants, the user here has put three major symptoms of the disease. After analyzing all the inputted symptoms, it diagnosed the potential disease and suggested which specialty of doctor to consult about the disease.

IV. Performance Analysis

In this section, we have analyzed the overall performance of our system, analyzed the results of different algorithms and parts of the system. All the experiments of the system were done in a high-performance machine with 32GB RAM, Intel Xeon processor, and 6GB Graphics card. The core language used in the experiment is the Python 3.6 with the Scikit-learn library [23]. Since we have our own independent Bangla disease classification dataset, we have used both the independent test set and K-fold Cross Validation (CV) [24] to validate the classifier. In K-fold CV, the total dataset randomly partitioned into K equal sized subsamples. Only one subsample works as test data and the remaining $K - 1$ subsamples are used as training data. The cross-validation process then repeated K times. We have used 10-fold CV in our experiment. The summary of the average 10-Fold score of our dataset is given in Table 3.

Table 3. Summary of average K-Fold score

Classifier	Average K-Fold Score
Decision Tree	0.9663
Random Forest	0.9753
Multinomial NB	0.9625
SVM	0.9844
AdaBoost	0.9687
kNN	0.9713

However, since we have our own independent Bangla test set, we have tested the model using that dataset too. We have used three performance evaluation metrics i.e. Accuracy, Precision (weighted avg.), and F1-Score (weighted avg.) in our experiment. Precision and F1-Score of each class are weighted by the number of samples from that class in our test set. The metrics can be defined as follows:

$$Accuracy = (TN + TP)/(TN+TP+FN+FP) \quad (1)$$

$$Precision = TP/(TP + FP) \quad (2)$$

$$Recall = TP/(TP + FN) \quad (3)$$

$$F1\ Score=2*((Precision*Recall)/(Precision+Recall)) \quad (4)$$

where:

TP = True Positive
TN = True Negative
FP = False Positive
FN = False Negative

Figure 9 shows the test accuracy result of the experiment graphically and Table 4 shows the summary of the result.

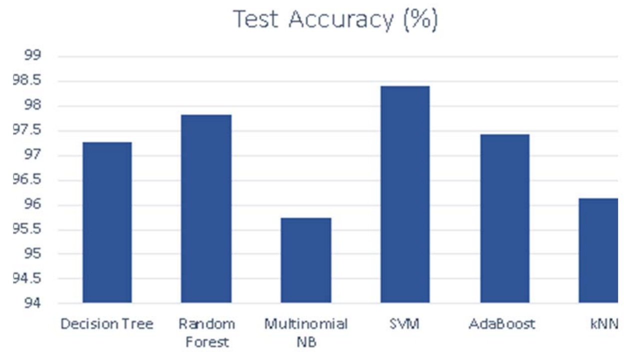

Fig. 8. Graphical representation of test accuracy

Table 4. Summary of experimental results

Classifier	Accuracy	Precision (weighted avg.)	F1-Score (weighted avg.)
Decision Tree	97.26	98.54	97.66
Random Forest	97.82	98.97	98.19
Multinomial NB	95.73	96.87	96.02
SVM	98.39	98.78	98.49
AdaBoost	97.42	98.89	97.88
kNN	96.13	97.27	96.41

Although all the classifiers have shown great performances with high accuracy, Figure 9 demonstrates in our experiment SVM shows the best performance with the highest accuracy of 98.39 per cent. In addition to that, Random Forest shows the second-best performance with the accuracy rate of 97.82 per cent where Multinomial NB shows the worst performance in this case with 95.73 per cent of accuracy rate. This clearly indicates SVM should be the core classifier of the system.

V. Conclusion and Future Work

This paper demonstrates the implementation process of a comprehensive machine learning based Bangla healthcare chatbot. We have demonstrated the steps to build the healthcare chatbot including the customized Bangla datasets. We have experimented with six different supervised machine learning algorithms where SVM shows the best performance.

Although we have reached our goal, the system has some minor limitations. The main limitation according to our study is the incapability of taking sufficient symptoms from the user. Users generally tend to put very few symptoms of their problems. This may lead to an incorrect decision. The authors are working on different approaches to solving this issue. Moreover, in our current experiment, we have used only generic machine learning algorithms. Our next strategy is to use Deep Learning (DL) approaches to make the system more robust. In addition, we are planning to do a user experience survey according to HCI discipline based on a robust online and offline questionnaire to improve the quality of the work. Moreover, we are planning to upload all the datasets and codes of our experiments in our GitHub repository for other researchers.

REFERENCES

[1] Murray, Christopher JL (2015/08/22). Global, regional, and national incidence, prevalence, and years lived with disability for 301 acute and chronic diseases and injuries in 188 countries, 1990–2013: a systematic analysis for the Global Burden of Disease Study 2013. The Lancet, 386, 743-800. doi: 10.1016/S0140-6736(15)60692-4

[2] Ethnologue. (2019). Summary by language size. [online] Available at: https://www.ethnologue.com/statistics/size .

[3] Orin , T. D. (2017). Implementation of a Bangla Chatbot (Unpublished Bachelor Thesis). BRAC University, Dhaka, Bangladesh.

[4] Paul, Anirudha & Latif, Asiful & Amin Adnan, Foysal & Rahman, Mohammad. (2018). Focused domain contextual AI chatbot framework for resource poor languages. Journal of Information and Telecommunication. 1-22. 10.1080/24751839.2018.1558378.

[5] Rarhi, Krishnendu & Bhattacharya, Abhishek & Mishra, Abhishek & Mandal, Krishnasis. "Automated Medical Chatbot" . SSRN Electronic Journal (2017). Ddoi: 10.2139/ssrn.3090881.

[6] Kowatsch, Tobias & Nißen, Marcia & Shih, Chen-Hsuan & Rüegger, Dominik & Volland, Dirk & Filler, Andreas & Künzler, Florian & Barata, Filipe & Haug, Severin & Büchter, Dirk & Brogle, Björn & Heldt, Katrin & Gindrat, Gindrat & Farpour-Lambert, Nathalie & l'Allemand, Dagmar. (2017). Text-based Healthcare Chatbots Supporting Patient and Health Professional Teams: Preliminary Results of a Randomized Controlled Trial on Childhood Obesity.

[7] Mrs. Rashmi Dharwadkar, Dr.Mrs. Neeta A. Deshpande "A Medical ChatBot". International Journal of Computer Trends and Technology (IJCTT) V60(1):41-45 June 2018. ISSN:2231-2803. www.ijcttjournal.org. Published by Seventh Sense Research Group.

[8] Oyebode, Oladapo & Orji, Rita. "Likita: A Medical Chatbot To Improve HealthCare Delivery In Africa". HCI Across Borders (HCIxB) (2018) . doi: 10.13140/RG.2.2.29159.09128.

[9] Fadhil, Ahmed and Gianluca Schiavo. "Designing for Health Chatbots." *ArXiv*abs/1902.09022 (2019): n. pag.

[10] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. IEEE transactions on systems, man, and cybernetics, 21(3):660–674, 1991.

[11] Tin Kam Ho. Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on, volume 1, pages 278–282. IEEE, 1995.

[12] Xu, Shuo & Li, Yan & Zheng, Wang. (2017). Bayesian Multinomial Naïve Bayes Classifier to Text Classification. 347-352. 10.1007/978-981-10-5041-1_57.

[13] Evgeniou, Theodoros & Pontil, Massimiliano. (2001). Support Vector Machines: Theory and Applications. 2049. 249-257. 10.1007/3-540-44673-7_12.

[14] Robert E Schapire. The boosting approach to machine learning: An overview. In Nonlinear estimation and classification, pages 149–171. Springer, 2003.

[15] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of machine learning. MIT press, 2012.

[16] Qaiser, Shahzad & Ali, Ramsha. (2018). Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents. International Journal of Computer Applications. 181. 10.5120/ijca2018917395.

[17] Rahutomo, Faisal & Kitasuka, Teruaki & Aritsugi, Masayoshi. "Semantic Cosine Similarity". The 7th International Student Conference on Advanced Science and Technology ICAST (2012).

[18] Chowdhury, Shammur & Alam, Firoj & Khan, Naira. (2018). Towards Bangla Named Entity Recognition. 10.1109/ICCITECHN.2018.8631931.

[19] Dozier, Christopher & Kondadadi, Ravikumar & Light, Marc & Vachher, Arun & Veeramachaneni, Sriharsha & Wudali, Ramdev. (2010). Named Entity Recognition and Resolution in Legal Text. 27-43. 10.1007/978-3-642-12837-0_2.

[20] "Blood Pressure Chart." Blood Pressure : Blood Pressure Chart, www.bloodpressureuk.org/BloodPressureandyou/Thebasics/Bloodpressurechart.

[21] "Disease Prediction Using Machine Learning With GUI", https://www.kaggle.com/neelima98/disease-prediction-using-machine-learning

[22] "Cloud Translation API Documentation | Cloud Translation API | Google Cloud." Google, Google, cloud.google.com/translate/docs/.

[23] Fabian Pedregosa, Ga¨el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikitlearn: Machine learning in python. Journal of Machine Learning Research, 12(Oct):2825–2830, 2011

[24] Jung, Yoonsuh & Hu, Jianhua. (2015). A K-fold Averaging Cross-validation Procedure. Journal of Nonparametric Statistics. 27. 1-13. 10.1080/10485252.2015.1010532.