International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 4 Issue 5, May 2015

DIGITAL MEDICAL CARE WITH ITERATIVE AND PROBABILISTIC DISEASE PREDICTION

Aanchal Oswal, VachanaShetty, Mustafa Badshah, RohitPitre

***Abstract*—Prediction is a rigorous and often quantitative-statement, forecasting what will happen under specific conditions. Similarly disease prediction is a prognosis of the possible outcomes of a disease, which one is likely to suffer from. In this paper, we focus on searching algorithms assembled for generalized disease prediction, based on symptoms and recent trends to conclude the efficiency and effectiveness of the algorithms, and also to increase their scalability.**

Index Terms—**Symptoms, Probability, Prediction, Recent Trends**

I. INTRODUCTION

This paper exemplifies how various methods can be used for predicting diseases. The dissimilarity between disease predicting methods and how methods such as linear search, neural networks and data mining are used in various ways is explained, and how they execute, depending on the size and type of data. It also shows how the recent trend in diseases affects the result of prediction of diseases.

II. OBJECTIVE & SCOPE

The objective of this proposed system is to develop an Android-based medical care system to deal with disease prediction, appointments and reminders, to ease the process of taking appointments, provide an easy interface to users and doctors alike, and store information securely in a data base. [1]

Disease diagnosis is limited to generalized diseases; complex diseases wouldn't be covered in this system. Medical records are confined to only text based records. Reports or laboratory results of different forms wouldn't be supported by this system. This application is supported by Android. [2]

Manuscript received May, 2015.

***First Author Aanchal Oswal**, Computer Science, KJ College of Engineering and Management Research, Pune, India, (email: aanchaloswal@gmail.com).*
***Second Author VachanaShetty**, Computer Science, KJ College of Engineering and Management Research, Pune, India, (email: vachanakshetty23@gmail.com).*
***Third Author Mustafa Badshah**, Computer Science, KJ College of Engineering and Management Research, Pune, India, (email: mustafab325@gmail.com).*
***Fourth Author RohitPitre**, Computer Science, KJ College of Engineering and Management Research, Pune, India, (email: rohitpitre3@gmail.com).*

III. MOTIVATION

There is a high perception value that almost 79% people believe that an E-Health Care Facility system could provide major benefits in managing health. This shows that people are ready to accept an E-Health Care Facility which would help them cater to their basic medical needs.

The study also shows that there are a large number of people (around 47%) who expressed their interest in using an online health facility. This gives us a high motivation to implement a system which caters basic medical needs of patients.

But only a few i.e. just 2.7% of online health care facility systems are being used as of now.

IV. EXISTING SYSTEM WITH LIMITATIONS

Existing systems are confined to a single hospital and the patients which are treated there. The data sharing is limited to only the doctors of that hospital.

Maximum number of disease diagnosis algorithms uses a simple search to find the most accurate disease as per the symptoms given by the user, where it searches linearly in the given schema and gives the result as to which disease has the highest probability from with regards to what the patient might be suffering from.

One of the simplest ways of searching for the disease is by mapping them to the entered symptoms using linear search.

If user gives a set of inputs, the linear search along with neural networks algorithm would work as follows:

- Get names of symptoms from database for ids entered by user.
- Store the name of Symptoms to the Array 'signs'.
- Get name and symptoms of all the Diseases from Database.
- Store name of each disease, and its corresponding symptoms all symptoms and the probability according to symptoms entered by user in the array 'diseases'.
- Sort the array according to probability.
- Display all the diseases with their symptoms and probability as a result.

For the above algorithm it checks for k symptoms present in each disease, where the number of diseases is n. The time complexity for the same is – **nO(k)**.[3]

Working of the above system-

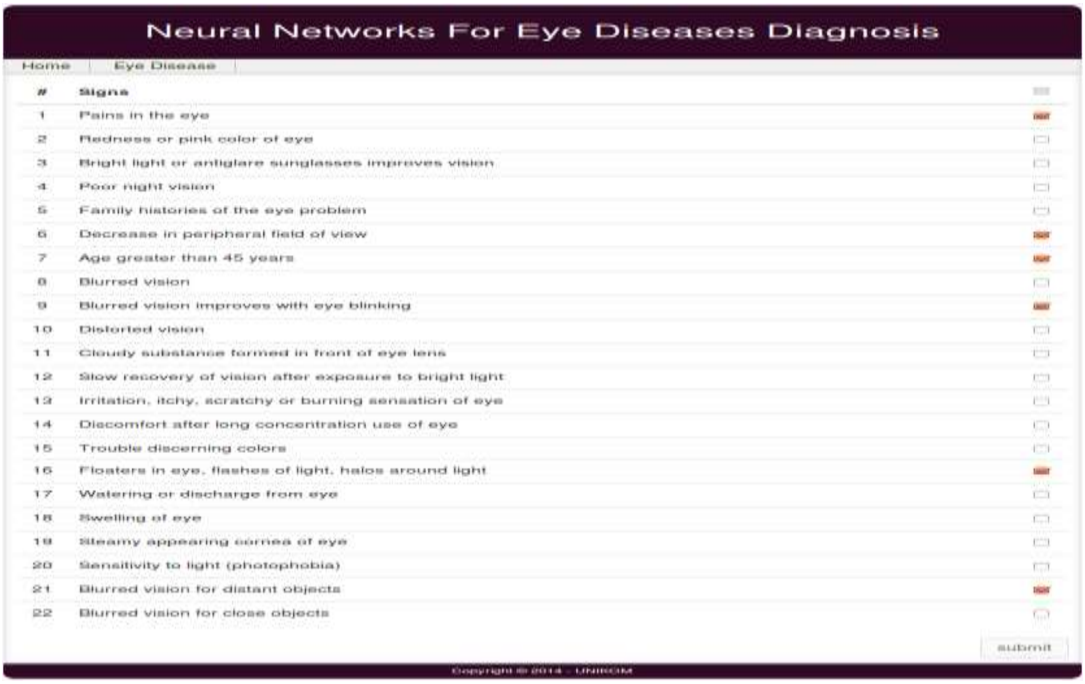

Fig. 1.- Select Symptom[3]

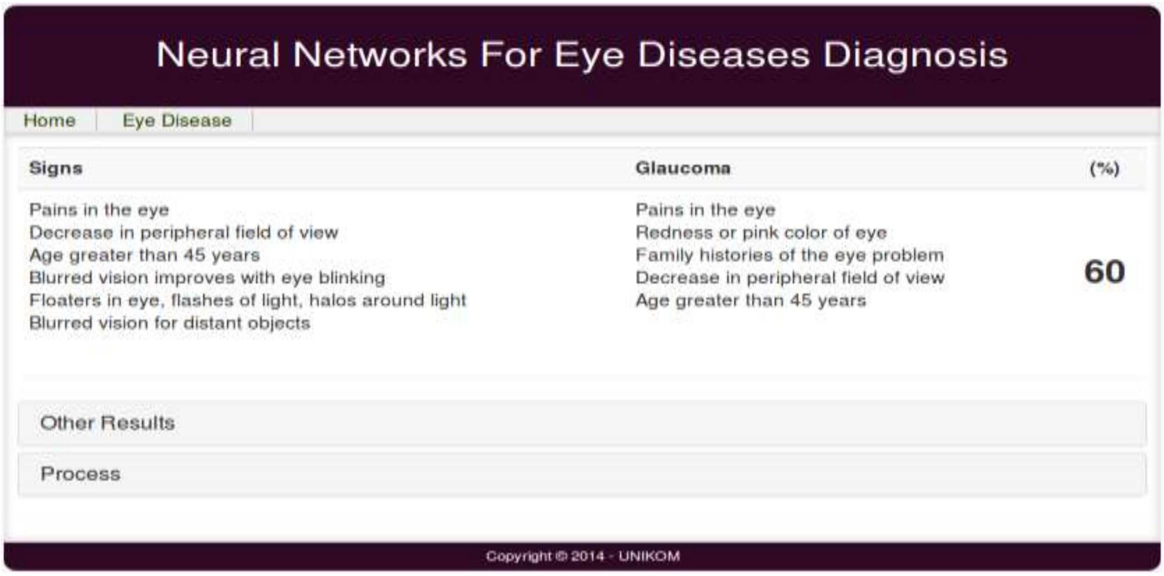

Fig. 2.- Result[3]

4], uses recent trends in case where there are multiple disease predictions because it becomes difficult to point to one disease in such cases where the symptoms are not unique. The time frame they use is of the previous 3 months, so that the prediction can be more accurate.

Disease	Probability	Differential Diagnosis	
Malaria	27.03	Overall	Past 3 Months
Influenza	18.09	Overall	Past 3 Months
Leptospirosis	17.8	Overall	Past 3 Months
Dengue fever	16.83	Overall	Past 3 Months
Gastroenteritis	3.13	Overall	Past 3 Months
showing 1 - 5 of 16		Page 1 2 3 4	

Fig. 3.- Symptoms based recent trends[4]

V. PROPOSED SYSTEM

Our system overcomes the drawback of limited data sharing as the data is centralized, secured and can be viewed by doctors of any hospital if and only if the patient is willing to share the data.

Our proposed system takes input from the data present in the predefined schema that contains information about the patients, doctors, hospitals, appointments, prescriptions, diseases and symptoms.

The schema consists for the following relations-

1. Signup
2. Patient
3. Doctor
4. Appointments
5. Prescription Alerts
6. Disease diagnosis

The schema was designed for an E-Health Care application; therefore it contains the relations for the patients, doctors, hospitals, appointments and prescriptions.

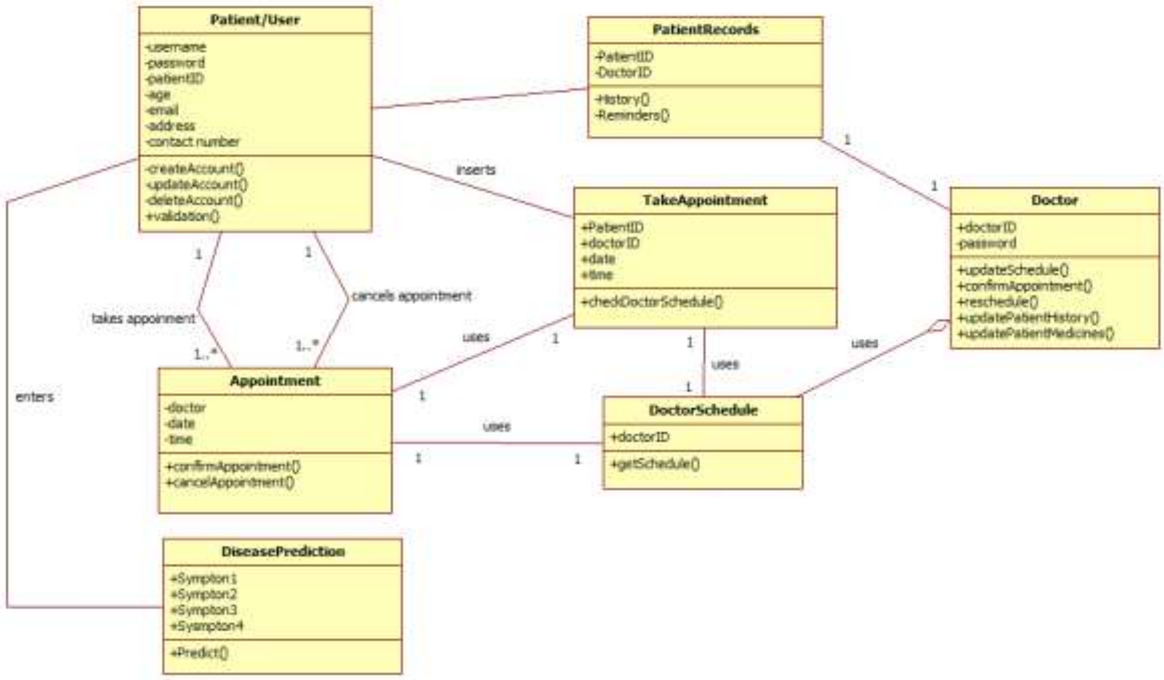

Fig. 4. - Schema

Our disease prediction is based on an iterative algorithm on a sorted database.

Disease diagnosis depends mainly on the following factor:

- **Symptoms**: A symptom is a phenomenon that is experienced by the individual affected by the disease, while a sign is a phenomenon that can be detected by someone other than the individual affected by the disease.

Following are the three main steps which explain the working of the algorithm-

1. **Symptoms**

Firstly the user has to enter the symptoms that he/she may be suffering from. Based upon the symptoms that the user gives, a calculation is performed, and the user gets a list of probable diseases.

2. **Calculate Probability**

This algorithm gives the probability for various diseases as per the symptoms entered by the user. The probability of each disease is calculated using the number of symptoms entered by the user and the total number of symptoms for the corresponding disease.

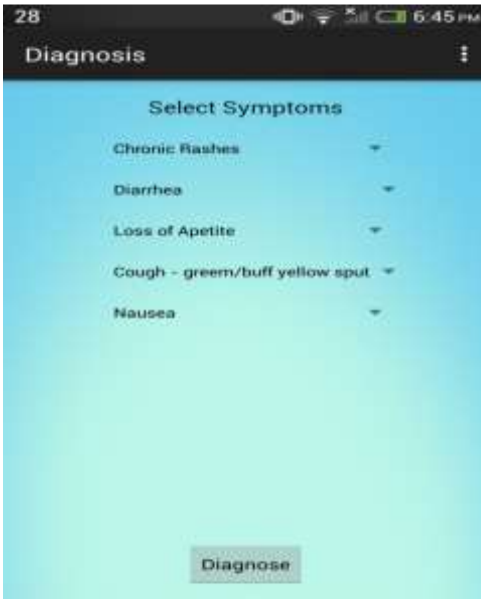

Fig. 5. - Select Symptoms

The probability for any disease can be given by the formula:

$$\mathbf{P_i = (\sum(W_{s1} + W_{s2+.....+}W_{sn})*T_{Di}) / 100}$$

Where,

P_i = Probability of disease (D_i)
W_{sn} = Weight of disease (D_i)
T_{Di} = Total number of symptoms for D_i

Fig. 6. - Probability Calculation

3. Recent trends

Our proposed symptom uses a similar technique used in [3], for calculating recent trend based on the previous medical records. Our system stores the diseases with highest probability in the recent trend relation and based on which if the user has similar symptoms, he/she will be notified.

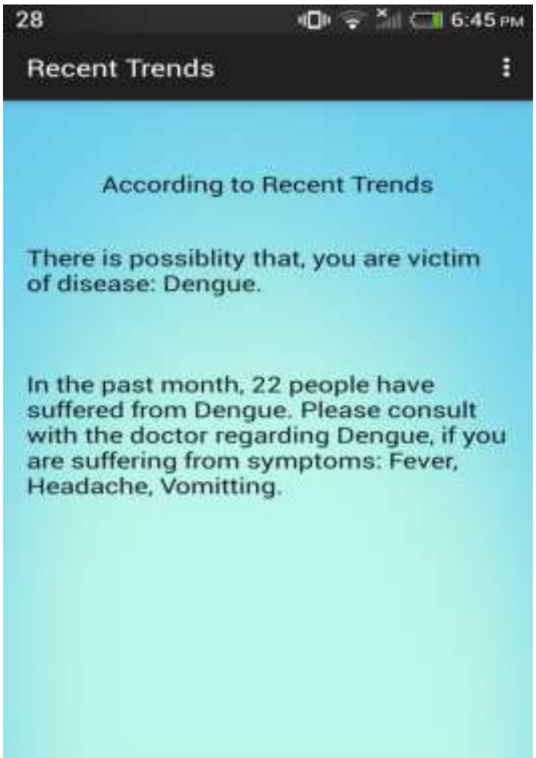

Fig. 7. - Recent Trends

VI. ALGORITHM

Input: Multiple Symptoms from the user ($S_1, S_{2,....,}S_n$)
Output: Probability of various diseases.
Steps:

1. Start
2. probability[] = 0
3. counter[] = 0
4. disease[] = 0
5. If symptoms == NULL the goto step 16
6. Map all the disease ids to D[] if Symptom (S_i) is in D_i
7. Find the max value of disease[] i.e the disease id
8. Increment counter for each D_i
   a. For each entry in D_i
      i. counter[disease[i]] = counter[disease[i]] + 1
9. Get the total disease weights for all the diseases in disease[] in d_total[]
10. for i = 0 to counter[].length
    a. If counter != 0
       i. probability[i] = (counter[i] * d_total[i])*100
       ii. probability_id[i] = i

11. Display all the probable diseases in decreasing order of their probabilities
12. Find out the disease with the highest probability.
13. Store the disease id with the current month in the recent history table.
14. For the diseases predicted check if any exists in the recent history table
    a. If yes
        i. Count the number of iterations of each disease predicted in current month.
        ii. Get the disease id with the max no of occurrences, fetch the symptoms of the recent trend's disease which are not entered by user and display the disease name and ask the user whether suffering from the remaining symptoms of the recent trend disease.
15. Error: Enter Correct Symptoms.

VII. TIME COMPLEXITY

To compare and measure the effectiveness of an algorithm, we need to measure the time complexity of the algorithm. To measure the time complexity, following points need to be taken into consideration

- Nesting of loops:
    - **Step 9**: Checking the list of all n diseases and incrementing their indexes till k number of symptoms are found.
        - **Complexity –O(kn)**
    - **Step 10**: Calculating the probability of n diseases with the corresponding m number of symptoms entered by the user.
        - **Complexity –O(mn)**
    - **Step 15**: Checking for recent trends, by searching for the occurrence of the disease with highest probability in step 10 in the list of x diseases in recent trends.
        - **Complexity–O(x)**
    - All other statements i.e. the conditional statements and other data retrieval statements, have much less complexity as compared to the above three steps. Hence they can be neglected.
    - Therefore the total time complexity – **O(kn)+O(mn)+O(x)** .

VIII. COMPARISON

ALGORITHM/ PARAMETER	**LINEAR SEARCH[2]**	**ALGORITHM**
Time complexity	nO(k)	O(kn)+O(mn)+O(x)
Scalability	Scalability can be handled, but is time consuming	Scalability can be handled, but is time consuming if the diseases increase
Time to execute	1-1.5 seconds	≈600 milliseconds
Probability Calculation	Complicated	Simple formula used
Recent Trends	Not Maintained	Maintained
Application	Specific Disease only	Multiple types of diseases
Limitations	Major diseases cannot be predicted. Eg. Cancer	Major diseases cannot be predicted. Eg. Cancer

Table 1. – Comparision

IX. FUTURE ENHANCEMENT

Every algorithm comes with certain kind of advantages and disadvantages. Some of these disadvantages can be overcome by making changes in the algorithm. Following are the limitations of our algorithm:

- **Scalability**: When a disease is added along with it symptoms, not only is the disease-symptom schema changed but also the mapping of diseases with its symptoms. As the number of diseases increase the symptom mapping becomes a tedious task.
- **Data Retrieval**: As and when the data increases, the retrieval becomes more time consuming as linearly searching through a large amount of data will take much more time as compared to smaller data.

X. CONCLUSION

This algorithm overcomes the limitation of linear search and other similar algorithms. The schema and the relations defined for this algorithm are predefined and are tested. Though there are limitations in the scalability, the algorithm is transparent and easy to implement.

REFERENCES

[1]Ming Li, Member, IEEE, Shucheng Yu, Member, IEEE, Yao Zheng, Student Member, IEEE, KuiRen, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE "Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 1, JANUARY 2013.

[2]International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 3 Issue 11, November 2014, Aanchal Oswal, VachanaShetty, Mustafa Badshah, RohitPitre, ManaliVashi, "A SURVEY ON DISEASE DIAGNOSIS ALGORITHMS".

[3]Githubrepository-https://github.com/imancha/ANN_Eye_Disease

[4] IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE, VOL. 16, NO. 6, NOVEMBER 2012 1287 Knowledge Discovery in Medical Systems Using Differential Diagnosis, LAMSTAR, and k-NN Rahul Isola, Student Member, IEEE, RebeckCarvalho, Student Member, IEEE, and Amiya Kumar Tripathy, Member, IEEE

First Author Aanchal Oswal is BE student of K.J. College of Engineering & Management Research,Pune.Her current research interests are data mining and android development.

Second Author VachanaShetty is BE student of K.J. College of Engineering & Management Research,Pune.Her current research interests are data mining and cloud computing.

Third Author Mustafa Badshah is BE student of K.J. College of Engineering & Management Research,Pune. His current research interests are data mining, image processing and web development.

Fourth Author RohitPitre is BE student of K.J. College of Engineering & Management Research,Pune. His current research interests are data mining and web development.