IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

E-Health Care with Prediction Analysis

Mohit Kunjir[1], Satish Kamble[2], Amol Swami[3], Priyanka Karad[4], Satvashila Bagal[5]

I.T. Dept. PVGCOET, India[1,2,3,4,5]

Abstract: During a disease outbreak or some natural or manmade disaster, it has been observed that several people suffer due to mismanagement and miscommunication between hospitals. This happens because most of the hospitals although are equipped with modern healthcare infrastructure, still rely on pen paper based record keeping of the patients. This not only causes wastage of resources but also complicates and increases the patient's bills. Having paper based records although helps in keeping track of an individual patient, but it gives us no knowledge about the health of a community as a whole. Our project tries to solve the above problems by automating the healthcare systems by providing a one stop close loop solution. Using healthcare automation a patient can easily be registered to a hospital and doctor can keep track of his disease and treat him better. The patient also would be able to carry his reports and prescription on the go without any hustle. Location based emergency response service can help greatly in providing medical assistance to the victims in the nick of time. By creating Electronic Medical Record of every patient we can also determine patterns in the diseases caused in the patients over a period of time and can try to trace its source. Using K-Means clustering algorithm we can also predict what disease a patient would have by taking his blood report and symptoms as input and verifying them against the historical data stored in database. The project will consist of 3 modules, viz, Patient and Hospital Data Accumulation for Automation purpose, Location based emergency services, discovering disease patterns and identifying drug and medicine usage.

Keywords: Disease Prediction, Hadoop, KNN.

1 INTRODUCTİON

The major enabler of healthcare information sharing is the Electronic Health Record (EHR) or the Electronic Medical Record (EMR), containing all the relevant patient healthcare data in a shareable form [1]. In healthcare delivery, the EHR serves integrating functions similar to that of the Bill of Materials (BOM) used in manufacturing. The penetration of EHRs into healthcare settings is low and spotty. While large health care institutions have significant investments in EHR-based computer systems, it is estimated that only a very small percentage (around 5 to 10%) of the family physicians in the India uses an EHR system in their daily practice. For a majority of health care settings (especially small clinics) paper-based records and fax-based communications is still the norm. The adoption of standards for information interchange will facilitate integration of disparate healthcare systems. However, implementations of healthcare data integration should not be simply geared to support human readability of medical reports, but should incorporate the formalism and details necessary for proper computer interpretability of healthcare information, such as those proposed in HL7's (Health Level 7) [2] clinical document architecture standard. Such measures would prevent the loss of information during data interchange that may otherwise occur due to differences in terms and codes and their semantics in the various healthcare vocabularies. Healthcare institutions would then be able to deal transparently with information obtained from external agencies as well as that generated by in-house healthcare information systems. Their applications could perform data mining of patient medical records for healthcare quality metrics, identify patients across populations for timely medical interventions, and check for compliance with preventive-service protocols. Health care systems had access to such an enormous armamentarium of diagnostic procedures, treatment options, and resources to improve the health care of individual patients and the health of populations. At the same time, health care providers find themselves faced with an ever-growing loss of autonomy. In parallel, throughout the world, the systems of health care delivery and financing are increasingly subject to resource constraints.

II. LİTERATURE SURVEY

A variety of messaging and information exchange standards like HL7 [2], DICOM [3], E1460, E1467, E1381, E1394, IEEE 1073, and others listed by HIMSS' Integrating the Healthcare Enterprise initiative and ANSI's Healthcare Information Technology Standards panel currently permits an enterprise to integrate the various health information systems and archive the data as an EHR/EMR. The figure below illustrates the classes of clinical information and some of the standards that link these classes. Independent healthcare institutions can submit orders and referrals via HL7 for healthcare services for their patients. DICOM standards enable the interchange of information between imaging systems and facilitate remote access for physicians at their clinic. With standards-based integration of information systems and authenticated remote access to reports and images, physicians can have access to the radiologist's report as well as the diagnostic images for review and patient counseling. One concern with the current standards, discussed

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

above, is that these standards deal with only syntactic issues. Due to the disparate nature of the vocabularies (medical terminologies) in various EHRs, it is also important to develop tools and techniques for semantic interoperability. In the 1970's with the increasing use of CT scanners and other image-based diagnostic devices that opened the way for large – scale deployment of software applications in the medical field and thus the need to interface machines and applications manufactured/produced by various companies was born. This translated into the need to specify a common image format for all medical imaging devices.

In 1983, ACR and NEMA delegated a committee to solve this problem and propose a standard that would allow the following:
- Data exchange concerning generated digital medical images between devices produced by different manufactures;
- The development of PACS (Picture Archiving and Communication Systems);
- The development of medical databases able to be interrogated using (geographically) distributed software applications;

In 1985 version 1.0 of the DICOM standard was published by ACR and NEMA, followed by 2 revisions in October 1986 and January 1988. Version 2.0 of the standard was issued also in 1988 and added a set of commands for displays, by introducing a new image identification scheme based on "data-elements" for a better characterization of the image parameters (NEAC, 2008). The last published version, 3.0, issued in 2000, contains a large number of changes and additions with respect to the previous ones. The DICOM standard facilitates medical imaging equipment's inter-operability by specifying:
- A set of protocols that compliant devices must withstand;
- The commands' semantics and syntax as well as the associated information format that can be transmitted using the protocol.

In a heterogeneous system, in order to integrate medical equipment's supplied by various manufactures, the use of DICOM and HL7 standards is compulsory. DICOM, as previously presented, is mainly dedicated to medical imaging, whereas HL7 **(Health Level Seven)** covers more general aspects of medical digital data processing and management. HL7 is used for transmitting data related to patient charts, files and other associated documents and audio recordings. The number "7" refers to the application layer, the 7th one from the OSI (Open Systems Interconnection) model system representation (HL7, 2008).

The HL7 standard was first published in 1987 by a group of companies manufacturing medical equipment. Version 2.0 succeeded in 1988, followed by versions 2.1, 2.2, 2.3 and 2.3.1 in 1990 to 1999. In 1994 ANSI (American National Standards Institute) officially recognized it as an industry standard. Currently, version 3.0 is on its way, with a draft already released.
The main objective of the HL7 standard is to produce a set of specifications that allows free communication and exchange of data between medical software applications in order to eliminate or reduce incompatibility among different applications. To achieve this, the following measures have been proposed:
- The standard must support information exchange between systems implemented in a large variety of development environments (technical) and communication environments. Its implementation must be possible in all the major existing programming languages.
- Immediate, single transaction, transfer must be available in the same time as file sharing/transfer based on multiple-transactions;
- The highest degree of standardization must be obtained when compared to the most often encountered cases of elements formatting; the standard must comply with the specific necessities of each medical field. Accordingly, the standard comprises situation specific tables, definitions and segments that can be customized (Z-segments)
- The standard must cope with variations suffered in time due to inherent technical progress and evolution;
- The standard evolution must be based on the experience already gained and on already existing and well known industrial protocols. Favours given to certain producers or companies must be avoided by all means.
- HL7 makes no presumptions related to the architecture of the medical informatics system, and does not try to resolve the architectural differences present in medical informatics systems. Due to this reason, HL7 cannot have a „plug and play" interface.
- A first interest of the HL7 workgroup was to use the standard as soon as possible.

Once published, HL7 was voted and recognized as a standard by the American National Standards Institute (ANSI) and the Accredited Standards Organization (ASO).
- Currently, the cooperation with other standardizing organization from the medical field (ACR/NEMA DICOM, ASC X12, ASTM, IEEE/MEDIX, NCPDP, etc.) has become a priority for HL7 and focus on a better development of medical informational systems has contributed to the group's joining to the ANSI HISPP (Health Information Systems Planning

Panel) process, ever since its debut, in 1992. The two standards, DICOM and HL7, form the basis of the informational integration of software based medical processes. In November 1998, Healthcare Information and Management Systems Society (HIMSS) and Radiological Society of North America (RSNA) founded the Integrating the Healthcare Enterprise (IHE) forum, with the declared goal of helping the integration of software application from various medical fields and domains. Its main objective is to ensure that in the course of the medical act all the information necessary in the decision making and taking processes are accurate and available in time for the medical specialist. Its purpose is not to define new standards, but to promote the use of the existing ones, namely DICOM and HL7. Currently the main focus is on radiology. The DICOM and HL7 standards provide the necessary means and technology for developing software applications, while IHE supervises their adoption into real-life medical world. IHE provides support for the medical software applications users by ensuring a better access to information and eliminating, as much as possible, confusions or misunderstanding when acquiring such applications. From the medical software application development point of view, the IHE specifications facilitate fast and safe releases of new products as well as simple mechanisms for implementing interfacing options with other, already existing ones. Therefore, in our project we try to create a proper standard on top of HL7 & DICOM which will enable us to store medical data in a shareable format without any ambiguity.

III. PROPOSED WORK

The system we propose primarily comprises of the three modules viz., Passive Data Accumulation module, Disease and Epidemic Prediction Module and Real Time Medical Emergency Service module.

Module 1: Passive data Accumulation

In the first module, i.e. Passive Data Accumulation Module, we collect two types of data from user. First is the patient's Demographic information (Personal Information). Second type of data is patient's medical history. Demographic Information can be fetched from Adhaar API or can be manually entered and edited by the patient any time. However, only a registered doctor is allowed to enter and edit a patient's medical history. The patient can only view his medical data. This data is stored in the database according to the recommended HL7 standard proposed to store clinical data of patient. This data can also be shared between hospitals and is also used for data analysis.

Module 2: Disease and Epidemic Prediction

The second module is data analysis module. Here we perform two types of analysis viz., Disease prediction and Finding disease spread pattern. The Disease Prediction sub-module uses KNN algorithm to predict what disease the user might have depending upon the blood report and his symptoms. The second sub module tracks the disease spread pattern using Map Reduce concept in a given geographical area and informs the hospitals in those areas about the same. Disease spread pattern shows how people living in a geographical area are getting affected by a particular disease. Currently, our KNN algorithm can predict a particular disease from a set of 4 fever related diseases, namely, Dengue, Jaundice, UTI and Typhoid. The dataset we obtained contained blood reports of the affected patients along with the symptoms. KNN model uses Euclidian distance as metric to find the distance between the dimensions of individual training and testing records.

Algorithm:

In this implementation, KNN is performed according to the below two phases:

1. Find K nearest neighbors: For a specific test record, an array with size K is created to keep the K nearest neighbors, and the way to achieve this goal is to go through the whole training set and update the array if there is a training record whose distance is smaller than the largest distance in the original array. At the end of this process, the K nearest neighbors are found. The below Java codes demonstrate how it is implemented:

```
// Find K nearest neighbors of testRecord within trainingSet
static TrainRecord[] findKNearestNeighbors(TrainRecord[] trainingSe,
TestRecord    testRecord,int K, Metric metric) throws Exception{
    int NumOfTrainingSet = trainingSet.length;
    if(K > NumOfTrainingSet){
      throw new Exception("K is larger than the length of training
         set!");
    }

    //Create an array with size K to store K nearest neighbors
    TrainRecord[] neighbors = new TrainRecord[K];
```

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

```
      //initialization, put the first K trainRecords into the above array
      int index;
      for(index = 0; index < K; index++){
        //calculate the distance between TrainingRecord and TestRecord
        trainingSet[index].distance =
           metric.getDistance(trainingSet[index], testRecord);
        neighbors[index] = trainingSet[index];
      }

 //go through the remaining records in the trainingSet to find K
 //nearest neighbors
 for(index = K; index < NumOfTrainingSet; index ++){
   //calculate the distance between TrainingRecord and TestRecord
    trainingSet[index].distance =
        metric.getDistance(trainingSet[index], testRecord);


  //get the index of the neighbor with the largest distance to
  //testRecord in array neighbors

   int maxIndex = 0;
   for(int i = 1; i < K; i ++){
      if(neighbors[i].distance > neighbors[maxIndex].distance)
         maxIndex = i;
   }

   //add the current trainingSet[index] into neighbors when its
   //distance is smaller than the distance of neighbors[maxIndex]
   //By this way, the array keeps the K nearest training records
   if(neighbors[maxIndex].distance > trainingSet[index].distance)
      neighbors[maxIndex] = trainingSet[index];
   }

   return neighbors;
```

2. Classify according to weights: After the K nearest neighbors are found, a Hash Map is created that adds the original weight with 1 / distance which is put into the map. If the Hash Map happens to meet a new label, <label, 1 / distance> is directly added into the map. Otherwise, an updated version which program will go through the whole Hash Map and find the label associated with the largest weight. Below are the Java codes which demonstrate the implementation:

```
// classify the class label by using neighbors
static int classify(TrainRecord[] neighbors){
      //construct a HashMap to store <classLabel, weight>
      HashMap<Integer, Double> map = new HashMap<Integer, Double>();
      int num = neighbors.length;

      for(int index = 0;index < num; index ++){
            TrainRecord temp = neighbors[index];
            int key = temp.classLabel;

            //if this classLabel does not exist in the HashMap, put
            //<key, 1/(temp.distance)> into the HashMap
            if(!map.containsKey(key))
                  map.put(key, 1 / temp.distance);

            //else, update the HashMap by adding the weight associating
            //with that key
            else{
                  double value = map.get(key);
                  value += 1 / temp.distance;
                  map.put(key, value);
            }
      }


  //Find the most likely label
  double maxSimilarity = 0;
  int returnLabel = -1;
  Set<Integer> labelSet = map.keySet();
  Iterator<Integer> it = labelSet.iterator();

  //go through the HashMap by using keys
  //and find the key with the highest weight
  //the key with the highest weight is the returned class label
```

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

```
while(it.hasNext()){
      int label = it.next();
      double value = map.get(label);
      if(value > maxSimilarity){
            maxSimilarity = value;
            returnLabel = label;
      }
}

return returnLabel;
```

Feature Extraction:

The blood report consisted of total of 19 features of which we have considered 4 features namely Hemoglobin, W.B.C. Count, Platelet Count and Bilirubin. We also have considered 16 symptoms which are unique for every disease.

Evaluation:

K	1
Metric Type	Euclidian
Accuracy	95%
Time(seconds)	0.133 seconds

This evaluation is based on 40 records given as input during training phase and 10 records during testing phase.
The running time complexity of the algorithm **is O(m*d*n)**
Where,

m is number of examples in training records (here 34)
d is number of dimensions (here 20)
n is number of examples in training set (here 10)

Module 3: Real Time Medical Emergency Service module
The third module is centered on providing emergency medical assistance during events like road accidents [4]. This module takes location of the victim, from the web browser using Google Maps Geolocation API, as input searches for any nearby hospital using Google Maps Places API and relays the location of the victim to the hospital. This location is then sent to the ambulance driver along with the optimal path to the victim using Google Maps Direction API. Any user need not be registered for using this feature. A user can also view hospitals nearby him on the basis of the specialty he wants.

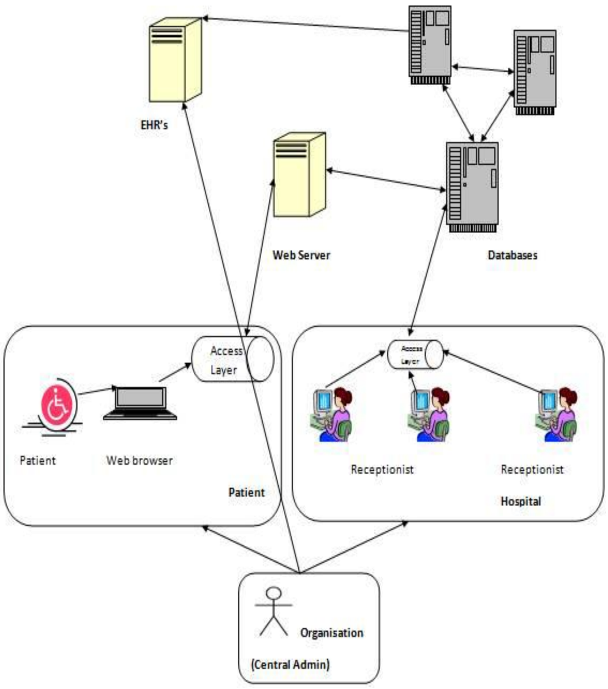

Fig 1: System Architecture

IV.METHODOLOGY

K-means algorithm uses an iterative process in order to cluster database. It takes the number of desired clusters and the initial Means (centroids or averages) as inputs and produces final means as output. Mentioned first and last means are the means of clusters. If the algorithm is required to produce K clusters then there will be K initial means and K final means. In completion, K-means algorithm produces K final means which answers why the name of algorithm is K-means. According to the algorithm we firstly select k objects as initial cluster centers, then calculate the distance between each cluster center and each object and assign it to the nearest cluster, update the averages of all clusters, repeat this process. Here we build a disease prediction model that initially predicts about 50 different diseases from patient's blood report and symptoms. Each disease is actually a cluster from K-Means point of view. We provide initial Mean, i.e. a blood report and set of symptoms which uniquely identify the every disease as training data to K-Means using which it builds the prediction model. After the model is built we provide a test blood report sample and test the prediction model. Disease Spread Pattern is identified using Map Reduce Technique on HIVE database. The input to this algorithm is taken from EMR data of patients. Particularly, we require the patient's location, age, gender, date of admission, disease identified and medicines prescribed for this analysis. Using this data and Map Reduce Technique we can find out how many number of patients were affected by a particular disease in a given area.

V. CONCLUSİON

Our System is useful for all Hospitals but also has some limitations. It gives details about various modules. The system successfully handles the database concepts. We can view the details for registered user. The system is online and work on client-server application. The system is platform independent, so it can work in any environment. So the conclusion is the system is useful but not the perfect one. Finally we can conclude that the system we have developed will eliminate the existing system drawbacks and limitations to maximum extent and provide the user with a product with a high quality, standard, and excellence.

REFERENCES

[1] Boris A. Kobrinskii, "E-Health and Telemedicine: Current State and Future Steps", 10 November 2014
[2] Ram D. Sriram, "The Role of Standards in Healthcare Automation", 5th Annual IEEE Conference on Automation Science and Engineering Bangalore, India, August 22-25, 2009
[3] Charalampos Doukas, Thomas Pliakas, and Ilias Maglogiannis," Mobile Healthcare Information Management utilizing Cloud Computing and AndroidOS" 32nd Annual International Conference of the IEEE EMBS Buenos Aires, Argentina, August 31 - September 4, 2010
[4] Yuanyuan Du, Yu Chen, Dan Wang, Jinzhao Liu, Yongqiang Lu," An Android-Based Emergency Alarm and Healthcare Management System" 978-1- 61284-704-7/11/$26.00 ©2011IEEE
[5] Alin CORDOS1 , Bogdan ORZA1 , Aurel VLAICU1, "Hospital Information System using HL7 and DICOM standards", 7, October 2010