International Journal of Engineering and Manufacturing Science.
ISSN 2249-3115 Volume 8, Number 1 (2018) pp. 119-133

http://www.ripublication.com

A Hospital Queuing-Recommender System for Predicting Patient Treatment Time Using Random Forest Algorithm

Roopa G M[1*], Dr. Nirmala C R[2]

Assistant Professor, Prof. & Head
Department of Computer Science and Engineering
Bapuji Institute of Engineering and Technology, Davangere -577004.
roopa.rgm@bietdvg.edu , crn@bietdvg.edu

Abstract

Today overcrowding of patients is one of the major challenges faced by hospitals. Unnecessary and annoying waits for long periods result in substantial human resource and time wastage and increase the frustration endured by patients. For each patient in the queue, the total treatment time of all the patients before him is the time that he must wait. It would be convenient and preferable if the patients could receive the most efficient treatment plan and know the predicted waiting time through a mobile application that updates in real time. Therefore, we propose a Patient Treatment Time Prediction (PTTP) algorithm using Random Forest (RF) algorithm to predict the waiting time for each treatment task for a patient. We use realistic patient data from various hospitals to obtain a patient treatment time model for each task. Base on this large-scale, realistic dataset, the treatment time for each patient in the current queue of each task is predicted. Based on the predicted waiting time, a Hospital Queuing-Recommendation (HQR) system is developed which calculates and predicts an efficient and convenient treatment plan recommended for the patient. We use an Apache Hadoop based server to achieve the aforementioned goals.

Keywords: BigData, Hadoop, Patient Records, Predicting models, Recommender system, Random Forest Algorithm

1. INTRODUCTION

Due to the huge increase in population hospitals are overcrowded because of this it becomes difficult for hospital management system to control and to minimize the patient waiting time while getting treatment in hospital. Management of patient queue and calculation of waiting time is very difficult task. Thus, a patient queue management and wait time prediction forms a challenging and complicated job because each patient might require different phases/ operations, such as a check-up, various tests, e.g., a sugar level or blood test, X-rays or a Computerized Tomography

(CT) scan, minor surgeries, during treatment. We call each of these phases /operations as treatment tasks or tasks. Each treatment task can have varying time requirements for each patient, which makes time prediction and recommendation highly complicated. A patient is usually required to undergo examinations, inspections or tests (refereed as tasks) according to his condition. In such a case, more than one task might be required for each patient. Some of the tasks are independent, whereas others might have to wait for the completion of dependent tasks. Most patients must wait for unpredictable but long periods in queues, waiting for their turn to accomplish each treatment task.

The massive unstructured data is called Big Data. Basically, the term big data not only means large volume of data but also other features that differentiate it from the concepts of ‘‘massive data or large volume of data’’. Now in present days very less amount of data is generated in structured form as compare to unstructured data e.g. Text files, sensor data, log data, web data, social networking data or different varieties of data. For Big Data management Hadoop is used.

Apache Hadoop is an open-source software framework used for distributed storage and processing of big data sets using the MapReduce programming model. It consists of computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common occurrences and should be automatically handled by the framework. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part which is a MapReduce programming model. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel. This approach takes advantage of data locality, where nodes manipulate the data they have access to. This allows the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

1.1 Existing System

- A parallel boosted regression tree algorithm was introduced for web search ranking. A multi-branch decision tree algorithm was also proposed based on a correlation-splitting criterion.
- A keyword-aware service recommendation method on MapReduce was proposed for big data applications. A travel recommendation algorithm that mines people's attributes and travel-group types was proposed.
- A Bayesian-inference-based recommendation system for online social networks was proposed, in which a user propagates a content rating query along the social network to his direct and indirect friends.

1.2 Problem Definition

- Most of the data in hospitals are massive, unstructured, and high dimensional. Hospitals produce a huge amount of business data every day that contain a great deal of information.
- Because of the manual operation and various unexpected events during treatments, a large amount of incomplete or inconsistent data appears.
- The time consumption of the treatment tasks in each department might not lie in the same range, which can vary according to the content of tasks and various circumstances, different periods, and different conditions of patients. For example, in the case of a CT
- scan task, the time required for an old man is generally longer than that required for a young man.
- There are strict time requirements for hospital queuing management and recommendation.

1.3 Proposed Solution

- The work focus on helping patients complete their treatment tasks in a predictable time and helping hospitals schedule each treatment task queue and avoid overcrowded and ineffective queues.
- The massive realistic data from various hospitals to develop a patient treatment time consumption model is collected. The realistic patient data are analyzed carefully and rigorously based on important parameters, such as patient treatment start time, end time, patient age, and detail treatment content for each different task.
- We identify and calculate different waiting times for different patients based on their conditions and operations performed during treatment.
- A Patient Treatment Time Prediction (PTTP) model is trained based on hospitals' historical data. The waiting time of each treatment task is predicted by PTTP, which is the sum of all patients' waiting times in the current queue. Then, according to each patient's requested treatment tasks, a Hospital Queuing-Recommendation (HQR) system recommends an efficient and convenient treatment plan with the least waiting time for the patient.

1.4 Objectives

- Hospital data from different task are gathered which consists of information like Registration, Checkup, Medicine, CT scan, Blood Tests etc.

- All the data collected from different hospitals are converted into the same dimensions i.e. choosing the same features of data such as gender, age, start time and end time to train the PTTP model.

- To train the PTTP model important new feature variables are calculated like patient time consumption for each treatment. After calculating the new features variable, the error and noisy data are removed.

- The PTTP algorithm is trained based on a Random Forest (RF) algorithm using features variable calculated for each treatment task, and the waiting time of each task is predicted based on the trained PTTP model.

- Then, HQR recommends an efficient and convenient treatment plan for each patient based on predicted time.

2. BACKGROUND THEORY

Tyree et al. [1] described that Patient queue management and wait time prediction form a challenging and complex job because each patient might require different operations, such as a checkup, various tests during treatment. So there are five major methodologies used in this system Big Data management with Historical Dataset, Preprocessing of data, Use Learning Algorithm PTTP(patient Treatment Time prediction)with base of improved RF.(Random Forest) Algorithm, Calculates the Waiting Time in Hospital Queue Recommendation. A random forest optimization algorithm is performed for the PTTP model. He also introduced a parallel boosted regression tree algorithm for search ranking. The queue waiting time of each treatment task is predicted using the trained PTTP model.

A parallel HQR system is introduced, and an efficient and convenient treatment plan is recommended for each patient. The patient may undergo various treatment operations such as CT scan, MR scan and a payment task. These set of treatment operations are submitted to decision maker and recommendation module via mobile interface. The predicted waiting Time of all of the treatment tasks is calculated by PTTP model. After this a treatment recommendation with least waiting time is advised.

Meng et al. [2] proposed a keyword-aware service recommendation method on MapReduce for big data applications. Hospital data is center generally stores the Structured and Unstructured Data. Most data used in the EMR is Structured Data which includes information of a patient, information of a treatment, diagnostic information and the reports. This above data is stored in the Hadoop cluster with the help of a JDBC/ODBC interface. To process the data in the system we need to give a connection to the database then we need to check an existence of a table i.e. if a Table

exists already then we need to add a Partition otherwise we need to create a Table and then we need to add a Partition. Then we need to check an availability of a data if a data is available then just update the data otherwise write the data and then we need to disconnect with the database. Hadoop is a framework which provides distributed processing of large data sets across cluster using a simple programming model. Mainly Apache Hadoop Framework consists MapReduce and Hadoop distributed file system. Hadoop distributed file system, as Map reduce provide a simple programming model well as other related projects. Large amount of data is created by users in daily life which requires huge amount of storage and various techniques to discover knowledge from data. Hadoop architecture is of two main components HDFS (Hadoop Distributed File System) and MapReduce for Big Data Analytics. There are various technologies belongs to Hadoop for storing large dataset, Apache pig is scripting language for processing of large data set, Hive is designed for OLAP is fast and Scalable, Scoop is used for import and export data from

3. SYSTEM OVERVIEW

Figure.1 describes the system architecture model of HQR System and includes the following modules:

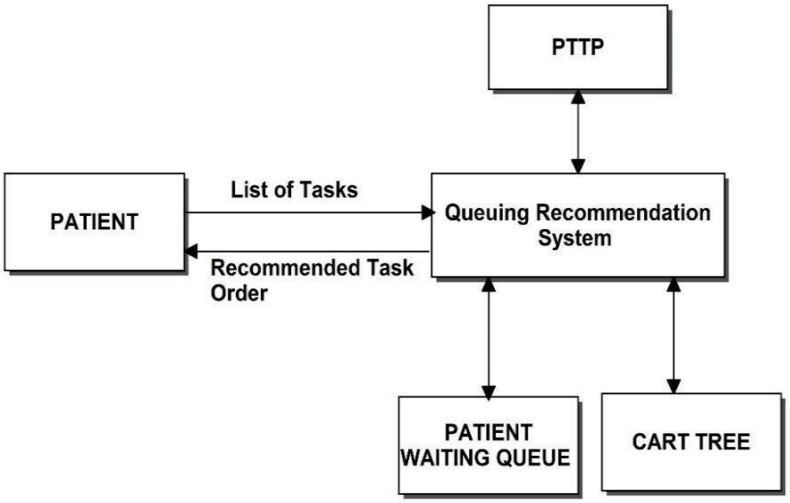


Figure 1. System overview of HQR System

- **Patient** - This module will contain all of the information which are required by the Queuing Recommendation for prediction of waiting time such as age, gender, treatment, tasks, etc.
- **PTTP** - This module is trained based on patient's data. The waiting time of each treatment task is predicted by PTTP, which is the sum of all patients waiting time in the current queue.

- **Queuing Recommendation System** - This module is the core module of the HQR system. According to each patient requested treatment and on the basis of trained PTTP, this module system will recommend efficient treatment order.
- P**atient Waiting Queue** - This module stores the data about patients who are in the waiting queue.
- **CART Tree** - The term Classification and Regression Tree (CART) analysis is an umbrella term used to refer to classification and regression tree analysis performed by the RF algorithm.

In a general Hospital use case diagram, there are 4 actors i.e. patient, scheduler, doctor and clerk. The patient can make or cancel appointment via scheduler. The doctor will provide medication to the user or he can check old medical records of the patients. The patient will make the payment for the bill which will be referred by the doctor to the clerk. Clerk can also check for any active insurance policy for the patient.

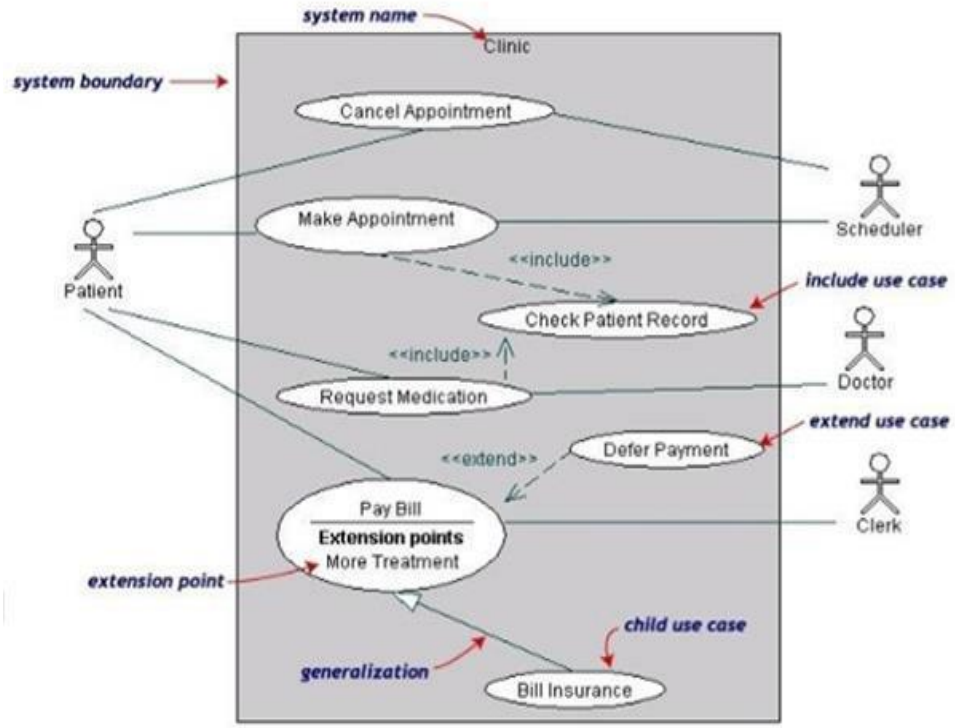
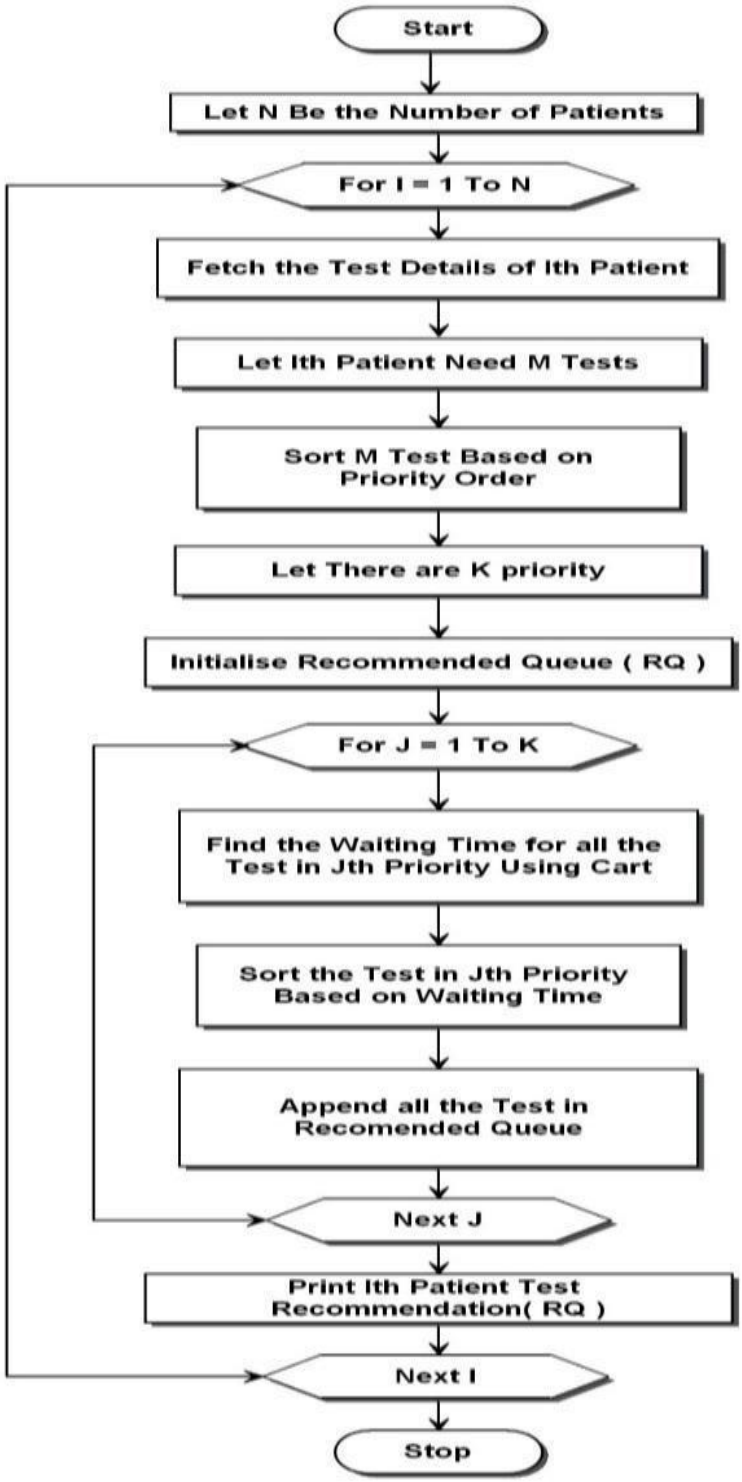


Figure 2: Generalized Use-case Diagram of HQR System

3.1: Workflow Procedure



3.2: Sequence of Actions

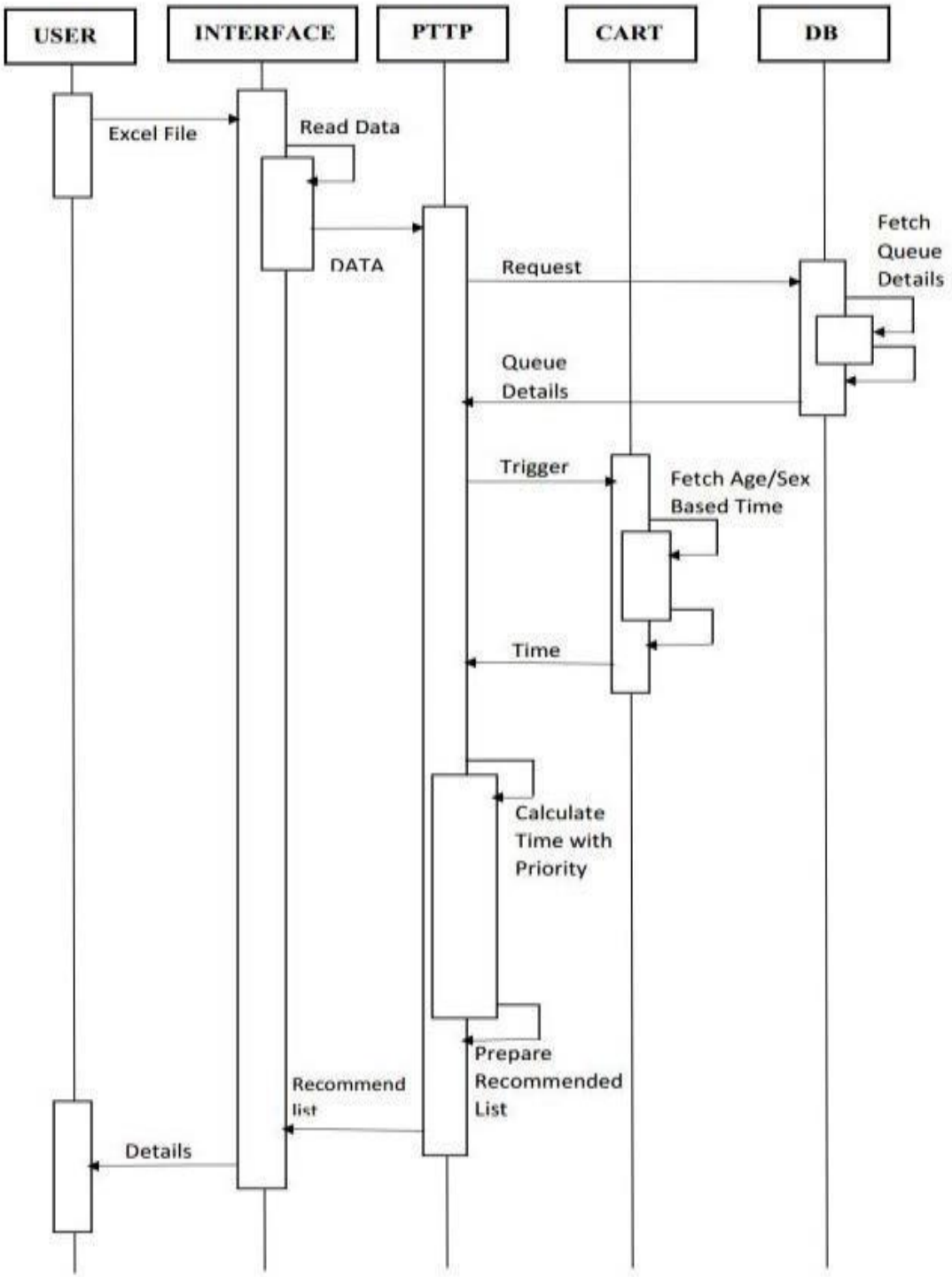


Figure 4: Sequence of actions

4. RESULTS AND DISCUSSION:

4.1: Installation Steps for Hadoop on Windows Using Cygwin

1. Download and run the Cygwin setup.exe file.
2. Once you start installing, goes on clicking next until we welcomed on page to select packages.
3. Choose OpenSSH installation along with Cygwin and proceed with the installation.
4. Now we installed Cygwin with OpenSSH2, set Environment Variable in Window.
5. Setup SSH daemon, start SSH daemon and setup the authorization keys.
6. Install JAVA and set Environment Variables.
7. Download a recent stable release of Hadoop from one of the Apache Download Mirrors.
8. Extract the Hadoop files and folder and edit the configuration files.

4.2: Procedures used for developing the HQR System:

To build the HQR System based on both patient and time characteristics, a PTTP algorithm is proposed. The PTTP model is based on an improved RF algorithm and is trained from the massive, complex, and noisy hospital treatment data. For building the system, the following procedures are carried out:

a) Prediction based on analysis and processing of massive noisy patient data from various hospitals is a challenging task. Most of the data in hospitals are massive, unstructured, and high dimensional. Hospitals produce a huge amount of business data every day that contain a great deal of information, such as patient information, medical activity information, time, treatment department, and detailed information of the treatment task. Moreover, because of the manual operation and various unexpected events during treatments, a large amount of incomplete or inconsistent data appears, such as a lack of patient gender and age data, time inconsistencies caused by the time zone settings of medical machines from different manufacturers, and treatment records with only a start time but no end time.

b) To train the patient time consumption model for each treatment task, we choose the same features of these data, such as the patient information (patient card number, gender, age, etc.), the treatment task information (task name, department name, doctor name, etc.), and the time information (start time and end time). Other feature subspaces of the treatment data are not chosen because they are not useful for the PTTP algorithm, such as patient name, telephone number, and address.

c) To train the PTTP model, various important features of the data should be calculated, such as the patient time consumption of each treatment record, day of week for the treatment time, and the time range of treatment time.

d) After calculating new feature variables of treatment data, the error and noisy data need to be removed. The treatment records with missing values for critical features are removed as incomplete data, such as patient gender, patient age, and task name. The treatment records with negative values of time consumption are removed as inconsistent data. To predict the waiting time for each patient treatment task, the patient treatment time consumption based on different patient characteristics and time characteristics must first be calculated. The time consumption of each treatment task might not lie in same range, which varies according to the content of tasks and various circumstances, different periods, and different conditions of patients. Therefore, we use the RF algorithm to train patient treatment time consumption based on both patient and time characteristics and then build the PTTP model.

e) After training the PTTP model for each treatment task using historical hospital treatment data, a PTTP-based hospital queue recommendation system is developed. An efficient and convenient treatment plan is created and recommended to each patient to achieve

4.3 Pseudo code for PTTP Process

```
for i =1 to k do
create training subset straini ←sampling(STrain);
    create OOB subset sOOBi ←(STrain−straini);
    create an empty CART tree hi;
    for each independent variable yj in straini do
            calculate candidate split points vs ←yj;
            for each vp in vs do
                    calculate the best split point (yj,vp)
            end for
                    append node Node(yj, vp) to hi;
                    split data for left branch RL(yj,vp) ←{x|yj ≤ vp};
                    split data for right branch RR(yj,vp) ←{x|yj > vp};
            for each data R in{RL(yj,vp), RR(yj,vp)}do
                    calculate ϕ(vpL|yj)←maxi ϕ(vi|y);
                if (ϕ(vp(L|R)|yj)≥ ϕ(vp|yj)) then
                    append subnode Node(yj,vp(L|R)) toNode(yj,vp) as multi-branch;
```

split data to two forks RL(yj,vpL) and RR(yj,vpR);
else
collect cleaned data for leaf node Dleaf ← (IL ≤ yj ≤ OL);
calculate mean value of leaf node c ← 1/k∑Dleaf ;
end if
end for
remove yj from strain;
end for
calculate accuracy CAi ← I(hi(x)=y)/ I(hi(x)=y)+PI(hi(x)=z)
for hi by testing sOOBi; end for
PTTPRF ←H(X,Θj)← 1/ k∑ ^{k} i=1 [CAi × hi];
return PTTPRF.

4.4 Pseudo code for Hospital Queuing Recommendation

create map Ts(X)← HashMap < string,double >;
for each Task-i in X do
create array Ui[]← patients-in-waiting of Task-i;
for each patient Uik in Ui do
predict time consumption Tik ← PTTPRF;
end for
calculate predicted waiting time Ti ← / Wi∑^{m} k=1 Tik;
append waiting time Ts(X)← < Task-i , Ti >;
end for 1
sort map Ts(X) in an ascending order;
for each < Task-i , Ti > in Ts(X) do
if (Task-i has dependent tasks) then
put records of the dependent tasks before Task-i;
end if
end for
return Ts(X).

4.5 Screen Shots

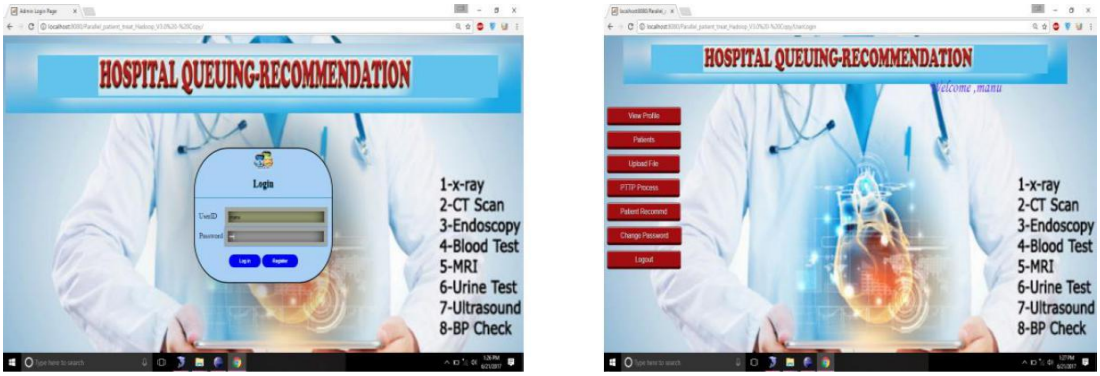


Figure : 5(a) Login Page 5(b) Welcome Page

Figure 5(a): User (the person who will control and manage the system) will start the system, he/she will welcome with the login screen where user can enter their valid credentials or new users can register them within the system.

Figure 5(b): Once user's credentials are verified, they will be log into the system and presented with several options to manage the system.

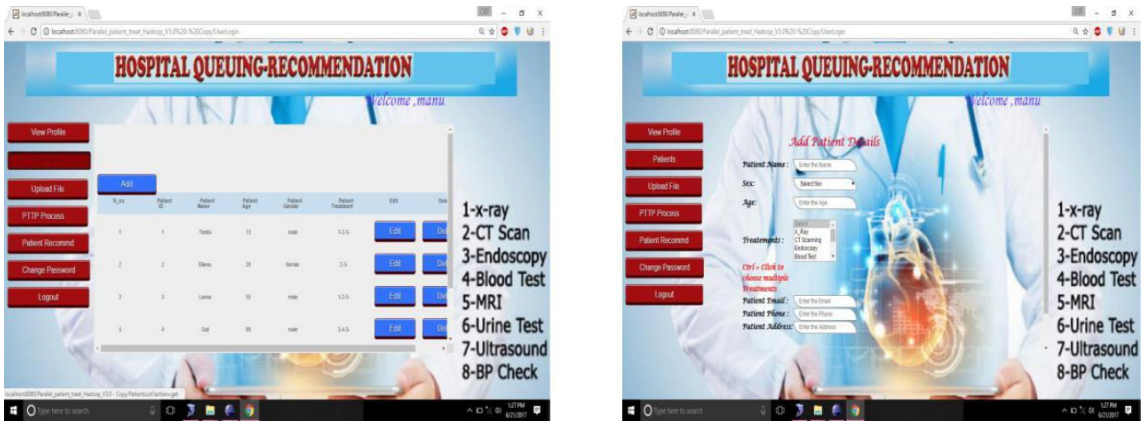


Figure :6(a) Patient Details 6(b) Add Patient Details

Figure 6(a): Patient tab leads the user to view, edit or delete the patient information. 6(b) gives the admin to add new patient details to the system.

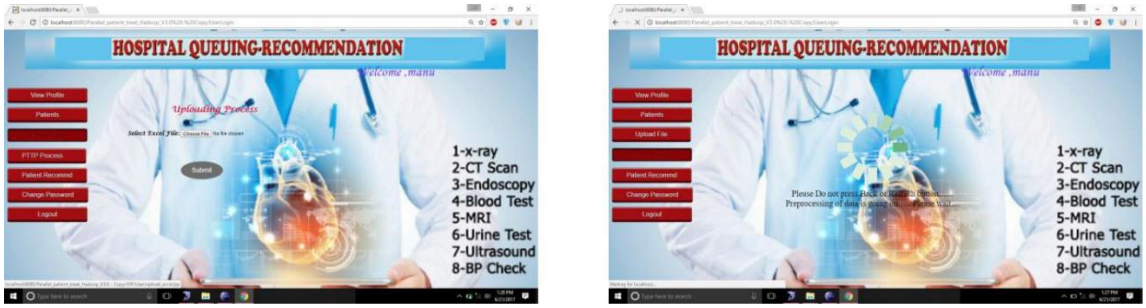


Figure :7 (a) Uploading Process 7(b) Running PTTP Process

Figure 7(a): illustrates to upload file where the user needs to select the excel file containing information about the patient which will be uploaded to the HDFS for processing. 7(b) Triggers the PTTP process option to apply on the uploaded patient data.

Figure 8: Predicted waiting time recommended to the user

5. CONCLUSION

A PTTP algorithm based on BigData environment is proposed to tackle queuing problem in hospitals for different tasks. A Random Forest optimization algorithm is performed for the PTTP model. The queue waiting time of each treatment task is predicted based on the trained PTTP model. A parallel HQR system is developed, and an efficient and convenient treatment plan is recommended for each patient which provides a high level of satisfaction to the user by reducing the patient waiting time at various stages of treatment.

Some of the future enhancements that can be done for the proposed system are:

1. A mobile application can be developed that will notify patient about their queue status.
2. A method can be included that will predict the time of their appointment using machine learning techniques.
3. A more convenient recommendation with minimized path-awareness can also be used.

REFERENCES

[1] R. Fidalgo-Merino and M. Nunez, ‘‘Self-adaptive induction of regression trees,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 8, pp. 1659–1672, Aug. 2011.

[2] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin, ‘‘Parallel boosted regression trees for Web search ranking,’’ in Proc. 20th Int. Conf. World Wide Web (WWW), 2012, pp. 387–396.

[3] N. Salehi-Moghaddami, H. S. Yazdi, and H. Poostchi, ‘‘Correlation based splitting criterionin multi branch decision tree,’’ Central Eur. J. Computer. Sci., vol. 1, no. 2, pp. 205–220, Jun. 2011.

[4] T.G. Morrell, L. Kerschberg, "Personal Health Explorer: A Semantic Health Recommendation System," in Data Engineering Workshops (ICDEW), 2012 IEEE 28th International Conference , Arlington, 2012.

[5] G. Chrysos, P. Dagritzikos, I. Papaefstathiou, and A. Dollas, ‘‘HC-CART: A parallel system implementation of data mining classification and regression tree (CART) algorithm on a multi-FPGA system,’’ ACM Trans. Archit. Code Optim., vol. 9, no. 4, pp. 47:1–47:25, Jan. 2013.

[6] E. Sezgin, S. Ozkan,, "A systematic literature review on Health Recommender Systems," in E-Health and Bioengineering Conference (EHB), Iasi, 2013.

[7] N. T. Van Uyen and T. C. Chung, ‘‘A new framework for distributed boosting algorithm,’’ in Proc. Future Generat. Commun. Netw. (FGCN), Dec. 2007, pp. 420– 423.

[8] Y. Ben-Haim and E. Tom-Tov, ‘‘A streaming parallel decision tree algorithm,’’ vol. 11, no. 1, pp 849–872, Oct. 2010. P. B. Kantor, L. Rokach, F. Ricci, B. Shapira, Recommender Systems handbook, Springer-Verlag New York, 2010.

[9] L. Breiman, ‘‘Random forests,’’ Mach. Learn., vol. 45, no. 1, pp. 5–32, Oct. 2001

[10] G. Yu, N. A. Goussies, J. Yuan, and Z. Liu, ‘‘Fast action detection via discriminative random forest voting and top-K subvolume search,’’ IEEE Trans. Multimedia, vol. 13, no. 3, 507–517, Jun. 2011.

[11] G. Adomavicius, A. A. Tuzhilin, "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions," IEEE Trans. on Knowl. and Data Eng., vol. 17, no. 6, pp. 734-749, 2005.

[12] Bamshad Mobasher, Honghua Dai, Tao Luo, Yuqing Sun and Jiang Zhu, "Integrating Web Usage and Content Mining for More Effective Personalization, " Electronic Commerce and Web Technologies LCNS, vo.1875, pp.165 - 176, 2000.

[13] C. C. Aggarwal,A. Hinneburg, D. A. Keim, On the surprising behavior of distance metrics in high dimensional space, Berlin Heidelberg: Springer, 2001.