Perspectives in Communication, Embedded-Systems and Signal-Processing (PiCES) – An International Journal
ISSN: 2566-932X, Vol. 2, Issue 9, December 2018
Proceedings of National Conference on Knowledge Discovery in Information Technology and Communication Engineering (KITE 18), May 2018

Hospital Queuing-Recommendation for Predicting Parallel Patient Treatment Time in a Big Data Environment

Rakesh B S, Pooja U, Monika, Akanksha Pandey

Department of Computer Science, Brindavan College Of Engineering, Bengaluru, INDIA, monikatk.13@gmail.com

Abstract:* One of the major challenges faced by hospitals are effectively managing patient queue to minimize patient wait delay and overcrowding. 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 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. Based 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. HQR calculates and predicts an efficient and convenient treatment plan recommended for the patient. Because of the large scale, realistic dataset and the requirement for real-time response, the PTTP and HQR system mandate efficiency and low-latency response.

***Keywords:* Apache spark; big data; cloud computing; hospital queuing recommendation; patient treatment time prediction.**

I. INTRODUCTION

A. Motivation

Currently, most hospitals are overcrowded and lack effective patient queue management. Patient queue management and wait time prediction form a challenging and complicated job because each patient might require different phases/ operations, such as a checkup, various tests, e.g., a sugar level or blood test, X-rays or a CT scan, minor surgeries, during treatment. We call each of these phases /operations as treatment tasks or tasks in this paper. 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 tasksare 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.In this paper, we 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. We use massive realistic data from various hospitals to develop a patient treatment time consumption model. 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. The work on of the patient treatment and wait model is illustrated in Fig. 1.

Fig. 1 illustrates three patients (Patient1, Patient2, and Patient3) and a set of treatment tasks required for each patient. Some tasks can be dependent on a previous one, e.g., surgery or bandage cannot be done before X-rays. Tasks { A; B; D} are required for Patient1, whereas task D must wait for the completion of B. Tasks {E; B; C; A} are required for Patient2, and tasks {D; E; C} are required for Patient3. Moreover, there are different numbers of patients waiting in the queue of each task, for example, 7 patients in the queue of task A and 5 patients in the queue of task B.

In this paper, 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.

The patient treatment time consumption of each patient in the waiting queue is estimated by the trained

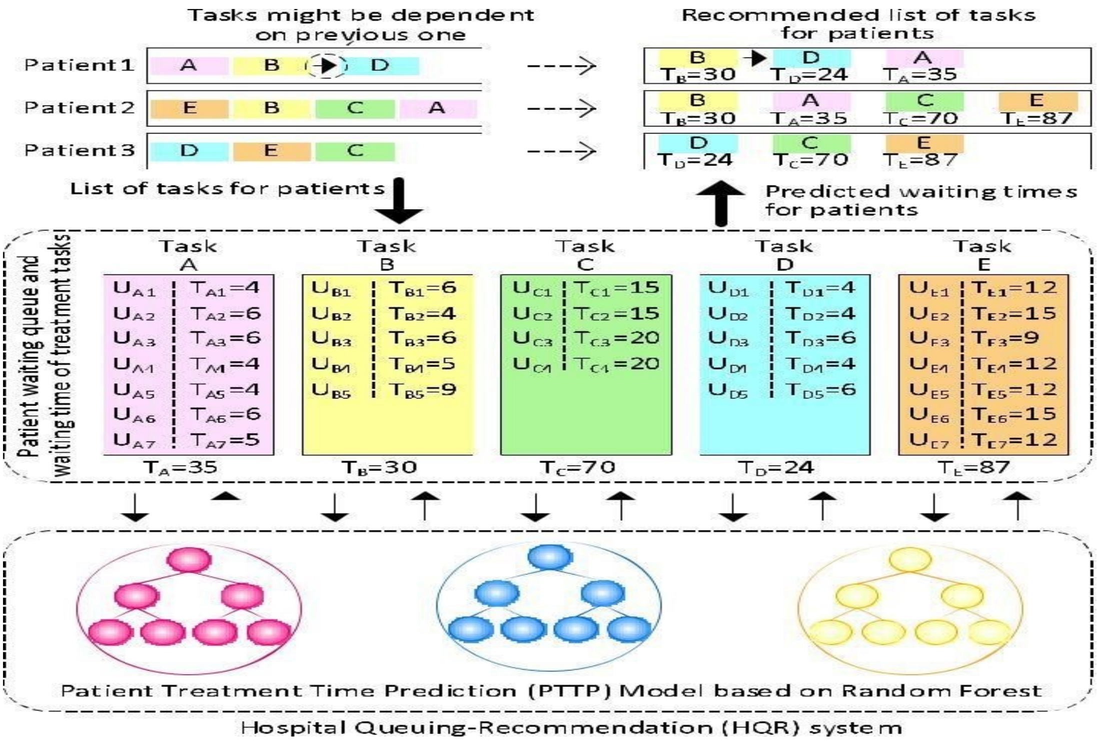

Tasks might be dependent on previous one
Recommended list of tasks for patients
Patient1
Patient2
Patient3
A B D
E B C A
D E C
B T_B=30 D T_D=24 A T_A=35
B T_B=30 A T_A=35 C T_C=70 E T_E=87
D T_D=24 C T_C=70 E T_E=87
List of tasks for patients
Predicted waiting times for patients
Patient waiting queue and waiting time of treatment tasks
Task A
Task B
Task C
Task D
Task E
U_{A1} T_{A1}=4 U_{A2} T_{A2}=6 U_{A3} T_{A3}=6 U_{A4} T_{A4}=4 U_{A5} T_{A5}=4 U_{A6} T_{A6}=6 U_{A7} T_{A7}=5
U_{B1} T_{B1}=6 U_{B2} T_{B2}=4 U_{B3} T_{B3}=6 U_{B4} T_{B4}=5 U_{B5} T_{B5}=9
U_{C1} T_{C1}=15 U_{C2} T_{C2}=15 U_{C3} T_{C3}=20 U_{C4} T_{C4}=20
U_{D1} T_{D1}=4 U_{D2} T_{D2}=4 U_{D3} T_{D3}=6 U_{D4} T_{D4}=4 U_{D5} T_{D5}=6
U_{E1} T_{E1}=12 U_{E2} T_{E2}=15 U_{E3} T_{E3}=9 U_{E4} T_{E4}=12 U_{E5} T_{E5}=12 U_{E6} T_{E6}=15 U_{E7} T_{E7}=12
T_A=35
T_B=30
T_C=70
T_D=24
T_E=87
Patient Treatment Time Prediction (PTTP) Model based on Random Forest
Hospital Queuing-Recommendation (HQR) system

Fig 1. Workflow of patient treatment and wait model

PTTP model. The whole waiting time of each task at the current time can be predicted, such as {T_A = 35(min); T_B = 30(min); T_C =70(min); T_D =24(min); T_E =87(min)}. Finally, the tasks of each patient are sorted in an ascending order according to the waiting time, except for the dependent tasks. A queuing recommendation is performed for each patient, such as the recommended queuing {B; D; A} for Patient1, {B; A; C; E} for Patient2, and {D; C; E} for Patient3.

To complete all of the required treatment tasks in the shortest waiting time, the waiting time of each task is predicted in real-time. Because the waiting queue for each task updates, the queuing recommendation is recomputed in real-time. Therefore, each patient can be advised to complete his treatment activities in the most convenient way and with the shortest waiting time.

B. Our Contribution

Our contributions in this paper can be summarized as follows.

A PTTP algorithm is proposed based on an improved Random Forest (RF) algorithm. The predicted waiting time of each treatment task is obtained by the PTTP model, which is the sum of all patients' probabletreatment times in the current queue. An HQR system is proposed based on the predicted waiting time. A treatment recommendation with an efficient and convenient treatment plan and the leastwaiting time is recommended for each patient

The PTTP algorithm and HQR system are parallelized on the Apache Spark cloud platform at the National Supercomputing Center in Changsha (NSCC) to achieve the aforementioned goals. Extensive hospital data are stored in the Apache HBase, and a parallesolution is employed with the MapReduce and Resilient Distributed Datasets (RDD) programming model.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 details a PTTP algorithm and an HQR system. The parallel implementation of the PTTP algorithm and HQR system on the Apache Spark cloud environment is detailed in Section 4. Experimental results and evaluations are presented in Section 5 with respect to the recommendation accuracy and performance. Finally, Section 6 concludes the paper with future work and directions.

II. PATIENT TREATMENT TIME PREDICTION ALGORITHM

To build the PTTP model based on 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.

A. Problem Definition And Data Preprocessing

a) Problem Definition

Prediction based on analysis and processing of massive noisy patient data from various hospitals is a challenging task. Some of the major challenges are the following:

1) 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.

2) 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.

3) There are strict time requirements for hospital queuing management and recommendation. The speed of executing the PTTP model and HQR scheme is also critical.

B. *Data Preprocessing*

In the preprocessing phase, hospital treatment data from different treatment tasks are gathered. Substantial numbers of patients visit each hospital every day. Let S be a set of patients in a hospital, and a patient who has been registered and his information is represented by si. Assume that there are N patients in S:

$$S=\{s_1; s_2; : : : ; s_N\};$$

where each patient si can have specific unchanged parameters, e.g., name, ID, gender, age, and address. Some of these parameters are useful to our analysis, whereas others are not.

Each patient can visit multiple treatment tasks according to his health condition. Let X jsi be a set of treatment tasks for patient si during a specific visit:

$$X|s_i=\{x_1; x_2; : : : ; x_K\};$$

where each treatment task record xi can consist of multiple information Y , e.g., task name, task location, department, start time, end time, doctor, and attending staff:

$$Y|xi=\{y_1; y_2; : : : ; y_M\};$$

where yj is a feature variable of the record of treatment task xi. Here, for a single visit, we have a single record for patient name, age, gender, and multiple records for treatment tasks, as shown in Table1.

The work on of the preprocessing task can be depicted by the following steps.

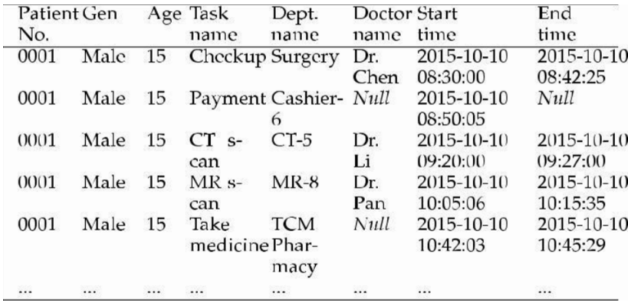
a) *Gather Data From Different Treatment Tasks*

Depending on statistics, the number of patients in a medium sized hospital lies between 8,000 and 12,000 per day, and the number of remedial treatment data records is between 120,000 and 200,000.

b) *Choose The Same Dimensions Of The Data*

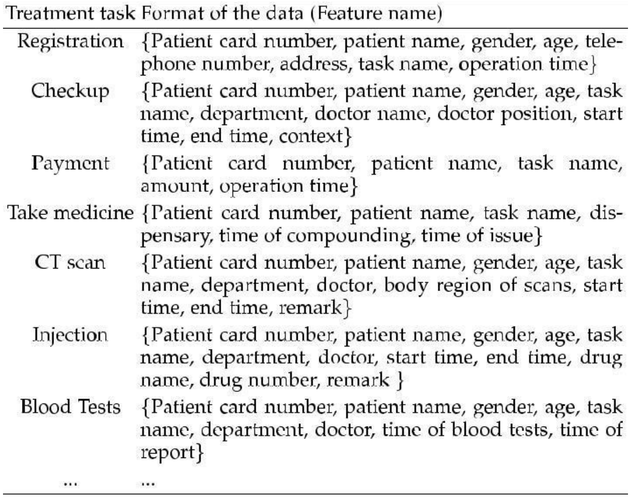
The hospital treatment data generated from different treatment tasks have different contents and formats as well as varying dimensions. To train the patient time consumption model for each treatment task, we choose the same features of these data, such as the patient information (patient card number, 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.

Table 1. Example of treatment records.



Patient No.	Gen	Age	Task name	Dept. name	Doctor name	Start time	End time
0001	Male	15	Checkup	Surgery	Dr. Chen	2015-10-10 08:30:00	2015-10-10 08:42:25
0001	Male	15	Payment	Cashier-6	*Null*	2015-10-10 08:50:05	*Null*
0001	Male	15	CT s-can	CT-5	Dr. Li	2015-10-10 09:20:00	2015-10-10 09:27:00
0001	Male	15	MR s-can	MR-8	Dr. Pan	2015-10-10 10:05:06	2015-10-10 10:15:35
0001	Male	15	Take medicine	TCM Phar-macy	*Null*	2015-10-10 10:42:03	2015-10-10 10:45:29
...	...	...	...	...	...	...	...

Table 2. Formats of the data for different treatment tasks



Treatment task	Format of the data (Feature name)
Registration	{Patient card number, patient name, gender, age, telephone number, address, task name, operation time}
Checkup	{Patient card number, patient name, gender, age, task name, department, doctor name, doctor position, start time, end time, context}
Payment	{Patient card number, patient name, task name, amount, operation time}
Take medicine	{Patient card number, patient name, task name, dispensary, time of compounding, time of issue}
CT scan	{Patient card number, patient name, gender, age, task name, department, doctor, body region of scans, start time, end time, remark}
Injection	{Patient card number, patient name, gender, age, task name, department, doctor, start time, end time, drug name, drug number, remark }
Blood Tests	{Patient card number, patient name, gender, age, task name, department, doctor, time of blood tests, time of report}
...	...

c) *Calculate New Feature Variables Of The Data*

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.

For example, in the treatment record of the CT scan task in Table 1, the start time is ``2015-10-10 09:20:00" and the end time is ``2015-10-10 09:27:00", the time consumption for this patient in the treatment is ``420 (s)", the day of the week is ``Saturday", and the time range is ``09".

d) Remove Incomplete And Inconsistent Data

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, for instance, if the end time of the treatment operation is before the start time, which can occur in cases when a start time is recorded by a human and an end time is shown by a machine. The types of data shown above are considered as noisy data in this paper. The features of the treatment data used in the process of employing the PTTP algorithm are presented in Table 3.

Table 3. Features of treatment data for the PTTP algorithm

No.	Feature Name	Value range of each feature subspace
y_1	Patient Gender	"Male", "Female".
y_2	Patient Age	The age of the patient.
y_3	Department	All departments in the hospital.
y_4	Doctor Name	All doctors in the hospital.
y_5	Task Name	Each treatment task in all treatment processes in the hospital.
y_6	Start Time	The start time of the treatment task.
	End Time	The end time of the treatment task.
y_8	Week	The day of week for the treatment time. The value is from Monday to Sunday.
y_9	Time Range	The time range of treatment time in a day. The value is from 0 to 23.
y_{10}	Time Consumption	(1) End time - Start time, such as a CT scan, an MR scan. (2) Time interval between one patient and the next in the same treatment, such as payment.

At the same time, the unselected data in each sampling period are composed as an out-of-bag (OOB) dataset. k OOB sets are constructed as a collection of SOOB:

$$S_{OOB} = \{S_{OOB1}; S_{OOB2} \ ; : : : ; S_{OOBk}\};$$

where k N , STrain 2 S, and SOOB 2 S. These datasets are used as testing sets after the training process to verify the classification or regression accuracy of each tree. The process of the training dataset random sampling for the RF model is shown in Fig. 2.

C. Constructing Training

a) Subsets For The PTTP Model

In the process of employing the PTTP model, the treatment time consumption of patients with different conditions and different environments in each treatment task are addressed. Due to the diverse nature of different medical tasks, the range of patient treatment time consumption cannot be measured by an absolute standard.

To improve the accuracy of the PTTP model, an improved RF algorithm is used to build the PTTP model. k training subsets are sampled from the original training dataset S in a bootstrap sampling process. N samples are selected from S by a random sampling and replacement method in each sampling period. After the current step, k training subsets are constructed as a collection of STrain:

$$S_{Train} = \{ s_{train1} \ ; s_{train2} \ ; : : : ; s_{traink} \}:$$

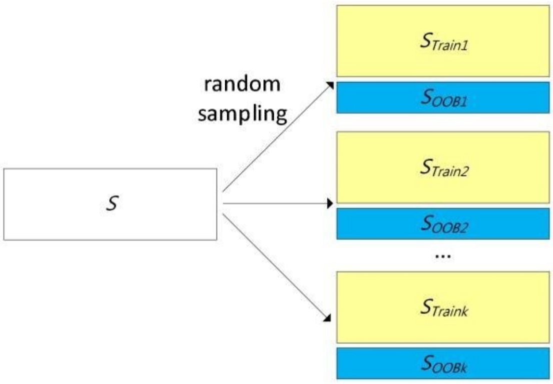


Fig 2. Process of training dataset random sampling for the PTTP model.

b) PTTP Model Based On The Improved RF Algorithm

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.

Because of the limitations of the original RF algorithm and the characteristics of hospital treatment data, the RF algorithm is improved in 4 aspects to obtain an effective result from large-scale, high dimensional, continuous, and noisy hospital treatment data.

1) All of the selected (cleaned) features of the data are used in the training process, instead of m features selected randomly, as is done in the original RF algorithm, because the features of the data are limited and the data are already cleaned of unnecessary features such as patient name, address, and telephone number.

2) Because the target variable of the treatment data is patient treatment time consumption, which is a continuous variable, a CART model is used as a meta-classifier in the improved RF algorithm. At the same time, some independent variables of the data are nominal data, which have different values such as time range (0 - 23) and day of week (Monday - Sunday). In such a case, the two-fork tree model of the traditional CART cannot fully react the analysis results. Therefore, to construct the regression tree model felicitously, a multi-branch model is proposed for the construction process instead of the two-fork model of the traditional CART algorithm.

3) Although we have removed part of the error in the preprocessing, other types of noisy data might also

exist. In some treatment tasks, the time consumption is the time interval between one patient and the next in the same treatment. For example, in a payment task, assume that the operation time point of the last patient in the morning is ``12:00:00" and the operation time point of the first patient in the afternoon is ``14:00:00". The time consumption of the former is ``7200 (s)" and is considered as incorrect data because it is larger than the normal value of ``100 (s)". However, the value ``7200 (s)" of time consumption has not always been incorrect data, such as in a blood examination task. Therefore, we cannot simply designate one value of time consumption as noisy data; each must be classified according to treatment data features. Then, we must identify and remove the noisy data. In calculating the average value of the data in each leaf node of the regression tree, noisy data are removed to reduce their influence on accuracy.

4) The original RF algorithm uses a traditional direct voting method in the prediction process. In such a case, a RF containing noisy decision trees would likely lead to an incorrect predicted value for the testing dataset. Therefore, in this paper, a weighted voting method is employed in the prediction process of the RF model. Each tree classifier corresponds to a specified reasonable weight for voting the testing data. A tree classifier that has high accuracy in the training process will have a high voting weight in the prediction process. Hence, the classifier improves the overall classification accuracy of the RF algorithm, and reduces the generalization error.

Compared with the original RF algorithm, our PTTP algorithm based on the improved RF algorithm, has significant advantages in terms of accuracy and performance.

i) Training Cart Regression Trees Of The RF Model

Because the patient treatment time consumption is the target feature variable of treatment data S, which is a continuous value, the type of the single decision tree in the RF model is a regression tree. Thus, a CART regression tree model is created for each training subset straini.

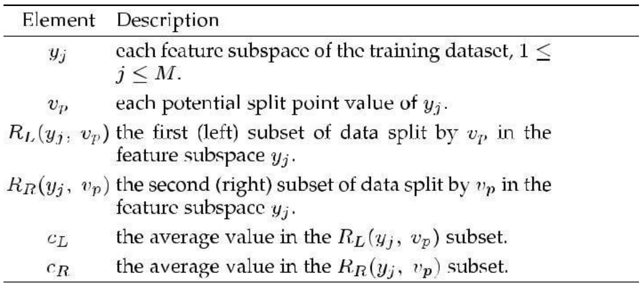
The first optimization aspect of the RF algorithm is in the growing process of each CART tree. All of the M features of each training data s_{traini} are used in the training process instead of the m features selected randomly as is done in the original RF algorithm. The main process of building the regression tree of CART is described as follows.

In such a case, the variable y_j with the smallest value of the loss function is selected as the best split feature, and the value v_p is used as the split point for y_j at the current splitting tree node.

ii) Split The Data Into Two Forks

Split the training dataset into two forks by v_p in the feature subspace y_j.

Table 4. Summary of the elements in eq. (1).



Element	Description
y_j	each feature subspace of the training dataset, $1 \leq j \leq M$.
v_p	each potential split point value of y_j.
$R_L(y_j;\ v_p)$	the first (left) subset of data split by v_p in the feature subspace y_j.
$R_R(y_j,\ v_p)$	the second (right) subset of data split by v_p in the feature subspace y_j.
c_L	the average value in the $R_L(y_j,\ v_p)$ subset.
c_R	the average value in the $R_R(y_j,\ v_p)$ subset.

$R_L(y_j;v_p)$ denotes the first (left) data subset and $R_R(y_j;v_p)$denotes the second (right) data subset. These subsets are denoted as follows:

$$R_{L(yj;\ vp)}=\{x|(y_j \leq v_p)\};$$

$$R_{R(y\ j;\ vp)} =\{\ x|(y_j > v_p)\}: \qquad (2)$$

iii) Construct Multi-Branch For The Cart Model

Some independent variables of data are nominal data, which have different values, such as the time range (0 - 23) and day of week (Monday - Sunday). Therefore, to construct the regression tree model felicitously, a multi-branch regression tree model instead of two-fork tree model is used constructing the CART, which is the second optimization aspect of the RF algorithm. After the tree node split into two forks by variable y_j and value v_p in step (2), the same variable y_j continues to be selected to calculate the best split point v_{pL} for the data in the left branch and v_{pR} for the data in the right branch. Taking the left branch as an example, the best split point calculated for the current feature subspace is denoted as follows:

$$\Phi(v_{pL}|y_j) = \max_i \Phi\ (v_i|y): \qquad (3)$$

The $\Phi(v_{pL}|y_j)$ is defined as follows:

$$\Phi(v_{pL}|y_j) = 2P_LP_R\textstyle\sum_{j-1}^{m} p(cj|yL)\ -\ p(c_j|y_R)|; \qquad (4)$$

where P_L and P_R are the ratios of the amount of data in the left branch and in the right branch to the entire volume of training data, respectively. $p(c_j|y_L\ )$ is the ratio of the volume of data that belong to class c_j in the left branch to the volume of data in the left branch.

If the split value of $\Phi\ (v_{pL}|y_j)$ is greater than the father node, namely $\Phi\ (v_{pL}\ |y_j)\ \Phi\ (v_p|y_j)$, then the left branch continues to split by the variable y_j and value v_{pL}. Otherwise, the remaining feature variables continue to be computed. The right branch is calculated similarly. Then, each node and its two subnodes are calculated successively. If the same variable split exists in both the parent node and the child node, a node merger operation should be done. Consequently, a multi-branch node of the tree is constructed. An example of multi-branch splitting for the CART model is shown in Fig. 3.

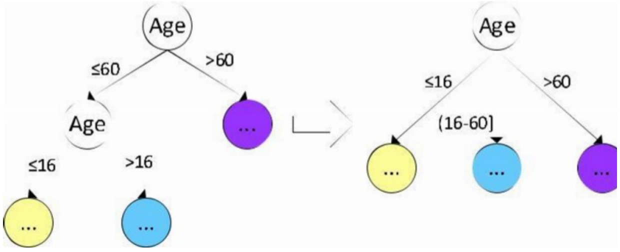


Fig 3. Example of multi-branch splitting for the CART model.

Repeat steps (1 - 3) until the data in each branch are classified in one class as a leaf node.

iv) Calculate Mean Value Of Leaf Nodes After Removal Of Noisy Data

Although we have removed part of the error data in the preprocessing, other types of noisy data mentioned above might exist. Therefore, the third optimization aspect of the RF algorithm is to reduce the influence that the noisy data have on the algorithm accuracy. A boxplot-based noise removal method is performed in the value calculation of each CART leaf node.

The data in the current leaf node are sorted in ascending order. Then, the values of three data points Q1, Q2, Q3 of the box-plot model are calculated, where Q2 is the median data point and Q1 and Q3 are the lower and upper four digits of the data, respectively. The inner limit of the noisy data is denoted as follows:

$$IL = Q1- 1:5(IQR) = Q1- 1:5(Q3-Q1): \qquad (5)$$

The outer limit of the noisy data is denoted as follows:

$$OL = Q3+1:5(IQR) = Q3+1:5(Q3\quad Q1): \qquad (6)$$

The data outside the range of {IL; OL} are removed as noisy data. After removing the noisy data, the average value c_j of the data y_j is calculated in each leaf node of the regression tree. This splitting process is repeated until all of the feature values are generated. A CART regression tree for the training subset S_{traini} is trained, and the tree model is denoted as follows:

$$hi(x, \Theta j)= \sum_{n=1}^{N} c_n I(x\in R_n) : \qquad (7)$$

where N is the number of leaf nodes of the tree, 2j is the target feature variable, and I ( ) is an indicator function. A meta CART regression tree of the PTTP model is shown in Fig. 4.

v) Calculate The Accuracy Of Each Tree

After each regression tree of the training subset S_{traini} is built, the testing subset S_{OOBi} is used to calculate the accuracy of the meta-classi er tree. The accuracy of a metaclassifier tree refers to the ratio of average number of votes in correct classes to all of the error classes, which are classified by the trained meta-classifier tree.

The PTTP model based on the random forest algorithm is shown in Fig. 5.

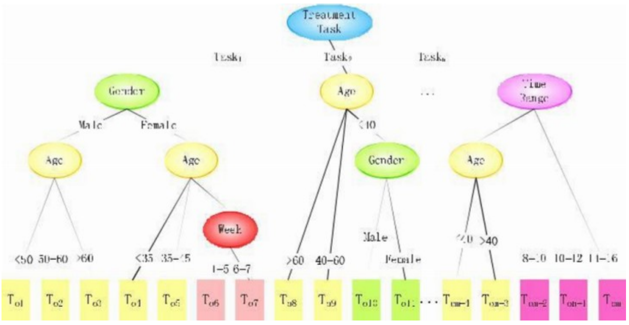


Fig 4. Meta CART tree of the PTTP model

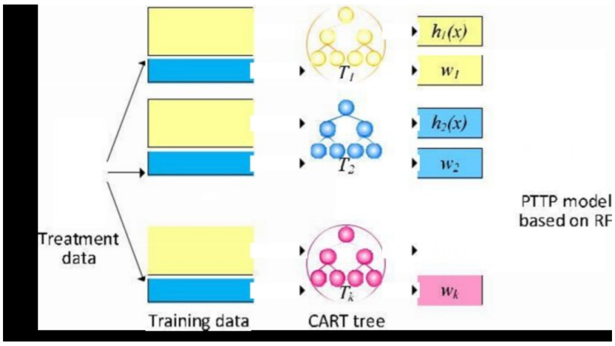


Fig 5. PTTP model based on the RF algorithm

c) *Hospital Queuing Recommendation System Based On PTTP Model*

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 intelligent triage.

Assume that there are various treatment tasks for each patient according to the patient's condition, such as examinations and inspections. Let Tasks = {Task1; Task2; : : : ; Taskn} be a set of treatment tasks that the current patient must complete, and let Ui ={ Ui1; Ui2; : : : ; Uim} be a set of patients in waiting the queue for $Task_i$. The process of the HQR system based on the PTTP model is shown in Fig. 6.

PTTP model according to the patient's characteristics (such as gender and age), time factors (such as the week and month of the current time), and other factors (such as treatment departments, available machines, and service windows). The patient treatment time consumption T_{ik} of patient U_{ik} in queue is denoted as follows:

$$T_{ik} = H\ (X_{ik}; \Theta j) \qquad (8)$$

$$=\frac{1}{k}\sum_{i=1}^{k} CA_i * h_i(x, \Theta j)] \qquad (9)$$

where X_{ik} is the treatment data of patient U_{ik} , Θj is all of the independent variables of X_{ik} , CA_i is the accuracy weight of tree hi, and hi(x; Θj) is a result of patient treatment time consumption predicted by a single CART regression tree.

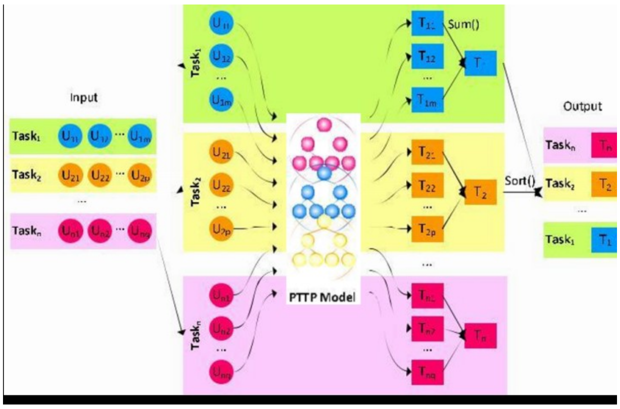


Fig 6. Process of the HDR system based on the PTTP model

Then, all of the predicted patient treatment time consumption of patients in the queue is added to obtain the waiting time of Task$_i$, which is denoted as T$_i$. The calculation formula of Ti is denoted as follows:

$$Ti = \frac{1}{Wi}\sum_{k=1}^{m} T_{\text{ik}} \qquad (10)$$

where W$_i$ is the number of service windows or workbenches that can provide a service for treatment task Task$_i$ in parallel, m is the number of patients waiting in the queue of Task$_i$, and T$_{ik}$ denotes the predicted waiting time for the patient-in-waiting Patient$_k$.

D. Parallel Implementation Of The Hqr System

Usually, there are a number of treatment tasks for each patient, and many patients waiting in the queue of each treatment task. Therefore, a parallel HQR system is implemented for each patient if there is more than one treatment task for the patients. The process of the parallel HQR system is shown in Fig. 7.

Assume that there are n treatment tasks for the current patient to complete and that there is a number of patients waiting in the queue of each treatment task. In the parallelization solution, n RDD objects are created to refer to the n treatment tasks. There is a number of partitions in each RDD object that refer to patients waiting in the queue of each task. Let partition Uij be the jth patient waiting for the ith treatment task.

S of the patient might generate in the ith task, as predicted by the trained PTTP model. In this step, the time consumption for each patient Uij is calculated with the k trained CART trees of the RF-based PTTP model in a shuffle() function, and the predicted patient treatment time consumption Tij is derived.

Step 2: The patient treatment time consumption of all ofthe patients in each task is added in a sum() function, and the predicted waiting time T i of each task is obtained. An RDD object (Taski; Ti) is created for each task.

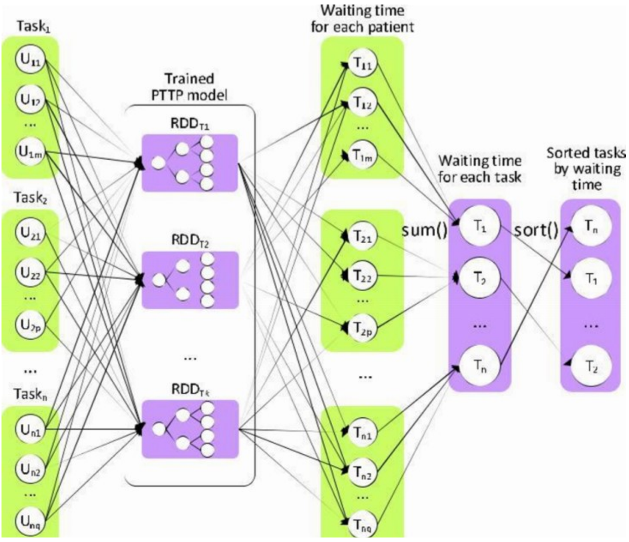


Fig 7. Parallelization recommendation process of the HQR system

Step 3: The predicted waiting times for all of the tasks for the current patient are sorted in ascending order with a sort() function. A new RDD object Ts is created to save the sorted waiting times of all of the treatment tasks. Hence, the parallel hospital queuing recommendation schema for the current patient is performed.

a) Average Waiting Time For Patients

To evaluate the efficiency of our HQR system, various experiments about average waiting time for patients in the with-HQR case with that in the without-HQR case are performed. Each case is under the treatment data with 5000 patients and 20,000 treatment records. We accounted and compared the average waiting time of patients in the with HQR case with that in the without-HQR case. The results of comparison are presented in Fig. 13.

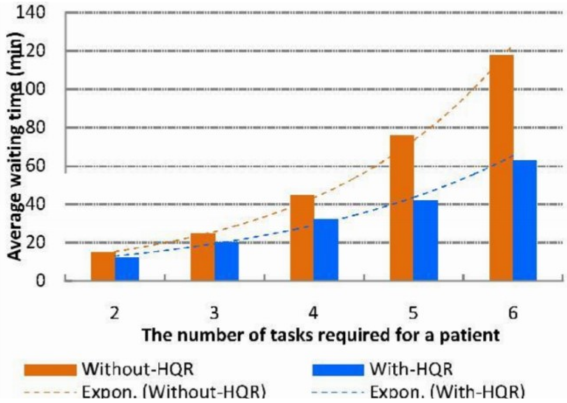


Fig 8. Average waiting time for patients.

It is easy to observe from Fig. 13 that the advantage of the average waiting time of patients in cases of with-HQR is greater than in cases of without-HQR. Moreover, the more patients treatment tasks are, the more obvious is for this advantage. When the number of tasks required for each patient is equal to 2, the average waiting time of each patient is approximately 15 min in the without-HQR case (the original case), while 12 min in the with-HQR case. When there are 6 treatment tasks required for each patient, the average waiting time is approximately 118 min in the former case, while 63 min in the latter case.

III. CONCLUSION

In this paper, a PTTP algorithm based on big data and the Apache Spark cloud environment is proposed. 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. Extensive experiments and application results show that our PTTP algorithm and HQR system achieve high precision and performance.

Hospitals' data volumes are increasing every day. The workload of training the historical data in each set of hospital guide recommendations is expected to be very high, but it need not be. Consequently, an incrementalPTTP algorithm based on streaming data and a more convenient recommendation with minimized pathawareness are suggested for future work.

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. WorldWide 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. Comput.Sci., vol. 1, no. 2, pp. 205 220, Jun. 2011.

[4] G. Chrysos, P. Dagritzikos, I. Papaefstathiou, and A. Dollas, ``HCCART: 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.

[5] 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.

[6] Y. Ben-Haim and E. Tom-Tov, ``A streaming parallel decision tree algorithm,'' J. Mach. Learn. Res., vol. 11, no. 1, pp. 849 872, Oct. 2010.

[7] L. Breiman, ``Random forests,'' Mach. Learn., vol. 45, no. 1, pp. 5 32, Oct. 2001.

[8] G. Yu, N. A. Goussies, J. Yuan, and Z. Liu, ``Fast action detection via discriminative random forest voting and top-K subvolume search,'' IEEETrans. Multimedia, vol. 13, no. 3, pp. 507 517, Jun. 2011.

[9] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes, ``Robust and accurate shape model matching using random forest regression-voting,'' IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1862 1874,Sep. 2015.

[10] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, ``Big data analytics framework for peer-to-peer botnet detection using random forests,'' Inf.Sci., vol. 278, pp. 488 497, Sep. 2014.

[11] S. Bernard, S. Adam, and L. Heutte, ``Dynamic random forests,'' PatternRecognit. Lett., vol. 33, no. 12, pp. 1580 1586, Sep. 2012.

[12] H. B. Li, W. Wang, H. W. Ding, and J. Dong, ``Trees weighting random forest method for classifying high-dimensional noisy data,'' in Proc. IEEE7th Int. Conf. e-Business Eng. (ICEBE), Nov. 2010, pp. 160 163.

[13] G. Biau, ``Analysis of a random forests model,'' J. Mach. Learn. Res., vol. 13, no. 1, pp. 1063 1095, Apr. 2012.