# A Parallel Patient Treatment Time Prediction Algorithm and Its Applications in Hospital Queuing-Recommendation

Suraj Bawankar[1], Dheerajkumar Pandey[2], Prince Rathore[3], Rajkush[4], A.A. Bamanikar[5]
[1,2,3,4] *Student, P.D.E.A.'s COE Manjari (Bk.), Pune, Maharashtra*
[5]*Professor, P.D.E.A.'s COE Manjari (Bk.), Pune, Maharashtra*

***Abstract*- Effective patient queue management to chop back patient wait delays and patient overcrowding is one altogether the foremost challenges featured by hospitals. In essential and annoying waits for long periods result in substantial human resource and time wastage and increase the frustration endured by patients. For each patient at intervals the queue, the complete treatment time of all the patients before him is that the time that he ought to wait. would possibly it'd be convenient and fascinating if the patients might receive the foremost economical treatment organize and perceive the expected waiting time through a mobile application that updates in real time. Therefore, we've an inclination to propose a Patient Treatment Time Prediction (PTTP) recursive to predict the waiting time for every treatment task for a patient. We've an inclination to use realistic patient information from varied hospitals to urge a patient treatment time model for each task. Supported this large-scale, realistic data-set, the treatment time for every patient at intervals the current queue of each task is foreseen. Sustained the expected waiting time, a Hospital Queuing Recommendation (HQR) system is developed. HQR calculates Associate in Nursing predicts a cost-effective and convenient treatment established steered for the patient.**

***Index Terms*- Patient Treatment Time Prediction (PTTP), Hospital Queuing Recommendation (HQR), Random Forest (RF).**

## I. INTRODUCTION

Patient queue management and wait time prediction kind a tough and complex job as a results of each patient may would like whole totally different phases/ operations, sort of an examination, various tests, e.g., a sugar level or biopsy, X-rays or a CT scan, minor surgeries, throughout treatment. Each treatment task can have varied time requirements for each patient that makes time prediction and recommendation extraordinarily tough. Variety of the tasks square measure freelance, whereas others might have to attend for the completion of dependent tasks. Most patients ought to stay up for unpredictable but long periods in queues, waiting their communicate accomplish each treatment task.

In this paper, we've a bent to focus on serving to patients complete their treatment tasks terribly} very certain time and serving to hospitals schedule each treatment task queue and avoid overcrowded and ineffective queues. We tend to use huge realistic data from various hospitals to develop a patient treatment time consumption model. The realistic patient data square measure analyzed rigorously and strictly supported necessary parameters, like patient treatment begin time, end time, patient age, and detail treatment content for each whole totally different task. We've a bent to work out and calculate whole totally different waiting times for numerous patients supported their conditions and operations performed throughout treatment.

## II. PROBLEM STATEMENT

Prediction based on analysis and process of massive noisy patient information from varied hospitals could be a difficult task .Most of the information in hospitals are massive, unstructured, and high dimensional. Hospitals that contain an excellent deal of information, like patient data, medical activity data, time, treatment department, and detailed information of the treatment task. Because of the manual operation and varied unexpected events throughout treatments, a large quantity of incomplete

or inconsistent information seems, like a lack of patient gender and age data, time inconsistencies caused by the time zone settings of medical machines from completely different manufacturers, and treatment records with only a start time but no end time.

## III. LITERATURE REVIEW

The first paper [1], A new algorithm for incremental construction of binary regression trees is presented. This algorithm, called SAIRT, adapts the induced model when facing data streams involving unknown dynamics, like gradual and abrupt function drift, changes in certain regions of the function, noise, and virtual drift. It also handles both symbolic and numeric attributes. The proposed algorithm can automatically adapt its internal parameters and model structure to obtain new patterns, depending on the current dynamics of the data stream. SAIRT can monitor the usefulness of nodes and can forget examples from selected regions, storing the remaining ones in local windows associated to the leaves of the tree. On these conditions, current regression methods need a careful configuration depending on the dynamics of the problem. Experimentation suggests that the proposed algorithm obtains better results than current algorithms when dealing with data streams that involve changes with different speeds, noise levels, sampling distribution of examples, and partial or complete changes of the underlying function.

The second paper [2], Gradient Boosted Regression Trees (GBRT) is the current state-of-the-art learning paradigm for machine learned web search ranking — a domain notorious for very large data sets. In this paper, we propose a novel method for parallelizing the training of GBRT. Our technique parallelizes the construction of the individual regression trees and operates using the master-worker paradigm as follows. The data are partitioned among the workers. At each iteration, the worker summarizes its data-partition using histograms. The master processor uses these to build one layer of a regression tree, and then send this layer to the workers, allowing the workers to build histograms for the next layer. Our algorithm carefully orchestrates overlap between communication and computation to achieve good performance. Since this approach is based on data partitioning, and requires a small amount of communication, it generalizes to distributed and shared memory machines, as well as clouds. We present experimental results on both shared memory machines and clusters for two large scale web search ranking data sets. We demonstrate that the loss in accuracy induced due to the histogram approximation in the regression tree creation can be compensated for through slightly deeper trees. As a result, we see no significant loss in accuracy on the Yahoo data sets and a very small reduction in accuracy for the Microsoft LETOR data. In addition, on shared memory machines, we obtain almost perfect linear speed-up with up to about 48 cores on the large data sets. On distributed memory machines, we get a speedup of 25 with 32 processors. Due to data partitioning our approach can scale to even larger data sets, on which one can reasonably expect even higher speedups.

## IV. ALGORITHM

### A. Algorithm

Input:

STrain: the training datasets;

k: the number of CART trees in the HQR model.

Output:

PTTP : The HQR system model based on PTTP algorithm

Step1: for i <- 1 to k do

Step 2: create training subset strain <- sampling(STrain);

Step 3: create OOB subset sOOBi <-(STrain - straini);

Step 4: create an empty CART tree hi;

Step 5: for each independent variable *yj* in *straini* do

Step 6: calculate candidate split points vs <- yj;

Step 7: for each *vp* in *vs* do

Step 8: calculate the best split point (yj,vp) <- arg $\min\sum_{x\varepsilon R_L} (y_i - c_L)^2 + \sum_{x\varepsilon R_R} (y_i - c_R)^2$];

Step 9: end for

Step 10: append node Node(yj,vp) to hi;

Step 11: split data for left branch $R_{L(yj,vp)}$ <- {x|yj ≤ vp};

Step 12: split data for right branch $R_{R(yj,vp)}$ <- {xjyj > vp};

Step 13: for each data R in {RL(yj;vp); RR(yj;vp)} do

Step 14: calculate 8(vpLjyj) maxi 8(vijy);

Step 15: if ($\phi(vp_{(L|R)}|yj) \geq \phi(vp|yj)$) then

Step 16: append subnode Node(yj,$vp_{(L|R)}$) to Node(yj,vp) as multi-branch;

Step 17: split data to two forks $R_L(yj,vp_L)$ and $R_R(yj,vp_R)$;
Step 18: else
Step 19: collect cleaned data for leaf node Dleaf <- (IL ≤ yj ≤OL);
Step20: calculate mean value of leaf node c 1/K $\sum D_{leaf}$ ;
Step21: end if
Step22: end for
Step23: remove *yj* from *straini*;
Step24: end for
Step25: calculate accuracy CAi <-I (hi(x)=y)/I (hi(x)=y)+$\sum$I (hi(x)=z) for hi by testing sOOBi;
Step26: end for
Step27: PTTP H(X,ɵj) <-1/k$\sum_i^K$ <- [CAi × hi];
Step28: return PTTP.

B. BLOCK DEIAGRAM OF SYSTEM

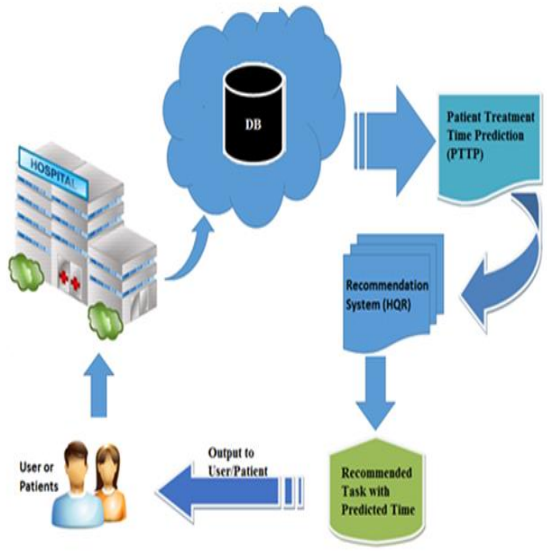


Figure 4.1.Block diagram

In this paper, a Patient Treatment Time Prediction (PTTP) model is trained supported hospitals' historical information. The waiting time of every treatment task is expected by PTTP that is that the ad of all patients' waiting times within the current queue. Then, consistent with every patient's desired treatment tasks further as schedule, a Hospital Queuing-Recommendation (HQR) system recommends Associate in Nursing economical and convenient treatment set up with the less waiting time for the patient/ user.

HARDWARE REQUIREMENT

| | |
|---|---|
| SYSTEM | : Pentium IV 2.4 GHz |
| HARD DISK | : 40 GB |
| FLOPPY DRIVE | : 1.44 MB |
| MONITOR | : 15 VGA colour |

SOFTWARE REQUIREMENTS
Operating System
Windows XP Professional
Java
Code is written in Java. The recommended Java version is JDK 1.6 release and the recommended minimum revision is 31 (v 1.6.31).
Backend
MySQL database

## V. ADVANTAGES

- Decrease the patients waiting time.
- In this technique, we've a tendency to specialize in helping patients complete their treatment tasks throughout a predictable time and helping hospitals schedule each treatment task queue and avoid overcrowded and ineffective queues.
- To improve the accuracy of the information analysis with continuous options, varied improvement strategies of classification and regression algorithms are proposed.

## VI. APPLICATION

- We will use this system in bus stations for tickets
- ATM's to predict the waiting time to user.
- For railway reservation system

## VII. CONCLUSION AND FUTURE SCOPE

In this project, a PTTP rule supported giant data then the Apache Spark cloud setting is planned. A random forest optimization rule is performed for the PTTP model. The queue waiting time of every treatment task is anticipated supported the trained PTTP model. A parallel HQR system is developed and an inexpensive and convenient treatment arranges is typically recommended for every patient. Intensive

experiments and application results show that our PTTP formula and HQR system succeed high preciseness and performance.

## VIII. ACKNOWLEDGMENT

Authors wish to acknowledge Principal, Head of department and guide of their project for all the support and help rendered. To express profound feeling of appreciation to their regarded guardians for giving the motivation needed to the finishing of paper.

## REFERENCES

[1] 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. 488497, Sep. 2014.

[2] S. Meng, W. Dou, X. Zhang, and J. Chen, ``KASR: A keyword-aware service recommendation method on MapReduce for big data applications,''IEEE Trans. Parallel Distrib. Syst., vol. 25, no. 12, pp. 32213231, Dec. 2014.

[3] 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. 387396.

[4] R. Fidalgo-Merino and M. Nunez, ``Self-adaptive induction of regression trees," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 8, pp. 16591672, Aug. 2011.

[5] G. Yu, N. A. Goussies, J. Yuan, and Z. Liu, ``Fast action detection via discriminative random forest voting and top-K sub volume search," IEEE Trans. Multimedia, vol. 13, no. 3, pp. 507517, Jun. 2011.

[6] N. Salehi-Moghaddami, H. S. Yazdi, and H. Poostchi, ``Correlation based splitting criterion in multi branch decision tree," Central Eur. J. Comput. Sci., vol. 1, no. 2, pp. 205_220, Jun. 2011.

[7] G. Chrysos, P. Dagritzikos, I. Papaefstathiou, and A. Dollas, ``HC-CART: A parallel system implementation of data mining classi_cation 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.

[8] 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.

[9] 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.

[10] 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.

[11] L. Breiman, ``Random forests," Mach. Learn., vol. 45, no. 1, pp. 5_32, Oct. 2001.

[12] A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment Jianguo Chen, Kenli Li, Senior Member, IEEE, Zhuo Tang, Member, IEEE, Kashif Bilal, Shui Yu, Member, IEEE, Chuliang Weng, Member, IEEE, and Keqin Li, Fellow, IEEE, 1045-9219 (c) 2016 IEEE.

[13] Srinivasan K. Department of Electronics and Instrumentation Engineering Sri Ramakrishna Engineering College Coimbatore, India hod-eie@srec.ac.in