INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

IJARIIT

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 3)
Available online at: www.ijariit.com

A parallel patient treatment algorithm and it's application in hospital queuing recommendation

Milan Bekal
milanbkl@gmail.com
BMS College of Engineering, Bengaluru, Karnataka

Abishek Manikandan
abishek.mani@gmail.com
BMS College of Engineering, Bengaluru, Karnataka

Ayushman Tewari
ayushman.kishu.tewari@gmail.com
BMS College of Engineering, Bengaluru, Karnataka

Pallavi G. B.
pallavi.cse@bmsce.ac.in
BMS College of Engineering, Bengaluru, Karnataka

ABSTRACT

A patient queue management system to minimize patient wait time, delays and patient overcrowding is one of the major challenges faced by hospitals. Redundant and bothersome waits for long periods result in substantial human resource and time wastage and increase the frustration endured by patients. Therefore, we recommend a Patient Treatment Time Prediction (PTTP) algorithm to predict the waiting time for each treatment task for a patient. Based on the predicted waiting time, a Hospital Queuing-Recommendation (HQR) system is developed. HQR calculates and predicts an efficient and suitable treatment plan recommended for the patient.

Keywords— *PTTP, HQR, Time management, Regression tree algorithm, Random forest algorithm*

1. INTRODUCTION

Currently, most hospitals are congested and lack effective patient queue management. Patient queue management and wait time prediction form a challenging and intricate job because each patient might require different phases of operations, such as a regular check-up, 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 a varying time requirement for each patient, which makes time prediction and recommendation highly intricate. A patient is usually required to undergo examinations, inspections or tests (referred 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 irregular but long periods in queues, waiting for their turn to complete each treatment task. We recommend a Patient Treatment Time Prediction (PTTP) algorithm to predict the waiting time for each treatment task for a patient. Based on the expected waiting time, a Hospital Queuing Recommendation (HQR) system is developed. HQR calculates and predicts an effective and suitable treatment plan recommended for the patient. The front end of this application is a UI which is manageable to patients in the form of a webpage. The data (account details) regarding the patient wait time and the number of tasks to be completed is displayed to the user on this webpage.

2. LITERATURE SURVEY

Most hospitals are currently congested and lack effective queue management for patients. Patient queue management and wait time prediction is a challenging and intricate job as each patient may require different phases of surgery, such as regular check-ups, various tests, such as sugar or blood tests, X-rays or CT scans, minor surgery, during treatment. In this paper, we refer to each of these phases/operations as treatment tasks or tasks. For each patient, each treatment task may require varying time, making time prediction and recommendation highly complex. A patient is usually required to undergo (referred to as tasks) examinations, inspections or tests to his condition. In such a case, each patient may need more than one task. Some of the tasks are independent, while others may have to wait for dependent tasks to be completed. Most patients have to wait in queues for irregular but long periods to complete each treatment task, waiting for their turn. A Patient Treatment Time Prediction (PTTP) algorithm is recommended to predict a patient's waiting time for each treatment task. A Hospital Queuing Recommendation (HQR) system is developed based on the expected waiting time. HQR calculates and predicts the patient's recommended effective and appropriate treatment plan. This application's front end is a UI that can be managed in the form of a webpage for patients. The patient waiting for time data (account details) and the number of tasks to be completed will be displayed on this website to the user.

As we were able to use our queue management systems to implement our web application. This helps you achieve a unified patient flow from an initial check-out entry point. The system allows process efficiency to be enabled by improving patient experience and enhancing overall operational excellence at healthcare facilities. This is one of the most capable business technologies expected to bring many positive changes to the Hospital queue management system world. Benefits:

- Management of patient queues more proficiently and appropriately.
- The system allows real-time queue status information
- Reduces patient wait time and service time substantially
- Makes service areas less crowded with patients
- Clearly shows the length of wait of each patient
- Real-time reporting allows effective management of hospital services.

This web-based system enables patients to join a queue readily and online check the status of the queue. [1] R. Fidalgo-Merino and M. Nunez et al mentioned different methods of classification and regression algorithms are proposed to increase the accuracy of data analysis with continuous features. Incremental construction of binary regression trees was presented with a self-adaptive induction algorithm. This brings us to reliability. Introduced a parallel boosted regression tree algorithm for web search ranking, the system is fully armed [2] S. Tyree K. Q. Weinberger et al, their research found the basis of a correlation-splitting criterion, a multi-branch decision tree algorithm was proposed. Other improved methods for tree classification and regression were proposed. [4]G. Chrysos, P. Dagritzikos et al, and [5] N. T. Van Uyen and T. C. Chung et al, [6] Y. Ben-Haim and E. Tom-Tov. The random forest algorithm [7] and [8] G. Yu, N. A. Goussies, J. Yuan, and Z. LiuL. Breiman et al described a collective classification algorithm based on a decision tree, a suitable algorithm for large data mining. The random forest algorithm is widely used in many fields such as quick action recognition through discriminative random forest voting and search for top-K sub-volume. A strong and accurate model with random forest regression vote. [9] C. Lindner, P. A. Bromiley, M. C. Ionita, and T. F. Cootes used a large data analytical framework for peer-to-peer botnet detection using random forests[10] K. Singh, S. C. Guntuku, A. et al, introduced the new results in these papers to show that the random forest algorithm is effective and applicable. In order to improve the correctness of the random forest algorithm, [11] Bernard et al, proposed a dynamic training method. [12]H. B. Li, W. Wang, H. W. Ding, and J. Dong et al, it was proposed to classify high-dimensional noisy data by a random forest method based on weighted trees. However, an outdated direct voting method is used in the voting process by the original random forest algorithm. In such a case, the random forest with noisy decision trees would likely result in the testing dataset having an incorrect predicted value. [13]G. Biau et al introduced different recommendation algorithms in related fields which were offered and applied. For large data applications, [14] Meng et al., proposed a keyword-aware service recommendation method for Map Reduce. [15] Yang ET. a travel recommendation algorithm was planned to undermine the attributes and travel types of people.[16] Jianguo Chen et al introduced a Bayesian-inference recommendation system for online social networks in which a user distributes a content rating query to his direct and indirect friends throughout the social network. [17]Adomavicius and K et al won introduced new multi-criteria rating system recommendation techniques. Tuzhilin and Adomavicius. [18]G. Adomavicius and A. Tuzhilin et al, Overview of the current generation of recommendation approaches, such as content-based, cooperative and hybrid approaches. However, in the existing studies, there is no operational prediction algorithm for the time consumption of patient treatment. The speed of data mining and analysing big data is a very important factor [19] X. Wu, X. Zhu, and G.-Q. Wu, and W. Ding, to predict the waiting time for each treatment task, we use the random forest algorithm to train the time consumption of patient treatment based on both patient and time characteristics and then build the PTTP model. Since the time consumption of patient treatment is a non-stop variable, the RF algorithm uses a Classification and Regression Tree (CART) model as a meta-classifier. Due to the inadequacies of the original RF algorithm and the characteristics of the patient data, the RF algorithm is improved in four aspects in order to obtain an effective result from large-scale, high-dimensional, non-stop and noisy patient data. [20]J. Dean et al had the idea of PTTP algorithm based on an improved RF algorithm, which has significant advantages in terms of accuracy and performance compared to the original RF algorithm. In addition, there is no existing research and recommendations on hospital queuing management. We are therefore proposing a PTTP-based HQR system. This paper is the first attempt to solve the issue of patient waiting time for hospital queuing service to the best of our knowledge. It is recommended that each patient receive a treatment queuing recommendation with an efficient and appropriate treatment plan and the minimum waiting time

3. 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 about the treatment task. Because of the manual operation and varied unexpected events throughout treatments, a large quantity of incomplete or inconsistent information exists, 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.

4. IMPLEMENTATION

The implementation of this project has the following:

4.1 Patient module

- List of Treatments and corresponding average treatment time should be added in the database.
- Assign the priorities to the treatments. Because some treatments are dependent on the previous treatment results.
- List of treatments and matching average treatment time, based on age and gender should be added in the database. Because treatment time is also dependent on the gender and the age of the patients.
- Patient data will be added to the database through GUI, which includes the patient name, age, gender and treatments.

4.2 Excel data process module

- Excel sheet data has to be prepared, which contains patient id, age, gender and treatments.
- The organized excel sheet is the input for the Parallel Treatment Time Prediction (PTTP) process.
- Excel sheet should be uploaded from the application, this data is read row by row and it is inserted into the database for the further process.

4.3 PTTP process module

- From the input data, which has been inserted from the excel sheet, patient allocated treatments will be taken into concern.
- Based on the priority of the treatment, and the average treatment time and the gender of the patient, the new treatment slot will be recommended.

The trained PTTP model projects the patient treatment time consumption of each patient in the waiting queue. The whole waiting time of each task at the current time can be predicted, such as fTAD 35(min); TBD 30(min); TCD 70(min); TDD 24(min); TE D87(min). Finally, the task of each patient is sorted in 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 fB; D; Eg: - for Patient1, fB; A; C; Eg: - for Patient2, and FD; C; Eg: - 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 suitable way and with the minimum waiting time.

Our process makes use of the Java development platform. The web API can be accessed with a client-server interface. In this project, we make use of the Tomcat server to run Java servlets. These servlets connect the database to the front –end application using HTTP service request.

The next step is to use Microsoft Open Database Connectivity (ODBC) that is a standard programming interface for application providers and database system developers. Using ODBC our Excel spreadsheets and plain text files can be turned into data sources.

Along with that, we make use of JDBC, which we use to connect the SQL database. SQL level database varies as we move from one database to another, but the JDBC will allow any query statement to be passed through it. The JDBC will sit on top of other SQL level APIs.

We use TCP/IP stack to connect the application to the hardware interface. The IP layer will allow the routing of large datagrams by breaking the large datagrams into smaller chunks. Once the network is established, we are good to go. The application software makes use of HTML, CSS, JavaScript and JSPs for the front-end. We make use of the DLL layer to deliver the message from one node to the next.

Decision tree or classification of regression trees is used to create a prediction model for our data set. Decision works on the principle of recursive partitioning. The dataset is divided into subsets by splitting the data one variable at a time. This algorithm is called SAIRT, adapts the induced model when facing data streams involving unknown data streams, like gradual and abrupt function drift. The individual regression trees operate using the master-worker paradigm. 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 sends this layer to the workers, allowing the workers to build histograms for this layer. Our algorithm judiciously orchestrates overlap between communication and computation to achieve respectable performance. In this project, we make use of DT (Decision Tree) map tasks. By using the concept of Entropy and Gini Index as the splitting criterion to build new DT's. A splitting criterion will describe the tree's best splitting variable as well as the variables threshold splitting. By using the classic forward selection method, the variable having the maximum absolute value is chosen as the next best splitting variable at each node. Next, we use the Streaming classification algorithm. This algorithm is designed for classifying large datasets and streaming data. A random forest is built to proficiently generate discriminative votes from individual interest points, and a fast top-K sub volume search algorithm is developed to find all active instances in a single round of search. Without meaningfully demeaning the performance, such a top-K search can be performed on down-sampled score volumes for more effective localization.

5. RESULT

As we were able to implement our web application using our queue management systems. This helps you accomplish a unified flow of patients from an initial entry point to check out. The system allows to enable process efficiency and increase overall operational excellence at healthcare facilities by improving the patient experience. This is one of the most capable business technologies, which is expected to bring many positive changes into the world of Hospital queue management system benefits:

- Management of patient queues more proficiently and appropriately.
- The system allows real-time queue status information
- Reduces patient wait time and service time substantially
- Makes service areas less crowded with patients
- Clearly shows the length of wait of each patient
- Real-time reporting allows effective management of hospital services

5.1 Login page

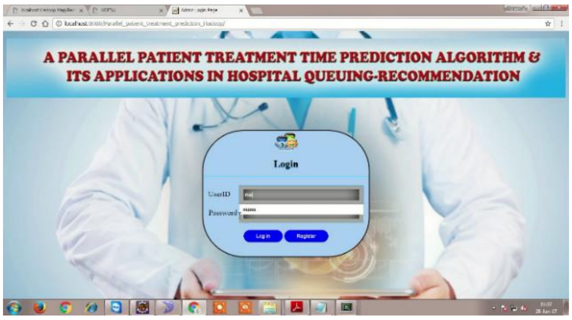

Fig. 1: Login page

5.2 Home page

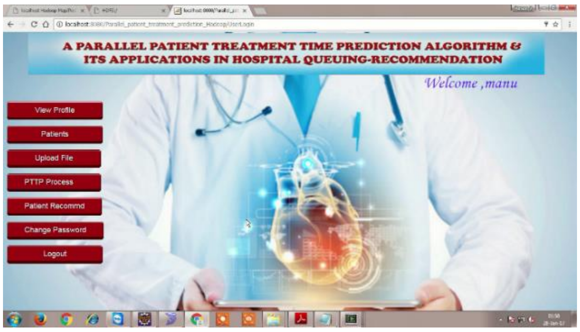

Fig. 2: Home page

5.3 View profile

Fig. 3: View profile

5.4 Edit profile

Fig. 4: Edit profile

5.5 View patient details

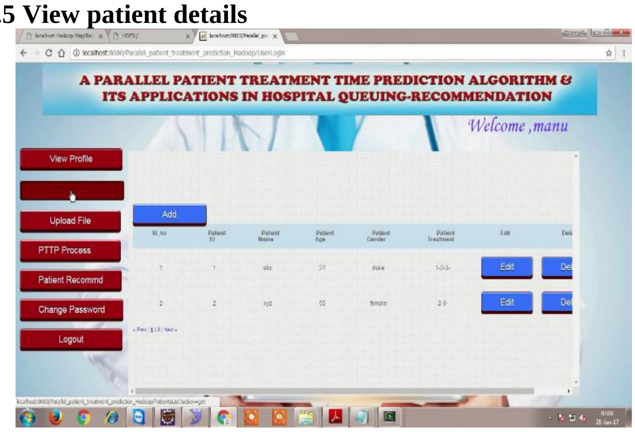

Fig. 5: View patient details

5.6 Add patient details

Fig. 6: Add patient details

5.7 Add patient successfully

Fig. 7: Add patient details

5.8 PTTP process

Fig. 8: PTTP process

5.9 Queuing recommendation

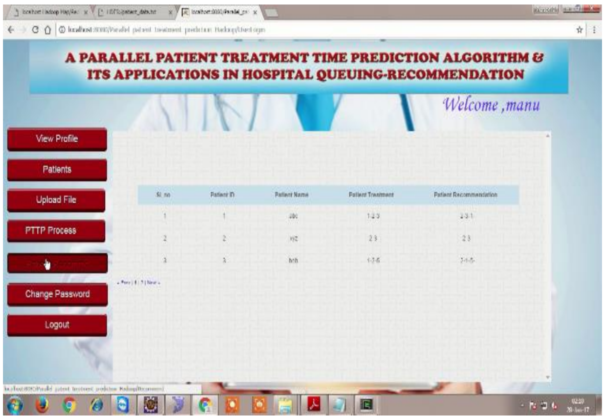

Fig. 9: Queuing recommendation

5.10 Change password

Fig. 10: Change password

6. CONCLUSION

In this project, a PTTP algorithm reinforced large information and so the Apache Spark cloud environment is scheduled. A random forest optimization algorithm is performed for the PTTP model. The queue waiting time of each treatment task is reinforced by the trained PTTP model. A parallel HQR system is developed, and convenient treatment plan is recommended for each patient. Rigorous experiments and application results show that our PTTP algorithmic rule and HQR system succeed with high meticulousness and performance.

7. REFERENCES

[1] Jianguo Chen; Kenli Li; https://ieeexplore.ieee.org/document/7458801
[2] R. Fidalgo-Merino and M. Nunez, ``Self-adaptive induction of regression trees," IEEE Trans. And S. Tyree, K. Q.

Weinberger, K. Agrawal, and J. Paykin, ``Parallel boosted regression trees for Web search ranking''
[3] S. Tyree, K. Q. Weinberger, K. Agrawal, and J. Paykin, ``Parallel boosted regression trees for Web
[4] Search ranking," in Proc. 20th Int. Conf. World Wide Web (WWW), 2012, pp. 387396.
[5] R. Fidalgo-Merino and M. Nunez, ``Self-adaptive induction of regression trees," IEEE Trans.
[6] Pattern Anal. Mach. Intell., vol. 33, no. 8, pp. 16591672, Aug. 2011.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] N. T. Van Uyen and T. C. Chung, ``A new framework for the distributed boosting algorithm," in Proc.
[12] Future Generat. Commun. Netw. (FGCN), Dec. 2007, pp. 420_423.
[13] Y. Ben-Haim and E. Tom-Tov, ``A streaming parallel decision tree algorithm," J. Mach. Learn.
[14] L. Breiman, ``Random forests," Mach. Learn., vol. 45, no. 1, pp. 5_32, Oct. 2001.
[15] A Parallel Random Forest Algorithm for Big Data in a Spark Cloud Computing Environment
[16] Jianguo Chen, Kenli Li, Senior Member, IEEE, Zhuo Tang, Member, IEEE, Kashif Bilal, Shui Yu.
[17] Member, IEEE, ChuliangWeng, Member, IEEE, and Keqin Li, Fellow, IEEE, 1045-9219 (c) 2016
[18] G. Adomavicius, A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions", IEEE Trans.Knowl. Data Eng., vol. 17, no. 6, pp. 734-749, Jun. 2005.
[19] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, "Data mining with big data", IEEE Trans.Knowl.Data Eng., vol. 26, no. 1, pp. 97-107, Jan. 2014.
[20] M. Zaharia et al., "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing", Proc. USENIXNSDI, pp. 1-14, 2012.

