Noname manuscript No.
(will be inserted by the editor)

Machine Learning For Surgical Time Prediction

Oscar Martinez* · **Carol Martinez*** · **Carlos Parra*** · **Saul Rugeles†** · **Daniel Suarez***

Received: date / Accepted: date

Abstract Operating Rooms (ORs) are one of the most expensive services in hospitals. A challenge to optimize the OR efficiency is to improve the surgery scheduling task, which requires the estimation of surgical time duration. It is typically estimated by the surgeon or by a programming department (based on people's experience). These methods usually include bias, such as overestimation of the surgery time, therefore increasing the operational cost of ORs. This paper analyzes a machine learning-based solution for surgical time predictions. We apply and compare four machine learning algorithms (Linear Regression, Support Vector Machines, Regression Trees, and Bagged Trees) to predict the surgical time duration of a tertiary referral university hospital in Bogotá, Colombia. Historical data from 2004 until 2019 was used to train the algorithms. Results show that tree-based algorithms are the ones with the lowest errors. On the other hand, our approach was compared with the manual experience-based method (currently used in the hospital), and results show that the manual method overestimates the surgical time in 82% of the cases.

Keywords Machine Learning · Surgical Time Prediction · Linear Regression · Support Vector Machine · Regression Trees · Assembly Methods

*Pontificia Universidad Javeriana,
Faculty of Engineering, Bogotá, Colombia
Tel.: +123-45-678910
E-mail: d-suarez@javeriana.edu.co
.
†Hospital Universitario San Igancio,
Surgery Department,
Bogotá, Colombia
Tel.: +123-45-678910
E-mail: sjrugeles@husi.org.co

Mathematics Subject Classification (2010) 68T05 · 68Q32 · 68U35 · 90B36

1 Introduction

Surgery service programming consists on assigning time blocks to the surgical procedures and assigning to them rooms available in the surgery service. Most hospitals have a limited quantity of operating rooms (OR) and specialized equipment [1]. This is why surgeries have to be carefully programmed along the day with specific time-blocks to allow the movement of the equipment and medical resources before and after the surgeries. The use of quantitative tools to support decisions is not widespread in hospitals, and the manual experience-based method of programming surgery services prevails [2, 3].

In the state of the art only a few studies have dealt with the surgery scheduling problem using predictive algorithms that minimize the surgical time estimation error. For instance, Master *et al.* [4] showed that Tree-based Machine Learning methods trained with surgeon's estimations of surgeries duration could provide significantly better estimates (60% of average accuracy) of the duration of pediatric surgeries, than averaging historic duration times or surgeons' estimations alone. In the analyzed hospital, surgeons are the ones that request specific time blocks for their surgeries. With that information, the programming department assigns the date, the room, and the time of the surgery.

Davies [5] in his thesis compared four ML algorithms (K-Nearest Neighbors, Linear Regression, Regression Trees, and SVM Regression) to estimate the overall occupation time of the operating room (i.e. the time to move specialized equipment; and anesthesia, surgery,

recovery, and cleaning times). As input, he uses 12 variables such as procedure, OR, team count (number of support staff), specialty, date; and nurse, surgeon, and anesthesiologist that will be involved in the surgery, among other variables. Most of them extracted from the medical history. His results showed that Regression SVM and Regression Trees algorithms have the lowest error in the prediction of operating room's occupation, 40 min and 43 min, respectively.

On the other hand, Fairley *et al.* [6] used a combination of ML estimators and heuristics programs (deterministic optimization) to estimate the occupation in the Recovery Unit of a hospital. The proposed strategy required the estimation of the surgical time, which was estimated using a Regression Tree algorithm. Ten variables were used as input, such as Age, Service, Patient class, Sex, among others. The system was tested on a six month period and results show 65% of accuracy when predicting surgical times. Edelman *et al.* [7] used a Linear Regression approach to predict the surgical time of surgeries developed in six different university hospitals. In their work, they used surgeons' presurgical duration estimates, patient age (coded in 10 year intervals), type of surgery (hospital codes), American Society of Anesthesiologists ASA physical status classification, and type of anesthesia (hospital codes), as input. Their results showed low prediction time errors (approximately 39 min) when only a few variables such as the age of patients, type of anesthesia, and presurgical duration estimates, were used.

Finally, Hosseini *et al.* [8] used a quadratic regression to estimate the surgical time of 15 specialties, using variables extracted from a data mining processing of the hospital's clinical histories. The variables used in that work, correspond to personal data and the patient's medical history, as well as procedures information performed in the hospital. The results show a good approximation of surgical time (RMSE between 31 to 94 minutes), using a Quadratic Regression model and 6 input variables: surgery type (4), procedure (49), ASA physical status classification, patient age, surgery scope (2), specialty.

Research carried out so far shows that machine learning algorithms can be used to predict surgery times. Following this direction, in this paper we present a comparison of three machine learning algorithms (Linear Regression, Support Vector Machines, Regression Trees, and Bagged Trees) to find the best algorithm for predicting surgery time duration for a third level hospital in Bogota, whose programming is based exclusively on experience. Therefore, this paper is an effort towards modeling experience-based criteria for programming surgeries.

The data used in this study correspond to real surgical times acquired throughout records of programmed surgeries of the operating room service from 2014 to 2019. Additionally, different interviews were conducted to identify key features that should be included in the models, which are not clearly found directly from the hospital database. Therefore, the models have been designed to learn the criteria followed by the programming department to define the time blocks of the surgeries. Different tests where conducted to choose the ML algorithm with the best performance based on the Mean Square Error and the training and testing time for the surgical time prediction problem.

The paper is organized as follows. Section 2 defines the surgery scheduling problem and presents the particularities of the process. Section 3 and Section 4 presents the database and explains the feature selection process. Section 5 provides details of the tested algorithms and training results. Finally Sections 6 presents conclusions.

2 The Surgery Scheduling Problem

Scheduling surgeries is a difficult task due to the variability of procedures; and the different people involved in the scheduling process like hospital administrators, patients, surgeons, anesthesiologists, nurses, among others, that could introduce bias and errors, increasing the complexity of surgery time prediction [3, 9]. A good surgery scheduling system should contemplate the previously mentioned factors in addition to complementary tasks (e.g. patient enlistment, instrumentation, cleaning, among others), and should seek for an optimal balance between occupation and service quality. The latter impacts the financial status of hospitals by maximizing the OR usage; and improves patients care by reducing the time between the dates when the surgery is requested and when it is performed [10].

Surgery service programming consists on assigning time blocks to the surgical procedures and assigning to them rooms available in the surgery service. The minimum programming time block represents the average time to prepare the OR. It starts from the moment the room is prepared (e.g. moving specialized equipment), until the patient's anesthesia ends. Each hospital defines its minimum programming time block according to its needs [2].

The study presented in this paper was conducted in an tertiary referral academic hospital with a basic Electronic Health Record (EHR) system. The hospital has an academic component that generates variations in the surgery time that increase the complexity of the problem. In this kind of hospitals, it is common that resident

doctors and students participate in the surgeries. However, in the surgery records, surgeries are commonly assigned to the surgeon professor and the information that indicates the number of students involved in the surgery is not usually present. These kind of situation shows us that the variability of surgery times in academic hospitals is higher, since students have to participate in the procedures as part of their training process, something that affects the duration of the procedures according to their experience.

Surgeries scheduling is conducted by the programming department which assigns time blocks to surgeries, based on experience. The following list describes the surgery programming steps followed by the hospital used in this study.

- Surgeons request time for a surgery. They provide patient ID, patient name, anesthesia type, procedure, required equipment.
- The head of the programming department assigns a time block for the surgery. There are specific time blocks in the morning for specific specialties and others in the afternoon. The day to performs the surgery is confirmed with the patient by a phone call.
- One week before the surgery, at the end of the week, the head of the programming department, the head of supplies, the head of medical resources, the head of nurseries, the head of the floor, the head of the OR, and the head of the cleaning department, get together to adjust the time blocks proposed by the programming department, and to define the final schedule of the surgeries for the next week. Once it is approved, the schedule is uploaded to the system and surgeons are notified
- The day before the surgery, the schedule is analyzed again by the surgery programming assistant and is confirmed with the patients by phone calls. After this, the schedule does not change.
- Finally, the defined time blocks are locked, and the system is updated with the final schedule. At the end of the day, a list with surgeries of the next day are sent to surgeons.

The output of the previously mentioned steps is the schedule of surgeries per OR, i.e. time blocks assigned for each of surgery. A time block (t_block) includes the time required for the anesthesia (t_a), the cleaning time (t_c), and the surgery time (t_s), as shown in Equation 1. In the hospital used in this research, the anesthesia time is a constant of 10 minutes. This time has been estimated based on experience and considering that 90% of the surgeries required general anesthesia. The cleaning time (t_c) is a constant of 25 minutes. This constant was estimated using a study of cleaning times and movement of the cleaning staff inside the hospital. Finally, the surgery time (t_s) is the time the surgeon takes to conduct the surgery (excluding anesthesia).

$$t\_block = t\_a + t\_c + t\_s \tag{1}$$

The programming process in the hospital used in this research depends on the average time duration of the surgeries (t_s). It is estimated based on the experience of the programming department staff according to the type of surgery and the surgeon. For them, information such as gender or patient demographic information are not relevant for scheduling the surgeries.

Therefore, this research contemplates a particular scenario with specific challenges to solve, such as the academic environment of the hospital, the restrictive sensible data usage, and a large number of categorical variables to model the problem, all these requiring a generic solution to solve the hospital's needs. This research aims to predict the surgical time for an academic hospital with an experience-based programming scheme. We use classic ML algorithms allowing traceability in the predictions and compare the ML performance versus the manual method used in the hospital.

3 Database

The database used in this research corresponds to real surgical records from December 2004 to April 2019. The information was extracted from the Electronic Health Record system (EHR) of the hospital.

The EHR has two essential modules for this research, the programming module and the surgical record module. The programming module contains information about patients and procedures, the surgical time, the operating room to develop the procedure, the day for the surgery, the anesthesia type, among others. On the other hand, the Surgical record module includes the surgeons records, surgery duration, and some information about process complications and surgery development. The raw database for the research had 206.587 records of 14 years.

The information per surgery contains patient personal and demographic variables such as home, city of birth, age, gender, health record; and specific information about the procedure and surgery staff such as surgeon, anesthesia type, surgical resources, diagnostic and comments. However, according to the programming department, some information such as gender or patient demographic information, are not relevant when they schedule the surgery.

Most of the information in the surgical database used in the research were categorical variables and represent a high challenge in the surgery scheduling research with ML, since ML algorithms are usually based on mathematical models and, therefore, need numerical variables to operate. Categorical variables do not have the property to be mathematically computable. This challenge was approached in the research with encoding techniques explained in *Surgical Time Prediction* section (Section 3),

3.1 Data Cleaning

Figure 1 describes the data cleaning methodology applied to the dataset. The raw database was an SQL query request from the Oracle Database, with conditions to avoid incomplete records. Different filters were applied to end up only with records that correspond to programmed surgeries with single procedures (82.472 records, i.e. 40% of the raw database).

Additionally, records with atypical surgical times were removed by thresholding the surgery duration field. An atypical record was defined as a prominent deviation in the surgical duration of a surgery, with respect to the mean surgical duration time of the specialty. The threshold was calculated according to Equation 2, where μ represents the mean surgical time of each specialty, and σ represents the deviation in the surgical time for each specialty. We assumed a normal distribution in the data for each specialty. Therefore, records with times beyond three standard deviations are considered atypical. This filter was applied to each specialty independently.

$$thres = \mu \pm 3\sigma \tag{2}$$

After applying the filters, the final database contains 81.248 records of single procedures surgeries from 25 specialties, with 27 fields: patients information like personal ID, age, gender, birthplace, contact number, among others. Also, surgeons information like personal ID, professional ID, profession, home, contact number, associated specialty. And finally the procedure information, such as procedure ID, anesthesia type, OR, patient destination, surgery time programmed, surgery time developed, specialized medical elements, complications record, vital signs notes, anesthesia notes, and post-surgical diagnosis.

.

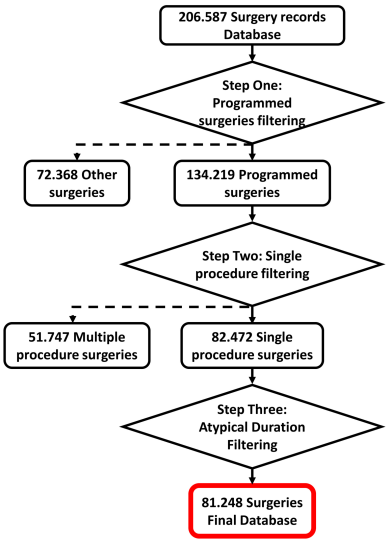

Fig. 1 Data cleaning process. Step One: emergency surgeries were removed (only programmed surgeries were used in the study). Step Two: surgeries with multiple procedure were removed (only surgeries with single procedures are used), Step Three: remove atypical surgical times per specialty, by thresholding. As a result, the database used in the research contains 81.248 records.

3.2 Variable Encoding

Most of the available variables are categorical. To use the categorical variables in the predictive models, it was necessary to code them in such a way that they became computable numerical variables that ML algorithms could interpret. This challenge is prevalent in ML systems, since a lot of information is distributed in categorical variables that are independent of each other and cannot be computed. The techniques used in this research to code those variables were the One Hot Encoding (OHE) and the Sequential Encoding (SE). OHE allows coding categorical variables with independence between them, which is very useful in most cases where it is necessary to avoid correlation biases that the system could learn. This technique could generate dimensional problems when the number of categories to be encoded are very large, since it increases the dimensionality of the problem in the same proportion.

On the other hand, sequential encoding SE codes the categories assigning them a sequential number. This type of coding does not present dimensionality problems but can generate correlation biases of variables, resulting on mathematical relationships that could exist between the coded categories. To understand this technique see [11]. Both methodologies were used to code all categorical variables following the independence criteria below: the variables of procedure, surgeon, specialty, and patient destination were coded using OHE. In con-

trast, the patient life stage variable was coded using SE.

The patient's age variable was coded into six classes according to the life cycle: early childhood, childhood, adolescence, youth, adulthood, and old age [12].

3.3 Variable Selection

Interviews with nurses, surgeons, and administrative programming personal, provided insights on the relevant features required in the study. The variables selected initially were procedure, surgeon, specialty (25 specialties associated with the surgery service), anesthesia type (General, Spinal, Local, Epidural, Sedation and Blocking), patient destination (hospitalized or ambulatory), patient age, and the surgical time. Seven inputs and 1 output (surgical time).

The surgical time is the variable the models will predict. The real time spend in the surgery is provided by the surgeon after the surgery is over. It represents the duration of the procedure from when the surgeon makes the first incision, until the last suture is made and the patient leaves the OR to the recovery room. This variable is filled in the hospital's EHR system at the end of each surgery.

From the 25 specialties, the 80% of the surgeries were developed in nine specific specialties: orthopedics and traumatology, general surgery, gynecology and obstetrics, neurosurgery, urology, plastic surgery, otorhinolaryngology, ophthalmology, and head and neck surgery. For this reason, only those specialties were the ones included in the study.

Apart from those variables, we decided to add a variable called *Surgeon Experience*. This variable represents the number of surgeries performed by the surgeon in the hospital, and was added as a variable to estimate their experience. This extra variable was added to improve the predictive performance of the machine learning system.

An Analysis of Variance (ANOVA) decomposition was performed to determine the relevance of the variables. The null hypothesis was the equality of the means of each variable and their interactions ($p > 0.001$). In this way, the relevance of each variable in the surgical time prediction was evaluated, and all those variables that obtained a p-value below 0.001 were chosen for modeling. From the seven variables initially selected, 5 variables with a p-value under 0.001 were selected. This variables were: procedure, surgeon, specialty, and patient destination (after surgery). Patient age and anesthesia type variables were not relevant to explain the surgical time variability ($p > 0.001$) .

Table 1 shows the final variables chosen and some descriptive information about them, like type of variable, number of levels in the categorical variables, and a short description of the variable.

4 Machine Learning for Surgical Time Prediction

Figure 2 describes the process followed for designing the algorithm for surgical time prediction including the data pre-processing, algorithm selection, and the steps followed to make predictions.

For the design phase, based on the literature mentioned in Sections 1 and 2, we compared three classic prediction algorithms and one assembly method. They were the Linear Regression, Regression Trees, Support Vector Regressors, and Bagging Regression Trees [6–8, 13]. These algorithms are known as supervised machine learning methods [13–15], where *a priori* examples of the inputs and corresponding outputs are required to train them. These algorithms are widely used in different diagnostic and classification tasks. However, some modifications of them are often used for the prediction of continuous variables such as time.

4.1 Machine Learning Algorithms

– **Linear Regression**[14]. It attempts to model the relationship between one or several predictor variables and a response variable. This method aims to predict results on a continuous scale instead of class tags like other supervised classification algorithms. The adjustment model of this algorithm is based on line equation, see equation 3, but some variations allow non-linear models.

$$\begin{aligned} y &= w_0 * x_0 + w_1 * x_1 + \cdots + w_n * x_n \\ &= \sum_{i=0}^{n} w_i * x_i \\ &= W^T * X \end{aligned} \tag{3}$$

– **Regression Trees**[15]. They are decision Trees variation where the response variable is continuous numerical and not categorical. A Regression Tree is recommended when many variables are related to each other in a complicated and non-linear way, where a robust or straightforward regression cannot explain the data variability. This algorithm is also recommended when the problem has many categorical variables, and the numerical methods cannot

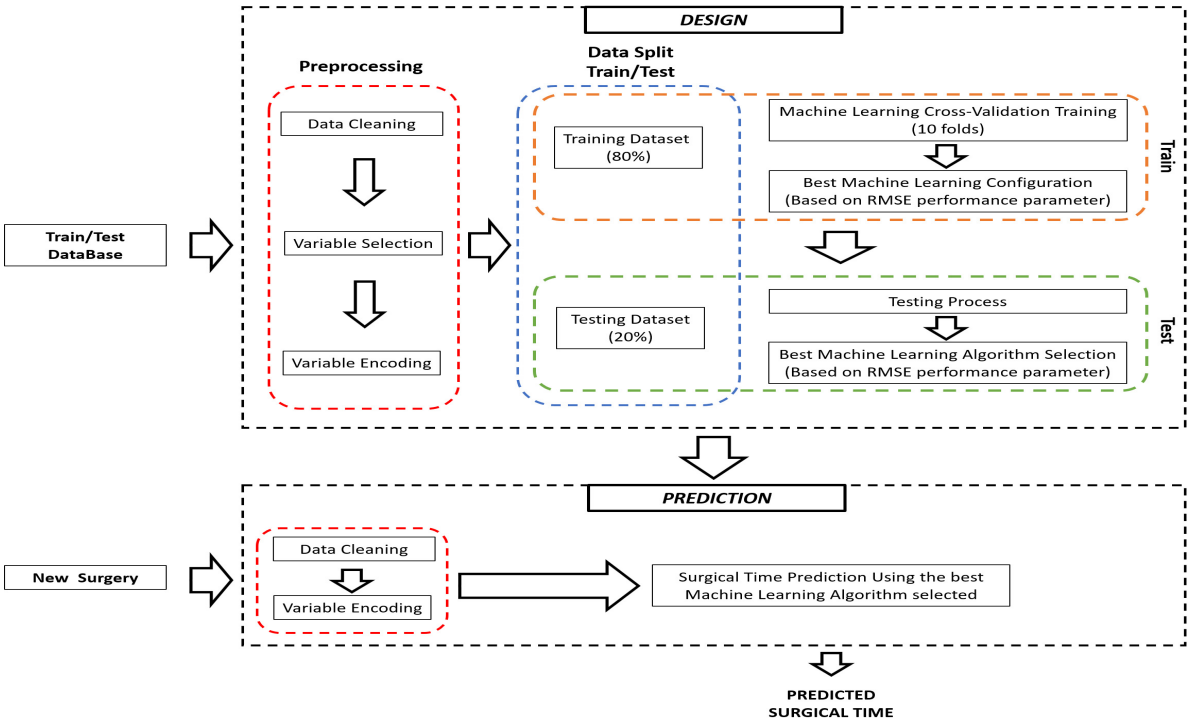

Fig. 2 Surgical Time Prediction Algorithm. Steps followed for the design of the algorithm, including the data pre-processing, algorithm selection, and the steps followed to make predictions.

Table 1 Final variables selected after surgical service staff interviews and ANOVA tests.

VARIABLE	TYPE	CATEGORICAL LEVELS	DESCRIPTION
Procedure	Categorical	~ 1200	Procedure unique code
Surgeon	Categorical	~ 300	Surgeon unique code
Surgeon Experience	Continuous	N/A	Number of surgeries performed by the surgeon in the database
Specialty	Categorical	9	Department responsible for the procedure (eg: Orthopedics)
Patient Destination	Categorical	11	Patient admission class (eg: Hospital, Outpatient)
Patient Life Stage	Categorical	6	Stage of life according to the patient's age
Surgical Time	Continuous	N/A	Duration of the surgery reported at the end of the surgery

model the inputs. The algorithm behind a Regressive Tree is based on subdividing the problem into smaller sets through nodes to reduce the variance of the data. Finally, once the variance of each of the tree branches is stabilized, the output variable is the average of all the samples grouping in each branch individually during the training. The formula that models the response variable in a regression tree is as follows.

$$y = \frac{1}{n} * \sum_{i=0}^{n} y'_i \tag{4}$$

Where y is the response variable modeled for any future observation that meets the conditions under a specific tree branch y'.

– **Bagged Regression Trees**[15]. To improve the performance of algorithms, exist assembly methods that allow grouping many algorithms to model the problem. These assembly methods are divided into Bagging and Boosting techniques. The principal difference between them is the training method to improve performance. In Bagging methods, many algorithms are trained in parallel with a random selection of the variables of the problem, and the general performance is the mean result of the whole group. In Boosting methods, a group of algorithms is trained sequentially, but each step of training adds a new machine learning method only in the worst execution of the previous algorithm, forming a chain of algorithms [15].

From the regression trees, it is possible to make groups of trees allowed a type of Bagging called Random Forest. The algorithm principle is focused on the creation of random trees that, when finished growing, the response variable is the averages the predictions of all trees. This technique generally has a better generalization performance than individual trees, which helps to reduce the variance of the model [14, 15].

– **Support Vector Machines**[15]. This method is widely used in classification. However, it is also possible to use it to make predictions under the same way of Linear Regression. In the SVM case, the error is minimized by the descending gradient optimization algorithm to determine the line that explains the data that not necessarily would be linearly predictable. This fitting is possible because SVM has the attribute to change their workspace to higher dimensions, where a problem that is not linearly predictable can be predictable. This type of algorithm is called "kernel-based" because transformations at higher orders are made using the kernel transformation tool.

4.2 Training Methodology

Each algorithm described above was trained and tested with an 80-20 model, where 80% of the data was used for training, while the remaining 20% was used for testing. The loss function for the ML algorithms and the performance criterion was the Mean Square Error (MSE) of prediction time, see Equation 5. Where N is the number of samples of the problem, y_i is the actual surgery duration, and y'_i is the predicted surgery duration. To maintain the same units of the response variable, we use the RMSE to show the error in minutes. Equation 6 shows the RMSE formula.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y'_i - y_i)^2 \tag{5}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y'_i - y_i)^2} \tag{6}$$

Three tests were elaborated to select the best scenario to apply the ML algorithms. These three test scenarios are described in Figure 3 and correspond to:

1. All the raw data in the database. In this scenario, we try to determine the ML capacity to predict time in a global environment without pre-processing. We use the Table 1 variables with all the categorical levels, meaning the usage of all the 25 specialties recorded in the database, all the surgeons involved throughout the recording time (levels $\sim$ 400), and all the procedures developed recorded in the database (levels $\sim$ 3000).
2. Nine specialties representing 80% of the surgeries of the database. In this scenario, we used the preprocessed data, which means the usage of variables and categorical levels showed in Table 1.
3. Specialties with most of the samples in the database. In this last scenario, the analysis includes only the data related to the Orthopedics and Traumatology specialties, which had 17.336 records, 105 surgeons involved, and 772 procedures associated. Figure 3 presents the way the testing and training processes was carried out in the research.

As mentioned in Section 2), to train and test the ML algorithms, categorical variables were transform into numerical ones. For the Linear Regression and Support Vector Machines algorithms, the One Hot encoding and Sequential encoding were used. While, for the Regression Trees and Bagged Trees, the categorical variables were created internally by the algorithm, before the model was constructed. Regression and Bagged Trees have the attribute to automatically using the categorical variables through implementing discrimination rules for each variable in the nodes of the tree [15].

4.2.1 Hyperparameter selection

Some parameters of each algorithm were evaluated and optimized to find the best configuration that provides the best prediction time. The configuration of hyperparameters improve the performance of the ML algorithms and, in this case, improve the accuracy of the surgical time prediction. To select the parameters, each algorithm was programmed in MatLab® 2018a software installed on a workstation with Windows 10 operating system. The workstation had an Intel® Xeon® CPU E5-2667, 256GB RAM, 700GB HDD and a NVIDIA Quadro K2000 GPU.

Table 2 presents the main hyperparameters that were changed for each algorithm.

Appendix A shows the results obtained adjusting the different hyperparameters per algorithm and the RMSE found in each of the three training scenarios. As a result of the hyperparameter selection, each algorithm ended up with the following configuration:

– Regression: Linear regressive model with Ordinary least squares (OLS) robust fit.
– Support Vector Machine: Quadratic Kernel, Box Constraint equals to 1, Kernel scale equals to 1, half-width of the epsilon-insensitive band equals to 0.1, with standardization and Sequential Minimal Optimization (SMO) routine
– Regression tree: Standard CART Algorithm to select the best split predictor, Merge Leaves method, MSE split criterion without prune, quadratic error tolerance equals to $1 * 10^{-6}$, and minimum leaf size equals to 36 observations

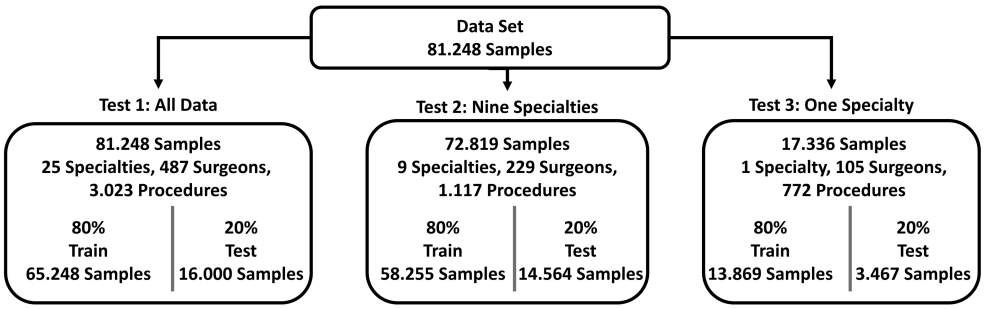

Fig. 3 Test scenarios. The figure shows the number of samples from each of the implemented cases to train/test each of the algorithms. As the scenario becomes more specific, fewer samples were available.

Table 2 Hyperparameters analyzed per algorithm. Some of the hyperparameters stayed on their default values.

ML ALGORITHM	HYPERPARAMETER	VALUE
Linear Regression	*Regresive Model*	Linear Relational Quadratic Relational Quadratic Step by Step
	Robus Fit	OLS Bisquare
Support Vector Machines	*Kernell*	Linear Quadratic Cubic Fine Gaussian Medium Gaussian Thick Gaussian
	Kernell Scale	1
	Epsilon	0.1
	Standardization	Yes No
	Optimization Method	SMO
Regression Trees	*Leaf Merge*	Yes
	Prune	No
	Error Tolerance	$1 * 10^{-6}$
	Surrogates	No
	Minimum Leaf Size	4 12 36
Bagged Trees	*Data Fraction in Bag*	0.1 0.5 1
	Sample Replacement	Yes No
	Method	Regression
	Predictors to Sample	All
	Minimum Leaf Size	4 12 36

– Bagged tree: Regression Bag Ensemble method, number of ensemble learning cycles equal to 100, sample replacement methodology in training process with data fraction of replacement to train the weak learners equal to 1, minimum leaf size equals to 12 observations, and learning rate for shrinkage equals to 1.

5 Results

This section presents the performance of the implemented ML algorithms during training and testing, evaluated using three different input data (three scenarios were tested, all specialties, relevant specialties, and one specialty). The algorithm and scenario with the best performance were selected. The performance of ML-based method for surgical time predictions was compared with the current experience-based method used in the hospital. For this test, a new validation dataset was created, which contains surgeries performed during August and September in 2019.

5.1 Algorithm selection

Table 3 summarizes the performance of the four ML algorithms when estimating surgery times. Each algorithm was tested in three different scenarios (described in Section 3) using as input data all the available specialties (scenario 1), using only the nine relevant specialties (scenario 2), or using one specialty (scenario 3). Table 3 shows the RMSEs (in minutes) obtained during training and testing. Based on the results, Bagged Trees is the algorithm that provides estimations with low RMSEs with the lowest computational cost. For the proposed application, the inference time is a significant parameter, since the hospital environment requires quick results to develop its administrative tasks. The best scenario for Bagged Trees' performance was the nine specialties scenario (relevant specialties). According to the results, Bagged Trees was selected as the best ML algorithm to predict the surgical time, and the final feature matrices were constructed based on procedure, surgeon, specialty, and patient destination (after surgery) as features, using the records of the nine

specialties that included 80% of the database. (orthopedics and traumatology, general surgery, gynecology and obstetrics, neurosurgery, urology, plastic surgery, otorhinolaryngology, ophthalmology, and head and neck surgery) as mentioned in Section 3

Table 3 shows an overall prediction error lower than 40 minutes in training and testing for the three different scenarios that were tested. Each tested scenario has different characteristics. For instance, using many variables could introduce much variability into the model, but it could include an ample feature space to identify and generalize. On the other hand, including a few variables would reduce the algorithms' discriminant power, but it could reduce the computational cost. In that way, a limited feature space based on the specific problem requirements would be the best solution. In this research, the nine-specialty scenario satisfied that purpose. That scenario allows a better performance of all the implemented ML algorithms, and it reduces the prediction error time by a couple of minutes compared to the other tested scenarios. In a hospital environment, a couple of minutes is enough to save a life and for this reason, that scenario was selected to continue with the analysis of the performance of the algorithm.

In the nine specialty scenario, Bagged Trees obtained the best performance with an overall prediction error of about 26 minutes and an inference time of 0.49 minutes. This shows that by using Bagged Trees algorithms the surgeries could be scheduled using tighter time blocks, which could benefits the hospital by reducing the time of unoccupied rooms and allowing better attention to patients.

5.2 ML vs experience-based methods

In this section, the performance of the Bagged Trees algorithm was compared versus the current manual method used in the hospital for surgery scheduling. For this test, a data set of new records acquired between August and September 2019 was used. This data set was extracted from the hospital's HCE system and the same procedures explained in Section 3 were used to clean the data.

These records were filtered from the nine specialties selected based on the scenarios results. The number of records of the data set was 765. The records contain the manual estimated time for scheduling the surgeries, the procedure code, the surgeon ID, the specialty ID, the patient destination (after surgery), and the real time spent in the surgery. The last-mentioned feature is used as ground truth for comparing the results.

As mentioned in the problem section, the hospital's manual method for scheduling surgeries is based on the experience and the surgeries' average execution time. Table 4 shows the RMSEs obtained for the manual method and the ones of the ML-based method. The overall results in the prediction of the nine specialties procedures show that the RMSE of the manual method was 94 minutes on average, while the RMSE of the Bagged Trees was 65 minutes on average.

Table 4 shows the RMSE on each specialty in the nine-specialty scenario, and evidence that the overall RMSE using Bagged Trees is 55% lower than the overall RMSE produced by the manual method.

Bagged Trees model has an RMSE range between 28 minutes and 108 minutes, and the manual method has an RMSE between 34 minutes and 102 minutes. There is a similar range between both. The upper limit in the error range of both models correspond to Neurosurgery specialty predictions. This is a challenging specialty to model because there are different types of complications or factors that increase the variability in the surgery. For instance, in most cases the standard deviation of the surgical time is twice the average time of this specialty. Therefore, if the Neurosurgery specialty is omitted from the predictions, the RMSE of the Bagged Trees model decreases and, in some cases, is equal or better than the manual method, demonstrating the positive impact of this type of algorithm for this application.

Figure 4 shows the behavior of the RMSE for a particular specialty comparing the Bagged Trees and the manual method. In this graph, there is a predominant negative error tendency (82% of the cases) and a reduced positive error tendency (18% of the cases) with the manual method (blue), compared to the light positive error tendency (63% of the cases) with the Bagged Trees model(red).

The behavior of the manual method responds to the way the time blocks are programmed by the programming department. They schedule time in blocks of 60 minutes, adding more time than required to overcome delays and to keep available time in case of complications. This criterion is an excellent way to prevent surgeries standstill in the waiting room. However, according to Table 4, the RMSE shows that the time frame predicted with the experience-based method for unforeseen events are usually up to 50% more of the time needed. This time could be 16% reduced, by using our ML-based method, the Bagged Trees.

Although Bagged Trees has an RMSE better than the manual process, it has also more probability of producing a positive error. It means that it tends to predict less time than the one needed. In the case of the hospital studied in this paper, the optimal requirement would be to avoid the delay in surgery by minimizing errors that imply a lack of time in the prediction. To satisfy

Table 3 Train/Test results of ML algorithms. The training/test errors are in minutes. The best results are in bold.

	ALL SPECIALTIES				NINE SPECIALTIES				ONE SPECIALTY			
Algorithm	**Train RMSE (min)**	**Training Time (min)**	**Test RMSE (min)**	**Testing Time (min)**	**Train RMSE (min)**	**Training Time (min)**	**Test RMSE (min)**	**Testing Time (min)**	**Train RMSE (min)**	**Training Time (min)**	**Test RMSE (min)**	**Testing Time (min)**
Linear Regression	36,60	7881,02	36,74	2,07	34,32	4643,58	30,84	1,71	35,53	3649,27	36,82	1,37
Quadratic SVM	34,81	25275,45	35,47	11,20	32,85	24533,24	30,27	8,01	34,02	2880,85	35,11	2,16
Regression Trees	37,03	2,93	36,82	0,56	34,12	2,18	27,94	0,34	36,11	0,84	36,48	0,18
Bagged Trees	35,85	4,93	35,79	0,71	**33,09**	**3,16**	**26,09**	**0,49**	35,17	1,34	35,75	0,43

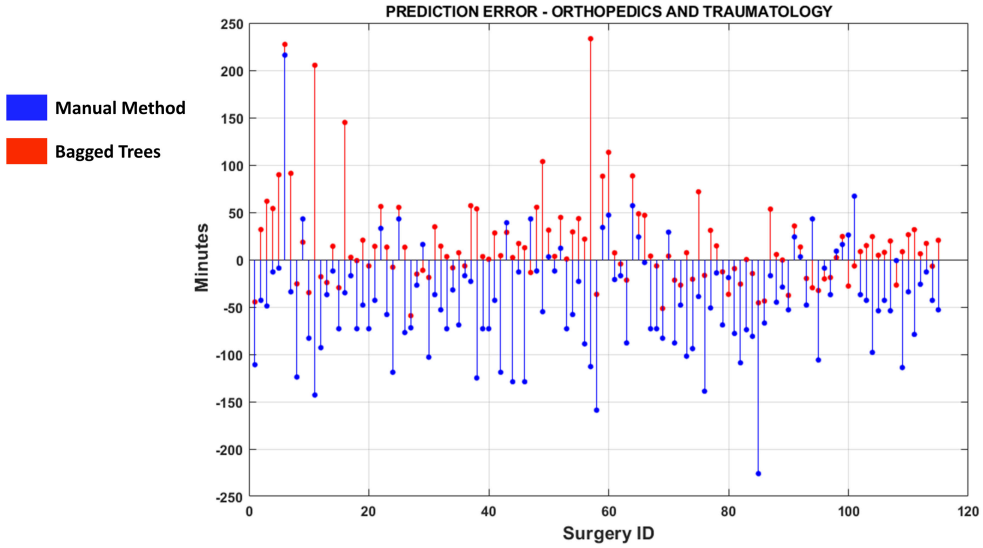

Fig. 4 Comparison of surgery to surgery prediction error of a specific specialty. The graph shows the distribution of the RMSE in the data set used to compare the manual method and Bagged Trees. The tendency to negative error is superior with the manual method, while Bagged Trees has a tendency to positive error.

this requirement, a time frame with a tolerance of 40 minutes could reduce the positive error by 70%, leaving it in a 20% occurrence percentage, which is comparable with the manual method. The 40 minutes of tolerance proposed to add in the Bagged Trees prediction also could be used for contemplate possible complications that requires more time to perform the surgery.

5.3 Discussion

The Bagged Trees model presented in this paper could reduce 35% the minimum time block for scheduling surgeries (from 60 minutes to 40 minutes)

Our results are comparable with the results obtained in the literature that reports reductions in the surgical estimation error between 15% and 40% using tree-based algorithms or linear regression methods [4, 6, 7]. We also obtained a comparable reduction in block-based programming models since the literature reports reductions up to 25% [7], and we achieved a decrease of 35%. It is necessary to clarify that the results in the literature respond to diverse programming environments, such as the use of the surgeon's criterion as an adjustment variable [4, 8], the estimation of the entire occupation time of the surgery room as a unit [5, 6], or the usage of sensitive data like medical records [4–6, 8]. In that way, we do not use any adjustment variable for the prediction based on the specialist's criteria, and we do not use the personal information of the patients and surgeons. We only estimate the surgical time based on the patterns and the basic information in the HCE system, as mentioned in section 3.

Despite the excellent performance of machine learning, there are particularities in the problem that cannot be entirely modeled by Bagged Trees. These particularities could be mitigated by directly intervening in the surgeries scheduling process and improving administrative practices. One of these improvements in the

Table 4 Comparison between the manual method and Bagged Trees. The table presents the RMSE in minutes for comparison in each of the relevant specialties using the validation data set registered during the months of August and September 2019.

SPECIALTY	MANUAL METHOD Test RMSE (min)	BAGGED TREES Test RMSE (min)
Orthopedics and Traumatology	71,69	51,76
General Surgery	76,31	70,46
Gynecology and Obstetrics	49,74	30,06
Neurosurgery	102,63	108,71
Urology	51,77	45,28
Plastic surgery	37,39	68,08
Otorhinolaryngology	69,71	70,93
Ophthalmology	34	38,61
Head and neck surgery	64,34	27,98

process is related to the use of digital tools to avoid loss of information relevant to subsequent studies. This could implies a direct impact on the usability of data since the number of samples and useful records could increase significant.

Finally, According to interviews there are tools to record the relevant information in the hospital, but these steps are often omitted since the tool is not user-friendly and requires much time and unnecessary information that represents an obstacle for the quick daily activities development. In that way, a point of future researches would be to improve the usability criteria by the final users. It would facilitate the administrative work, would prevent the loss of information, and would enhance the quality of service in general.

6 Conclusions

In this paper we have explored different ML algorithms for predicting the surgical time duration. The analysis conducted shows that by using the Bagged of Tree algorithm the error rate in the prediction of surgical time could be reduced by up to 16% and could reduce the minimum programming block up to 35%. It is important to highlight that the good performance of the algorithm is related to the appropriate cleaning and selection of variables processes that were conducted. The ANOVA test helped eliminating redundant variables that can generate noise in the system.

The use of categorical variables is a challenge today in machine learning. There are many techniques to encode those variables but those methods could produce a high dimensionality problem in tasks with many categorical variables and many categorical levels. In this regard, our results show that the prediction of continuous variables, like time, using grouping and discrimination methods, like Bagged Trees, with categorical input variables, has better performance than methods based on mathematical models. We validate that the prediction error can be significantly reduced using ML algorithms, which would imply a better use of physical resources and a possible improvement in surgical service quality.

In future works, Bagged Trees could be implemented in a software tool that could be proved in the programming area. Additionally, we recommend modeling complementary times of the surgeries, like anesthesia time, cleaning time, and recovery time. This to improve the time prediction accuracy and to have a complete model of the hospital environment.

Acknowledgements This research was developed in collaboration with the Hospital Universitario San Ignacio (HUSI), Colombia. We thank the managers, administrative and professional staff who gave us their support to complete the investigation. We also thank the Pontificia Universidad Javeriana for providing us with founding (PPTA 7796) and facilities necessary to carry out the research.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Zhu S, Fan W, Yang S, Pei J, Pardalos PM (2019) Operating room planning and surgical case scheduling: a review of literature. Journal of Combinatorial Optimization 37(3):757–805, DOI https://doi.org/10.1007/s10878-018-0322-6
2. Velasco N, Barrera D, Amaya C (2012) Logística hospitalaria: Lecciones y retos para colombia. La salud en colombia Logros, retos y recomendaciones (1):309–343
3. Litvak E, Long M, Prenney B, Fuda K, Levtzion-Korach O, McGlinchey P (2007) Improving patient flow and throughput in california hospitals operating room services. Guidance document prepared for the California Healthcare Foundation (CHCF)
4. Master N, Zhou Z, Miller D, Scheinker D, Bambos N, Glynn P (2017) Improving predictions of pediatric surgical durations with supervised learning. International Journal of Data Science and Analytics 4(1):35–52

5. Davies SI (2004) Machine learning at the operating room of the future : a comparison of machine learning techniques applied to operating room scheduling. PhD thesis
6. Fairley M, Scheinker D, Brandeau ML (2018) Improving the efficiency of the operating room environment with an optimization and machine learning model. Health Care Management Science pp 1–12, DOI https://doi.org/10.1007/s10729-018-9457-3
7. Edelman ER, van Kuijk SM, Hamaekers AE, de Korte MJ, van Merode GG, Buhre WF (2017) Improving the prediction of total surgical procedure time using linear regression modeling. Frontiers in Medicine 4(JUN)
8. Hosseini N, Sir MY, Jankowski CJ, Pasupathy KS (2015) Surgical duration estimation via data mining and predictive modeling: A case study. AMIA Annual Symposium proceedings AMIA Symposium 2015:640–648
9. Guerriero F, Guido R (2011) Operational research in the management of the operating theatre: A survey. Health Care Management Science 14(1):89–114, DOI https://doi.org/10.1007/s10729-010-9143-6
10. May JH, Spangler WE, Strum DP, Vargas LG (2011) The surgical scheduling problem: Current research and future opportunities
11. Harris DM, Harris SL (2007) Digital Design and Computer Architecture. Morgan Kaufmann
12. Minsalud (2019) Páginas - ciclo de vida. URL https://www.minsalud.gov.co/proteccionsocial/Paginas/cicloVida.aspx
13. Wang Y, Li H, Jiang Y, Dong Y, Shen H, Zhi H, Jiang F, Wang Y, Dong Q, Ma S (2017) Artificial intelligence in healthcare: past, present and future. Stroke and Vascular Neurology 2(4):230–243, DOI https://doi.org/10.1136/svn-2017-000101
14. Raschka S, Mirjalili V (2017) Python Machine Learning : Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow. Packt Publishing
15. Kumar A (2016) Learning Predictive Analytics with Python. Packt Publishing, Birmingham

Appendix A Hyperparameters selection

Hyperparmeter selection of the ML algorithms. Next tables present the results obtained in the hyperparameter fit tests for each test scenario. Results are showed as RMSE error time in minutes for each tuning configuration.

Table 5 Experiments developed to determine the best regression parameters to use.

Regresive Model	Robust Fit	RMSE Test 1	RMSE Test 2	RMSE Test 3
Linear	OLS	***36,33***	***34,32***	***35,53***
	Bisquare	37,60	35.02	35,98
Relational	OLS	Inf	Inf	Inf
	Bisquare	Inf	Inf	Inf
Quadratic	OLS	36,63	34,85	35,55
	Bisquare	37,78	34.93	36,71
Relational Quadratic	OLS	Inf	Inf	Inf
	Bisquare	Inf	Inf	Inf

Table 6 Experiments developed to determine the best SVM regression parameters to use.

Kernell	Standarization	RMSE Test 1	RMSE Test 2	RMSE Test 3
Linear	Yes	39,78	36,98	35,33
	No	41,94	74,56	54,13
Quadratic	Yes	***34,81***	***32,85***	***34,02***
	No	Inf	Inf	Inf
Cubic	Yes	36,49	33,43	34,15
	No	Inf	Inf	Inf
Fine Gaussian	Yes	35,02	35,31	36,99
	No	37,54	40,15	34,32
Medium Gaussian	Yes	37,45	36,36	36,23
	No	36,20	42,09	37,22
Thick Gaussian	Yes	40,14	44,92	43,69
	No	40,89	49,73	45,96

Table 7 Experiments developed to determine the best Regression Trees parameters to use.

Minimum Leaf Size	RMSE Test 1	RMSE Test 2	RMSE Test 3
4	39,61	36,87	38,81
12	37,87	34,74	37,34
36	***37,02***	***34,12***	***36,11***

Table 8 Experiments developed to determine the best Bagged Trees Regression parameters to use.

Data Fraction in Bag	Sample Remplacement	Minimum Leaf Size	RMSE Test 1	RMSE Test 2	RMSE Test 3
0,1	Yes	4	46,02	36,60	37,96
		12	39,55	37,74	38,77
		36	41,09	39,73	39,84
	No	4	38,87	36,59	37,96
		12	39,47	37,57	38,75
		36	41,26	39,75	39,67
0,5	Yes	4	35,99	33,38	35,31
		12	36,23	33,71	35,50
		36	37,14	34,67	36,42
	No	4	35,95	33,37	35,43
		12	36,18	33,52	35,26
		36	37,18	34,40	36,19
1	Yes	4	***35,85***	33,20	35,25
		12	36,02	***33,09***	***35,17***
		36	36,24	33,51	35,45
	No	4	37,99	35,47	35,72
		12	38,46	35,57	35,38
		36	38,81	35,81	35,21