Eline R. Tsai, Derya Demirtas, Nick Hoogendijk, Andrei N. Tintu* and Richard J. Boucherie

Turnaround time prediction for clinical chemistry samples using machine learning

https://doi.org/10.1515/cclm-2022-0668
Received July 11, 2022; accepted September 12, 2022;
published online October 12, 2022

Abstract

Objectives: Turnaround time (TAT) is an essential performance indicator of a medical diagnostic laboratory. Accurate TAT prediction is crucial for taking timely action in case of prolonged TAT and is important for efficient organization of healthcare. The objective was to develop a model to accurately predict TAT, focusing on the automated pre-analytical and analytical phase.

Methods: A total of 90,543 clinical chemistry samples from Erasmus MC were included and 39 features were analyzed, including priority level and workload in the different stages upon sample arrival. PyCaret was used to evaluate and compare multiple regression models, including the Extra Trees (ET) Regressor, Ridge Regression and K Neighbors Regressor, to determine the best model for TAT prediction. The relative residual and SHAP (SHapley Additive exPlanations) values were plotted for model evaluation.

Results: The regression-tree-based method ET Regressor performed best with an R^2 of 0.63, a mean absolute error of 2.42 min and a mean absolute percentage error of 7.35%, where the average TAT was 30.09 min. Of the test set samples, 77% had a relative residual error of at most 10%. SHAP value analysis indicated that TAT was mainly influenced by the workload in pre-analysis upon sample arrival and the number of modules visited.

Conclusions: Accurate TAT predictions were attained with the ET Regressor and features with the biggest impact on TAT were identified, enabling the laboratory to take timely action in case of prolonged TAT and helping healthcare providers to improve planning of scarce resources to increase healthcare efficiency.

Keywords: machine learning; medical diagnostic laboratory; prediction; turnaround time.

Introduction

Turnaround time (TAT) is an essential performance indicator of a medical diagnostic laboratory [1]. Accurate TAT prediction enables laboratories to take timely action in case of prolonged TAT and helps healthcare providers to improve planning of scarce resources to increase healthcare efficiency. Physicians can plan their tasks according to the predicted TAT, while the laboratory receives less inquiries by physicians about availability of the test results, saving time of laboratory technicians.

There are numerous studies on measuring TAT and investigating factors affecting TAT, for example [2–5]. However, we found only one study on predicting TAT for a specific sample, namely the time in post-analysis between result generation and result reporting [6].

The diagnostic process is complex with different test-mix and workload dependent processing times, varying set-up, wait-to-batch and cycle times of the instruments, and sample prioritization. Therefore, TAT prediction requires advanced algorithms. A machine learning (ML) model may effectively learn the relation between various features and TAT. ML is already used extensively in other medical fields. For example, an exponential increase in the number of studies using ML in operating room planning has been observed [7]. However, there are very few studies on ML to predict TAT or investigate factors affecting TAT in medical diagnostic laboratories [5, 6].

The aim of this study was to develop a ML model to accurately predict TAT in a medical diagnostic laboratory, focusing on the automated pre-analytical and analytical phase in the laboratory. Accordingly, this study identified features with the biggest impact on TAT.

***Corresponding author: Dr. Andrei N. Tintu**, Department of Clinical Chemistry, Erasmus MC, University Medical Center Rotterdam, PO Box 2040, 3000 CA, room Na-405, Wytemaweg 8, 3015 CN Rotterdam, The Netherlands, Phone: +31 6 219 515 49,
E-mail: a.tintu@erasmusmc.nl
Eline R. Tsai, Center for Healthcare Operations Improvement and Research (CHOIR), University of Twente, Enschede, The Netherlands; and Department of Clinical Chemistry, Erasmus MC, University Medical Center Rotterdam, Rotterdam, The Netherlands
Derya Demirtas, Nick Hoogendijk and Richard J. Boucherie, Center for Healthcare Operations Improvement and Research (CHOIR), University of Twente, Enschede, The Netherlands

Materials and methods

Setting

We considered TAT in the fully automated part of the clinical chemistry laboratory of the Erasmus University Medical Center Rotterdam (Erasmus MC) that consists of one cobas 8,100 and two cobas 8,000 analyzer lines (c8100, c8000, Roche Diagnostics International Ltd, Rotkreuz, Switzerland). Figure 1A provides a schematic overview of the laboratory workflow.

The c8000 analyzer lines consist of four modules, ISE, c702, c502 and e801, on which the actual diagnostic testing is performed. At each module, the samples are pipetted, after which they are transferred to the incubator disc of that module (Figure 1B). In the incubator disc they are mixed with reagents. After the incubation time, the results become available. In the meantime, after pipetting is completed, the rack is routed to the next module on its route. Incubation times are fixed per test. We define the TAT of a sample as the time between the arrival at the c8100 for pre-analysis and generation of results from both the collection tube and its aliquots.

Data extraction and preparation

Data were extracted from January to March 2019 from both the cobas IT middleware (cITm, Roche Diagnostics International Ltd, Rotkreuz, Switzerland) and the c8000 log files, containing the priority level, required tests, and date and time of the various activities. Samples with a duration of more than 40 min in the c8100 or the c8000 were considered outliers and therefore removed from the analysis.

Features

Table 1 describes the 39 features obtained from the dataset, with $t \in \{2, 5, 10, 15, 20, 25, 30\}$ minutes. The features Centrifuged, CO_2-L, HasBatchTest and Priority are binary, where a value of 1 means that the sample was centrifuged at the c8100, requires the bicarbonate (CO_2-L) test, has a batch test or is a high priority sample, respectively. The other feature values lie in a wider range of integers, such as the number of samples that arrived in the past $t \in \{2, 5, 10, 15, 20, 25, 30\}$ minutes.

Methods

Python 3.9 with package PyCaret 2.2.0 was used to compare multiple regression models for TAT prediction. PyCaret has 25 built-in regression models that fall into multiple categories depending on their parameters (Table 2, [8]). The category Linear Models considers regression problems where the target value is expected to be a linear combination of the features. Linear Regression and Ridge Regression use different loss functions for penalizing the difference between the actual and predicted target value [9]. Kernel Ridge regression combines Ridge regression and classification with the kernel trick, therefore learning a linear predictor in the space induced by the kernel [10]. Support Vector Machines construct a set of hyper-planes to separate the samples, aiming to maximize the gap between a hyperplane and the nearest data points to decrease the generalization error [9]. The K Nearest Neighbors Regressor predicts the target value of a new instance on the basis of the target values of its K nearest neighbors [9]. The Decision Tree Regressor builds a regression tree where the default function to evaluate the quality of a split in PyCaret is the mean

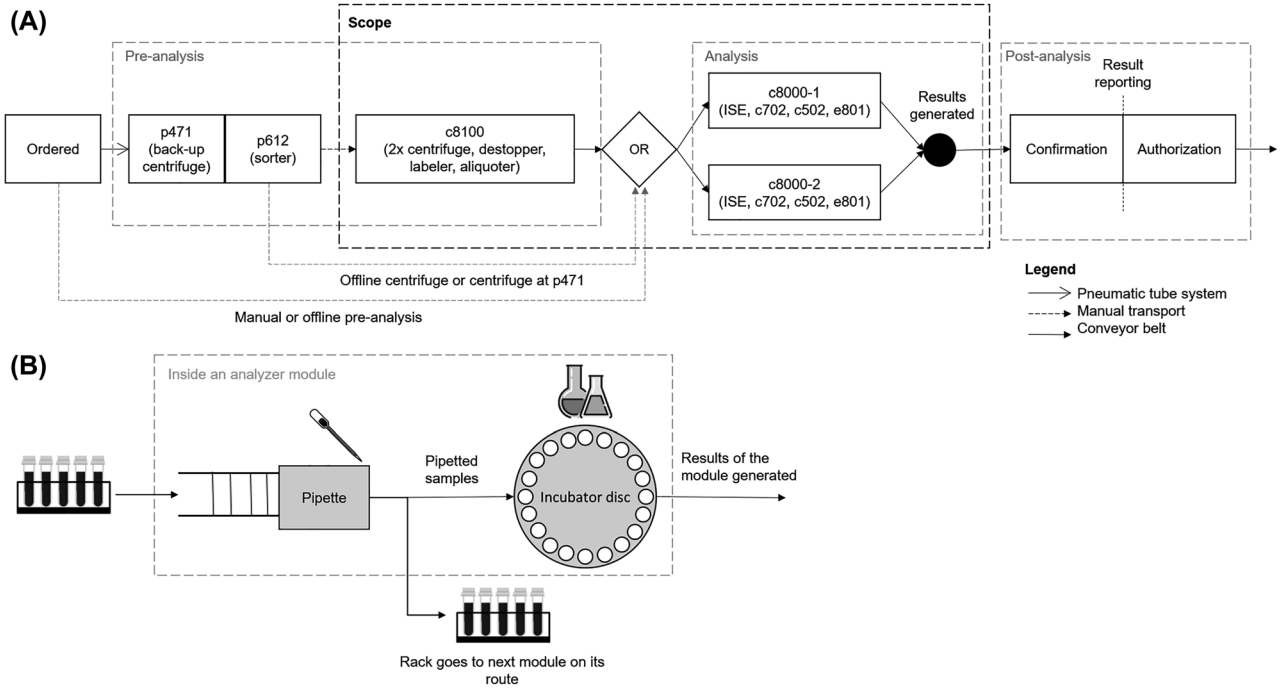

Figure 1: Schematic overviews.
(A) Schematic overview of the main sample flow in Erasmus MC. (B) Schematic overview of the sample flow in an analyzer module.

Table 1: List of included features.

Feature	Description	Type
Centrifuged	Whether the sample requires centrifugation on c8100 or not.	Binary
CO_2-L	Whether the sample requires the bicarbonate test or not.	Binary
HasBatchTest	Whether the sample contains a batch test or not.	Binary
Priority	Sample priority	Binary
InC8100EnteredAna-t	Number of samples that entered the analyzer line the sample is allocated to in the last t minutes.[a]	Integer
InC8100EnteredC8100-t	Number of samples that entered the c8100 during the last t minutes.[a]	Integer
InC8100EnteredC8100Ana-t	Number of samples that entered the c8100 during the last t minutes[a] which are allocated to the same analyzer line.	Integer
InC8100EnteredPrePre-t	Upon sample arrival in c8100, this is the number of samples that entered the p471 during the last t minutes.[a]	Integer
InC8100WorkloadAna	Upon sample arrival in c8100, this is the workload of the analyzer line the sample is allocated to.	Integer
InC8100WorkloadC8100	Workload c8100 upon sample arrival.	Integer
InC8100WorkloadC8100Ana	Number of samples in c8100 upon sample arrival which are allocated to the same analyzer line.	Integer
InC8100WorkloadPrePre	Upon sample arrival in c8100 this is the number of samples that are in the p471.	Integer
NumAliq	Number of tubes. When only collection tube: NumAliq=1.	Integer
NumModule	Number of c8000 modules allocated to.	Integer
Numtest	Number of ordered tests.	Integer

The type "integer" includes integer values of 0 and larger. ([a]) $t \in \{2, 5, 10, 15, 20, 25, 30\}$ minutes.

Table 2: Regression models built into PyCaret.

Model	Category	Subcategory
Linear Regression	Linear Models	Classic linear regressors
Lasso Regression		Regressor with variable selection
Elastic Net		
Least Angle Regression		
Lasso Least Angle Regression		
Orthogonal Matching Pursuit		
Bayesian Ridge		Bayesian regressor
Automatic Relevance Determination		
Random Sample Consensus		Outlier robust regressor
TheilSen Regressor		
Huber Regressor		
Ridge Regression		Miscellaneous
Passive Aggressive Regressor		
Kernel Ridge	Kernel Ridge Regression	
Support Vector Regression	Support Vector Machines	
K Neighbors Regressor	Nearest Neighbors	
Decision Tree Regressor	Decision Trees	
Random Forest Regressor	Ensemble Methods	
Extra Trees Regressor		
AdaBoost Regressor		
Gradient Boosting Regressor		
MLP Regressor	Neural Network Models	Supervised models
Extreme Gradient Boosting	Gradient Boosting Extension	
Light Gradient Boosting Machine		
CatBoost Regressor		

squared error. Ensemble Methods combine the predictions of multiple base predictors to improve model performance [10]. The Extra Trees (ET) Regressor builds an ensemble of regression trees, where the arithmetic average of the predictions of the individual trees is the final prediction. At each node in the tree, a random subset of features is considered, for which random feature values are generated. A node is split based on the best feature and cut-point combination in terms of variance reduction. The Gradient Boosting Regressor

sequentially enhances the performance of a model by fitting an additive model on the negative gradient of a chosen loss function [10]. Neural Networks consist of several hidden layers containing neurons, where each neuron transforms the values from the previous layer into a new value based on a weighted summation of the previous values followed by an activation function [9].

PyCaret was used to split the data, and train, evaluate, compare, and tune these models. The dataset was divided into two sets, a training set, and a test set, by random splitting. Several ratios for splitting the data were tested, namely 95–5%, 90–10%, 80–20% and 70–30%, respectively, to determine the best one for our experiment. When comparing the models and tuning the parameters, the average performance measures obtained from 10-fold cross validation on the training set were considered to get a more accurate view of model performance. The tuned model was then fit on the whole training set and its performance was evaluated on the test set.

The ML model was trained to predict the time until pipetting is completed, namely $TAT_{pipet,i}$. For TAT prediction, the largest incubation time out of all ordered tests of a sample was added to the prediction, i.e., $TAT_i = TAT_{pipet,i} + \max\{\text{incubation time sample } i\}$.

Performance of a model was quantified in terms of mean absolute error (MAE), mean absolute percentage error (MAPE), and R^2 value:

$$\text{MAE} = \frac{\sum_{i=1}^{N}\left|\text{TAT}_{pipet,i} - \widehat{\text{TAT}}_{pipet,i}\right|}{N},$$

$$\text{MAPE} = \frac{100}{N}\frac{\sum_{i=1}^{N}\left|\text{TAT}_{pipet,i} - \widehat{\text{TAT}}_{pipet,i}\right|}{TAT_{pipet,i}},$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(\text{TAT}_{pipet,i} - \widehat{\text{TAT}}_{pipet,i}\right)^2}{\sum_{i=1}^{N}\left(\text{TAT}_{pipet,i} - \overline{\text{TAT}}_{pipet,i}\right)^2},$$

where $\text{TAT}_{pipet,i}$ denotes the actual value for sample i, $\widehat{\text{TAT}}_{pipet,i}$ denotes its predicted value and

$$\overline{\text{TAT}}_{pipet,i} = \frac{1}{N}\sum_{i=1}^{N}\text{TAT}_{pipet,i}$$

denotes the mean of the actual values. Of these performance measures, R^2 is the most informative in terms of how often did the model correctly predict TAT [11]. It can be interpreted as the proportion of the variance in TAT that is predictable from the features [11]. MAE and MAPE are commonly used measures in ML [9] and provide a more natural and intuitive measure of model performance [12, 13].

Default model parameters in PyCaret were used to compare the models in Table 2. After model selection, the parameters of the best model were tuned to increase model performance. PyCaret uses random grid search (RGS) for parameter tuning. RGS selects a predefined number of parameter combinations according to some probability distribution, for which the model performance is evaluated, after which the best parameter combination is selected [14]. PyCaret tunes the values of the main model parameters and various additional parameters.

Model evaluation

The relative residual plot was used to visualize the distribution of the prediction error. It considers the relative difference between the actual and predicted value for each sample i in the test set:

$$\text{Relative residual}_i = \frac{\text{TAT}_{pipet,i} - \widehat{\text{TAT}}_{pipet,i}}{\text{TAT}_{pipet,i}}.$$

SHAP (SHapley Additive exPlanations) values [15] were plotted to interpret how the selected model makes predictions, i.e., to analyze how and to what extent the features influence TAT_{pipet}. SHAP value analysis is particularly useful when the relation between the features and predictions is not straightforward, such as when using ensemble methods. Let f denote the original prediction model, x an individual observation and G the set of all features. In our case $f(x) = TAT_{pipet}$. To evaluate the predictions made by f when only a subset of the features $S \subseteq G$ is included we calculate

$$\mathbb{E}[f(x)|x_S],$$

which corresponds to the conditional expected value of the prediction given that only the attribute values corresponding to the features in S are known. SHAP values were calculated with the trained model. For each sample in the test set, the contribution of each feature to the prediction was determined, thus the value of adding a feature i to S was evaluated. The SHAP value for feature i and observation x is calculated as follows [15]

$$\phi_i(f,x) = \sum_{S\subseteq G\setminus\{i\}}\frac{(|S|+1)!\,(|G|-|S|)!}{|G|!}\left(\mathbb{E}[f(x)|x_{S\cup\{i\}}] - \mathbb{E}[f(x)|x_S]\right).$$

Results

The initial dataset contained 96,126 samples. After data cleaning, 90,543 samples remained, while 5.8% of the initial dataset was considered as outliers and discarded. The historical average TAT and TAT_{pipet} were 29.96 and 19.30 min, respectively. The distributions of the duration in the c8100, the duration in the c8000, TAT_{pipet} and TAT were right-skewed (Supplementary Figure 1).

Out of the regression models in PyCaret, the ET Regressor [16] performed best in terms of MAE, R^2 and MAPE (Table 3). The top 7 performing models in terms of MAE were also the top 7 performing models in terms of R^2 and MAPE. The mean TAT_{pipet} in the training set was 19.29 min. Testing the different ratios for splitting the dataset showed that splitting the data into a 95% (86,015 samples) training set and a 5% (4,528 samples) test set performed best in terms of MAE, R^2 and MAPE.

Parameter tuning

The ET Regressor has three main parameters to tune: the number of trees generated (T), the number of features randomly selected at each node (F), and the minimum number of samples for splitting a node (n_{min}) [15]. Their default values are T=100, F = all features, and n_{min}=5.

Additional parameters considered by PyCaret include the minimum number of samples allowed in a

Table 3: Performance of regression models in the model comparison phase for predicting TAT_{pipet}.

Model	MAE, min	R^2	MAPE (%)
Extra Trees Regressor	2.48	0.509	12.31
Random Forest Regressor	2.56	0.494	12.71
CatBoost Regressor	2.61	0.485	12.81
Extreme Gradient Boosting	2.61	0.482	12.84
Light Gradient Boosting Machine	2.74	0.439	13.43
K Neighbors Regressor	2.76	0.394	14.09
Gradient Boosting Regressor	2.92	0.355	14.24
MLP Regressor	3.03	0.344	14.89
Ridge Regression	3.08	0.285	14.95
Bayesian Ridge	3.08	0.285	14.95
Linear Regression	3.08	0.285	14.95
Automatic Relevance Determination	3.08	0.284	14.96
Least Angle Regression	3.11	0.276	15.12
TheilSen Regressor	3.04	0.257	14.69
Orthogonal Matching Pursuit	3.24	0.237	15.84
Support Vector Regression	3.05	0.180	14.79
Elastic Net	3.37	0.168	17.44
Lasso Regression	3.37	0.164	17.49
Huber Regressor	3.55	0.102	17.46
Lasso Least Angle Regression	3.80	0.000	19.78
Decision Tree Regressor	3.45	−0.050	17.27
Random Sample Consensus	5.41	−0.798	27.92
Passive Aggressive Regressor	5.83	−1.543	28.69
AdaBoost Regressor	8.86	−1.614	51.61
Kernel Ridge	Insufficient memory		

Reported values are average cross validation scores on the training set.

leaf node and the maximum tree depth. Using the tuned values of the three main parameters and the default values of the other parameters gave better model performance than setting all parameters to their tuned value (Table 4).

Figure 2 shows the behavior of model performance when varying the values of T, F or n_{min}, while using default values for the other parameters. MAPE shows similar behavior to MAE (Supplementary Figure 2). The training time increases linearly as T and F increase, while it decreases exponentially as the n_{min} increases (Supplementary Figure 2).

We found a non-monotone relation between the performance of the model in terms of the number of features randomly selected at each node and the minimum number of samples for splitting a node. Furthermore, the model performance does not greatly improve after T=350, while the training time increases linearly. Therefore, we set T to 350, while varying F between 4 and 15, and n_{min} between 2 and 6.

Table 4 shows the parameter combination that resulted in the highest R^2 value and the combination resulting in the lowest MAE and MAPE. We selected the parameter settings that resulted in the highest R^2 value as this explains a higher fraction of the variation of TAT by the features.

Model evaluation

The ET Regressor was fit on the entire training set and its performance was evaluated on the test set. This resulted in R^2 of 0.52, MAE of 2.42 min and MAPE of 11.71%, where the historical average TAT_{pipet} over the test set was 19.41 min. Accordingly, for predicted TAT ($TAT_i = TAT_{pipet,i}$ + max{incubation time sample i}), we obtained R^2 of 0.63, MAE of 2.42 min and MAPE of 7.35%, where the historical average TAT over the test set was 30.09 min. The model takes 0.38 s to predict TAT for all the test set samples.

The relative residual plot for TAT_{pipet} (Figure 3) shows that 59.5% of samples had a relative residual between −0.1 and 0.1, and 88.2% of samples had a relative residual between −0.25 and 0.25. For predicting TAT, 76.6% of samples had a relative residual between −0.1 and 0.1, and 95.7% of samples had a relative residual between −0.25 and 0.25. Using a myopic prediction policy in which each TAT is predicted to be the mean TAT, 42.6% of the samples had a relative residual between −0.1 and 0.1, and 84.8% had a relative residual between −0.25 and 0.25. Thus, the model had sufficiently small errors for predicting TAT and had 1.8 times (76.6 vs. 42.6%) more samples with a relative residual between −0.1 and 0.1 as compared to using the myopic policy.

Table 4: Results of parameter tuning.

Method	n_{min}	T	F	Values other parameters	MAE, min	R^2	MAPE, %
PyCaret	7	200	6	Tuned	3.00	0.332	14.82
	7	200	6	Default	2.49	0.508	12.24
Manual	2	350	6	Default	2.39	0.530	11.79
	2	350	4	Default	2.38	0.529	12.77

Method = PyCaret: Performance ET Regressor using PyCaret parameter tuning. Method = manual: Parameter combination resulting in the highest R^2 value and the combination resulting in the lowest MAE and MAPE from selection of parameters based on Figure 2. Reported values are average cross validation scores on the training set.

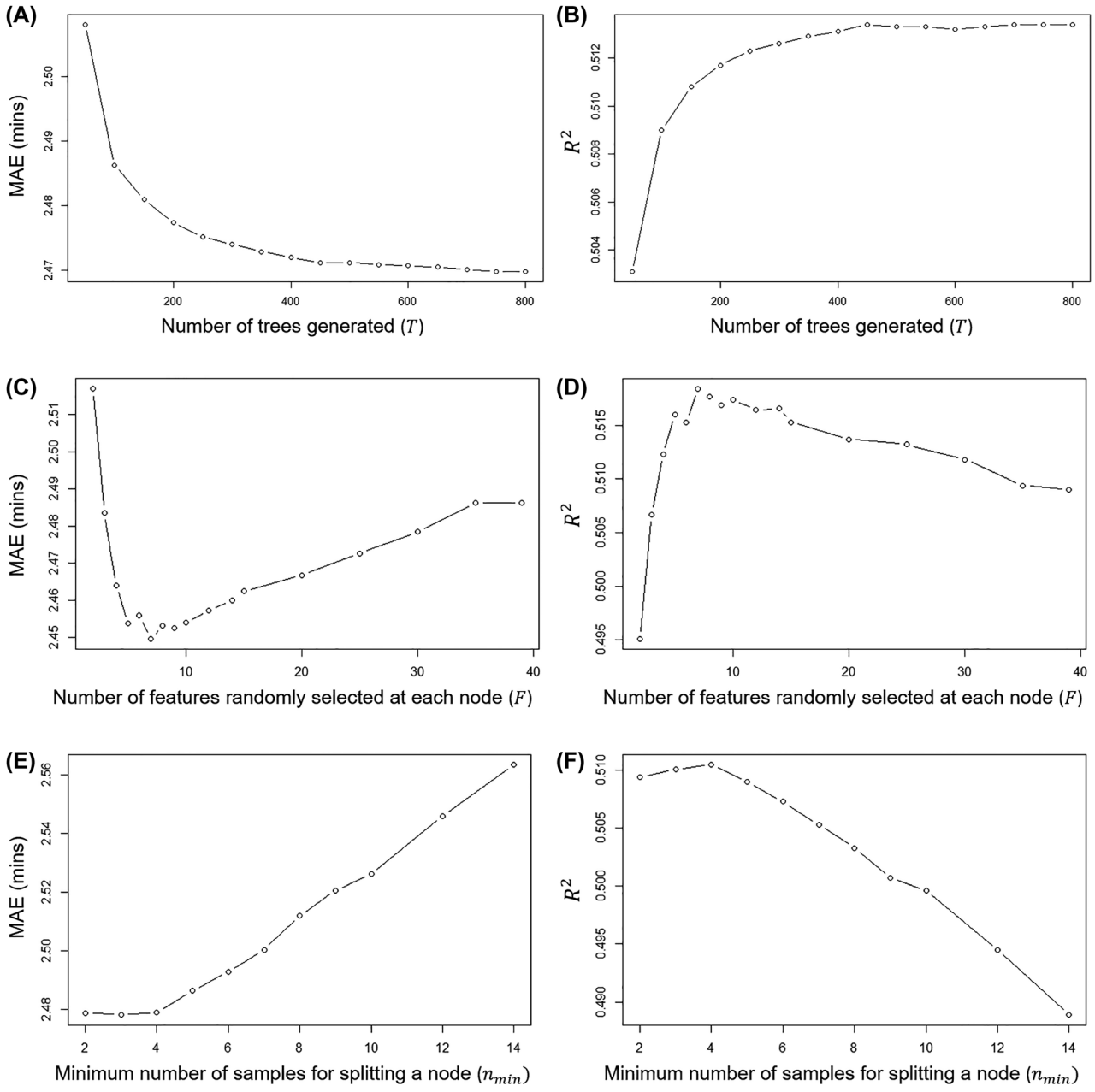

Figure 2: Relation between the parameters and model performance in terms of MAE and R^2.
When varying one parameter, the default values of the other parameters were used: number of features randomly selected at each node = 39, number of trees generated = 100, minimum number of samples for splitting a node = 5. Reported values are average cross validation scores on the training set.

The SHAP values of the top 20 features that influenced the model are shown in Figure 4. Computation time of the SHAP values was approximately 3.5 days on a 16 GB RAM computer cluster. The most important three features were the number of modules a sample is allocated to, the workload in pre-analysis (the c8100) upon sample arrival, and whether the sample requires the CO_2-L test or not.

Discussion

In this paper, we presented a method to accurately predict TAT of clinical chemistry samples (R^2=0.63, MAE=2.42, MAPE=7.35%, historical average TAT=30.09 min) and identified features with the biggest impact on TAT. This is the first TAT prediction study for predicting the time

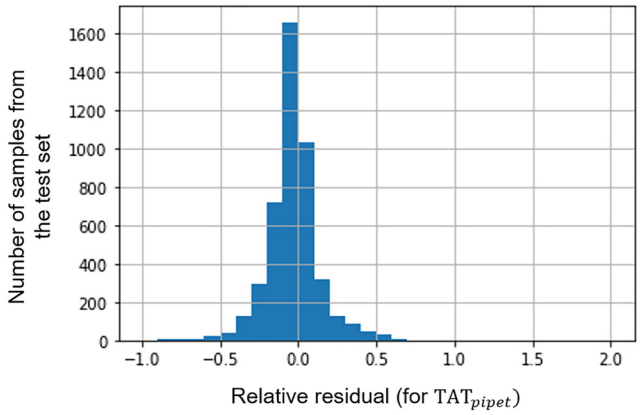

Figure 3: Distribution of the relative residual for predicting TAT_{pipet}.

from the start of the automated pre-analytical phase until the end of the analytical phase for medical diagnostic laboratories. The proposed methodology is widely applicable for TAT prediction in medical diagnostic laboratories and potentially beyond.

Our study showed that the top three features for predicting TAT_{pipet} were the number of modules a sample is allocated to, the workload in pre-analysis upon sample arrival, and whether the sample requires the CO_2-L test or not. The more modules a sample is allocated to, the higher TAT_{pipet} due to an increase in waiting, travel, and setup time in the c8000. The higher the workload in pre-analysis upon sample arrival, the higher the waiting times in the c8100. Samples requiring the CO_2-L test are prone

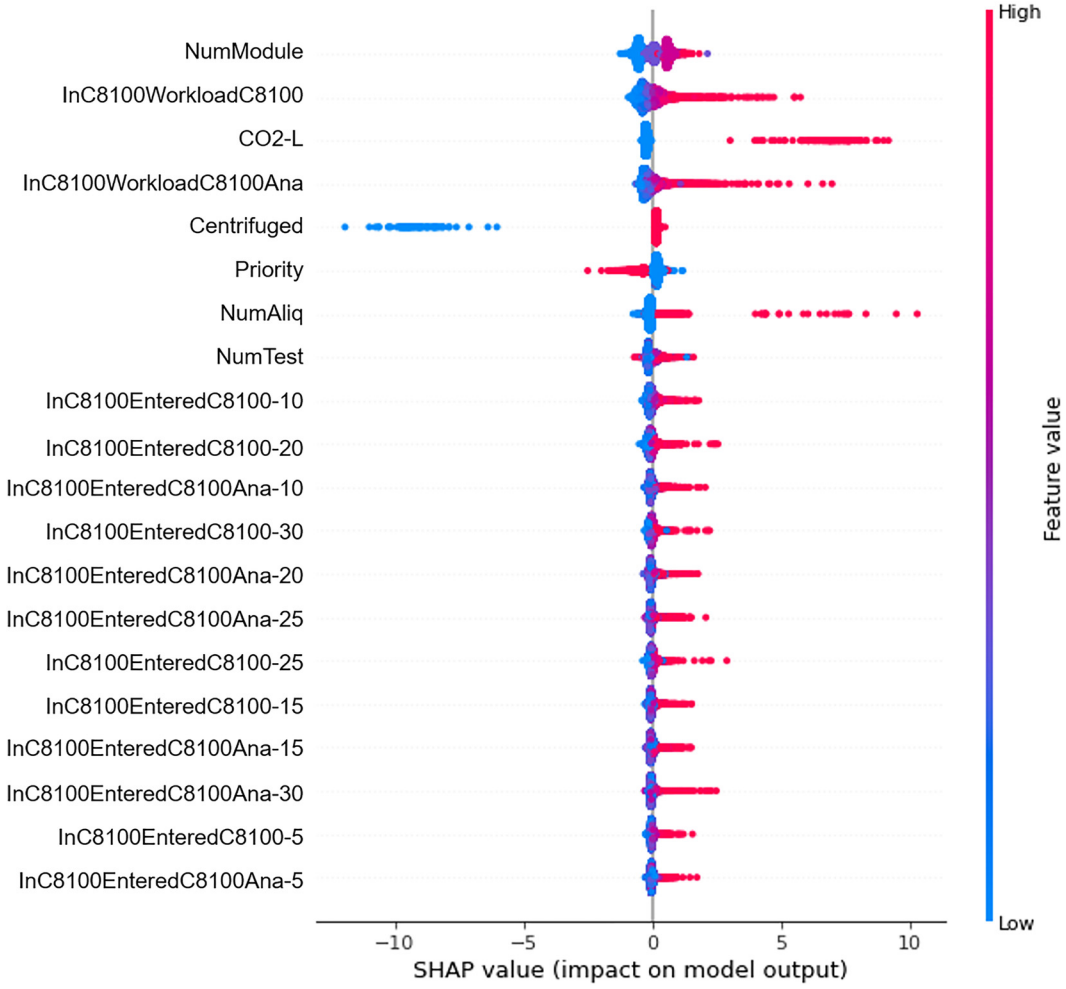

Figure 4: SHAP values of the top 20 features, sorted in decreasing order of importance, based on the tuned model and applied to samples in the test set.
The x-axis indicates whether the feature value resulted in an increase (left) or decrease (right) in the prediction of TAT_{pipet}, and to what extent. Each dot represents a sample, while the color indicates whether the feature value was high (red) or low (blue) for this sample. The features are ordered from highest to lowest mean absolute value of the SHAP values.

to a higher TAT as they are not allowed to be uncapped for too long before analysis. After centrifugation, these samples wait in the c8100 for a laboratory technician to pick them up and manually uncap them before being inserted into a c8000. Understanding which features prolong TAT can help laboratories to reduce TAT.

Our study has similarities and differences as compared with [6], who predicted the post analysis TAT. The authors included the requesting department name, weekday of the events starting from specimen collection to delivery of report to the patient, and the hour of the day these events occurred, as features. Our study showed that the workload upon sample arrival is important for TAT prediction. In the absence of workload information, the arrival weekday and hour of the day are good proxies for the workload, e.g., it is typically busier at 10 a.m. on a Wednesday than 10 a.m. on a Sunday. Using the workload for TAT prediction is especially important if the arrival pattern of the samples is not monotone. We also included sample characteristics and test-mix related features, which are expected to be more important for TAT prediction in pre-analysis and analysis than for post-analysis.

We defined 39 features to be used in TAT prediction models. One of the feature choices we made is the value of t, to determine the number of samples that arrived in the past t minutes at a particular phase or instrument in the testing process. In general, the behavior of TAT lags behind the behavior of the workload. Furthermore, samples arriving within a short time before a sample have a bigger influence on TAT of this sample as compared to samples that arrived much earlier and may almost be finished. It is possible to do feature selection before running the ET Regressor to reduce the number of included features. As the ET Regressor is a tree-based algorithm and as tree-based algorithms naturally filter out less significant features, applying *a priori* feature selection is not necessary.

The ET Regressor has several advantages. It is suitable for problems with a high dimensional feature space as considered in this paper, due to the generated ensemble of diverse trees [17, 18]. In this research, for example, the feature InC8100EnteredC8100-30 had a minimum value of 1 sample and a maximum value of 340 samples and the feature InC8100WorkloadPrePre had a minimum value of 0 samples and a maximum value of 365 samples. The ET Regressor is also computationally efficient [16]. Furthermore, the high level of randomization of the ET Regressor drastically reduces the variance [16], and thus the probability of overfitting. At the same time, to minimize bias, the ET Regressor uses the full original training set instead of bootstrap replicas [16].

We tuned the three main parameters of the ET Regressor to find the best performing combination for our study. The described parameter tuning approach is widely applicable, while the best parameter settings depend on the study. The best choice for T (number of generated trees) is a trade-off between computation time and accuracy, i.e., the more trees generated, the better the accuracy but the higher the computation time [16]. The smaller F (number of features evaluated at a node), the stronger the randomization of the trees and the weaker the dependence of the structure of the trees on TAT values of the training set samples [16]. When F=1, "totally randomized trees" are generated in which the feature and feature values are chosen completely independent of the target variable [16]. When F = all features, then the randomization of the algorithm is only through the randomly selected feature values [16]. Our study used F=6, meaning that the trees have a high level of randomization, while still allowing the model some freedom to select relevant features. The noisier the output, the higher the optimal value of n_{min} (minimum number of samples for splitting a node) to minimize overfitting by creating smaller trees [16]. Our study shows that for our case the optimal value of n_{min} is 2, which is lower than the default value 5. Comparing the average cross-validation scores on the training set with performance on the test set, we observed that model performance is similar. Therefore, we expect to have a good estimate of the performance of the model on an unseen dataset.

Samples with an unexpectedly high duration in the c8100 or c8000 were discussed with the laboratory technicians. These high delays were not explainable under regular circumstances and with our feature set. Possible explanations are short term machine maintenance or human interaction due to a potential error. A cut-off value was chosen that is a trade-off between removing too many samples and leaving in samples that could skew the results due to unexplainable factors. We pragmatically chose a cut-off value that is a multiple of 10 min and that resulted in removing approximately 5% of the samples, resulting in removing samples with a duration of more than 40 min in the c8100 or c8000 (5.81%).

The relative residual plots showed that our model is more likely to overpredict than underpredict TAT. This conservative prediction is preferred as the laboratory rather reports a higher than actual TAT to the post-analysis staff and physicians than a lower than actual TAT.

A limitation of this study is that the data is right-skewed. This is generally not an issue for non-parametric methods [18] such as the ET Regressor. Performance of the parametric methods could have been improved by first transforming the data to have a Gaussian distribution.

A direction for future research is to study whether model performance can be improved by techniques such

as bagging, boosting, and stacking. Other directions for future research are to broaden the scope of TAT and use these predictions to improve healthcare efficiency. Ideally, we predict the time between phlebotomy and result reporting. For this, one would also need the time spent between the p471 and c8100. The time spent between arrival at the p471 and arrival at the p612 typically takes less than 2 min, therefore not having a big contribution to TAT. However, samples are manually transported between the p612 and c8100, making accurate TAT predictions challenging. One can obtain insights into the time spent between the p612 and c8100 by including the staffing levels at the time the sample leaves the p612. One would also need data on result reporting. As the vast majority of Erasmus MC test results are automatically confirmed and reported, the result generation timestamp differs typically only a few seconds from the result reporting timestamp. To optimize sample routing, when a sample can be tested on multiple analyzer lines, the laboratory can assign the sample to the analyzer line with the shortest predicted TAT.

Conclusions

The diagnostic process is complex with various factors affecting TAT. ML techniques allow for accurate laboratory TAT predictions and the identification of features with the biggest impact on TAT, enabling the laboratory to take timely action in case of prolonged TAT and helping healthcare providers to improve planning of scarce resources to increase healthcare efficiency.

Research funding: None declared.
Author contributions: All authors have accepted respon sibility for the entire content of this manuscript and approved its submission.
Competing interests: Authors state no conflict of interest.
Informed consent: Not applicable.
Ethical approval: Not applicable.

References

1. Tsai ER, Tintu AN, Demirtas D, Boucherie RJ, de Jonge R, de Rijke YB. A critical review of laboratory performance indicators. Crit Rev Clin Lab Sci 2019;56:458–71.
2. Patel S, Smith JB, Kurbatova E, Guarner J. Factors that impact turnaround time of surgical pathology specimens in an academic institution. Hum Pathol 2012;43:1501–5.
3. Chauhan KP, Trivedi AP, Patel D, Gami B, Haridas N. Monitoring and root cause analysis of clinical biochemistry turn around time at an academic hospital. Indian J Clin Biochem 2014;29:505–9.
4. Fei Y, Zeng R, Wang W, He F, Zhong K, Wang Z. National survey on intra-laboratory turnaround time for some most common routine and stat laboratory analyses in 479 laboratories in China. Biochem Med 2015;25:213–21.
5. Thiha S, Shewade HD, Philip S, Aung TK, Kyaw NTT, Oo MM, et al. Factors associated with long turnaround time for early infant diagnosis of HIV in Myanmar. Glob Health Action 2017; 10:1–7.
6. Eminağaoğlu M, Vahaplar A. Turnaround time prediction for a medical laboratory using artificial neural networks. Bilişim Teknol Derg 2018;11:357–68.
7. Bellini V, Guzzon M, Bigliardi B, Mordonini M, Filippelli S, Bignami E. Artificial intelligence: a new tool in operating room management. Role of machine learning models in operating room optimization. J Med Syst 2020;44:1–10.
8. Ali M. PyCaret: an open source, low-code machine learning library in Python [Internet]. Available from: https://www.pycaret.org; 2020.
9. Shalev-Shwartz S, Ben-David S. Understanding machine learning: from theory to algorithms. New York: Cambridge University Press; 2014.
10. Murphy KP. Probabilistic machine learning: an introduction. MIT Press; 2022.
11. Chicco D, Warrens MJ, Jurman G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci 2021;7: e623.
12. Hyndman RJ, Koehler AB. Another look at measures of forecast accuracy. Int J Forecast 2006;22:679–88.
13. Willmott CJ, Matsuura K. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Research 2005;30:79–82.
14. Bhat PC, Prosper HB, Sekmen S, Stewart C. Optimizing event selection with the random grid search. Comput Phys Commun 2018;228:245-57.
15. Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Adv Neural Inf Process Syst 2017;30:4766–75.
16. Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. Mach Learn 2006;63:3–42.
17. Pinto A, Pereira S, Correia H, Oliveira J, Rasteiro DMLD, Silva CA. Brain tumour segmentation based on extremely randomized forest with high-level features. In: Proc Annu Int Conf IEEE Eng Med Biol Soc EMBS; 2015:3037–40 pp.
18. Moosmann F, Triggs B, Jurie F. Fast discriminative visual codebooks using randomized clustering forests. Adv Neural Inf Process Syst 2006;19:985–92.

Supplementary Material: The online version of this article offers supplementary material (https://doi.org/10.1515/cclm-2022-0668).