



Predicting Patient's Waiting Times in Emergency Department: A Retrospective Study in the CHIC Hospital Since 2019

Nadhem Ben Ameer, Imene Lahyani, Rafika Thabet, Imen Megdiche,
Jean-Christophe Steinbach, Elyes Lamine

► To cite this version:

Nadhem Ben Ameer, Imene Lahyani, Rafika Thabet, Imen Megdiche, Jean-Christophe Steinbach, et al.. Predicting Patient's Waiting Times in Emergency Department: A Retrospective Study in the CHIC Hospital Since 2019. MEDI 2022 - International Conference on Model and Data Engineering, Nov 2022, Le Caire, Egypt. pp.44-57, 10.1007/978-3-031-23119-3_4 . hal-03935078

HAL Id: hal-03935078

<https://imt-mines-albi.hal.science/hal-03935078>

Submitted on 19 Jul 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Predicting Patient’s Waiting Times in Emergency Department: A Retrospective Study in the CHIC Hospital Since 2019

Nadhem Ben Ameer^{1,4}, Imene Lahyani^{1,2(✉)}, Rafika Thabet^{3,4},
Imen Megdiche^{4,5}, Jean-christophe Steinbach⁶, and Elyes Lamine^{3,4}

¹ National School of Engineers of Sfax, University of Sfax, Sfax, Tunisia
`{nadhem.benameur, imen.lahyani}@enis.tn`

² ReDCAD Laboratory, Sfax, Tunisia

³ Centre Génie Industriel, IMT Mines Albi, Toulouse University, Albi, France
`{rafika.thabet, elyes.lamine}@mines-albi.fr`

⁴ Institut National Universitaire Champollion, ISIS, Toulouse University,
Castres, France
`imen.megdiche@univ-jfc.fr`

⁵ IRIT, Institut de Recherche en Informatique de Toulouse, Toulouse University,
Toulouse, France
`imen.megdiche@irit.fr`

⁶ Centre Hospitalier Intercommunal Castres-Mazamet, Castres, France
`jean-christophe.steinbach@chic-cm.fr`

Abstract. Predicting patient waiting times in public emergency department rooms (EDs) has relied on inaccurate rolling average or median estimators. This inefficiency negatively affects EDs resources and staff management and causes patient dissatisfaction and adverse outcomes. This paper proposes a data science-oriented method to analyze real retrospective data. Using different error metrics, we applied various Machine Learning (ML) and Deep learning (DL) techniques to predict patient waiting times, including RF, Lasso, Huber regressor, SVR, and DNN. We examined data on 88,166 patients’ arrivals at the ED of the Intercommunal Hospital Center of Castres-Mazamet (CHIC). The results show that the DNN algorithm has the best predictive capability among other models. By precise and real-time prediction of patient waiting times, EDs can optimize their activities and improve the quality of services offered to patients.

1 Introduction

Emergency Department Overcrowding (EDW) is a long standing issue affecting all healthcare systems. This is mainly caused by the imbalance between supply

and demand for emergency services, which is becoming overwhelming due to the ageing population and the lack of human resources. ED overcrowding implies different consequences such as waiting times, delayed treatment or ambulance diversion. Prolonged waiting times may reduce the quality of care and increase the likelihood of adverse outcomes for patients with serious illnesses. Patient satisfaction is also affected, and more patients leave before being visited by a physician. Thus, waiting time is a key metric for measuring ED efficiency.

Measuring waiting time is a key challenge that was addressed by several works [3,5,6]. In [3], they test different machine learning techniques using predictive analysis applied to two large datasets from two emergency departments in medium-sized public hospitals located in average-populated areas of north-central Italy. [5] develop deep learning models alongside or instead of queuing theory techniques to predict waiting time in a queue using real low patient acuity data collected from electronic medical records in an ED in Saudi Arabia. [6] propose to use statistical learning algorithms to significantly improve waiting time predictions using routinely collected administrative data from a large Australian public tertiary health care centre.

In this paper, we propose a retrospective study of waiting times in the Emergency Department of a French hospital of Castres Mazamet CHIC. In our approach, we experiment with different machine learning techniques on a large amount of real data covering patient characteristics and emergency characteristics. We also evaluate the sliding window technique to improve prediction accuracy due to quality issues.

The particularity of our work is a specification of waiting time prediction techniques according to the organization of this hospital and for real data suffering from some quality problems.

Our aim is to identify the learning technique that provides the most accurate and real-time previsions of waiting times for patients arriving to the ED. We evaluate the accuracy of Lasso, Random Forest, Support Vector Regression, Artificial Neural Network using two forecasting error measures: the mean squared error and the mean absolute error.

The paper is structured as follows: in Sect. 2 we introduce base study concepts. Then in Sect. 3, we address some work in the literature. In Sect. 4, we describe our methodology. We present the results in Sect. 5. Section 6 summarises the study and the findings and identifies future directions.

2 Preliminaries

In this research, we design predictive models for patients' waiting times in the ED of the CHIC Hospital. The waiting time was defined in collaboration with the chief of Emergency Department and the chief of the Information System of the hospital. The waiting time is declined according to the existing waiting rooms: we have internal and external waiting times as described in Fig. 1. At CHIC ED, the waiting room is adjacent to the ED entrance, and there are no barriers for patients to exit the ED once their wait time exceeds their willingness to wait. In

Fig. 1, we describe the simple ED patient flow. Upon arrival, The patient goes to one of the three registration counters, where further personal information (e.g., address, occupation) are collected, and the patient arrival record registration is created. After the registration, the patient may experience a queue to see a triage nurse who records some basic identification details, presents a complaint, describes the nature of the visit, and assigns a triage level and a treatment box. After the triage, the patient proceeds to the designated waiting area if not called in for treatment immediately. A patient must wait in a designated waiting area until being called in to see a physician or an intern, or a treating nurse may come out to the waiting area and take vital signs and periodically check on the patient status while waiting to be called in for treatment. Different ED teams generally treat patients in different ED areas. For example, ED also operates a Short-Term Hospitalization Unit (STHU) specifically designed for short-term treatment (often completed within 24 h). This inpatient unit under ED management provides health services to patients initially assessed, triaged, and treated in the ED. Most often, patients with selected medical conditions that require further observation, assessment, and reassessment but are unlikely to require a prolonged admission to the hospital are assigned to STHU. These patients are not considered by this study.

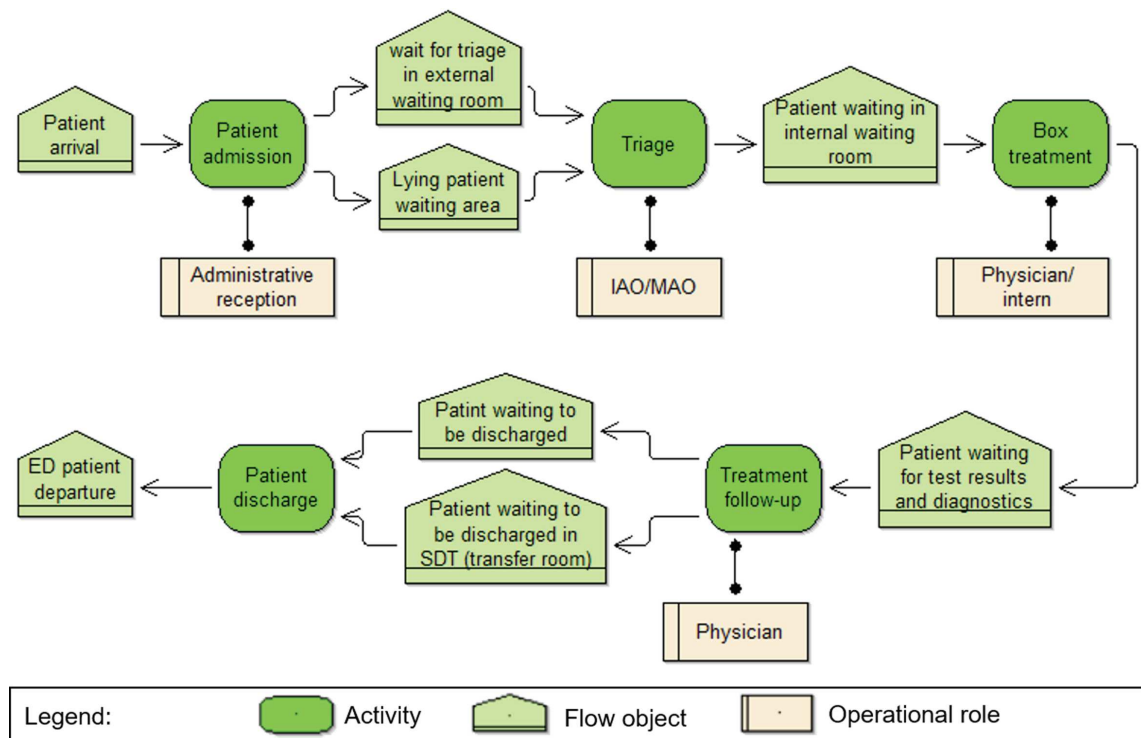


Fig. 1. Business process model of ED patient flow

3 Related Work

This section provides a comprehensive review of the works related to waiting time prediction in public emergency department rooms (EDs). Previous studies have demonstrated that prolonged waiting periods cause patients' frustration, anxiety and dissatisfaction [10]. In addition, patients are concerned about the quality of the information provided in EDs and how to interpret it [4]. Indeed, the current ED waiting times reporting systems are limited and generally based on rolling average or median estimators. Waiting time is one of the most important quality indicators for emergency department [8], several studies proposed various approaches to optimize patients waiting times prediction and enhance EDs resources management efficiency. [5] combined deep learning with queuing theory to predict waiting time in a queue using real low patient acuity data. They provided a guideline for queue waiting time analysis not only in healthcare but also in other sectors considering model understand and feature extraction process. [3] implemented a combination of approaches including machine learning and various categories of features to predict patient waiting time in real time. [2] used ordinal logistic regression and data mining techniques to develop models to classify ED patients in terms of their waiting and treatment times. Finally, [6] implemented quantile regression and statistical learning algorithms for a large set of predictors using comprehensive EMR from the ED information system. They developed models to predict ED wait time for patients with low acuity assigned to the ED waiting room and differentiated the low acuity patients into waiting room and non-waiting room patients for the purpose of waiting room prediction.

Our study differs from previous research on this topic as we predicted patients waiting times specific to CHIC ED and we implemented various ML and multi-DL optimization algorithms to improve accuracy. We used sliding window algorithm to detect outliers in data. In addition, we accounted for various predictors by extracting various categories of features such as patient's related features, queue-based and patient condition's severity score-based features. Our models with multiple ML and Multi-DL optimization algorithms can be applied in evaluating the wait times of patients with in the emergency department. Consequently, the models proposed in this study can be used to provide insights to medical staff in the emergency room to determine patient waiting time in the queue using EHR data.

4 Methodology

In this section, the extracted data and the used dataset are described, and our methodological process is detailed. As presented in Fig. 2, our methodology includes five main steps: data cleaning, feature engineering, dataset analysis, dataset preprocessing, Machine Learning models, and model evaluation. In the following, a detailed description of each step will be presented.

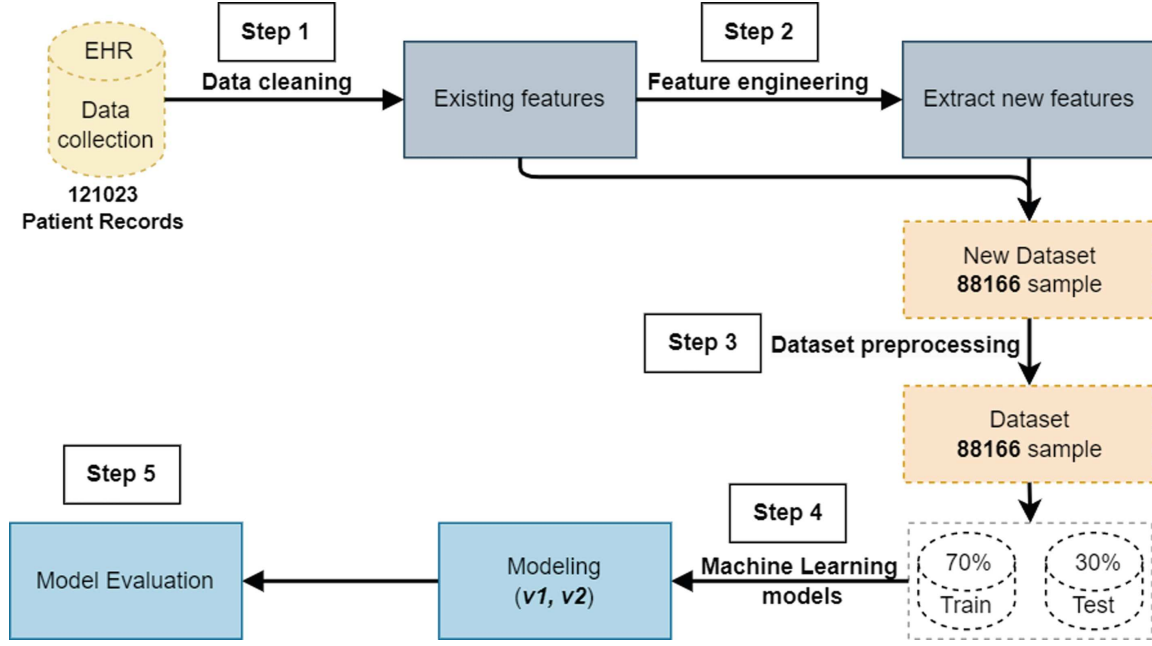


Fig. 2. Proposed methodology

4.1 Data Description

We used real data from the CHIC hospital ED. Electronic Health Records (EHR) tend to be complex and challenging to manage [9]. The Conceptual Data Models (CDM) of the databases used contains more than 1000 entity tables, from which 22 tables were selected to work on ED activities. We examined data on 121,023 patient’s arrivals of the hospital ED spanning the period from January 2019 to April 2022. The extracted raw data contain patients-related information and their associated ED episodes, from the registration and triage process to ED discharge or hospital admission timestamps. In order to build a relevant set of features, we essentially proceeded in three ways: the identification of relevant features previously approved in the scientific work for waiting time prediction, the exchange with the staff of the emergency services and the staff of the IT department of the CHIC hospital, and finally, the use of the data collected by the information system (databases and reports). The raw data was then cleaned and used to extract useful features for predicting patient waiting times specific to the CHIC hospital context. The features included in the data are summarized in Table 1.

4.2 Data Cleaning

The extracted raw data has several quality issues. It contains duplicate, missing, irrelevant data and outliers. In order to solve these problems, we proceeded in 4 phases:

- **Duplicated samples removal:** The extracted raw data contains few redundant patient records (PR) since the data used is the combination of the

Table 1. Description of raw data features

Features	Type	Features	Type
Patient Record PR_id	Numerical	Age	Numerical
Sex	Categorical	Postal code	Categorical
ED patients count	Numerical	Patient arrival status	Categorical
Raison for admission	Categorical	Severity score	Categorical
Arrival entry mode	Categorical	Arrival exit mode	Categorical
Patient arrival ts	Datetime	Ts nurse triage	Datetime
Nurse location ts_start	Datetime	Nurse location ts_end	Datetime
Internal waiting ts_start	Datetime	Internal waiting ts_end	Datetime
Box ts_entry	Datetime	Box ts_quit	Datetime
Box type	Categorical	Pattern code	Categorical
Transfert room wait ts_start	Datetime	Transfert room wait ts_end	Datetime
PR discharge ts	Datetime	PR exit ts	Datetime
Patient exit ts	Datetime	STHU patient count	Numerical
Triage ts validity	Binary	Internal waiting time calculability	Binary
Internal waiting time deb ts	Datetime	Internal waiting time end ts	Datetime

extractions from different resources (ED database, hospital database). The library used to remove duplicates is Pandas, which contains various built-in data preprocessing methods. The discarded sample size is 344 samples.

- **Filter patients by age:** In this research, we found it relevant to work on predicting waiting times for adults. Firstly, because children always have priority over adults, so they don’t wait very long. And secondly, children have separate waiting rooms and treatment boxes from adults, from which they are not drawn into the flow of adult patients. After filtering data by age (≥ 12), The discarded sample size is 18,879 samples.
- **Formatting dates:** Extracting data from multiple databases results in a heterogeneous dates format. Most of the extracted features listed in Table 1 are of type Datetime. Therefore, we performed a data formatting step to ensure that all datetime features have the same format (dd/MM/YYYY hh:mm).
- **Clean “Ts nurse triage” feature:** The data extraction period extends from January 2019 to April 2022 during the Covid-19 pandemic. This causes delays in the storage of patient records in the CHIC databases due to material or human matters. The feature “Ts nurse triage” (Table 1) is a special feature that corresponds to the timestamp of patient records creation in the ED database. As a result, there were some cases where the triage timestamp was severely delayed and calculating patient wait time was impossible. To reduce the number of discarded samples, we used the “Nurse location ts_start” feature (Table 1) because the triage process begins when the patient is in the nurse care area. Unfortunately, the “Nurse location ts_start” feature contains missing values, so it is not possible to recover all patient records with invalid triage timestamps. After the cleaning process, the discarded sample size is 280 samples.

- **Filter patients by ED waiting time predictability:** The predictability of patient waiting time depends on datetime features listed in Table 1. Because most of these features contain missing or invalid values, a cleanup process is required to ensure predictability of ED patient waiting time. We used the redundant features where there was timeline overlap in the patient path (e.g. the “Internal waiting ts_star” feature and the “Box ts_entry” feature must have the same values as the patient leaves the internal waiting room when he enters the assigned box for him)

4.3 Feature Engineering

The cleaned data was used to generate the dataset for modeling patient waiting time prediction algorithms. We considered a set of 30 features, as listed in Table 2. The patient-related features are already included in the raw data. It includes patient’s demographic information such as age and sex, and information about the patient’s status at arrival, the reasons for admission and information about nurse diagnosis (pattern code). To account for daily variations and highlight the seasonal effect, we investigated arrival time variables namely the time of day along with the day of the week, the week, the month, the quarter and a Boolean feature to identify weekend patient arrivals.

The higher the priority of new arrivals, the more significant their impact on the waiting time. Patients with high severity scores (score 0 is the most severe score, score 5 is patients with mental disorders) consume human and material resources for a long periods, which causes patients with the least severe scores to wait for longer periods. Thus, we developed six features corresponding to severity score used in CHIC ED. To calculate Condition severity score based (CS)-based features for each observation, we counted the number of patients based on their severity score in the admission timestamp.

Alongside the features commonly used in the healthcare literature, we also explored the use of queue-based features [3,6], which report the current state of the ED, to improve prediction accuracy, as suggested Queue-based features capture the influence of new ED arrivals on the waiting time of patients within the ED. Queue-based features determine the crowding level of the ED system. They measure the queue of patients waiting at various stages, from registration to discharge (e.g., the queue in internal waiting room or for transfer to the short-term hospitalization units (STHU)). To derive internal queue-based features, we used different datetime features in Table 1 to count patient in different area in the admission timestamp for each observation.

Following [3], we incorporated the rolling average of the waiting times of the last 15 patients who presented to the ED as an additional candidate feature, as well as the bed occupancy (the ratio of occupied bed in the hospital) and the waiting rooms occupancy (the ratio of patients in waiting rooms by the total number of patients in the ED).

Table 2. Feature description

Categories	Features
Patient-related (P)	Age Sex Residence Arrival mode Patient status at entry Reason for admission Triage score Pattern code
Related to time of arrival (TA)	Hour of day Day of week Week Month Quarter Is_weekend
Condition severity score based (CS)	Number of patients score_0 Number of patients score_1 Number of patients score_2 Number of patients score_3 Number of patients score_4 Number of patients score_5
Queue-based (Q)	Patients in external waiting room Patients in internal waiting room Treated patients waiting for discharge Discharged patients waiting for exit Patients in short-term hospitalization units (STHU) STHU_saturated ED patients count
ED and hospital indicators (I)	Bed occupancy Waiting rooms occupancy Wait times rolling average

All the process of feature engineering requires implementing algorithms for each category of features. Due to space limitation, we present only Algorithm 1 that calculate the patient number in the external waiting room.

Algorithm 1: Patient count in external waiting room algorithm

```
VARIABLES : df_slice : DATAFRAME
              df_sorted : DATAFRAME
              ranges : DICTIONNARY
              range_min, range_max, row_index : INTEGER
              other_patient_arrival_ts, other_patient_TS_nurse_triage,
patient_arrival_ts : DATETIME
              PR_id, other_patient_PR_id : INTEGER
              counter, index : INTEGER
INPUT      : df_sorted, PR_id, patient_arrival_ts, ranges
OUTPUT     : counter, index
begin
    counter  $\leftarrow$  0
    range_min  $\leftarrow$  ranges["min"]
    range_max  $\leftarrow$  ranges["max"]
    index  $\leftarrow$  range_min
    df_slice  $\leftarrow$  df_sorted.iloc[range_min : range_max]
    FOR row_index in range (range_min, range_max) do
        other_patient_arrival_ts  $\leftarrow$  df_slice.loc[row_index,
        "Patient_arrival_ts"]
        other_patient_TS_nurse_triage  $\leftarrow$  df_slice.loc[row_index,
        "Ts_nurse_triage"]
        other_patient_PR_id  $\leftarrow$  df_slice.loc[row_index, "PR_id"]
        IF patient_arrival_ts < other_patient_arrival_ts do
            Break
        IF other_patient_PR_id  $\neq$  PR_id and
        other_patient_arrival_ts  $\leq$  patient_arrival_ts  $\leq$ 
        other_patient_TS_nurse_triage do
            counter  $\leftarrow$  counter + 1
        IF counter  $\equiv$  1 do
            index  $\leftarrow$  row_index
    Return counter, index
end
```

The new dataset included 88,166 observations and 30 variables. The variables included in the dataset are summarized in Table 2.

We implement other feature engineering scripts and make our code available on GitHub.¹

4.4 Dataset Preprocessing

Data preprocessing is considered as the main step in any ML classification process. It refers to the transformations applied to data before feeding it into our ML algorithms. The goal is to convert the raw data into a clean dataset. We had to apply a sliding window technique and normalization to deal with the presence of outliers in the used numerical features. As for missing values issue, we

¹ <https://github.com/nadhembenameur99/CHIC-ED-patients-waiting-times-prediction.git>.

applied an imputing technique. Data encoding is applied to categorical features to convert them to numerical format, as ML algorithms require numerical data.

- **Sliding window concept for outlier detection** [1]: Firstly, the sliding window size is set to h and the data size is set to N . Then a sample formed by data from i to $i + h$, where $1 \leq i \leq N - h$, is used to estimate the upper and the lower bands of values inside the window. The window slide by one, the sample is updated, and the new upper and lower bands are produced.
- **Missing values imputation:** For imputing missing data, the most frequent imputation method is used to impute “Residence” and “Pattern code” features. In this approach, we impute each missing value by the most frequent value for each feature.
- **Data normalization:** The goal of normalization is to transform features to be on a similar scale. This improves the performance and training stability of the model. This transformation converts all the features into the same scale and harmonizes the data structure.
- **Categorical data encoding:** For encoding categorical data, the frequency encoding method is used to encode “Residence” and “Pattern code” features. In this approach, we simply encode each unique value by its frequency. This method is used to ignore the natural ordered relationship between integer values and to give more importance to the most frequent values in patient waiting time prediction. As for the other categorical features, they are already in numerical mode, so there were no need to use an encoder.

4.5 Machine Learning Models

The next step is to develop learning regression algorithms. The objective of this study is, therefore, to apply more preprocessing methods and then to compare supervised techniques based on linear regression (Huber, Lasso and SVR) which are sensitive to outliers, with a decision tree’ based ensemble method (RF) and a deep neural network’ based method. These algorithms are widely used [3, 5, 6] because they flexibly explore connections in the data, identify relevant features, are compatible with large datasets, and prevent overfitting [6]. As it is a supervised approach, we split the initial dataset into a training dataset (70%) and a test dataset (30%). To emphasize the relevance of the features, we built two versions of prediction models (V1 and V2) according to each feature category (Table 2) to be included. As for the sets of features to be included in the different versions of the predicting model, they are defined as follows:

1. Models V1: This includes Patient-related (**P**), Related to time of arrival (**TA**), Condition severity score based (**CS**) and ED and hospital indicators (**I**) categories for training the prediction algorithms ($P + TA + CS + I$).
2. Models V2: This version includes features used for training models V1 plus the Queue-based (**Q**) features ($P + TA + CS + I + Q$).

We implemented the selected learning techniques by using Anaconda distribution of Python programming languages for data science, Tensorflow 2, Keras and scikit-learn libraries.

To further improve our training model, parameter tuning is conducted using the Grid Search for hyperparameter optimization. For each algorithm, we chose to tune the most important parameters. In the case of SVR, the type of kernel and its degree are adjusted. For RF, the number of trees in the forest is used in addition to the maximum depth of the trees. Finally, when using Lasso, we tuned the alpha constant, controlling the L1 regularization strength.

- **Random Forest (RF)** [7]: RF is a learning technique that generates and combines binary decision trees, while also aggregating the results. Decision trees are constructed by using a bootstrap sample of the training data and randomly choosing a subset of features at each node [3]. As for the fine-tuning, in our case we set the minimum number of samples required to be at a leaf node **min-samples-leaf = 3**, the number of trees in the forest **n-estimators = 40** and the rest as default.
- **Lasso**: Lasso is a regularized linear regression method that can be used to select significant parameters from features. Lasso has the advantage of reducing the number of features in the model. So if a dataset has many features, Lasso can identify and extract the most important ones. However, when a set of features is highly correlated, Lasso tends to randomly choose one of the multicollinear features, regardless of context. For the fine-tuning, in our case we set the alpha constant **alpha = 0.02** and the rest as default.
- **Huber Regressor**: The Huber regressor is a linear ML regression technique that is robust to outliers. The idea is to use a different loss function than traditional least squares. In statistics, the Huber loss is a specific loss function that is often used for situations of robust regression problems where outliers are present that can affect the performance and accuracy of the least-squares-based regression. For the fine-tuning, in our case we used the default parameters.
- **Support Vector Regression**: SVR belongs to the family of generalized linear models that aim to make a predictive decision based on a linear combination of features derived from features. SVR provides the flexibility to define how much error is acceptable in the model and finds an appropriate line (or hyperplane in higher dimensions) to fit the data. For SVR fine-tuning, in our case we set the kernel type to polynomial **kernel = ‘poly’** and the rest as default.
- **Multi-Layer Perceptron (MLP)**: A multi-layer perceptron (MLP) is a feed-forward artificial neural network that generates a set of outputs from inputs. It is characterized by multiple layers of input nodes connected as a directed graph between the input and output layers. As for the network architecture, we set two hidden layers, 25 neurons for the first hidden layer, 18 neurons in the next hidden layer, and one neuron in the output layer.

5 Experimental Results

In this section, the experimental results and model evaluation are presented. Python programming language is used in the implementation phase and a PC

Table 3. Results of the models performance evaluation

	Algorithms	Rolling average	LR	RF	Lasso	Huber	SVR	MLP (Adam)	MLP (Rmsprop)
Model V1	MAE	48.52	43.49	38.97	43.58	41.86	40.62	37.62	37.6
	RMSE	75.53	57.85	53.92	57.93	58.54	59.48	55.86	56.14
Model V2	MAE	48.52	43.03	38.86	43.1	41.51	39.97	37.18	37.29
	RMSE	75.53	57.67	53.73	57.75	58.66	58.89	56.01	55.75

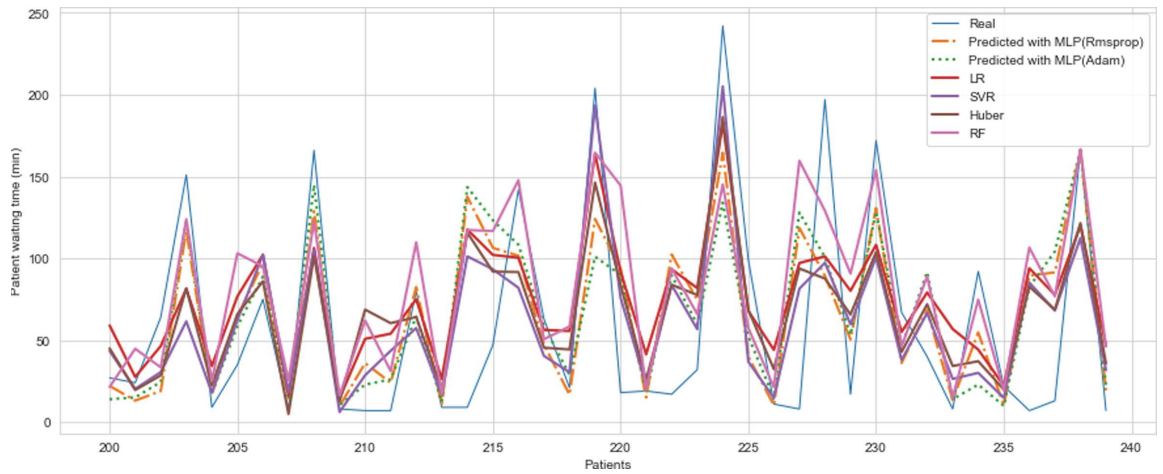
with an Intel (R) Core (TM) i7-10870H processor with 16 GB of RAM. Regression metrics such as MAE and RMSE are used to evaluate the performance of the system. These metrics, most commonly used in the literature, are useful for comparing model performance. MAE represents the absolute difference between the predicted and observed values and its robustness to outliers in the dataset. The RMSE is widely used because of its theoretical significance in statistical modelling [3]. On the other hand, RMSE is sensitive to outliers and for this reason we decided to use MAE with the RMSE.

The results of all the metrics used for the models evaluation (MAE, RMSE) are presented in Table 3 for both model versions (V1 and V2).

Analyzing the results from Table 3 regarding the performance of the models, the model MLP (Adam) outperformed all the other with the lowest value of MAE equal to 37.18 and RMSE equal to 56.01. RF is a method based on Decision Tree and apply bagging algorithms, we noticed that the bagging method is suitable to the dataset used with many outliers, as it has the lowest value of RMSE equal to 53.73.

Concerning the results of ML models applied on dataset with Queue-based features included (models V2), the results have improved but they are very close to those of models V1 comparing the evaluation metrics used.

Figures 3 and 4 show the actual and the predicted patient waiting time for models V1 and V2, respectively.

**Fig. 3.** Waiting time predicted vs. actual for Models V1

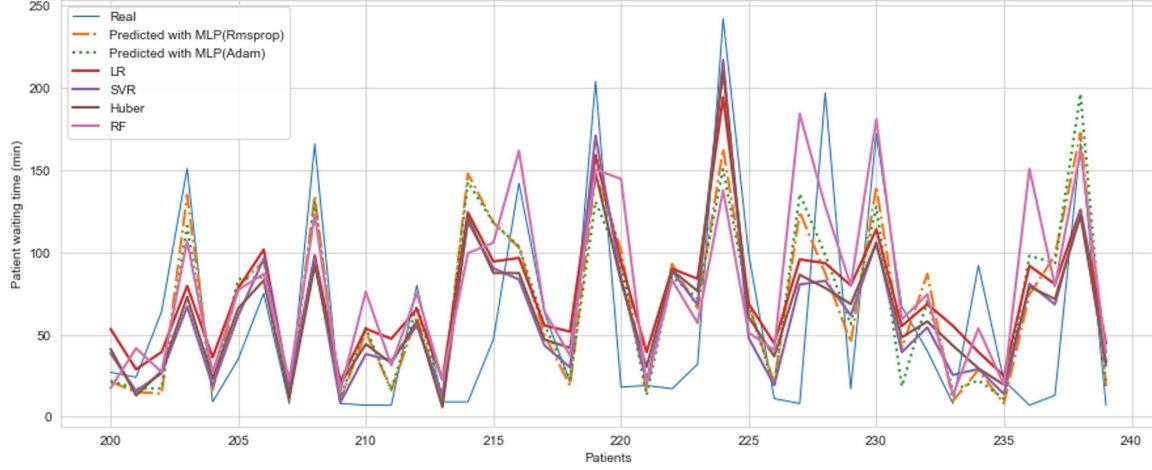


Fig. 4. Waiting time predicted vs. actual for Models V2

The behaviour of the prediction curves is almost the same for both model versions. Models V2 are slightly better in predicting waiting time values that tend to be in the range of outliers. This confirms that Queue-based features can enhance models performance, and that the proposed set of features can be extended further with other categories of features, in order to reduce the prediction error threshold. The high variance in the target feature (waiting time) prevent models from effectively train and causes overfitting. The trained models tend to underestimate the prediction when compared to the real values. RF and MLP models in V1 and V2 are capable of following data trends and learn better than other regression models. This is a promising result in a complex and dynamic context like an ED.

From the experiments, MLP (Adam) outperformed RF, Lasso, Huber regressor and SVR in the MAE metrics whereas RF outperformed Lasso, Huber regressor, SVR MLP (Adam) and MLP (Rmsprop) models in the RMSE metrics. For a real time case study, MLP (Adam) and RF are a good algorithms regarding to the features used in this study. We had to find a good compromise between the two constraints in our design which are a minimum error in predictions and less execution time.

6 Conclusion and Perspectives

This paper proposed a comparative study between ML/DL prediction models, namely Random Forest, Lasso, Huber regressor, SVR and MLP models (with Adam and Rmsprop optimizer) to evaluate ED patient's waiting times using real data. We used queue-based features combined with patient-related, condition severity-based, ED and hospital indicators, and time of arrival based features to optimize patient waiting time prediction along with ML/DL techniques instead of traditional approaches (Rolling average).

The study contributes to the current literature as an extensive and structured comparative analysis that establishes the real value of different learning

techniques in complex and dynamic environments like EDs. We have demonstrated the accuracy and suitability of predictive models in a real application context. This study also offers a practical contribution through the use of real data from an ED.

The results showed that the MLP model achieved better prediction accuracy (based on the MAE metric) than the traditional approaches. Future and extended work of this research are as follows: More information from ED could be implemented into the model, like parameters for medical staff schedules. In addition, different datasets from other hospitals and sites could be implemented. Furthermore, various machine learning algorithms could be further applied, including linear and nonlinear regression like the XGBoost ML algorithm. Finally, meteorological and event-related features could be used to extend the feature set and integrate environmental effects.

References

1. Amor, L.B., Lahyani, I., Jmaiel, M.: Recursive and rolling windows for medical time series forecasting: a comparative study. In: 2016 IEEE Intl Conference on Computational Science and Engineering, CSE 2016, and IEEE Intl Conference on Embedded and Ubiquitous Computing, EUC 2016, and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering, DCABES 2016, Paris, France, 24–26 August 2016, pp. 106–113. IEEE Computer Society (2016). <https://doi.org/10.1109/CSE-EUC-DCABES.2016.169>
2. Ataman, M.G., Saryer, G.: Predicting waiting and treatment times in emergency departments using ordinal logistic regression models. *Am. J. Emerg. Med.* **46**, 45–50 (2021)
3. Benevento, E., Aloini, D., Squicciarini, N.: Towards a real-time prediction of waiting times in emergency departments: a comparative analysis of machine learning techniques. *Int. J. Forecast.* (2021)
4. Hemaya, S.A., Locker, T.E.: How accurate are predicted waiting times, determined upon a patient’s arrival in the emergency department? *Emerg. Med. J.* **29**(4), 316–318 (2012)
5. Hijry, H., Olawoyin, R.: Predicting patient waiting time in the queue system using deep learning algorithms in the emergency room. *Int. J. Ind. Eng.* **3**(1) (2021)
6. Pak, A., Gannon, B., Staib, A., et al.: Forecasting waiting time to treatment for emergency department patients. *OSF Preprints* **26** (2020)
7. Patil, P., Thakur, S.D., Kasap, N.: Patient waiting time prediction in hospital queuing system using improved random forest in big data. In: 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT) (2019)
8. Soremekun, O.A., Takayesu, J.K., Bohan, S.J.: Framework for analyzing wait times and other factors that impact patient satisfaction in the emergency department. *J. Emerg. Med.* **41**(6), 686–692 (2011)
9. Tomar, D., Agarwal, S.: A survey on data mining approaches for healthcare. *Int. J. Bio-Sci. Bio-Technol.* **5**(5), 241–266 (2013)
10. Ward, P.R., et al.: ‘waiting for’ and ‘waiting in’ public and private hospitals: a qualitative study of patient trust in south Australia. *BMC Health Serv. Res.* **17**(1), 1–11 (2017)