Towards predicting waiting times in Hospital Emergency Rooms

Rui Tiago Eirinha de Almeida
rui.eirinha@ist.utl.pt

Instituto Superior Técnico, Lisboa, Portugal

October 2020

Abstract

The prediction of waiting times at hospital's emergency rooms is something that has a big impact for citizens. To decide what hospital to go to at a specific time can change the way we deal with unfortunate events in our lives and of those close to us. We address this issue by initially collecting data from the National Healthcare Service – forming a real-world dataset – in order to find relevant information about the waiting times for care at the emergency rooms of hospitals in the Lisbon area; we also include in this study a process of sentiment analysis applied to data collected from the social network Twitter, creating a new feature that can be seen as a "proxy sentiment" that is extracted from each collected *tweet*. We then prepare this waiting times data for time series analysis, and perform exploratory data analysis, finding first evidences of seasonal behavior in the daily variations of waiting times. In order to tackle the forecasting problem, we make use of traditional time series models like Auto Regressive Integrated Moving Average (ARIMA) and compare their performance for different time horizons of forecast. We do this to address patient and hospital concerns, i.e. for the patient is likely to be more impactful the forecast in short-term, whereas the hospital need longer-term forecasts to better manage their resources. To address the seasonal behavior of the waiting times, we use the open source software Prophet from Facebook, an additive model capable of fitting non-linear trends concerning the seasonality period. We then finally conduct an initial approach, within the context of Probabilistic Graphical models, using the Hidden Markov model to capture multimodal dynamics present in the time series.

Keywords: waiting times, time series, forecasting, arima, probabilistic graphical models, hidden markov model

1. Introduction

The emergency department (ED) of an hospital is often affected by overcrowding due to high demands for emergency care, representing the largest source of hospital admissions. As a result, patients' satisfaction tends to reduce due to the unexpected waiting time, the health provider increases its workload, and the quality of treatment and prognosis is consequently affected [1][2]. In these conditions, it is very likely for patients waiting in the emergency room to have less favourable outcomes, and the ED's resources eventually reach an unsupportable level of demand, which may lead to severe situations regarding the patient health condition [1][3][4]. Hence, the management of the EDs inflow and their resources are seen as a field of interest in medical research and a critical issue by both patients and hospital staff.

The importance of admission forecasting is intrinsically related to the optimization of hospital resources. In fact, early identification of emergency department patients who are likely to require admission may contribute to a better patient-flow in emergency department [5], avoiding the problem of overcrowding and leading to an upgrade on managing ED resources. The scope of this thesis does not follow the concept of admission forecasting, because we aim to address the problem of emergency rooms waiting times between the time of triage and the possible admission. Moreover, our data is completely anonymous. We do not have access nor authorization to use data from patients as it happens in [3] study, for instance.

The goals of this thesis are twofold: (1) collect and prepare relevant datasets for prediction of waiting times at ED; (2) explore predictive models for these waiting times. To accomplish these objectives we begin by collecting data from the National Healthcare Service, using web scraping techniques, over regular time intervals, so we can gather information about those waiting times. We are also interested on gathering information from other type of sources that potentially contain relevant information to forecasting. The process of collecting data is a crucial step in the first development stages as it eventually determines if we can even tackle the

question(s) that we formulate for this study, which might be, for instance, whether the data is available or not in the first place [6].

We exploit the possibility of having behavioral changes on waiting times in emergency rooms that can be explained as a result of the influence of external factors (exogeneous variables). The use of trending Internet searches and *tweets* has already been proposed for real-time prediction of outbreaks, which is known as the field of "digital epidemiology" [7][8]. During the analysis process, we also create a new feature from Twitter data by applying a process of sentiment analysis [9], capable of computing a polarity value from each collected *tweet*, and it can be seen as a "proxy sentiment" incorporated in those mass media and social media *tweets*.

The time series forecasting problem is addressed by using and comparing two main approaches: (1) traditional times series forecasting methods as tools of statistical models; (2) probabilistic Graphical models, more specifically the Hidden Markov model. For the first point, regarding the scope of this thesis, we use the Auto-Regressive Integrated Moving Average (ARIMA) models (including ARIMAX, which is able to incorporate exogenous variables in the model). Additionally, we also include in this study other statistical form of time series model: the Facebook Prophet package, which enable us to explore some components of the time series like seasonality.

The use of Hidden Markov model for this study is motivated by the necessity of covering potential multimodal dynamics in the time series data, which are a sort of behaviors that ARIMA models are not able to capture. Thus, the Hidden Markov model exploits an hidden regime switching process, where we know/learn the probability of the regime switching, and for each regime it is captured a different dynamics from the associated time series.

2. Data Collection

The data used for this study is collected from three different sources: National Healthcare Service (SNS), Twitter and Portuguese Institute for Sea and Atmosphere (IPMA). In the case of Twitter and IPMA, we were able to collect data through the available APIs provided by corresponding sources. In other hand, to collect data from SNS, we create and automate a process of web scraping, which is basically about extracting pieces of data from web pages, in a quick, efficient and automated manner, offering data in a more structured and easier to us format and storing it in a central local database or a spreadsheet for later retrieval or analysis [10][11]. Among the multiples techniques and methodologies available, we used Scrapy, which is a robust and high level web framework, written in Python, widely used for scraping data from various sources.

2.1. SNS Emergency Room Waiting Times and Queue Size

Since we are concerned about people's behavior on their daily life, information gathered from emergency rooms at some specific hospital plays an initial and important role, because we can get precious insights, possibly new research question, on city patterns.

We collect data from `http://tempos.min-saude.pt` web page, belonging to the SNS domain, here we can access information about waiting times and queues on hospitals' emergency rooms. It is worth mentioning that we are focus on (public) hospitals in Lisbon that have real-time data available. Therefore, the data is collected from four hospitals: Hospital Santa Maria, Hospital Dona Estefânia, Hospital São José and Hospital São Francisco Xavier. Regarding the scope of this thesis, we use only data from Hospital Santa Maria.

Each one of these hospitals has a different web page layout with tables representing the urgency type in the Manchester coding scheme. They contain relevant information about waiting times and the amount of people waiting on emergency rooms at those hospitals, the correspondent emergency level and the type of speciality. Figure 1 illustrates one example retrieved from Hospital Santa Maria web page. The data collected is then stored i a CSV file with following headers: "Hospital" (ID of the hospital, for example 216 corresponding to Hospital São Francisco Xavier), "Urgency Type" (table's title, like "Urgência Geral", the emergency room), "Service" (speciality type), "Emergency Stage" (contains the different states of healthcare severity (Emergente - 5, Muito Urgente - 4, Urgente - 3, Menos Urgente - 2, Não Urgente - 1)), "Waiting Time" (average waiting time on the last 2 hours), "People Waiting" (how many people are waiting), "Acquisition Time" (when does the data extraction occur). This data is collected every 10 minutes.

Figure 1: Waiting Times' table from Hospital Santa Maria's web page.

2.2. Mass Media and Social Data

For simplicity of access and uniform data processing, we collected mass media news from their Twitter outlet instead of having to deal with diverse web formats and changing platforms. Also, we checked that mass media Twitter outlets do provide a frequent stream of posts.

Twitter dataset is build upon a web scraping task, written in Python, and it is available on an open source project on GitHub (`https://github.com/taspinar/twitterscraper`). The whole dataset has information from eighteen Portuguese mass media users. Using this tool, we can get relevant attributes from each *tweet*: "User" (twitter handle), "Text" (the *tweet* itself), "Likes" (number of *likes*), "Retweets" (number of *retweets*), "Data Time" (*tweets*'s date).

2.3. Weather Forecasts

Portuguese Institute for Sea and Atmosphere (IPMA) dataset contains information about weather forecasting and, fortunately, IPMA website provides an API that allows us to get some attributes about it: "date time" (register date of the weather temperature), "tempC" (hourly temperature throughout the day), "maxtempC" (maximum temperature value for each day), "mintempC" (minimum temperature value for each day). This data is collected once a day.

3. Exploratory Data Analysis

3.1. SNS Dataset

The 'raw dataset' has 7 columns, almost 3 million rows and about 1.57×10^6 of those have missing data, so they are removed. In fact, we have a huge amount of data that is removed and this case is mostly related to the most emergent stage ('Muito Urgente'), each of which has a `NaN` value assigned to the corresponding 'Waiting Time' value, as well as in the emergency stage 4, which occasionally occurs for certain periods of time during the day, mostly in the morning. Therefore, the dataset does not contain any information related to waiting times at the emergency stage 5. Naturally, we expect no waiting time for a situation as severe.

The final dataset has information from November 15 2017 to June 15 2018.

3.1.1 People Waiting

Regarding the distribution of this variable, from Figure 2, we see that the queue sizes have a similar distribution for all emergency stages. It is more common to have a small amount of people waiting on the queue and we see that its range gets larger at the emergency stages 2 and 3, probably because there are less cases for the emergency stage 1 and, at the stage 4, we hope there are not so many people waiting. Considering the whole historical data, the queue, at emergency stages 2 and 3, is more likely to be congested.

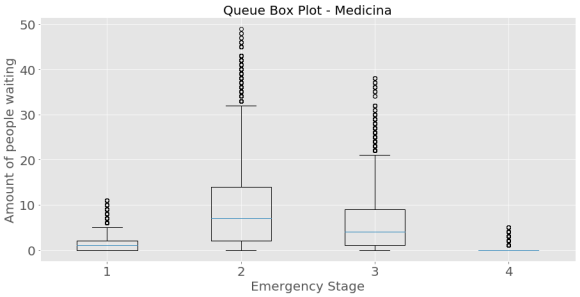
We also want to understand if there are any patterns in time series analysis. Figure 3 presents the queue evolution over time concerning two cases: Figure 3(a), where all values from variable 'People Waiting' are used. This is the 'raw view' of the time series; Figure 3(b), where we apply a 'daily mode' to the values of the variable 'People Waiting'. Instead of plotting a massive amount of points (former point), which does not contribute to any pattern-finding and is difficult to interpret, we enhance the graphical display of the time series in a way we believe allows for a deeper insight into this variable and summarizes its original representation. It shows the most frequent number of people waiting on queue for each day; note that we do not use the mean. 'People Waiting' is a discrete variable, so it does not make sense to get an average number for the queue.

It is quite clear that there is a periodic pattern and we might say that, usually, emergency stages 2 and 3 have similar behavior over time. Also, at the beginning of the months December and April, we have a peak affluence of people at emergency rooms on the emergency stage 3; note that there is a little gap between February and March. This was due to a extraction problem in data collection, where we were unable to reach the web pages.

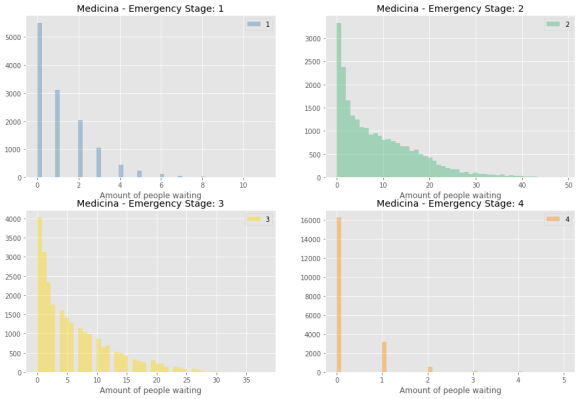
3.1.2 Waiting Time

Figure 4 illustrates the waiting times distribution by emergency stage. From Figure 4(a), we see that the distribution is positively skewed by observing the long tail in the top side of each emergency stage, and this information is complemented with Figure 4(b). The distribution for each emergency stage seems natural even though we expected less occurrences for larger waiting times.

This data has a massive amount of points and, therefore, it is more difficult to identify some kind of behavior. So we try to overcome this issue by smoothing the time series. Smoothing is usually done to help us better see patterns in time series by smoothing out the irregular roughness to see a clearer signal. To do so, we apply the moving average method, which consists on determining, at each point in time, averages of observed values that surround a particular time. Figure 5 shows the time series behavior from November 15 2017 to June 15 2018. From figure 5(a), we see a clear up trend, between December and January, for emergency stage 1 and another up trend from middle of January to middle of February, including again emergency stage 1 and also both emergency stages 2 and 3.

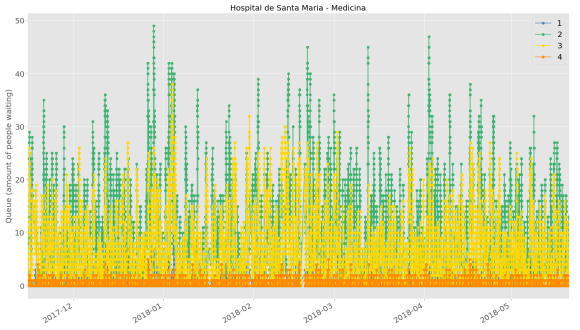


(a) Box plot - Queue distribution by emergency stage.

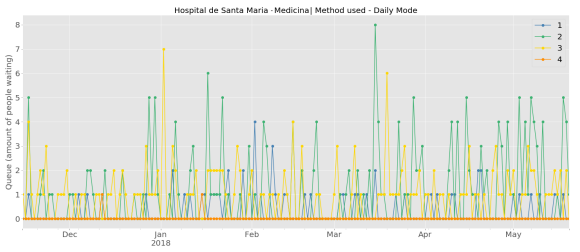


(b) Histograms for each emergency stage - Queue distribution.

Figure 2: Queue Distribution in Hospital Santa Maria.



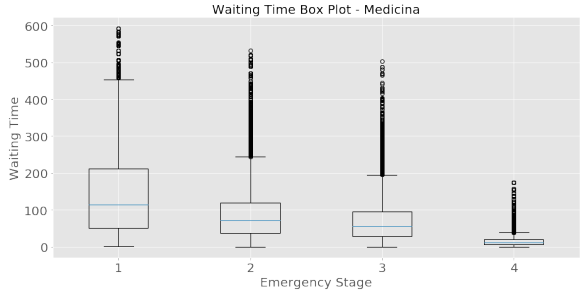
(a) Time Series Plot - global view.



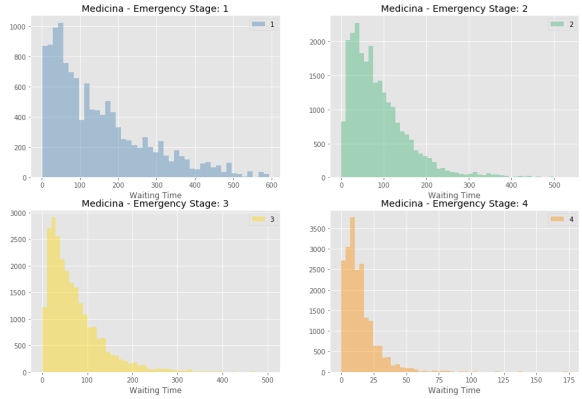
(b) Time Series Plot - mode view.

Figure 3: Queue variation over time in Hospital Santa Maria.

We also have a very interesting visualization in Figure 5(b) that shows an 'average behavior' of the time series in a day interval. So we first take the



(a) Box plot - Waiting times distribution by emergency stage.



(b) Histograms for each emergency stage - Waiting times distribution.

Figure 4: Waiting Times Distribution in Hospital Santa Maria.

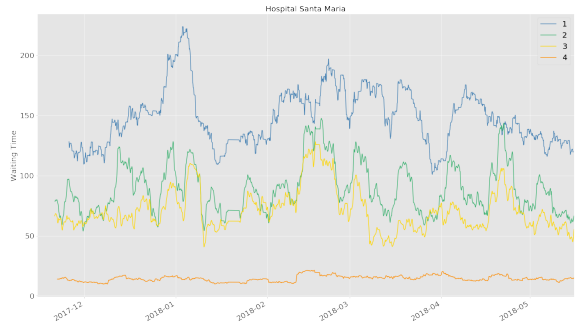
average waiting time for each minute concerning all values on this time series, and then we apply a moving average with a window size of one hour to get a clear signal. Hence, from Figure 5(b) we see that, on average, waiting time increases between midday and midnight, and then decreases until it reaches midday again. This behavior is more evident for the 3 less severe emergency stages.

To better analyse the periodicity in our time series, we inspect its autocorrelation plot, which addresses th issue of time dependence between time points. In Figure 6, We can confirm see that there is a high autocorrelation at 1 and then slowly declines, continues decreasing until it becomes negative. The peak around lags 140 and 280 might suggest some daily seasonal behavior; note that one day corresponds to roughly 144 consecutive points in our time series.

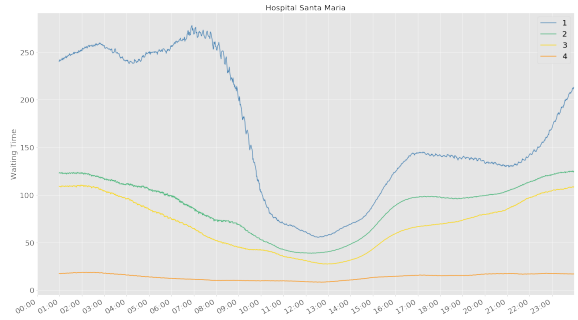
4. Methods

4.1. First modelling approaches

One of the main purposes of this project calls for time series analysis, where we try to correctly forecast the waiting time values using some classic statistical time series modelling approaches like Auto Regressive Integrated Moving Average (ARIMA) models. To perform this task, it is necessary to understand what kind of behaviors are present on



(a) Time Series Plot - Waiting evolution, by emergency stage, over time using moving average with size window of 5 days



(b) Time Series Plot - Average waiting time evolution during a day, by emergency stage, using moving average with size window of 1 hour.

Figure 5: Time Series Behavior in Hospital Santa Maria.

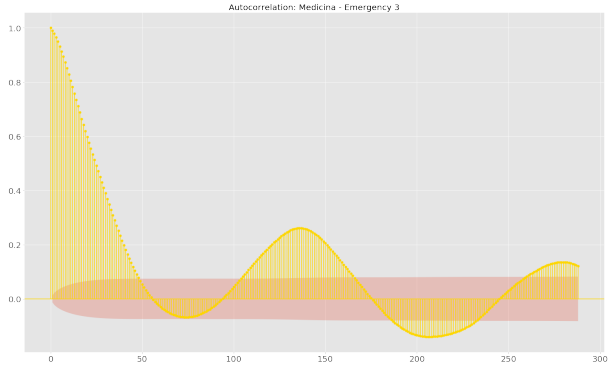


Figure 6: Autocorrelation Plot - Medicina — Emergency Stage 3.

time series. In the previous section, we took a look at some plots which gave us some insights on time series properties; note that we are considering only the emergency stage 3 and the medicine speciality of Hospital Santa Maria.

The temporal structure of the time series adds an order to the observations. This means that there are some important assumptions associated to the regularity of the time series that must be handled in order to correctly model this structure. Hence, one important property that characterizes the regularity of the time series is *stationarity*. When modelling, it makes the assumption that the joint distribution of a set of observations $\{y_{t_1}, y_{t_2}, \ldots, y_{t_k}\}$ is exactly the same as the joint distribution of the time-shifted set of observations $\{y_{t_{1+l}}, y_{t_{2+l}}, \ldots, y_{t_{k+l}}\}$. This gives us statistical properties that allows us to use various models for forecasting. Essentially, a stationary time series does not change its statistical properties over time, it implies statistical equilibrium or stability with constant mean and variance, and autocovariance does not depend on the time. In terms of basic tests to figure out whether the data is stationary or not, the Augmented Dickey-Fuller (ADF)– *unit root test* –, which is a statistical hypothesis test, is commonly used for this purpose. In this test, the null hypothesis is that the time series is not stationary. The alternative hypothesis – rejecting the null hypothesis – defines the time series as stationary.

In Section 3, we saw that the original time series plot does not provide any noticeable pattern at the first sight. In fact, although we can see several spikes throughout the series, which may imply some outliers, the time series seems to vary around a fixed level. Moreover, applying the ADF test to the data, we see that it succeeds to reject the null-hypothesis that a unit root is present, and hence we have a stationary time series. Additionally, as a complement, Figure 7 illustrates the original time series alongside its rolling mean and rolling standard deviation. It is possible to see that the original time series does not present a visible trend, and both rolling mean and rolling standard deviation share a stable behavior throughout the series, which means that the original time series is likely to already follow the assumption of stationarity. Nonetheless, in order to make this time series more stationary, it is considered a good practice to apply some transformations (take the log and square root, smoothing, differencing, etc.) and re-evaluate the behavior of the transformed time series with those stationarity tests.

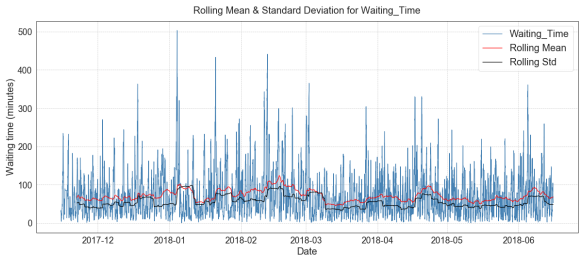


Figure 7: Time Series Plot - Original Time Series alongside its rolling mean and rolling standard deviation.

We also include in this study Facebook Prophet package. In comparison to those classic statistical

time series models, Prophet uses a slightly different statistical form of time series model, which are component models (or additive models), and it allows us to estimate effects from day of week, day of year, time of the day, holiday, and include trend trajectory. We use this package as an alternative method to those mentioned before and, specifically, to explore and address the potential seasonal behavior of the time series.

4.2. Validation metrics and model selection

For model selection, we choose to work with two methods: *Akaike's Information Criteria* (AIC) and cross-validation. In the first case, model selection solution is done by identifying the model that minimizes information loss. This evaluation is given by

$$AIC = -2\log(L) + 2(k+2), \tag{1}$$

where L denotes the likelihood of the data, and k is the number of parameters in the model, which includes the variance of the residuals, σ^2. The idea is to penalize the sum of squared residuals due to the addition of parameters in the model. In this regard, the best model is the one with the minimum value of the AIC.

In other hand, cross-validation is a search metric where the model score results from the average of forecasting error, e, across k folds. For this method, training data is split into k folds, each one containing a training and test sets. The model is trained on the training sets across the k folds, followed by an evaluation of the predictive accuracy on the corresponding test sets. Because we are dealing with time series, cross-validation is conducted differently regarding the standard k-fold cross-validation. It has to do to sequential dependencies between observations [12]. In this sense, cross-validation metric might be the best solution since it concentrates its efforts on forecasting performance which is crucial when selecting the best model. However, the downside is that it is more expensive to compute than the AIC solution, because the model must be trained k times, leading the process to massive computational times. Note that the training set contains more than 20000 time-units. For that reason, cross-validation is not an option for our case.

Therefore, in order to select the best model, we first split our dataset D into train set T and test set T_{test}, 90% and 10% of its samples, respectively. From the train set T, 10% of its samples (from the last part of the train set) are selected to create a validation set V, and the remaining part (train subset T_s) is used to train. In the case of the regression models, the AIC criteria is used to select the best model among a specific family (we consider AR, ARMA and ARIMA). After that, we compute the validation error for each one of the final candidates using the validation set V. The best model is the one with the lowest validation error. To calculate this error, we use the root mean square error (RMSE) as it penalizes large errors.

4.3. Regression models

4.3.1 Autoregressive Integrated Moving Average (ARIMA) model

ARIMA is a consequent extension of the ARMA model (we address this model in more detail in our thesis) with a dth order of *differencing*. As a result of this differencing operation, the 'Integrated' term is added to the original name. Usually, ARIMA model is written as ARIMA(p, d, q), where p is the order of autoregressive term, q stands for the order of the moving average term, and d denotes the order of differencing. The model is expressed as follows:

$$\begin{aligned} y_t^{'} = c + \phi_1 y_{t-1}^{'} + \cdots + \phi_1 y_{t-p}^{'} + \\ + \theta_1 \epsilon_{t-1} + \cdots + \theta_q \epsilon_{t-q} + \epsilon_t \end{aligned} \tag{2}$$

where $y_t^{'}$ denotes the differenced series.

The differencing operation is used to perform detrending in time series data, it computes the difference between successive observations. Thus, we have

$$y_t^{'} = y_t - y_{t-1}, \tag{3}$$

which denotes the first order differencing.

ARIMA model has a good performance only in short-term forecasting, meaning that its forecasting power decreases as the forecasting horizon increases. ARIMA forecast converges to the mean of the observations as the forecast horizon grows [12].

4.3.2 ARIMA model with explanatory variables (ARIMAX)

Regarding the scope of this thesis, we also want to explore the influence of external factors on our dependent variable of interest. Hence, ARIMAX accounts for that external factor by adding the exogenous regressive covariate.

In terms of mathematical definition, this model has a similar formulation in comparison to the other ARIMA models, simply differing on the addition of that variable X. Thus, we have:

$$\begin{aligned} y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=1}^{q} \theta_j \epsilon_{t-j} + \\ + \epsilon_t + \sum_{m=1}^{M} \beta_m X_{m,t}, \end{aligned} \tag{4}$$

where the term $\sum_{m=1}^{M} \beta_m X_{m,t}$ represents the effect of the exogenous variables, X_m, on the dependent/endogenous variable, y_t. β_m is an additional parameter associated to the new explanatory variable.

The data collected from the National Health Service plays a central role as it contains the time series data of waiting times from Lisbon's hospitals. Additionally, and concerning the matter of this section, we have data from other sources such as Twitter and IPMA, with which we are able to create new features for the corresponding datasets and use them to define the explanatory variables of ARIMAX model.

The datasets mentioned above contain different types of features, whose sets of values have different magnitudes as well, so it is likely to compromise the performance of the model. For this reason, before using ARIMAX model to fit our data, we need to do feature scaling for every variable (endogenous and exogenous). In this case, we use Standardization (or Z-Score Normalization). With this process, the features are transformed in such a way that their distributions are converted to a standard normal distribution with a mean of zero ($\mu = 0$) and a standard deviation of one ($\sigma = 1$).

4.3.3 Prophet

Prophet is a forecasting tool made available by Facebook, and its procedure is described by an additive regression model (or decomposable time series model) comprising trend, seasonality and holidays [13]:

$$y(t) = g(t) + s(t) + h(t) + \epsilon(t) \tag{5}$$

where $g(t)$ is the trend function, which automatically detects changes in trends by modelling non-periodic changes in the value of time series; $s(t)$ denotes periodic changes, i.e. weekly and yearly seasonality; $h(t)$ is the holiday effect. The error term, $\epsilon(t)$, represents something that the model cannot explain. In our case, we are interested in capture the seasonal behavior of the time series.

The idea of Prophet differs from traditional time series model in the sense that it "frames the forecasting problem as a flexible regression model as curve-fitting exercise" instead of explicitly account for the temporal dependencies in data structure.

4.3.4 Training and validation

To train the ARIMA models, we used the train subset T_s. It has roughly 2.4×10^4 samples and contains time series data from November 15 2017 to May 06 2018. After fitting a model, we proceeded with the model diagnostic checking, where we conducted a visual inspection of the residual (estimates of the true error) plots. These diagnostic graphs allow us to check whether the underlying assumptions, previously stated, are true. Considering y_i as the observed value and $\hat{y}_i$ as the fitted value, the residuals are then described by the following expression:

$$e_i = y_i - \hat{y}_i, i = 1, 2, ..., N, \tag{6}$$

where N is the length of the time series in T_s.

Regarding the scope of this thesis, and taking into account the contrast between the performances of model fitting and forecasting, we decided to evaluate and compare several models in order to get an overview of different approaches when modelling this time series data and how they relate to each other.

Within the range of ARIMA and Prophet models, we used 20 models to fit data. These models are summarized alongside their train and validation sets performances in the thesis. The models were initially divided into two groups, where we addressed the ARIMA and Prophet models in one side and investigated ARIMAX model with different exogenous variables in the other. Additionally, we also decided to include the same models with a log transformation applied to the data, which contributed for adding two more groups of models.

During the fitting procedure, for each ARIMA model, we conducted a grid search process, based on the AIC criteria, to select the best order for each model parameter (p, q and d). This process was the first step for comparing the different models and evaluating the performance of model fitting in the train subset, T_s. Then, we proceeded with the aforementioned model diagnostic checking. In general, models had similar outcomes for their residuals (the models that fitted data in which the logarithm was not applied had worse results for residuals); note that, for this evaluation, we are not considering the Prophet model, which is address in the end of this section. Hence, we illustrate the example of ARIMA model with parameters $p = 3$, $d = 1$ and $q = 9$, and application of the log transformation to the original time series. Figure 8 present the residual plots of this model, where it seems that we have two different sides in terms of normality conditions, which is an element we have to take into account regarding the underlying assumptions. In the top plot, it seems that the residuals fluctuate around a mean of zero and have a constant variance. Also, the autocorrelation plot shows that the residuals are not autocorrelated, and hence indicates that there is no pattern in residuals data. In other hand, the histogram and normality Q-Q plots suggest that the distribution is skewed, and data is likely to have outliers. In fact, normality Q-Q plot clearly shows deviations from the straight line, which indicates a non-normal distribution of the residuals. Finally, the lower left plot shows how residuals and fitted values relate to one another. In this case, we can see that the variance is not constant throughout the series, and endorses the idea of having outliers due to

some isolated points in the graph. In conclusion, we can assume that there is some information remaining in the model's residuals. However, and despite these issues, we decided to continue the evaluation of the identified model and investigate its forecasting performance before handling possible outliers. Moreover, this inconsistency in residuals might also suggest another level of complexity in the data that ARIMA model is not able to capture.

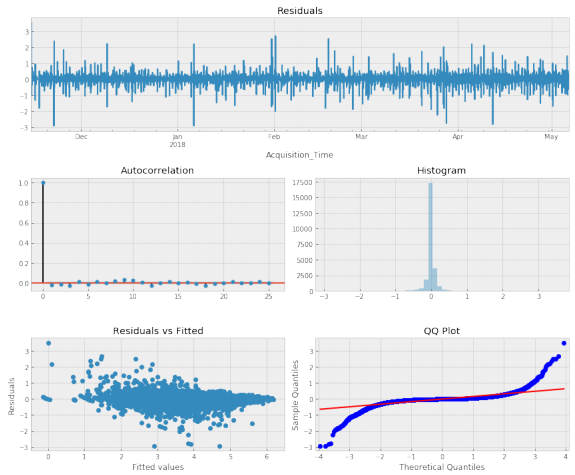


Figure 8: Residual plots from ARIMA(3,1,9) model.

Figure 9 shows the fitted values and forecasts produced by ARIMA(3,1,9) model, where we can see that the sample fit of train subset looks well suited (see the blue and yellow lines). However, looking at the forecasts for the validation set (green line), it is clear that the forecasting performance of this model deteriorates comparing to its sample fit. It lacks of information throughout validation time for making more accurate forecasts.

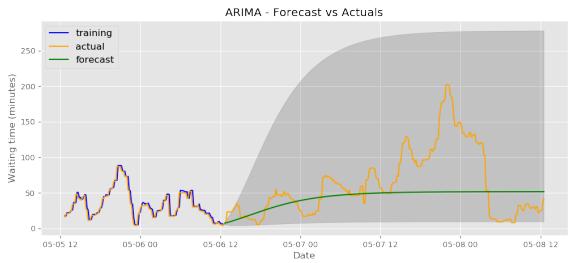


Figure 9: Forecasting performance of ARIMA(3,1,9) model for two days of forecasting steps.

Because it is an different approach in comparison to the ARIMA models, we also include Prophet (without the log transformation applied to original time series) in this visual analysis. In this case, it contrasts with the other models, because it captures the daily seasonal behaviour of the time series, so we decided to visualize the residuals plot and the forecasts for an horizon of two days. In the case of the residuals, looking at Figure 10, we can clearly see that there is some information left over. The residuals present autocorrelation, the variance does not seem to be constant looking at the bottom left graph. The histogram and the Q-Q plot show us, in other hand, that the residuals are close to a normal distribution. Lastly, in Figure 11 is possible to see this roughly daily behavior produced by the model, and it seems to follow the variation of the observed values (yellow line) in the validation set. Moreover, Prophet achieved the best performance results for the 2 days-forecast horizon. These results are described with detail in the thesis full document, where it is possible to compare all the performance results of the models.

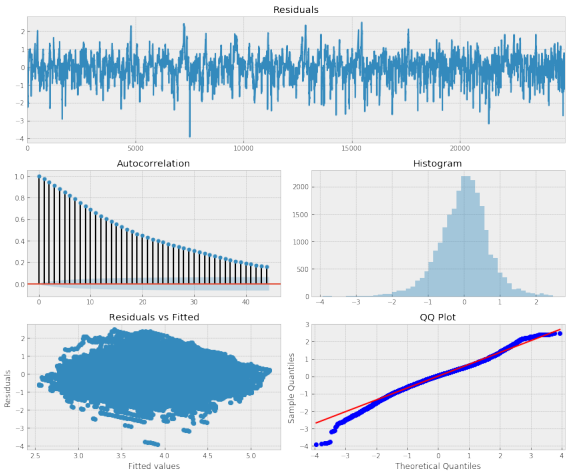


Figure 10: Residuals plots for the Prophet model.

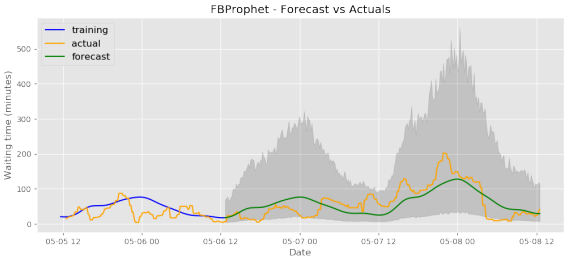


Figure 11: Forecasting performance of Prophet model for two days of forecasting steps.

4.4. Hidden Markov Model (HMM)

Lastly, we address the HMM approach for modelling the waiting times in Hospital Santa Maria at emergency stage 3 as an alternative technique in comparison to the previous enunciated models. The Hidden Markov Model is a state space model in the sense that there is an underlying state that we cannot observe and the corresponding output that we can observe. It broadens our perspective in

such a way that we can explore this idea of having an observation that is not in the same physical or conceptual category as the underlying state that is producing it [14].

4.4.1 Modelling waiting times and results

For the purpose of modelling our time series data, and taking into account the existence of a regime switching process, we considered the "proxy sentiments", associated with the computed polarities (-1, 0 and 1) in the Twitter dataset, as a starting point for defining the number of states. Thus, given the distribution of these polarities within Twitter dataset, we were able to initialize two model parameters, π and A, before applying the Baum-Welch algorithm, with the initial state distribution and transition probabilities, respectively. Emission probabilities are considered to follow a normal distribution, so for each state the Baum-Welch algorithm estimates a set of parameters, $\phi = (\mu, \sigma^2)$, that are believed to govern the distribution of the observations generated by a particular state.

Figure 12 presents the visualization of true observations from the validation set plotted alongside the observations generated by the 3-state Hidden Markov model.

We also included in this analysis an Hidden Markov model without parameter initialization and with a selection of number of states (7) produced by a grid search process, where the selection of the best model was made according to the AIC criteria within a range of ten states. In Table 1, it is presented the results of these models ("GHMM" stands for Gaussian Hidden Markov model). For the validation part, we sampled the corresponding observations from the trained HMMs.

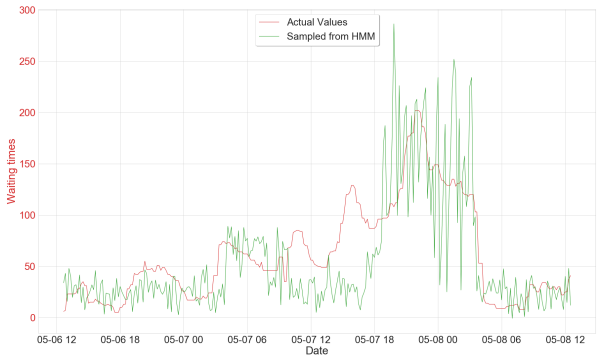


Figure 12: Observations generated by the 3-state Hidden Markov model.

4.5. Test set results

After finding the models that better performance on the validation set for the different forecast horizons, we train the same models using the exact same setup of parameters and the whole training set. The test set results can be seen in the Table 2. The most interesting results come from ARIMAX model not only because it performs well in the test set (even in the case of an horizon of two days in comparison to the other models), but also because the exogenous variable "daily weather temperatures" reveals to be statistically significant with respect to the waiting times (dependent variable), with a p-value of 0.07.

Table 1: Train and validation results of Hidden Markov Model.

	TRAIN		VALIDATION					
			1 hour		2 hour		2 days	
Model	Log Likelihood	AIC	RMSE	Std. dev. error	RMSE	Std. dev. error	RMSE	Std. dev. error
GHMM3	-110972.82	221973.64	22.56	16.03	17.7	15.78	40.46	39.84
GHMM7	-94708.57	189541.14	21.94	10.01	16.64	12.34	38.63	45.2

Table 2: Test set results.

	TEST					
	1 hour		2 hour		2 days	
Model	RMSE	Std. dev. error	RMSE	Std. dev. error	RMSE	Std. dev. error
ARIMAX	3.31	3.06	5.83	5.03	59.49	50.46
Prophet	6.86	6.46	6.36	6.08	49.12	43.95
GHMM7	8.05	5.59	7.46	6.89	58.36	55.23

5. Conclusions

In this thesis, we addressed the problem of time series analysis and forecasting on the waiting times for care of emergency rooms in Lisbon, Portugal. We performed extensive data collection, data preparation and cleaning, creating a large and high quality datasets, and tested predictive models for time-correlated data for Hospital de Santa Maria, medicine emergency rooms, focusing on the emergency level "Urgent". We began our study by creating and automating a web scraping procedure so we could collect this data from National Healthcare Service; the data collected also covered other hospitals, and might be relevant to address the same problem for those cases in future work.

To model waiting times, we experimented several approaches within the context of ARIMA models. Comparing the results of all these models, wee see that, for short-term forecasting (up to 2 hours), the overall performance is similar, being ARIMAX model, with daily weather temperatures as exogenous variable, the one with the best results. For the validation set results, taking into account the fitting procedure and its descriptive statistics, we saw that it did not reach statistical significance. However, when the model estimated its parameters using the whole training set, the exogenous variable increased its impact on the waiting times variable.

Due to the low complexity of these models, splitting the time series data into relevant periods of time (e.g.: yearly seasons, typical "season diseases") can be useful not only for improving the

model performance but also to more clearly interpret the results.

Considering the Prophet model, we see that this method is able to improve the results of long-term forecasting as it captures the daily seasonal behavior of the time series data, corroborating with our prior analysis where we found this potential nature of the data. The model we assumed with Prophet can be more complex in a future work by adding other effects (eg.: holidays, specific events in the calendar, change points in the series, etc.).

The results of the former mentioned models endorse the use of probabilistic graphical model approach in order to be able to capture the multimodal dynamics of the time series. The application of the Hidden Markov model conceptualizes this idea of having a time series with different regimes that are evolving over time, and this model is used for detecting the changes in those regimes and interpret the underlying structure of the time series. For future work, we can assume different approaches concerning the initialization of the model parameters, as well as the incorporation of other variables that we believe to belong to the same underlying state.

We can also use in the future Long Short-Term Memory networks (LSTMs) to model the time series data. In this case, due to its recurrent nature, the idea is to continuously feed the estimates of the observations with previous information from the other observations. The difference in this model is the capability of capture longer-term dependencies in a sequence. However, due to the complexity of the model, we can lack of interpretability when analysing the results, which is a major drawback for healthcare applications.

The process of sentiment analysis should be improved in the future in such a way that we can extract the context of the *tweet*. Taking into consideration this "feature", it allows us to better interpret the analysis of the data, which might contribute to new insights, and add evidence to the models.

References

[1] Hye Jin Kam, Jin Ok Sung, and Rae Woong Park. Prediction of Daily Patient Numbers for a Regional Emergency Medical Center using Time Series Analysis. *Healthcare Informatics Research*, 16(3):158, 2010.

[2] Izabel Marcilio, Shakoor Hajat, and Nelson Gouveia. Forecasting Daily Emergency Department Visits Using Calendar Variables and Ambient Temperature Readings. *Academic Emergency Medicine*, 20(8):769–777, August 2013.

[3] Predicting hospital admission at emergency department triage using machine learning. 13.

[4] Caroline Berchet. Emergency Care Services: Trends, Drivers and Interventions to Manage the Demand. OECD Health Working Papers 83, OECD Publishing, August 2015.

[5] Justin Boyle, Melanie Jessup, Julia Crilly, David Green, James Lind, Marianne Wallis, Peter Miller, and Gerard Fitzgerald. Predicting emergency department admissions. *Emergency medicine journal : EMJ*, 29:358–65, 06 2011.

[6] Rachel Schutt and Cathy O'Neil. *Doing Data Science — Staright Talk From the Frontline.* O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2014.

[7] Sherry Towers, Shehzad Afzal, Gilbert Bernal, Nadya Bliss, Shala Brown, Baltazar Espinoza, Jasmine Jackson, Julia Judson-Garcia, Maryam Khan, Michael Lin, and et al. Mass media and the contagion of fear: The case of ebola in america. *PLOS ONE*, 10(6):e0129179, Jun 2015.

[8] Miguel Won, Manuel Marques-Pita, Carlota Louro, and Joana Gonçalves-Sá. Early and Real-Time Detection of Seasonal Influenza Onset. *PLOS Computational Biology*, 13(2), 2017.

[9] Bing Liu. *Sentiment Analysis — Mining Opinions, Sentiments, and Emotions.* Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2015.

[10] Richard Lawson. *Web Scraping with Python.* Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, 2015.

[11] Dimitrios Kouzis-Loukas. *Learning Scrapy — Learn the art of efficient web scraping and crawling with Python.* Packt Publishing Ltd., Livery Place, 35 Livery Street, Birmingham B3 2PB, UK, 2016.

[12] Shumway Robert H. and Stoffer David S. *Time Series Analysis and Its Applications: With R Examples. Springer texts in statistics.* Springer, 2011.

[13] Sean J. Taylor and Benjamin Letham. Forecasting at Scale. *The American Statistician*, 72(1):37–45, January 2018.

[14] Daphne Koller and Nir Friedman. *Probabilistic graphical models principles and techniques.* MIT Press, 2012.