# Experimental Comparison and Tuning of Time Series Prediction for Telecom Analysis

André Pinho (apinho@student.dei.uc.pt), Helena Silva (lena@alticelabs.com), Pedro Furtado (pnf@dei.uc.pt)

University of Coimbra, Altice Labs, Portugal

**Abstract.** Prediction of consumption is fundamental in telecommunications, for efficient management of network resources, and for guaranteeing quality of service. In this work we investigate the use of time series models to forecast consumption. Two time series forecasting algorithms are compared, Auto-Regressive Integrated Moving Average (ARIMA) and Prophet, launched by Facebook in 2017. We also developed a simple automated parameterization solution for ARIMA, which is important in practical deployments. The work described was developed in the context of tool development effort within Altice Labs that provides actual software to associated Telecom operators, in collaboration with University of Coimbra. To validate results we used real data from a Telecom operator. The forecast results showed that ARIMA was better than Prophet with a Mean Absolute Percentage Error (MAPE) of 3.71% in the three-month forecast and 4.14% in the twelve-month forecast.

**Keywords:** ARIMA, Forecasting, Prophet

## 1 Introduction

Currently, telecom operators face competition from other operators and from new services made available through the internet. Operators need to be one step ahead of competition, and they need to offer reliable services to avoid migration of customers and a fall in profits. One important opportunity is to create tools to analyze the huge and valuable data that they collect using data science techniques, with great potential for decision support. In this context, consumption forecasting is critical to provide information that helps the operator efficiently plan and manage network resources and provide an improvement in quality of service.

Consumption forecast in telecommunications presents its challenges, as it is necessary to deal with seasonality, trends, and with the variation of the number of clients. Taking into account these challenges, this work developed at Altice Labs in collaboration with University of Coimbra had two objectives. First, to compare two time series forecasting models, ARIMA and Prophet, in order to verify which one best fits the context of telecommunications. Since ARIMA was superior and since it requires manual parameterization of the model, the second objective was to create an automatic parameterization mechanism. This mechanism consists of a set of steps, selection,

adfa, p. 1, 2011.

transformation, parameterization by exhaustive search, and application of the model chosen in forecasting. With the choice of model and the automated parameterization, the approach is ready to be integrated in decision support tools to be used by managers that do not need to know any details of the forecasting model to use it. Instead, they simply view charts with the forecasts they desire, and the approach adjusts automatically the parameterization to changes in behavior of the data series. In terms of software details, the tool was developed in python, and in particular using its time-series libraries statsmodels [1] and fbprophet [2], containing the ARIMA and Prophet methods respectively.

This paper is divided into 7 sections. Section II presents state of the art. Section III discusses how to apply time series forecasting in the context of telecommunications, reviewing the relevant details of ARIMA and Prophet. In section IV we describe the need for manual parameterization of the time series model, and in section V proposes an automatic parameterization approach for ARIMA using exhaustive search. Section VI reports and analyses our experimental results. Finally, section VII concludes and discusses future work.

## 2 State of the art

In this section we review works on time-series forecasting in the context of Telecommunications that are most related to ours.

In 2008, S. T., & Sampaio, R. J. B proposed a model to predict short-term consumption of a telecommunications service [3]. Since service consumption presents a non-linear behavior caused by the existence of tendency and seasonality, the authors used two neural network algorithms, the Multilayer Perceptron and the Radial Basis Function network (RBF). The per-month dataset was divided into two sets, training set consisting of a history of 3, 4 and 6 months, and test set consisting of only 1 month ahead. The metric used in the evaluation of the results was Mean Squared Error (MSE). From the experiments performed for different historical periods, the Multilayer Perceptron model presented better prediction quality, although with worse computational performance.

In 2015, Wang, M., Wang, Y., Wang, X., & Wei, Z proposed a model based on the Auto-Regressive Integrated Moving Average (ARIMA), with the objective of predicting performance in telecommunications [4]. In this study they used monthly aggregated data corresponding to periods of two and a half years. The data of the first two years was used for analysis of the time series and for training the model, and the remaining six months were used for validation of the model. The method used to verify the seasonality of the time series was the analysis of the statistical properties, mean, variance and correlation coefficient, verifying if they remain constant over time. In our study it was verified that the series was non-stationary, so it had to be adapted using the typical manual procedure we describe later. The model obtained by [4], had an average error of 1% for five months.

In 2017, Hideaki Hayashi compared the performance of Prophet and ARIMA in the different context of prediction of number of flights in the United States [5].

Prophet was inferior to ARIMA with the parameters configured manually, and superior to ARIMA with the parameters configured automatically with the values by default. The author concluded that the ARIMA method, unlike Prophet, requires manual configuration of the model parameters in order to have good results. This means that the ARIMA requires a lot of knowledge in the domain to be configured manually.

The two works [4] and [5] do show that ARIMA could be the best choice for time-series analysis and forecasting in Telecom and other contexts, but has the big drawback that it requires manual configuration, which is undesirable for integration into a managerial decision support tool as we desired. Furthermore, in our work it was important to evaluate the two alternatives (ARIMA and Prophet) in the context of real Telecom consumption data, and to devise which to integrate into a decision support tool and how to automate its use.

# 3 Application of time series forecasting in the context of telecommunications

In telecommunications, time series forecasting is typically applied in the forecast of consumption and also in the detection of anomalies in real time. The remainder of this section reviews the concepts of stationarity, the Auto-Regressive Integrated Moving Average (ARIMA) [6] [7] and Prophet [2]. The later was launched by Facebook in 2017 to allow its use by people with less knowledge in the field, since ARIMA requires manual tuning of fundamental parameters.

## 3.1 Stationarity in ARIMA

An important concept in the application of the ARIMA method is stationarity, since the model can only be constructed with stationary time series. A series is stationary if its statistical properties remain constant over time. The existence of trend and seasonality are two of the reasons that lead the series to be non-stationary [6] [7].

There are two methods that allow you to check whether a series is stationary or not. The first method consists in the graphical visualization of the variation of the statistical properties of the series, such as the moving average (calculation at each instant of the average of the values corresponding to the last seasonal period, typically of twelve consecutive months) and the moving standard deviation over time. If the properties of the series do not change over time, then the series is stationary. The second method, the Dickey-Fuller test, assumes that the null hypothesis is that the series is non-stationary. This test calculates the value of the statistical test and some critical values for different levels of confidence. If the value of the statistical test is less than the critical value, then the series is stationary [6].

Differentiation [6] [7] is one of the existing techniques that allows us to deal with seasonality and trend of the time series, bringing it closer to stationarity in time. At each instant in the series, differentiation subtracts the original observation, $Y_t$, from that of the previous instant, $Y_{t-1}$, using the following formula:

$$Y'_t = Y_t - Y_{t-1} \qquad (1)$$

### 3.2 Time series forecasting with ARIMA and with Prophet

This subsection begins by describing the Auto-Regressive (AR) and Moving Average (MA) models, before describing the Auto-Regressive Integrated Moving Average (ARIMA) method.

1. The AR model [7] extracts the influence of the values of the previous periods from those of the current period. This model is developed using the following linear equation.

$$Y_t = \mathrm{c} + \varphi_1 . Y_{t-1} + \ldots + \varphi_p . Y_{t-p} + e_t \quad (2)$$

The parameter p indicates the AR order in the model and represents the delayed time period of the dependent variable. The remaining parameters of the equation, φ which represents the AR coefficient, y, which is the observed value, e the deviation of the series at the current instant, c is a constant [7].

2. The MA model [7] extracts the influence of the error terms from the previous period in the current period. This model is developed using the following linear equation:

$$Y_t = \mathrm{c} + e_t + \theta_1 . e_{t-1} - \ldots - \theta_q . e_{t-q} \quad (3)$$

The parameter q indicates the MA order in the model and represents the delayed forecast errors. The remaining parameters of the equation, θ which represents the MA coefficient, y, the observed value, e the deviation of the series at the current instant, c is a constant [7].

3. The non-seasonal ARIMA model [6] [7] consists of three components, AR, Integrated (I) and MA, each component represented by a positive integer parameter, p, d and q respectively. These three components are combined in the following linear equation:

$$Y_t = c + \varphi_1 . Y_{d\ t-1} + \ldots + \varphi_p . Y_{d\ t-p} + e_t + \theta_1 . e_{t-1} - \ldots - \theta_q . e_{t-q} \quad (4)$$

I component [6], represented by parameter d and indicating the number of times the series has been differentiated to approximate stationary in time; AR component [4] [6], represented by parameter p means the delayed time period, estimated by Autocorrelation Function (ACF); MA component [4] [6] represented by parameter q indicates the order of the MA component and represents the delayed forecast errors, estimated by the Partial Autocorrelation Function (PACF).

It should be noted that parameters p and q are determined when the respective functions, ACF and PACF, cross the upper confidence interval for the first time. The confidence interval of the two functions is calculated as $\pm 1.96 / \sqrt{n}$, where the variable n, corresponds to the size of the historical data [6] [7]. Finally, seasonal ARIMA [7] extends the previous model, combining its components along with the seasonal component.

Time series forecasting with Prophet is more automated, due to its ability to find automatically inflection points in the data originated by changes in trend. A novelty of this method in relation to the previous one is the possibility of accommodating the existence of seasonal festive periods. The method combines three components, the trend, the seasonality and the festive periods, each modelled by some function [8]:

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

The trend component, g(t), is modelled by a logistic function. The seasonality component, s(t), by a Fourier series. The festive periods, h(t), are adjusted by parameterization in the model. Finally, the error term, εt, represents the changes originated by circumstances that are not accommodated by the model [8]. Further information on the formulation details of each of these components can be found in [8].

## 4 Manual Parameterization of the Time Series

Figure 1 shows real data from a Telecom company graphically. It includes the variation of the number of recharges (used in telecommunications services), and the volume of internet data consumed (data) over the period from 2014 to 2017. There is a tendency of recharge decrease and data consumption increase. Seasonality exists there as temporary increases and decreases in certain months of each year. In the series of recharges, there is a lower consumption in the months of February, October and November, and a higher consumption in the months of January, August and December. First 3 years were used for analysis and training, the fourth year for prediction testing.

Figure 2 shows the stationary test of the time series of recharges using 2 methods. The figure above shows graphical visualization of moving average and deviation. The second, shows Dickey-Fuller statistical test presented below the figure.

**Fig. 1.** Consumption variation over 4 years

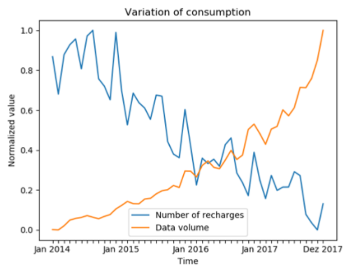
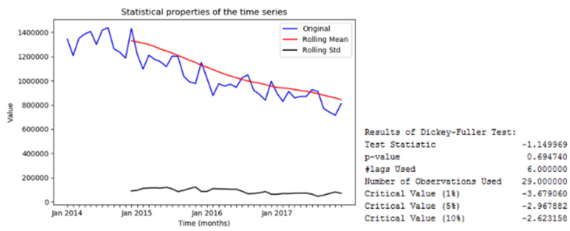
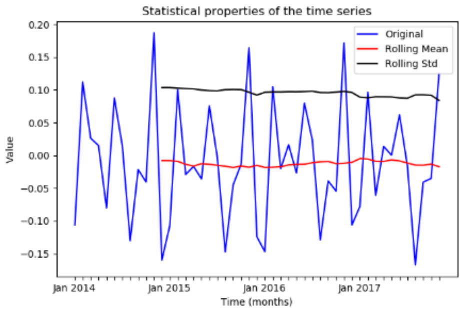



**Fig. 2.** Stationarity test of time series




We can see a decrease of the average over time. This variation over time indicates that the series is non-stationary. The result of the statistical test has a value greater than the critical value with a confidence level of 95% (-1.149969> -2.967882), also indicating that the series is non-stationary in time. Then a first-order differentiation transformation was applied to make the time series stationary in time. Figure 3 presents the stationary test of the series after this transformation.

**Fig. 3.** Stationarity test of the transformed time series

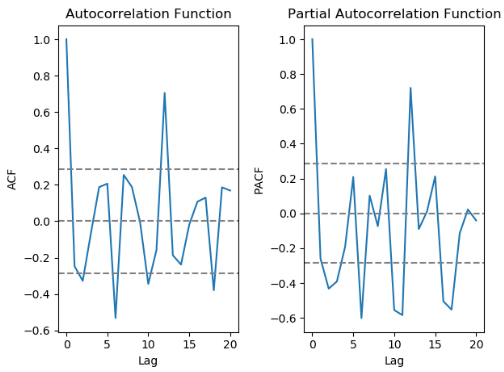



```
Results of Dickey-Fuller Test:
Test Statistic                 -4.815318
p-value                         0.000051
#lags Used                     10.000000
Number of Observations Used    24.000000
Critical Value (1%)            -3.737709
Critical Value (5%)            -2.992216
Critical Value (10%)           -2.635747
```

From the analysis of the previous graph, it is verified that the statistical properties of the time series have become approximately constant over time. The result of the statistical test is also less than the critical value for a 95% confidence level (-4.815318 <-2.992216), also indicating that the time series approached stationarity. Given that a differentiation was required to approximate the series of stationarity, then the parameter d of the model is equal to one.

The next step is to determine the values of the parameters p (order of the AR component of the model) and q (order of the MA component of the model) of ARIMA. To estimate these two parameters, we used the functions Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). Figure 4 shows the graphs of these two functions.

**Fig. 4.** Graphs ACF and PACF



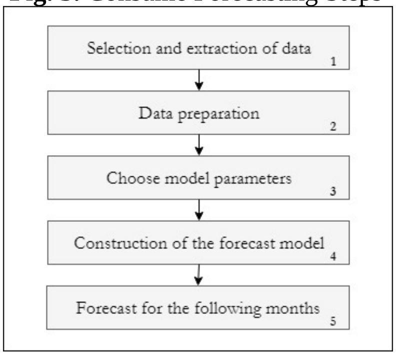

The graph on the left side of Figure 4, representing the ACF function, allows estimation of the value of parameter p, by looking at the position where the function crosses the confidence interval for the first time. In the graph this happens between p = 0 and p = 1. The graph on the right side is the PACF function, which allows to obtain the value of parameter q, also by crossing the function with the confidence interval. In the graph above, it happens between q = 0 and q = 1. In the experimental section we will compare the performance achieved by the forecasting model for this dataset using each of four possible parameter combinations (p = 0, d = 1, q = 0), (p = 1, d = 1, q = 0), (p = 0, d = 1, q = 1) and (p = 1, d = 1, q = 1).

Note that Prophet is much simpler to use than ARIMA, because of its ability to automatically find inflection points in the data, that is, points where trend changes [6].

## 5 Automation of the ARIMA model using exhaustive search

Automatic parameterization of the seasonal ARIMA model is important to allow its integration in a tool that can be used by managers without requirement of manual configuration or even knowledge of the details. It also accommodates changes in the behavior of the series. The proposed approach follows the steps shown in Figure 5.

**Fig. 5.** Consume Forecasting Steps




1. Selection of data, extraction and aggregation by month of a three-year historical period, related to the data that is intended to be forecasted.

2. Data preparation, subdivided into three tasks. The first converts the original date format to the format required by the template. The second constructs the input data structure in two-column model with date and the data consumption. The third applies a logarithmic transformation required by the method, which allows to attenuate the trend of the time series [6].

3. Choice of model parameters: This step aims to test and evaluate several models in order to choose the one that minimizes the forecast error. In this step, all combinations of values for parameters p, q and d are generated. The first two parameters can be zero, one or two, and parameter d can be zero or one [6]. This defined range of values takes into account manual parameterization tests as the ones presented in our experimental results, and also a certain flexibility to adapt to future data. The seasonality parameter is defined as 12 (yearly), corresponding to the seasonal period. After generating the combinations of parameters, each model instantiated is tested with historical values. The stationarity of the time series is a requirement for the application of the ARIMA method. Therefore, when the model is constructed with a non-stationary time series, the combination of parameters tested in this iteration is discarded, and the test advances to the next parameter combination.

4. Construction of the forecast model: after the choice of parameters, the forecast model is constructed with a recent three-year history, and with the parameters identified in the previous step.

5. Forecast of consumption of the following months: starts after the construction of the chosen model and consists of predicting the following months. The expected values are converted back to the scale of the original values by performing an exponential transformation (reverse operation of the logarithmic transformation).

## 6 Experimental Analysis

In the experiments presented in this section, we used real telecommunication data to compare the accuracy of seasonal ARIMA and Prophet methods when forecasting Telecom data, and to validate the parameterization mechanism of the proposed model.

### 6.1 Experimental Setup

The data used in this evaluation comes from a medium-sized telecommunications operator. Data was aggregated by month for a time period of four years. It consisted of consumption data, recharges (used in telecommunications services) and the volume of internet data consumed (data). Those are the same datasets already described in section IV. For both datasets we used the first three years for training and the fourth for testing the model.

The performance test (evaluation of execution time) was performed on a development machine with the following characteristics:

- Operating system: Windows 8 de 64 bits
- Processor: i5 de 2.50 GHz
- Memory RAM: 8.00 Gb
- Disk HDD: 297 Gb

### 6.2 Comparison between ARIMA and Prophet on Recharge dataset

Table 1 shows the RMSE and MAPE obtained using ARIMA with different combinations of parameter values, and Table 2 compares the forecasting errors of ARIMA and Prophet, choosing the best ARIMA result.

**Tab. 1.** Comparison of ARIMA errors on Recharges, 12 months

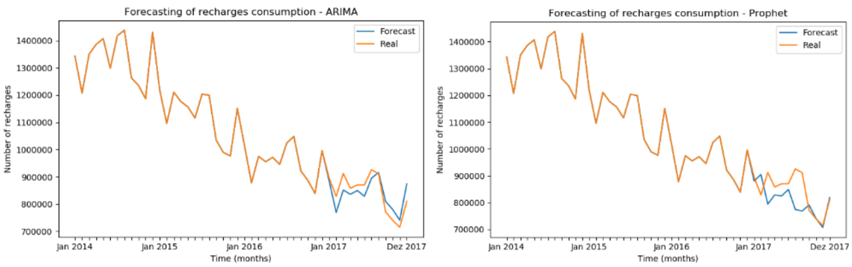
| ARIMA (p, d, q) | RMSE | MAPE |
|---|---|---|
| (0, 1, 0) | 4.08e+4 | 4.28 |
| (1, 1, 0) | 4.04e+4 | 4.24 |
| (0, 1, 1) | 3.99e+4 | 4.19 |
| (1, 1, 1) | 3.96e+4 | 4.14 |

**Tab. 2.** ARIMA versus Prophet on Recharges, 12 months

| Method | RMSE | MAPE |
|---|---|---|
| ARIMA (1, 1, 1) | 3.96e+4 | 4.14 |
| Prophet | 5.59e+4 | 6.3 |

From these results we can see that parameter configuration in ARIMA improves the error (RMSE or MAPE), and that ARIMA forecasting errors are much lower than Prophet for the Recharge dataset. Figures 6 show the forecast results graphically, comparing real to predicted values.

**Fig. 6.** Recharges Forecasting with ARIMA




From visual inspection of the results graphs we can see that both methods are able to deal effectively with both seasonality and trends. It is also clear that ARIMA outperforms Prophet for this dataset.

### 6.3 Comparison of various configurations

Table 3 presents the results of forecasting 12 months of internet data consumption using ARIMA and Prophet.

**Tab. 3.** ARIMA vs Prophet on Data Consumption, 12 months

| Method | RMSE | MAPE |
|---|---|---|
| ARIMA | 3.99e+13 | 8.50 |
| Prophet | 2.23e+13 | 9.88 |

Table 4 shows the comparison results for recharges and for data consumption (consume) over 3 months. In this case the training data was a whole year and the forecasting covered the next 3 months. Four forecasting runs were used, corresponding to the 4-year trimesters, the tables show the average and standard deviation of the errors over the four runs.

**Tab. 4.** ARIMA vs Prophet on Recharges and Data Consumption, 3 months

| Method | RMSE | | MAPE | |
|---|---|---|---|---|
| | Mean | Maximum | Mean | Maximum |
| ARIMA - Recharges | 3.69e+4 | 4.97e+4 | 3.71 | 5.01 |
| Prophet - Recharges | 7.63e+4 | 1.05e+5 | 7.82 | 11.08 |
| ARIMA - Data | 2.59e+13 | 5.12e+13 | 6.99 | 17.0 |
| Prophet - Data | 4.67e+13 | 6.35e+13 | 8.11 | 13.26 |

We can see from the tables that Data consumption was slightly more challenging than Recharges for both methods (higher average error). ARIMA was always better than Prophet (lower average error, almost half for both datasets), although ARIMA results have a higher standard deviation when compared with Prophet. Note that while for Recharges the sum of average error plus standard deviation or the maximum error were always smaller for ARIMA, in the case of Data Consumption one of the trimesters had the highest error for ARIMA, maximum error = 17% against 13% for Prophet, and the average error plus standard deviation is slightly higher for ARIMA. This was however an exception, as we could see by the results ARIMA was consistently better than Prophet for all other 12 and 3 months forecasting cases for both datasets.

### 6.4 Validation of automated ARIMA configuration

This section presents the validation results of the automatic parameterization mechanism of the ARIMA model described in section V. Table 5 shows the error of each combination tested by the exhaustive search approach regarding the 3-month and 12-months forecast, for both datasets tested. The exhaustive search parameter configuration approach obtains these errors for all cases and then chooses the one with lowest error, which is shown in bold. In all the tests it is verified that the implemented mechanism was able to automatically find the combination of parameters that minimizes the prediction error in the historical data.

The parameters chosen by the automatic approach for these 4 cases (Recharges and Data Consumption, 12 and 3 months forecasting) were always consistent with the established manual parameter configuration found (discussed and exemplified in sections III and IV). Note that the established manual approach involves iterative human inspection, running the Dickey-Fuller test or choosing visually, then successively differentiating and again testing using Dickey-Fuller or visual inspection until the decision thresholds are met. The exhaustive search simply replaces that tedious process with the automated version, with good results for these datasets and forecast objectives.

Another relevant issue is the runtime of the exhaustive search procedure, since it has to test a significant number of alternatives. We test the procedure runtime next.

**Tab. 5.** Automated parameter values finding by exhaustive search, 3 and 12 months

| **ARIMA (p, d, q)** | **Recharges (RMSE)** | | **Data consumption (RMSE)** | |
|---|---|---|---|---|
| | **3 months** | **12 months** | **3 months** | **12 months** |
| (0, 0, 0) | Discarded | Discarded | Discarded | Discarded |
| (0, 0, 1) | Discarded | Discarded | Discarded | Discarded |
| (0, 0, 2) | Discarded | Discarded | Discarded | Discarded |
| (0, 1, 0) | 3.87e+4 | 4.08e+4 | 3.98+e13 | 5.67e+13 |
| (0, 1, 1) | **3.69e+4** | 3.99e+4 | 4.02+e13 | 4.82e+13 |
| (0, 1, 2) | 3.75e+4 | 4.03e+4 | 3.84+e13 | 5.89e+13 |
| (1, 0, 0) | Discarded | Discarded | Discarded | Discarded |
| (1, 0, 1) | Discarded | Discarded | Discarded | Discarded |
| (1, 0, 2) | Discarded | Discarded | Discarded | Discarded |

| | | | | |
|---|---|---|---|---|
| (1, 1, 0) | 3.86e+4 | 4.04e+4 | 4.18e+13 | 5.13e+13 |
| (1, 1, 1) | 3.77e+4 | **3.96e+4** | **2.59e+13** | **3.99e+13** |
| (1, 1, 2) | 3.75e+4 | 4.03e+4 | 2.84e+13 | 5.55e+13 |
| (2, 0, 0) | Discarded | Discarded | Discarded | Discarded |
| (2, 0, 1) | Discarded | Discarded | Discarded | Discarded |
| (2, 0, 2) | Discarded | Discarded | Discarded | Discarded |
| (2, 1, 0) | 3.76e+4 | 4.06e+4 | 3.94e+13 | 6.48+e13 |
| (2, 1, 1) | 3.88e+4 | 3.97e+4 | 3.55e+13 | 7.36+e13 |
| (2, 1, 2) | 6.58e+4 | 4.01e+4 | 3.96e+13 | 1.19e+14 |

Table 6 shows the runtime of the automatic procedure (testing a significant set of parameter values alternatives) and compares it to the runtime of a single test as a ground truth. Note that the single test run is only used here for comparative reference, because running a single test automatically is not sufficient to configure the parameters, it must be done by a user inspecting the result, and most of the times, as happened for the datasets tested in this experimental section, will require further iterations of differentiating and running the test again.

**Tab. 6.** Runtime of parameter configuration by automated exhaustive search

| **Approach** | **Time (s)** | |
|---|---|---|
| | **Mean** | **Maximum** |
| Recharges Forecast (automated) | 6.38 ± 0.11 | 6.57 |
| Data consumption (automated) | 6.21 ± 0.39 | 6.92 |
| Single test | 0.30 ± 0.16 | 0.55 |
| Single test | 0.61 ± 0.03 | 0.66 |

These results show that the exhaustive search for parameter configuration takes about 6 secs, which is perfectly acceptable for the practical purposes of the decision support tool we were developing. The user has to wait for only 6 secs before the forecasting model does the forecast, since it is searching for the correct parameters. Nevertheless, we note that, although the procedure works fine for the sizes of datasets that our tool works with, it is important to develop improved parameter finding approaches in the future, to be able to handle much bigger datasets. Therefore, we identify as future work the possibility of improving the automated parameter configuration procedure. A simple way to scale to large datasets would involve sampling, to reduce the dataset to a size that is tractable by exhaustive search, the other alternative would be to apply heuristics to reduce the search space, and a third alternative would involve both. We reserve this study for future work.

## 7 Conclusion and future work

In this work we studied the application of time series forecasting methods ARIMA and Prophet to real Telecom data, with the aim of integrating the best performing one in a practical tool for decision support. The two methods were presented along with their parameterization processes. These two methods were compared in the forecast of consumption using real telecommunications data. Since the ARIMA model requires a great deal of knowledge in its parameterization, an automatic parameterization procedure was proposed and validated. This allows the approach to do every step of the data forecasting pipeline automatically. This way the approach was integrated into a tool used by managers to view the forecasts without requiring any knowledge of the data preparation and parameterization process for ARIMA forecasting.

This work also showed that, despite the great variation of consumption during the year, due to the existence of seasonality and trends, it is possible to make approximate forecasts with consumption data in the telecommunications area. From the results obtained and the comparison of the two time series methods, it was possible to obtain a minimum MAPE of 3.71% in the three-month forecast and of 4.14% in the twelve-month forecast. Of the two methods tested, the ARIMA model presented better prediction results in relation to Prophet.

For future work we intend to develop models of the Holt-Winters time-series method and compare with the methods studied in this work. In addition, we intend to investigate alternatives that improve the exhaustive search method in the automatic parameterization of ARIMA, using heuristics.

## References


[1] StatsModels Statistics in Python, available in http://www.statsmodels.org/devel/generated/statsmodels.tsa.arima_model.ARIMA.html, consulted in 2018-03-20.

[2] Prophet Forecasting at scale, available in https://facebook.github.io/prophet/, consulted in 2018-04-16.

[3] BRANCO, S. T., & DE SAMPAIO, R. J. B. A New Artificial Neural Networks Forecast Model in Telecommunications, consulted in 2017-10-28.

[4] Wang, M., Wang, Y., Wang, X., & Wei, Z. (2015). Forecast and Analyze the Telecom Income based on ARIMA Model. The Open Cybernetics & Systemics Journal, 9(1), consulted in 2018-03-15.

[5] Is Prophet Really Better than ARIMA for Forecasting Time Series Data, available in https://blog.exploratory.io/is-prophet-better-than-arima-for-forecasting-time-series-fa9ae08a5851, consulted in 2018-05-16.

[6] A comprehensive beginner's guide to create a Time Series Forecast, available in https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/, consulted in 2018-03-15.

[7] MATH6011: Forecasting, available in https://www.southampton.ac.uk /~abz1e14/papers/Forecasting.pdf, consulted on 2018-03-22.

[8] Taylor, S. J., & Letham, B. (2017). Forecasting at scale. The American Statistician, (just-accepted), consulted in 2018-04-16.

[9] MAE and RMSE - Which Metric is Better, available in https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d, consulted in2018-03-20.