# Time series analysis and forecast in the telecommunication market prices

Tiago Alvim - 95584
*Universidade de Aveiro*
*DETI*
Aveiro, Portugal
tiago.alvim@ua.pt

Vasco Costa - 97746
*Universidade de Aveiro*
*DETI*
Aveiro, Portugal
vascovc@ua.pt

Applications of Artificial Intelligence
2022/2023 - 41019 - AIA
Sónia Gouveia

***Abstract*—Time series analysis and forecast is one of the most in vogue subjects these days. This can be explained by the implication they can have in the real life and the potential economic gains. For that reason this project was developed as to forecast the price of electronic devices with different machine learning methods.**

**The models developed and tested were *LSTM* and *Convolutional LSTM*, the *Prophet* software developed by Facebook, the *ARIMA* and *SARIMA*. A *Multivariate LSTM* was also studied with a consumer perspective in mind to try and predict the lowest, the average or the highest price of the equipment.**

**The results that were achieved are in some cases good and in other cases showed some flaws with results that made no sense in the real world. In this aspect the data itself made the task somewhat more difficult as it was from real world and not synthetic. Despite that, the task at hand is considered to have been successfully explored.**

***Index Terms*—Time Series Forecast; Telecommunication Devices Price; LSTM; ConvLSTM; Prophet; ARIMA; SARIMA; Multivariate LSTM**

## I. Introduction

Time series analysis is a multidisciplinary topic that embraces the fields of statistics and probabilities. It combines them and shows how close they can be in some situations [1]. Its uses are also diverse with many real world applications such as in speech recognition or sleep stage classification. As this data is taken over the course of time and it often contains temporal dependencies it is often more difficult to analyse. The usage of Deep Learning models can simplify this task reducing the need for hand-crafted features [2].

The field of time series as also seen a big development in the past quarter of a century with a substantial amount of publications being made [3]. In particular by the financial sector, one of the main areas pushing it forward for both risk assessment and portfolio management [4]. The environmental sector as also spared no effort in this way to answer questions as is energy consumption in the most diverse of places, in buildings and campuses, for instance [5].

The present works intends to explore time series forecasting and analyses for electronic telecommunications devices with real world data. This can prove to be a relevant task for both consumers and sellers as they can have a better understatement of the current market and make wiser decisions.

## II. Dataset

[1] For the task of exploring different ways to process time series a dataset was provided with both shorter and longer periods of data collection, being only the longer series analysed here. Regarding all the instances prices of multiple sellers on electronic devices across several days.

As some data is missing and time dependency is something that requires attention when it comes to this subject, the interpolation and extrapolation of the missing days was made to simplify the task at hand. With different methods to do so being possible and implemented to study, requiring only one variable to change in the code.

Although all the code was developed to be used for every instance of products available in the dataset only one instance was studied. Namely, the data in respect to the *Samsung Galaxy A51 128*GB with **ID 23**. In Fig. 1 the data made available of the product is represented for different companies.

By using the methods of interpolation presented in the *pandas* package [6, 7] it was firstly studied how changing the option made the dataset different. In this case it is possible to visualise in Fig. 2 that the changes between the ways of filling in the gaps with the options of *time* and *spline* are in no way visually different. As it was verified with the other options, not presented here as they are of little importance. Due to this fact, the option of considering the way of *time* to perform the interpolation was chosen as it is mentioned in the literature that it works best on daily and higher resolution data.

As the dataset that was presented consists on multiple prices for one device two of the most straight forward approaches consist on either a univariate perspective, as in order to predict the future values for any given company. Or in a multivariate approach, taking in consideration the behaviour of other companies to be able to advice price changes.

### *Evaluation Metrics*

To evaluate how different models perform, meaning, how well they modulate reality and how close they come to it. It is possible to consider some numerical metrics.

[1] The work division is presented on a different file, `work_division.pdf`, that is included on the delivered folder as well as the other files mentioned here.

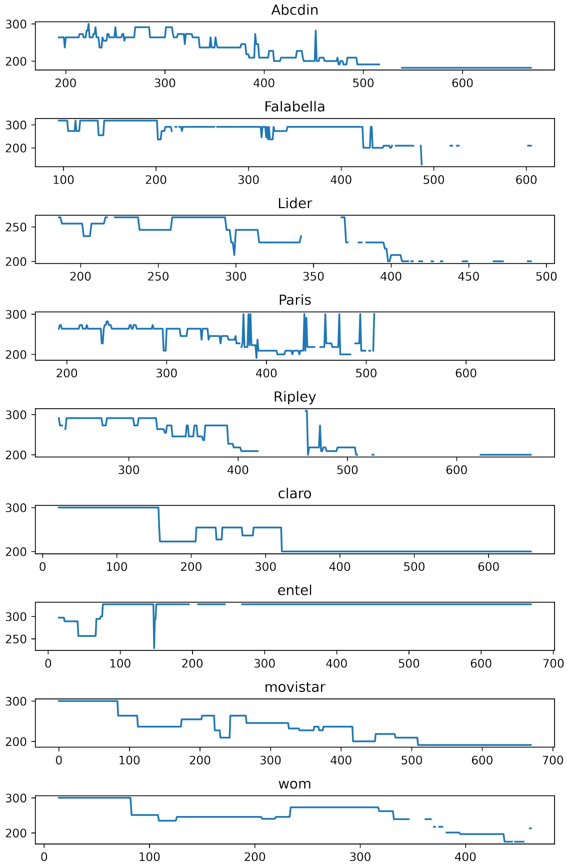
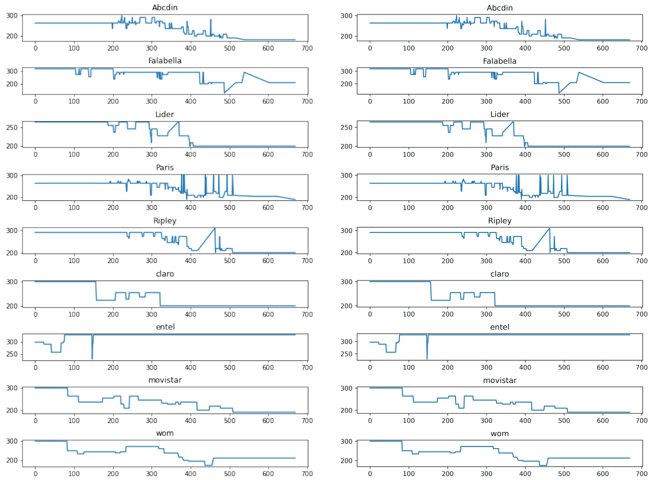


Fig. 1: Representation of the data regarding the product 21 with the its price on a certain day. The blank spaces indicate that it is a day with missing data.



(a) Interpolation - *time*

(b) Interpolation - *spline*

Fig. 2: Representation of the data after interpolation. On the left with the option of *time* and on the right with the option of *spline*.

One of the most common ways of measuring the goodness of the result is the **Mean Absolute Error (MAE)**, defined by the absolute difference between the forecasted, $y_i$, and true values, $x_i$, **MAE** $= \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$. Portraying how inaccurate the results are. However, by not considering the scale of the difference its meaning can be misleading.

In addition to this metric it is also possible to analyse the average of the error squares, **Mean Squared Error (MSE)**, **MSE** $= \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$. While overcoming some problems that **MAE** presents it also comes with its downsides when in low data volume, for instance. By taking the squared root of the **MSE**, the **Root Mean Squared Error (RMSE)** is calculated. Being this possible to normalise, by dividing the **RMSE** by the mean value or the range of actual values, **NRMSD**, in this work the mean version was considered [8].

Also of interest would be to perform the evaluation of the **Akaike information criterion (AIC)**. This measures the loss of information and penalises the complexity of a model, favouring models with less parameters. Due to being difficult to calculate for all the models the main idea of this metric was taken in consideration but not calculated.

One aspect that none of this metrics have in mind is the shape similarity of the predicted series and the real tendencies. For this, the **RdR** score is a way of helping decide if the results are random or have a true meaning [9]. This method unfortunately wasn't possible to consider due to the incompatibility between the type of models developed here and the required ones to use the metric.

## III. LSTM

LSTM is short for Long Short-Term Memory Recurrent Neural Networks, a type of network that can be used in respect to time series. It is a method that is capable of learning long-term dependencies with feedback connections. The main part of the model consists on a memory cell that maintains a state over time. This cell can then have information inserted or removed according to the logic gates, a multiplication operation and a sigmoid mechanism [10, 11].

### *A. Implementation of the most basic LSTM*

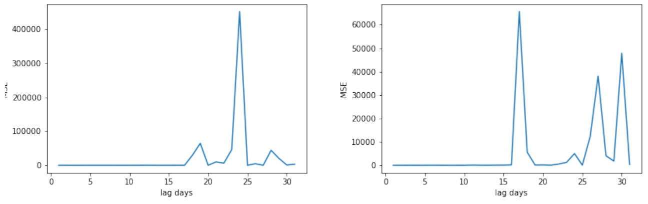
[2] By using a simple LSTM, with only a single hidden layer and an output one, it was possible to construct a model.

For this method there is a parameter that needs to be decided. The number of prior days to consider to make a prediction, *lag*. To decide which was the number that better

[2] Code available on `lstm_simple.ipynb`

portrayed this the method was tried on a range from just considering the day before to considering a month, 31 days.

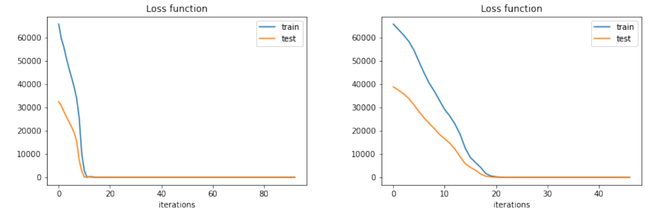
This was tested on the companies of *Abcdin* and *movistar*, considering a train portion of the dataset of 70% and the rest left out for later testing. To decide the best number of previous days to consider the metric of **MSE** was chosen on the predicted series for the training portion. This lead to the better number of days being 3 for *Abcdin* and 1 for *movistar*. In Fig. 3 the **MSE** result on the train portion is displayed for each of the *lag* intervals considered.



(a) *Abcdin* (b) *movistar*

Fig. 3: The value of **MSE** on the training set for each company in respect to the number of prior days in consideration for the method *LSTM*.

By choosing the best number of prior days the method was further taught by increasing the number of epochs from 1000 to 10000 and keeping the batch size of 72, changing however the early stop from 5 to 10 epochs of the monitorization of loss. If the value of loss did not change the fitting would stop. On Fig. 4 the loss is presented and visually one can conclude that it converged.



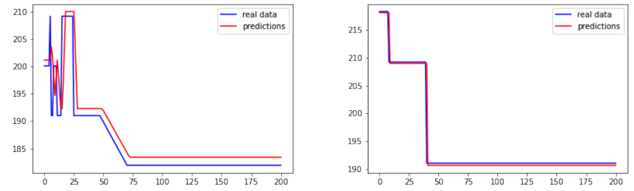
(a) *Abcdin* (b) *movistar*

Fig. 4: The loss graph for both companies on the training for the method *LSTM*.

Applying this method on the test portion of the dataset left out lead to the time series that is presented on Fig. 5. From here it is quite clear that the model suits *movistar* quite well, 5b, and that the shape on *Abcdin* is somewhat a little off.

One can measure how off the values predicted are and the metrics are portrayed on Table I. Where for *movistar* the results obtained are very promising and even for *Abcdin* but to a less extent. As one could have already expected from the visual representation where for *movistar* the lines are basically overlaping.

The results for this method seemed to be too good to be true as it is one of the most simple methods and yet the results are very good and almost perfect. A possibility for this is that an error might have been made but it was double checked and not found.



(a) *Abcdin* (b) *movistar*

Fig. 5: The predicted series for both companies for the test portion of the data for the method *LSTM*.

TABLE I: Results of the metrics with the best number of lag days for the method *LSTM* on both the prediction for the train and test portion of the data.

| | | Abcdin | Movistar |
|---|---|---|---|
| MAE | Train | -0,09368 | -0,00376 |
| | Test | 1,559356 | -0,21359 |
| MSE | Train | 81,40396 | 20,42741 |
| | Test | 10,26191 | 2,121359 |
| NRMSE | Train | 0,035634 | 0,018136 |
| | Test | 0,017227 | 0,007469 |

### B. ConvLSTM

[3] As the subject of LSTM was already being studied the topic of convolutional LSTM was also found and decided to be given a go. This method differs from the one presented before by the addition of a convolution operator in the state-to-state and input-to-state transitions. The size of the convolution operator determines the preference between faster or slower changes in the series [12].

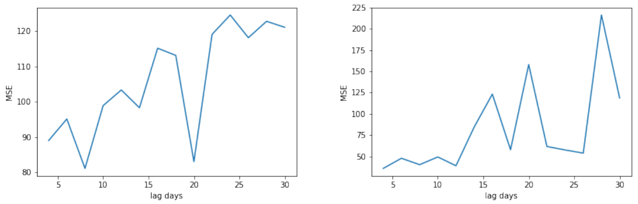
Maintaining the same structure of analyses, same number of epochs and early stopping technique, but now changing the number of days to be the even numbers between 4 and 30, inclusive, the reshaping method required it to be even. Now, the best number of days for *Abcdin* was 4 and for *movistar* was 8, the graphics of the **MSE** values in respect to the number of previous days considered can be found on Fig. 6.

For the best number of prior days, the number that lead to a lower **MSE**, the retraining with a more tolerable patience lead to the convergence of the loss parameter, Fig. 7.

The resulting series that was predicted can be found on Fig. 8 and the metrics obtained on Table II. From here it can be concluded that the simple *LSTM* lead to better results.

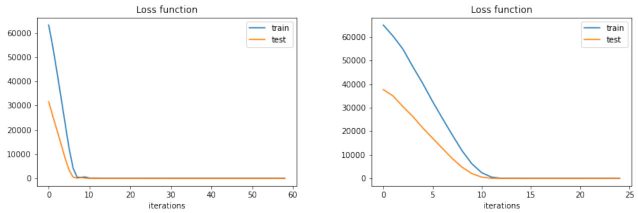
As the results were considered good enough and due to computational restrictions the search for other better parameters was not done, namely, for the size of the convolutional kernel.

[3] Code available on `lstm_conv.ipynb`



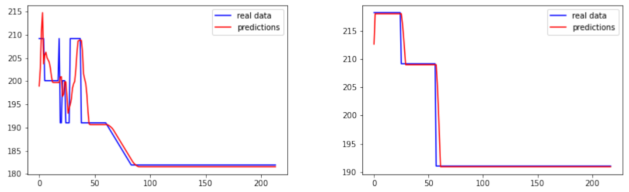
(a) *Abcdin* (b) *movistar*

Fig. 6: The value of **MSE** on the training set for each company in respect to the number of prior days in consideration for the method *convLSTM*.



(a) *Abcdin* (b) *movistar*

Fig. 7: The loss graph for both companies on the training for the method *convLSTM*.



(a) *Abcdin* (b) *movistar*

Fig. 8: The predicted series for both companies for the test portion of the data for the method *convLSTM*.

TABLE II: Results of the metrics with the best number of lag days for the method *convLSTM*.

| | | Abcdin | Movistar |
|---|---|---|---|
| MAE | Train | -0,00979 | 0,199179 |
| | Test | 0,045672 | 0,172422 |
| MSE | Train | 95,20742 | 44,2786 |
| | Test | 11,32866 | 4,019528 |
| NRMSE | Train | 0,038337 | 0,026613 |
| | Test | 0,017996 | 0,010186 |

## IV. Prophet

[4] The team at Facebook's Core Data Science developed this open-source procedure with the intention to take in consideration effects of early, weekly, and daily seasonality in addition to holidays. Working better on larger datasets and with a strong seasonality presence [13].

The model has in its design the ability to be intuitive for users to be able to adjust the model parameters without having great knowledge of the underlying part. By combining a decomposable time series model in separate components, trend, seasonality and holidays it is possible to frame forecasting problems as a curve-fitting exercise. Removing some importance of the inferential advantages of other generative models but with the gain of practical advantages as flexibility [14].

This method was tested on the companies of *Abcdin* and *movistar* as it was also done on the previous point. These companies were chosen as they presented the data that was considered as being more interesting in terms of the overall trend. In addition they also present some complex visual variability and enough data points for the least use of interpolation. The train portion of the data was set to 70%, this value was decided as it is a typical value and to be able to compare with the previous methods.

### A. *Out of the box parameters*

Firstly it was studied how the default parameters of the method behaved in this situation. In Fig. 9 the time series is represented, in black the training data for the method and in red the test portion. In blue the predicted time series is portrayed, the confidence interval and the predicted value. From here it is clear that in both situations the overall tendency is followed, not working so well for the test portion of the data as in 9a it is expected for the value to lower and it stagnates in reality. Despite that in 9b the general tendency is to drop, the predicted values are inside the confidence interval.

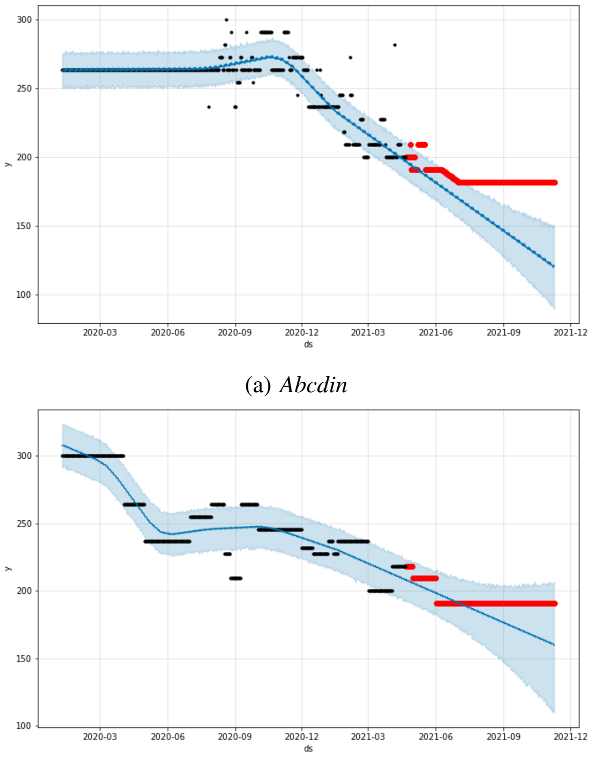
In Fig. 10 the trend is presented for both of the companies and their weekly seasonality. Being visible a general downward tendency for both companies, as it could be expected for the sale of electronics components due to obsolescence. With both agreeing on a higher value of the item on Thursdays.

By analysing the metrics, Table III, it is possible to evaluate how the method performed. Verifying how it represents the series in the train portion of the data and on the test part that was left out. In comparison with the two previous methods, the results here achieved are worse for both companies.

### B. *Hyperparameter search*

By performing hyperparameter search and cross validation this method was more deeply studied. The part of cross validation consists on using the train portion of the dataset and only training until a certain percentage $k$, and testing on the remaining train portion of the dataset. Then the value of $k$ is increased, meaning there is more part of the dataset to

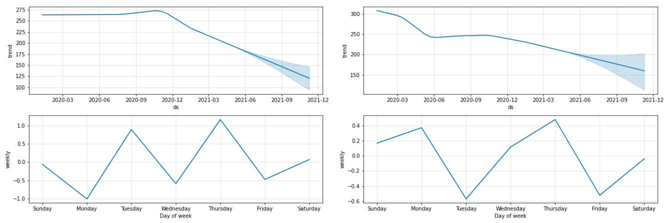
[4] Code available on `prophet.ipynb`



(a) *Abcdin*

(b) *movistar*

Fig. 9: The time series obtained for the different companies with the default parameters of *Prophet*. In black the points of the data used for the train portion are represented with the red points being the test values. In blue the confidence interval for the predicted time series is presented with the middle line for the estimated value.



(a) *Abcdin*

(b) *movistar*

Fig. 10: The trend and the weekly seasonality for the different companies with the default parameters of *Prophet*.

TABLE III: Results of the metrics with the default parameters for *Prophet*.

| | | Abcdin | Movistar |
|---|---|---|---|
| MAE | Train | 0,00438 | 0,00284 |
| | Test | -27,42384 | -11,08013 |
| MSE | Train | 66,98166 | 103,00976 |
| | Test | 1061,47104 | 223,82921 |
| NRMSE | Train | 0,03865 | 0,04870 |
| | Test | 0,17514 | 0,07668 |

train on and less to test for cross validation. This procedure is then repeated and the average of a metric is taken to see the overall performance of the combination of parameters.

This way of preforming cross validation is different from the methods purposed usually on the literature as it was felt that it was relevant to study how the model handled longer term predictions. As it could be possible to have a model that would fit the train data and the close predictions well and stable and be erratic for the more distant future, Fig. 11 would be an example. Where the close predicted, blue, values are close to the actual data, in black, but in red the prediction becomes erratic and does not come close to the actual data. This method was however not found in other works and was based on intuition.

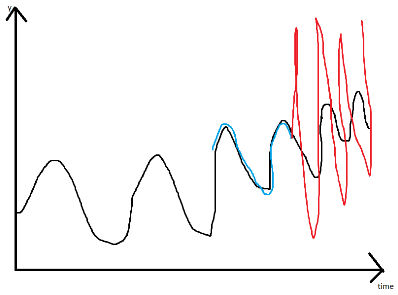


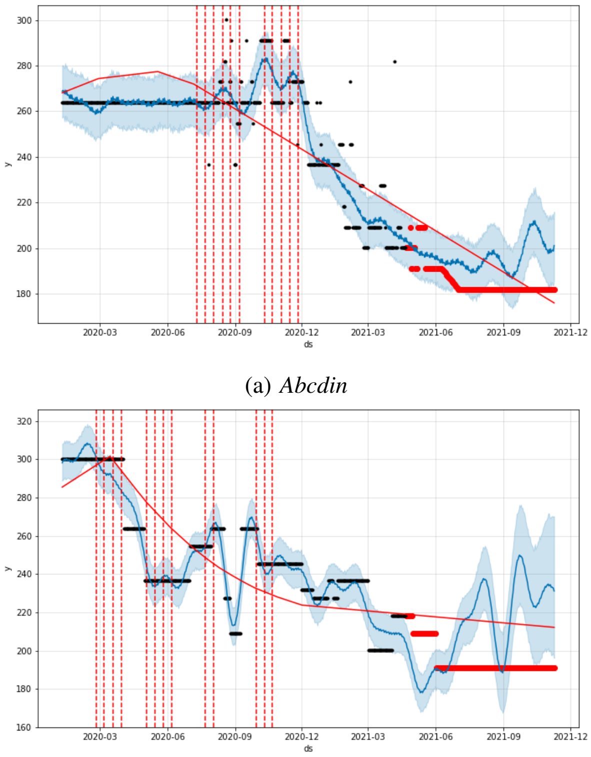
Fig. 11: Representation of a time series prediction. In black the actual data is presented with the blue line being a close prediction and in red the representation of the model fitting for later instances.

The hyperparameter search is done by performing the cross validation on the different combinations of values for the elements that can be tuned. In this case, the cross validation was chosen to be done in three folds with the search on the parameters of *n_changepoints*, *yearly_seasonality* and *changepoint_range*, respecting usual values of this parameters as was found in the literature. Monitoring the metric of **MSE** and preferring the combination of parameters that lead to a lower value.

The time series is presented in Fig. 12 and it is evident that the result is better on the train data as some overfitting might be occurring. Specially in 12b a very radical behaviour is of notice at the part of the test portion. However, a significant aspect is also how closer in scale the values are for 12a.

In Fig. 13 the trend and the weekly seasonality are presented as in the case before but now the yearly seasonality is also presented. From here it is possible to conclude the same as before that on Thursdays the price is usually higher. It is also clear that there is a clear tendency for price maintenance and slight decrease for the future of the series. From the yearly tendency the information that can be drawn is less clear as there is only information regarding two years. Despite that, around May both companies decrease their prices.

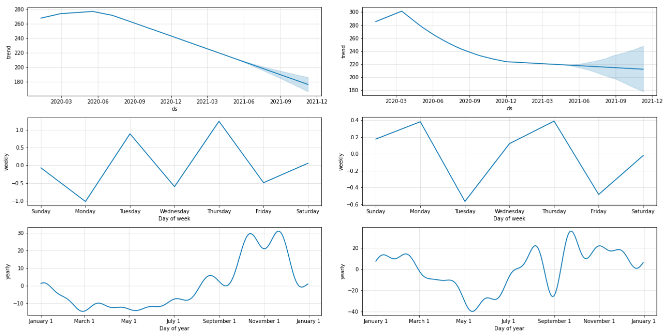
From the Table IV the metrics obtained this time, with the best hyper parameters, are shown. As one can see these results are better than the ones presented on Table III for the company



(a) *Abcdin*

(b) *movistar*

Fig. 12: The time series obtained for the different companies with the best hyperparameters of *Prophet*. In black the points of the data used for the train portion are represented, with the red portion being the test values. In blue the confidence interval for the predicted time series is presented with the middle line representing the estimated value. The vertical red dotted line indicates where the change points occur.



(a) *Abcdin*

(b) *movistar*

Fig. 13: The trend, the weekly and yearly seasonality for the different companies with the best hyperparameters of *Prophet*.

TABLE IV: Results of the metrics with the best hyperparameters for *Prophet*.

| | | Abcdin | Movistar |
|---|---|---|---|
| MAE | Train | -0,00006 | 0,00203 |
| | Test | 10,58319 | 16,86512 |
| MSE | Train | 54,58443 | 39,28158 |
| | Test | 188,04402 | 983,84227 |
| NRMSE | Train | 0,03489 | 0,03007 |
| | Test | 0,07372 | 0,16075 |

*Abcdin*. However, for the company *Movistar*, the train results improved slightly but the test portion that was left out, led to worse results. Showing clearly that overfitting is occurring. This can be a result of the innovative way of performing cross validation that was done.

## V. ARMA

ARMA stands for Auto Regressive Moving Average which is a model of forecasting where different methods of autoregression (AR) and moving average (MA) are applied on a stationary time series. When this fluctuates it does so on uniformly around a particular instance. As a first approach the AR parameters are estimated and with these the MA ones are followed [15].

This model is often represented by 1 where $p$ is the order of the autoregressive polynomial and $q$ is for the moving average polynomial order. With $\phi$ representing the autoregressive model's parameters and $\theta$ the parameters for the moving average, being $c$ a constant and $\epsilon$ the error term [16].

$$X_t = c + \epsilon_t + \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{1}$$

### A. ARIMA

[5] On the same line of thought of ARMA, ARIMA comes with the addition of an integration part (I), the number of times the times series needs to be differentiated to become stationary. Because of this, it is one of the most widely models when it comes to real life time series analysis as they rarely are stationary [17].

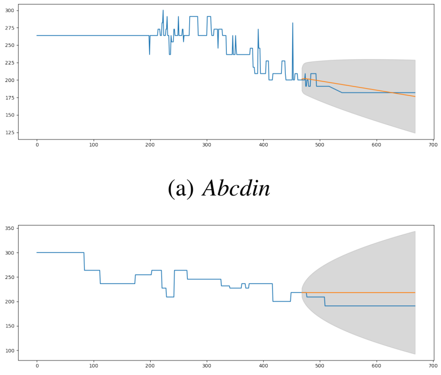
The *autoArima* model used can automatically search for the best set of parameters given the best combination that leads to a lower value of **AIC**. With the best set a model is trained on the 70% of the dataset and tested on both parts, the train and test parts. The metrics obtained are presented on Table V.

TABLE V: Results of the metrics for *ARIMA*.

| | | Abcdin | Movistar |
|---|---|---|---|
| MAE | Train | -81,36742 | -31,07067 |
| | Test | 3,74997 | 23,30746 |
| MSE | Train | 4849,94058 | 1258,26182 |
| | Test | 45,19152 | 609,71453 |
| NRMSE | Train | 0,32886 | 0,17019 |
| | Test | 0,03615 | 0,12662 |

[5] Code available on `ARIMA.ipynb`

The visual representation of the series and the predicted values can be found on Fig. 14. Here the orange predicted line and the grey confidence interval can be compared to the blue original data.



(a) *Abcdin*

(b) *movistar*

Fig. 14: The predicted series for both companies for the test portion of the data for the method *ARIMA*

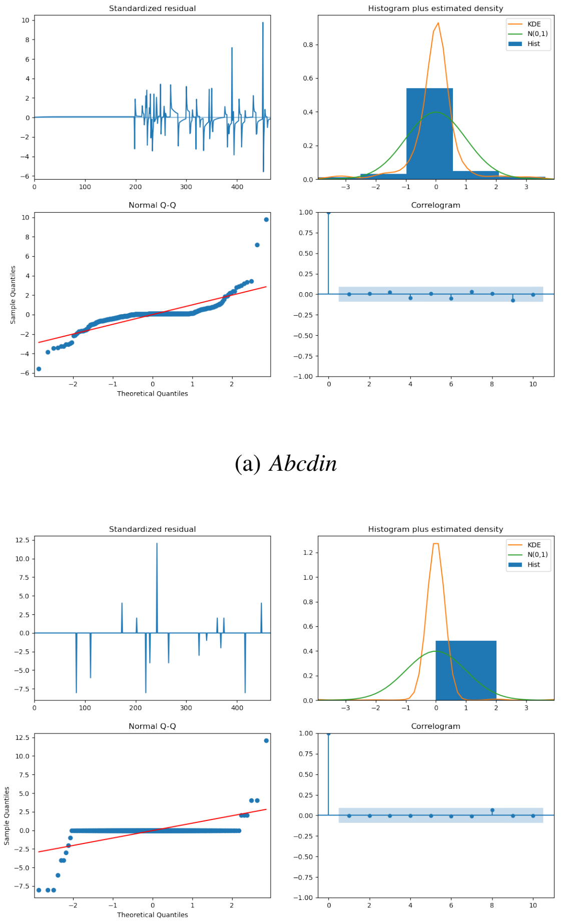
Moreover in Fig. 15 the diagnostics for this model are presented and one can validate what the was somewhat anticipated by the numerical values on Table V and by the visual representation, 14. For the company *movistar* the results are in fact unsatisfactory with the errors not following a normal distribution at all. Therefore the significance of these models is to be questioned.

## B. SARIMAX

The last point presented cannot however deal with seasonality and exogenous variables. For that, the model of SARIMA was constructed with the addition of extra components to deal with additional lags, monthly and hourly. This results in the inclusion of more parameters in the equation of ARMA, 1. To make the model more responsive it is also possible to include a direct term to represent the exogenous factors, external data, leading to the model of SARIMAX [18, 19].

[6] For the implementation of this method the same line of thought was used, meaning that both companies, *Abcdin* and *movistar*, were studied. By transforming the daily data into monthly the Fig. 16 can be obtained, this is done by taking the average of the values of the month.

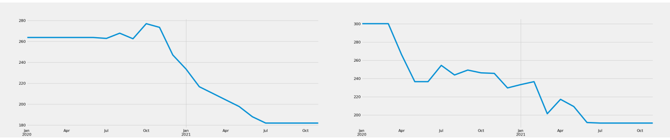
To test the series stationarity the Dickey-Fuller(ADF) and the Kwiatkowski-Phillips-Schmidt-Shin(KPSS) tests were performed. The result obtained from the ADF test, p-values of 0.89 and 0.33, respectively for *Abcdin* and *movistar*. Portraying that both of the series are non stationary and from the KPSS test, p-values of 0.023 and 0.018, respectively, that the hypothesis of them not being stationary cannot be rejected. Therefore to turn the series strictly stationary the first order differencing is taken, taking the differences between successive realisations of the time series.



(a) *Abcdin*

(b) *movistar*

Fig. 15: Diagnostic of the *ARIMA* model for both companies.



(a) *Abcdin*

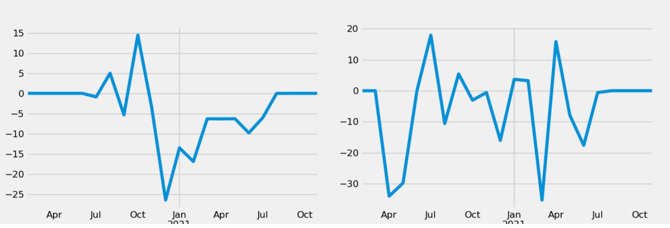
(b) *movistar*

Fig. 16: Data transformed to use in the *SARIMAX* model.

After differentiating and performing another Dickey-Fuller test, presented in Table VI, the p-value obtained for *movistar* makes the null hypothesis be rejected, $H_0$, and since the p-value of *Abcdin* is very close too, depending on the significance level considered, meaning that the data can be considered stationary. The data before becoming stationary can be seen in Fig. 17.

Another way to confirm data stationarity is by the graphs in the Figs. 18 and 19. Here the values of the partial correlation and correlation, respectively, are mostly presented within the blue box, the confidence interval, and close to 0. Reassuring

[6] Code available on `SARIMA.ipynb`



(a) *Abcdin* (b) *movistar*

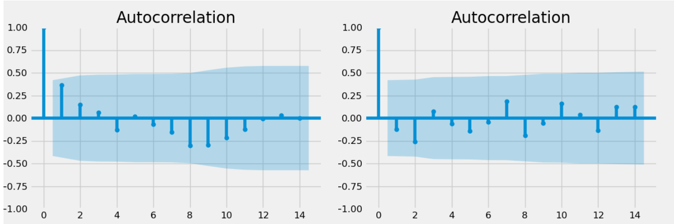
Fig. 17: Data transformed to be stationary in *SARIMAX*.

TABLE VI: Dickey-Fuller test to confirm data stationarity in *SARIMAX*.

| Dickey-Fuller Test | | |
|---|---|---|
| Compannу | Abcdin | movistar |
| Test Statistic | -2,38593 | -4,17338 |
| p-value | 0,14575 | 0,00073 |
| Lags used | 9 | 1 |
| Number of observations | 12 | 20 |
| Critical Value(1%) | -4,13783 | -3,80921 |
| Critical Value(5%) | -3,15497 | -3,02165 |
| Critical Value(10%) | -2,71338 | -2,65071 |

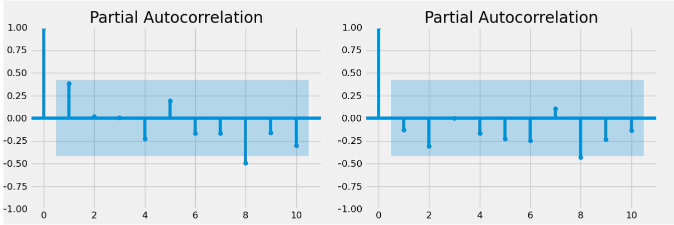
the ADF results mentioned previously.

Hyperparameter search was not done and for that the parameters used already in ARIMA were used. The model was then trained on the whole dataset and the final 30% were predicted, with the prediction results presented on Fig. 20. Here, the orange predicted line almost follows the real data, in blue, but in a more erratic behaviour. Table VII portrays numerically the metric results for the models in the overall series. With this technique presenting a better result for *Abcdin*. To note that by doing the training in this way it is not possible to compare this method with the others presented.



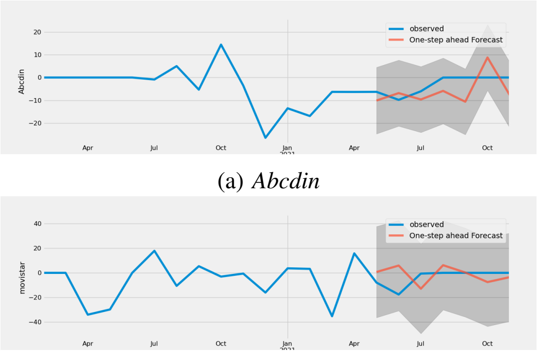
(a) *Abcdin* (b) *movistar*

Fig. 18: Autocorrelation of the data.



(a) *Abcdin* (b) *movistar*

Fig. 19: Partial Autocorrelation of the data.



(a) *Abcdin*

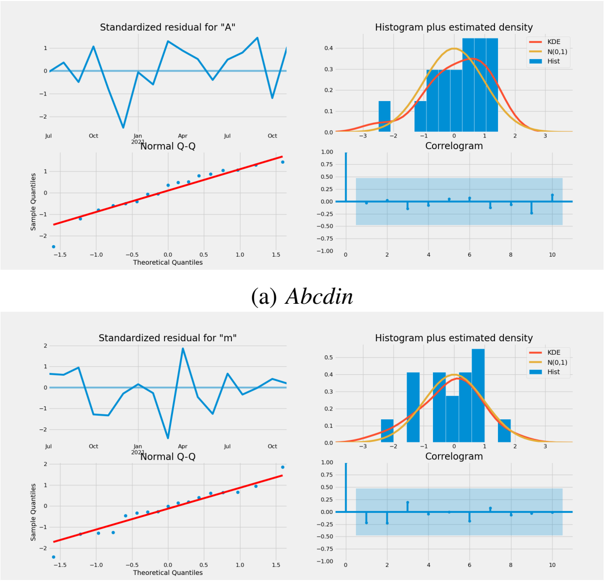
(b) *movistar*

Fig. 20: Prediction of the model *SARIMAX* on the last 30% of the dataset.

TABLE VII: Results of the metrics for *SARIMAX*.

| | Abcdin | movistar |
|---|---|---|
| MAE | -2.82889 | 2,18625 |
| MSE | 45,57433 | 126,18765 |
| RMSE | 6,75088 | 11,23333 |

It is also possible to diagnose the model with the graphs presented on Fig. 21. Firstly, it is clear, from the Q-Q plot, that the values follow approximately a normal distribution. The histograms also aids in the perception of the normal distribution followed and on the asymmetry present for *Abcdin*. From here it is possible to confirm that the models obtained are significant.



(a) *Abcdin*

(b) *movistar*

Fig. 21: Diagnostic of the *SARIMA* model for both companies.

## VI. Multivariate LSTM

With the dataset that was presented it also made sense to have a buyers perspective in mind. That is, having multiple sellers and the only difference being on the price to pay, not regarding different clauses like the warranties provided.

[7] With this in regard it is possible to combine all the series for the different companies and take the lower, the average or the maximum value as being the label to consider for the features given. Resulting in a new series, the minimum, average or maximum value with a set of features.

In Fig. 22 the heat-map of the correlation between the different sellers and the minimum, average or maximum value is presented. This perspective is interesting to validate who is the most pricey seller, and it is clear that it is *Entel*, with a negative value of correlation in the minimum, 22a, and the most positive value in the maximum, 22c. Normally the best deals can be found on *claro* or *movistar*, for this product. With *Lider* being where the average value is more commonly found, 22b.

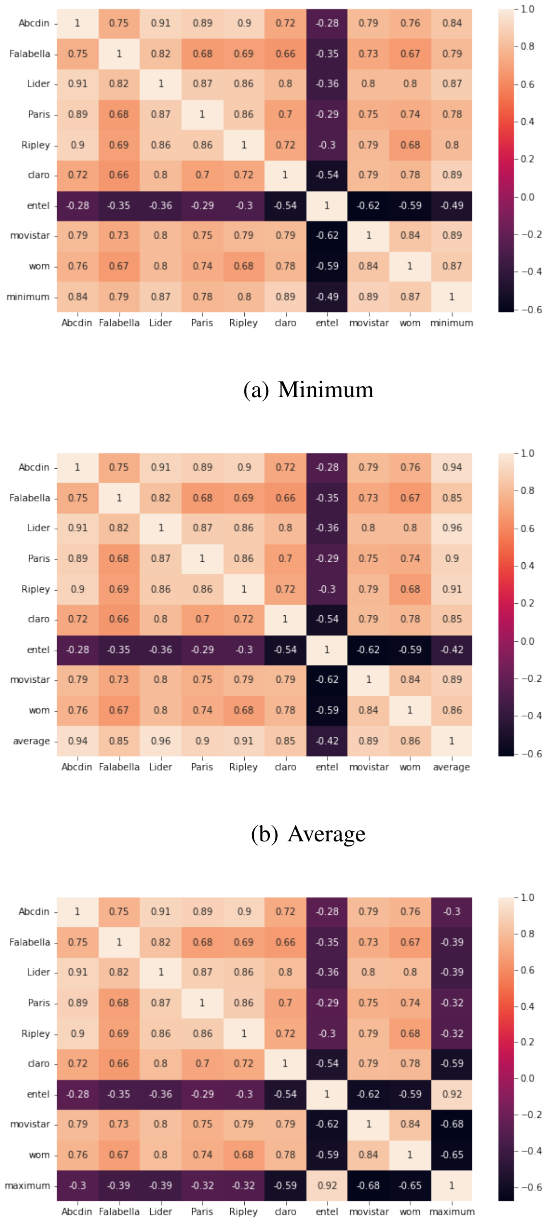
To work on this idea the implementation was followed from [20] with the addition of the search for the best number of prior days to consider by seeing the value of **MSE**. Following the same procedure as was done earlier on the *LSTM* topic with the same train to test division, 70/30, respectively. On Fig. 23 the values of **MSE** are presented for each number of days of information considered, from here, the minimum error is obtained for the minimum by using 7 days, the average with 2 days and the maximum by 7 days of previous information.

However, this time the epochs for the search of the best number of days was chosen to monitor the loss for 10 iterations, stopping if it didn't change. For the train of the best model the number of iterations to monitor was increased to 20 to be sure of achieving convergency. On Fig. 24 the graph of loss is presented for the three situations considered. With the train portion converging and the test not being of interest so the high value of the orange line in 24a not being importance here.

Fig. 25 displays the predicted series for both the test part that was left out, on the left, and on the right for the whole series, training and test combined. As it is very clear that for the situations that the minimum and the maximum values were to be predicted that the algorithm fails. However, when considering the average value the result is not as bad. With the overall shape being followed. Despite that, all the series present very abnormal spikes both on the positive and negative direction.

Table VIII comes to confirm how poorly this method performed on this case of study by putting the errors in a numerical aspect. For the train and test portion separated and the overall series.

The idea itself, to be able to predict the product value in the future, the minimum or the maximum would be of extreme relevance. However, it was not possible to achieve a model that can reliably predict it. With only the predictions of the average



(a) Minimum

(b) Average

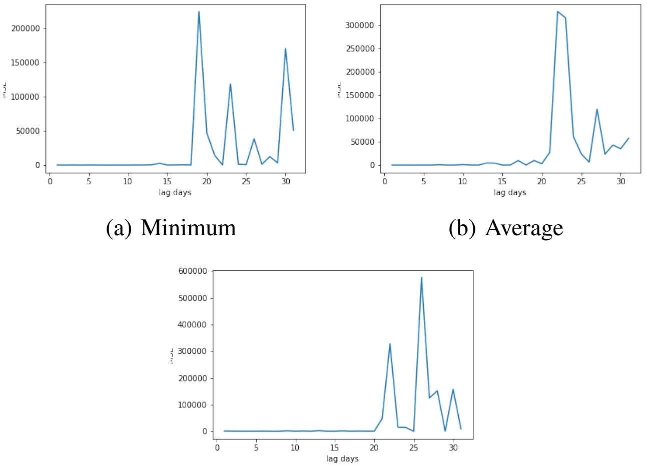
(c) Maximum

Fig. 22: Heat-map correlation between the different companies and the new column, minimum, average or maximum daily value.

TABLE VIII: Results of the metrics with the default parameters for the *Multivariate LSTM* for the monitoring of the minimum, average and maximum.

| | MAE | | | MSE | | | NRMSE | | |
|---|---|---|---|---|---|---|---|---|---|
| | Train | Test | All | Train | Test | All | Train | Test | All |
| Minimum | -0,376 | 271,111 | 81,070 | 78,352 | 788408,781 | 236577,481 | 0,039 | 4,916 | 2,302 |
| Average | -0,018 | 0,766 | 0,217 | 25,447 | 7,657 | 20,110 | 0,019 | 0,013 | 0,018 |
| Maximum | 2,734 | -62,227 | -16,754 | 1368,569 | 4132,827 | 2197,847 | 0,113 | 0,196 | 0,144 |

[7] Code available on `lstm_multivariate.ipynb`

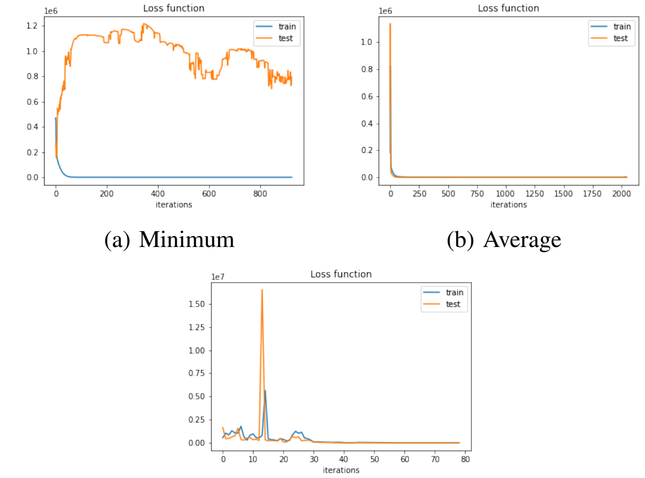


(a) Minimum

(b) Average

(c) Maximum

Fig. 23: The value of **MSE** in respect to the number of prior days in consideration for the method *Multivariate LSTM*.



(a) Minimum

(b) Average

(c) Maximum

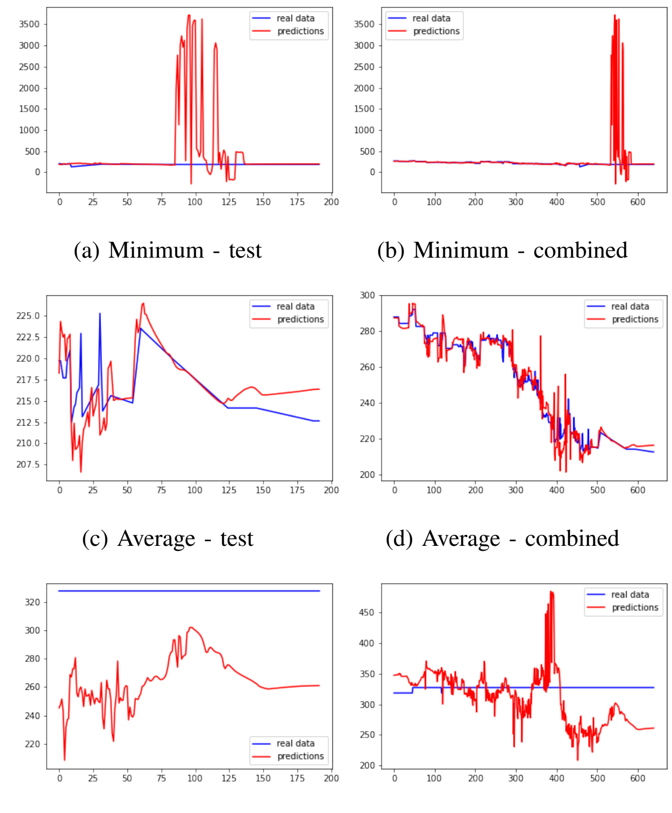
Fig. 24: The loss graph for the different cases for the *Multivariate LSTM*.

value having some closeness to the real world. More complex models can possibly deal with this task being that of future work.

## VII. Related works

The subject addressed on this work is one of the most researched topics in many areas and in special the finance sector as it can result on either the wins or losses of substantial amounts of money. With both time series analyses and forecasting being of interest on an investment, (only time series forecasting was studied here).

On the subject of e-commerce and electronic equipment, some papers have already been proposed on methods to incorporate news and the fluctuation of prices with sales influence. Showing promising results that lead to a more accurate forecast [21].



(a) Minimum - test

(b) Minimum - combined

(c) Average - test

(d) Average - combined

(e) Maximum - test

(f) Maximum - combined

Fig. 25: The graphs of the predicted values and the actual series for the different cases for the *Multivariate LSTM*. On the left for the whole series and on the right for just the test portion.

In respect to the electronic devices other works have been developed in respect to when they are expected to fail or need maintenance. By making it preventive and effectively higher costs repairs costs can be decreased [22]. In addition to this point, another concern these days is about the future potential of the electronic waste. In order to be able to predict the amount and how to better manage it [23].

As this dataset was given by the teaching from a private dataset it was not possible to consider other works and compare with the metrics here obtained.

## VIII. Conclusions

One of the points taken in consideration while developing this project was the ease for extending the research to other items. As well as creating methods that are modular to be more easily studied in future work.

With the implemented models it is possible to compare some of them, the different *LSTMs* the most simple and the convolutional with *Prophet*. Where the most basic *LSTM* was the best performer for the presented metrics. Despite that, it is interesting to verify the information that can be extracted from the *Prophet* model. As it has in consideration the exact days to relate with other external events its easier. Where future work could be developed to analyse the tendencies before the usual sales periods like *Black Friday* or *Cyber Monday*.

Another point one must consider is that the data here treated is of the real world and therefore it is harder to work with and even evaluate as one can have little representative data. For that it can be expected to have larger errors as it was seen while evaluating the predictions with the training data.

The addition of the ARMA family of processes is an interesting inclusion as it adds a statistical inference to the typical processes of machine learning used that are only evaluated based on their predictive performances exclusively.

The ability to predict the product value based on the diverse series from each seller would be of extreme interest but it wasn't successfully obtained. This method should in the future be compared with the series itself, not considering all the vendors.

The project goals were in this way achieved. Leading to a better understandment of the different ways to analyse time series.

## References

[1] Emanuel Parzen. "An Approach to Time Series Analysis". In: *The Annals of Mathematical Statistics* 32.4 (1961), pp. 951–989. DOI: 10.1214/aoms/1177704840. URL: https://doi.org/10.1214/aoms/1177704840.

[2] John Cristian Borges Gamboa. "Deep Learning for Time-Series Analysis". In: *CoRR* abs/1701.01887 (2017). arXiv: 1701.01887. URL: http://arxiv.org/abs/1701.01887.

[3] Jan G. De Gooijer and Rob J. Hyndman. "25 years of time series forecasting". In: *International Journal of Forecasting* 22.3 (2006). Twenty five years of forecasting, pp. 443–473. ISSN: 0169-2070. DOI: https://doi.org/10.1016/j.ijforecast.2006.01.001. URL: https://www.sciencedirect.com/science/article/pii/S0169207006000021.

[4] Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu. "Financial time series forecasting with deep learning : A systematic literature review: 2005–2019". In: *Applied Soft Computing* 90 (2020), p. 106181. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2020.106181. URL: https://www.sciencedirect.com/science/article/pii/S1568494620301216.

[5] Chirag Deb et al. "A review on time series forecasting techniques for building energy consumption". In: *Renewable and Sustainable Energy Reviews* 74 (2017), pp. 902–924. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2017.02.085. URL: https://www.sciencedirect.com/science/article/pii/S1364032117303155.

[6] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[7] *Pandas.dataframe.interpolate#*. URL: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.interpolate.html.

[8] Vijaysinh Lendave. *A Guide to Different Evaluation Metrics for Time Series Forecasting Models — analyticsindiamag.com*. https://analyticsindiamag.com/a-guide-to-different-evaluation-metrics-for-time-series-forecasting-models/. [Accessed 17-Jan-2023].

[9] M.Sc. Dave Cote. *RdR score metric for evaluating time series forecasting models — dave.cote.msc*. https://medium.com/@dave.cote.msc/rdr-score-metric-for-evaluating-time-series-forecasting-models-1c23f92f80e7. [Accessed 17-Jan-2023].

[10] Ralf C. Staudemeyer and Eric Rothstein Morris. *Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks*. 2019. DOI: 10.48550/ARXIV.1909.09586. URL: https://arxiv.org/abs/1909.09586.

[11] King Luna says: *What is LSTM - introduction to long short term memory*. Dec. 2022. URL: https://intellipaat.com/blog/what-is-lstm/.

[12] Xingjian Shi et al. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting". In: *CoRR* abs/1506.04214 (2015). arXiv: 1506.04214. URL: http://arxiv.org/abs/1506.04214.

[13] *Forecasting at scale.* URL: https://facebook.github.io/prophet/.

[14] Sean J Taylor and Benjamin Letham. "Forecasting at scale". In: (Sept. 2017). DOI: 10.7287/peerj.preprints.3190v2. URL: https://doi.org/10.7287/peerj.preprints.3190v2.

[15] Jason Gordon. *Autoregressive moving average (ARMA) - explained.* Apr. 2022. URL: https://thebusinessprofessor.com/en_US/research-analysis-decision-science/autoregressive-moving-average-arma-definition.

[16] Stephanie. *Arma model*. Jan. 2021. URL: https://www.statisticshowto.com/arma-model/.

[17] TrainDataHub. *Clear explanations of AR, MA, Arma, and Arima in times series analysis*. Jan. 2022. URL: https://medium.com/@ooemma83/clear-explanations-of-ar-ma-arma-and-arima-in-times-series-analysis-9a72ff569dee.

[18] Brendan Artley. *Time Series Forecasting with Arima&nbsp;, Sarima and SARIMAX*. June 2022. URL: https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6.

[19] statsmodels. *Statsmodels.tsa.statespace.sarimax.SARIMAX¶*. Jan. 2023. URL: https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html.

[20] Jason Brownlee. *How to develop LSTM models for time series forecasting*. Aug. 2020. URL: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/.

[21] Kuo-Kun Tseng et al. "Price prediction of e-commerce products through Internet sentiment analysis". In: *Electronic Commerce Research* 18.1 (Mar. 2018), pp. 65–88. ISSN: 1572-9362. DOI: 10.1007/s10660-017-9272-9. URL: https://doi.org/10.1007/s10660-017-9272-9.

[22] Lili Guan, Jinglong Guan, and Jiacheng Li. "Time-to-Failure Prediction of Electronic Devices Based on Hawkes Point Process". In: *2021 4th Artificial Intelligence and Cloud Computing Conference*. AICCC '21. Kyoto, Japan: Association for Computing Machinery, 2022, pp. 179–185. ISBN: 9781450384162. DOI: 10.1145/3508259.3508285. URL: https://doi.org/10.1145/3508259.3508285.

[23] Yang Yang, Jian Zhang, and Weizhe Feng. "A prediction on electronic waste resource with time series model". In: *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 2. 2010, pp. V2-551-V2–555. DOI: 10.1109/ICCASM.2010.5620680.