Netflix Stock Market Forecasting

Leonor Coelho
Master's in Data Science and Engineering
University of Coimbra
Coimbra, Portugal
mariacoelho@student.dei.uc.pt

Tiago Fernandes
Master's in Data Science and Engineering
University of Coimbra
Coimbra, Portugal
tfernandes@student.dei.uc.pt

***Abstract*—The Stock Market is a Time Series that varies throughout time and depends on multiple factors. Predicting the future is hard, and predicting the Stock Market is even harder, although there are models that can achieve reasonable results, it is far from predicting with good accuracy. The models implemented in our paper don't achieve the best results, but it is a start in this area. Our best model accomplished a Mean Squared Error value of 1618.165.**

***Index Terms*—Time Series, Stock Price, Trend, ARIMA, DES, Prediction**

I. Introduction

The stock price of a company is a value that changes easily: if there are a lot of buyers, the price increases; on the other hand, when the number of investors increases, the stock price decreases.

The stock market price is a time series since it is a set of sequential observations through time. People that invest in the market should be able to decide where to invest their money so they can make a bigger profit [1]. The analysis and forecasting of this time series can help in this decision: forecasting the stock price can help to predict if it will increase or decrease and investors can decide if they should buy or sell their shares.

Netflix is a paid streaming platform, although they first sold DVDs, and has already 214 million subscribers all over the world. In 2002 they became a public company and their first profit was in 2003. Since then, Netflix entered Fortune 500 rank and is ranked 164^{th}.

Forecasting stock market returns is difficult because of market volatility. Recent advances offer useful tools in forecasting noisy environments like stock markets, taking into account their nonlinear behaviour.

This paper is structured as follows: first, we present a literature review about this topic in section II; then, in section III we explain our approach, and which dataset and tools we used; the results and their discussion are presented in section IV; in section V we show our conclusions.

II. Literature Review

This paper [2] summarizes over 100 articles that try to forecast stock market prices using soft computing. The authors reviewed those papers and summarized which methods their authors used and which data they inputed. The most common approaches to forecast were Artificial Neural Networks, linear and multi-linear regression, ARMA and ARIMA, Genetic Algorithms, and others. Some of the authors used specific techniques to choose which variables should be inputed, like stock index opening or closing, that are the most common, daily minimum or maximum price, transactions volumes, a combination of opening and closing price of previous days, and others. In this paper, more than 60% of the surveyed articles use feed-forward neural networks (FFNN) and recurrent networks. The authors concluded that neural networks and neuro-fuzzy models are suitable for stock market forecasting.

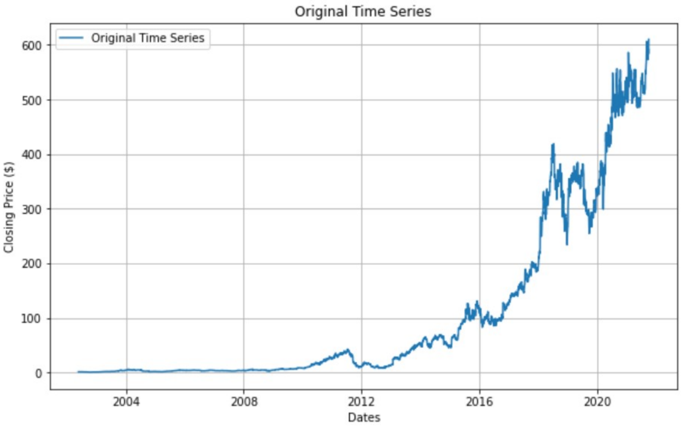

Fig. 1. Netflix's Stock Market since 2002

III. Materials and Methods

The main goal of this paper is to characterize and forecast a stock market time series. To achieve this goal, we used a dataset from Kaggle[1] with Netflix's stock market information since 2002, represented in Figure 1.

To characterize this TS, we estimated the trend using different approaches (polynomial functions, Moving-Average, Locally Weighted Scatterplot Smoothing and differencing), tried to find the seasonality with the Sample Autocorrelation Sequence and tested if our TS is stationary or non-stationary with the Augmented Dickey-Fuller Test.

Since this TS is non-stationary, as we will show in Section IV, we used Autoregressive Integrated Moving-Average (ARIMA) and Double Exponential Smoothing (DES) to do forecasting.

In Subsection III-G, we explain the experiments we made.

[1] https://www.kaggle.com/pritsheta/netflix-stock-data-from-2002-to-2021

A. Trend

It is possible to estimate the trend of a TS following different approaches. The TS has a trend when there is an increasing or decreasing slope or when it is constant.

A simple way to model the trend is using a **polynomial function** represented in Equation 1, where $e(n)$ is the erractic component.

$$X(n) = \beta_0 + \beta_1 n + \beta_2 n^2 + \beta_3 n^3 + ... + e(n) \tag{1}$$

We tried linear, quadratic and cubic regressions. (experiments)

Another way to estimate the trend is using a local smoothing approach, like the **Equally Weighted Moving-Average (EWMA)** filtering, Equation 2 where $\sum \omega_k = 1$.

$$\hat{t}(n) = \sum_{k=-\frac{M-1}{2}}^{\frac{M-1}{2}} \omega_k x(n+k) \tag{2}$$

Locally Weighted Scatterplot Smoothing (LOWESS) is a generalization of regression estimation and MA. It combines the two previous approaches and fits a smooth curve to the data.

Differencing the TS can give us the TS without the trend and without the seasonality. If we differentiate the TS once (1^{st} order differencing in Equation 3), we obtain the signal without its trend. Differencing the TS one more time (2^{nd} order differencing in Equation 4) give us the signal without the trend and the seasonality.

$$\Delta x(n) = x(n) - x(n-1) = y(n) \tag{3}$$

$$\Delta^2 x(n) = \Delta x(n) - \Delta x(n-1) \tag{4}$$

B. Decomposition Model

The decomposition of the TS can be used to inform forecasting models about the problem. In this paper, we decomposed our signal using 3 basic models: Additive, Multiplicative and Pseudo-Multiplicative.

The Additive Model (Equation 5) assumes that trend, season and erratic components are independent. The Multiplicative Model (Equation 6) is used when the magnitude of the seasonality and the erratic component change with the trend. The Pseudo-Additive Model (Equation 7) combines the previous two models. It can be useful when the TS has values close to or equal to 0.

$$x(n) = tr(n) + sn(n) + e(n) \tag{5}$$

$$x(n) = tr(n)sn(n)e(n) \tag{6}$$

$$x(n) = tr(n)(sn(n) + e(n) - 1) \tag{7}$$

C. Seasonality

Seasonality exists when the TS experiences regular and predictable changes within a specific period.

The **Sample Autocorrelation Sequence (ACS)** is a tool used to find repeating patterns. It provides the correlation between the TS and a lagged version of itself over successive time intervals. This autocorrelation can be represented by

$$r_{XY} = \frac{\sum_{n=0}^{N-T-1} (x(n) - \overline{x})(x(n+T) - \overline{x})}{\sum_{n=0}^{N-1} (x(n) - \overline{x})^2} \tag{8}$$

where $r \in [-1, 1]$ and T is the delay. The value of r can have 3 different meanings:

- if r is positive, the TS changes the same way in $x(n)$ and $x(n+T)$
- if r is negative, the TS changes in oposite directions in $x(n)$ and $x(n+T)$
- if r is 0, there is no relation between the TS values

For $T > 0$, if the correlogram (ACS plotted in function of T says that the TS values are not correlated, it means that the TS has no seasonality. When regular peaks are visible in the correlogram, the difference between peaks represents the seasonality.

D. Stationarity

A TS is stationary when its properties do not depend on time. It means that time series with a trend or seasonality are not stationary because these two properties affect the TS at different moments.

The Augmented Dickey-Fuller (ADF) test is a popular one to determine the presence of a unit root in the TS. This helps to understand if the TS is stationary or not. The test null and alternative hypotheses are:

- H0 (null hypothesis): The TS has a unit root.
- H1 (alternate hypothesis): The TS has no unit root.

If we fail to reject the null hypothesis, the series is non-stationary.

E. ARIMA

The Autoregressive Integrated Moving Average (ARIMA) is fitted to the TS to better understand the data and to predict future points [3]. Usually is applied to non-stationary TS. In this model, the future values are a linear combination of past values and errors. The ARIMA model is given by

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + ... + \phi_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - ... - \theta_q \varepsilon_{t-q} \tag{9}$$

where Y_t is the actual value, ε_t is the error, ϕ_i and θ_j are coefficients, p is the number of time lags of the Autoregressive (AR) model and q is the order of the Moving Average (MA). The ARIMA model has a parameter d that represents the degree of differencing (usually 1 or 2).

There is an extended version of this model that deals with seasonality (SARIMA). We will only focus on ARIMA because, as we will explain in Section IV, our TS has no seasonality.

F. Exponential Smoothing

In the simple MA, the past observations are weighted equally. The **Exponential Smoothing (ES)** is an extrapolation method that assigns exponentially decreasing weights over time. The Simple ES (SES) is given by

$$\hat{x}(n) = c_0x(n-1) + c_1x(n-2) + ... + c_{N-1}x(0) \quad (10)$$

where $c\{c_i\}_{i=0}^{N-1}$ are the weights and $c_i = \alpha(1-\alpha)^i$, with $i = 0, 1, ..., N$. For $0 < \alpha < 1$, the weights decay at a geometric rate and $\sum c_i = 1$. This method should be applied to stationary time series. It is common to find α between 0.1 and 0.3.

SES should not be used in non-stationary TS, the Double and Triple ES (DES and TES, respectively) are extended versions of the Simple one. The DES can deal with trend and TES can deal with trend and seasonality. Again, because our TS only has a trend, we will be focused only on DES. DES can be defined with Equations 11, 12 and 13. α and γ are the data smoothing factor and the trend smoothing factor, respectively.

$$L(n) = \alpha x(n-1) + (1-\alpha)(L(n-1) + T(n-1)) \quad (11)$$

$$T(n) = \gamma(L(n) - L(n-1)) + (1-\gamma)T(n-1) \quad (12)$$

$$\hat{x}(N-h) = L(N) + hT(N) \quad (13)$$

G. Experiments

As the dataset has various features, we choose only the *Closing Price*, since it is the most stable and represents the stock value at the end of the day.

1) Trend: We estimated the trend with polynomial functions (linear, quadratic and cubic), Moving Average with $M \in [5, 45, 91]$ and data augmentation, LOWESS with same values of M and by differencing the TS. Then we decomposed our TS with the additive, multiplicative and pseudo-additive models.

2) Decomposition Model: We tried to understand which type of model we were dealing with, using the polynomial fit we also experimented.

3) Seasonality: We tried to find the seasonality using the ACS.

4) Stationarity: We confirmed the stationarity of our signal.

5) ARIMA: We divided our TS in train/test with 94%/6% and did a grid-search to find the best parameters for ARIMA. We tested $p, q \in [0, 1, ..., 9]$. The metrics used were AIC, BIC and MSE.

6) DES: We divided our TS in train/test with 94%/6% and did a grid-search to find the best parameters for DES. We tested $\alpha, \beta \in [0.1, 0.11, ..., 0.99]$. The metrics used were AIC, BIC and SSE.

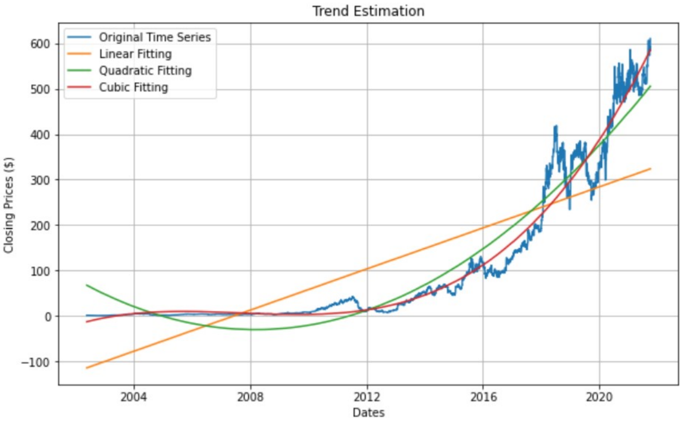

Fig. 2. Netflix's Stock Market since 2002

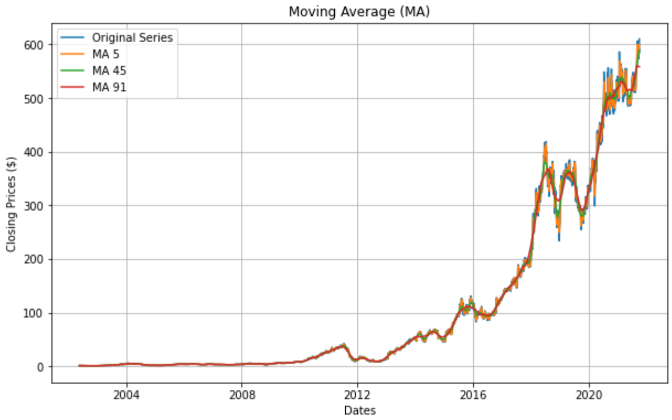

Fig. 3. Netflix's Stock Market since 2002

IV. Results and Discussion

A. Trend

1) Polynomial Fitting: We tried 3 types of polynomial fitting, and in the end, the best estimator was the cubic one, with the Mean Squared Error (MSE) of 776.399. If we tried higher degrees, it would overfit the TS. We can see these 3 degrees in Figure 2.

2) Moving Average: Regarding the MA, we experimented with 3 different odd values, since we want to predict a value for each data point, so need to have odd numbers. The results are in Figure 3 and a close-up in Figure 4.

3) LOWESS: In the experients with LOWESS, we tried the same values as in MA to be able to compare them. In Figure 5, we can see those 3 window sizes, and in Figure 6 a close-up. In the end, the results are pretty similar (Figure 7, but one big difference to notice is the time that takes to perform both algorithms (to run the 3 values, MA took 4.83s while the LOWESS too 54.4s).

4) Differencing: This method to obtain the trend of the TS is calculated by subtracting the actual value from the previous one. We can see in Figure 8 the values closer to zero for that reason.

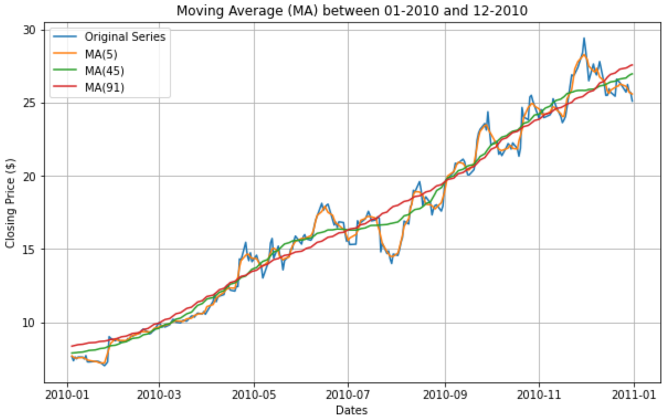

Fig. 4. Netflix's Stock Market since 2002

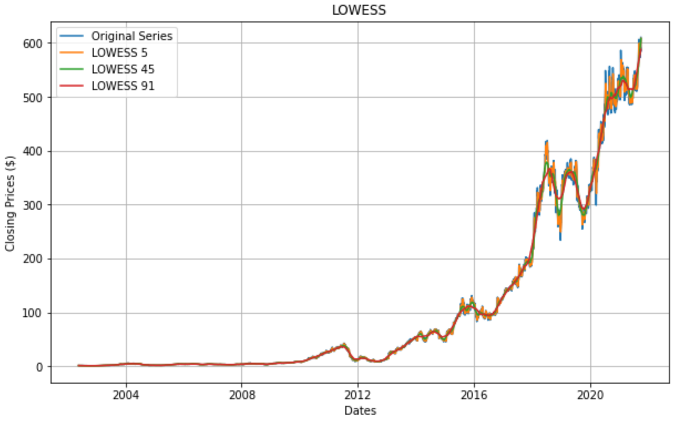

Fig. 5. Netflix's Stock Market since 2002

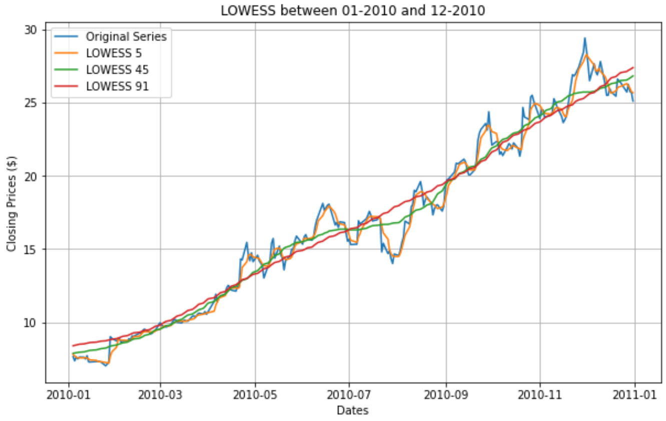

Fig. 6. Netflix's Stock Market since 2002

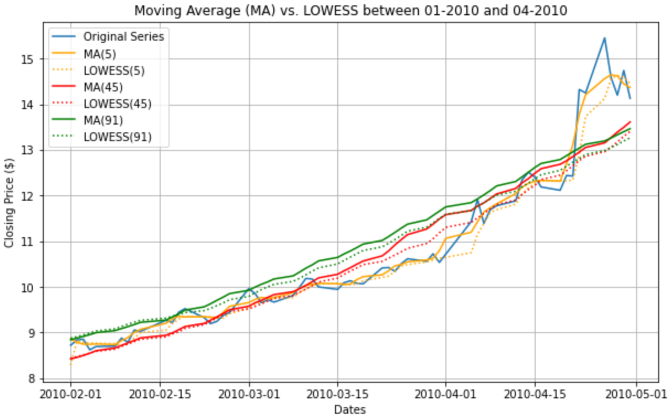

Fig. 7. Netflix's Stock Market since 2002

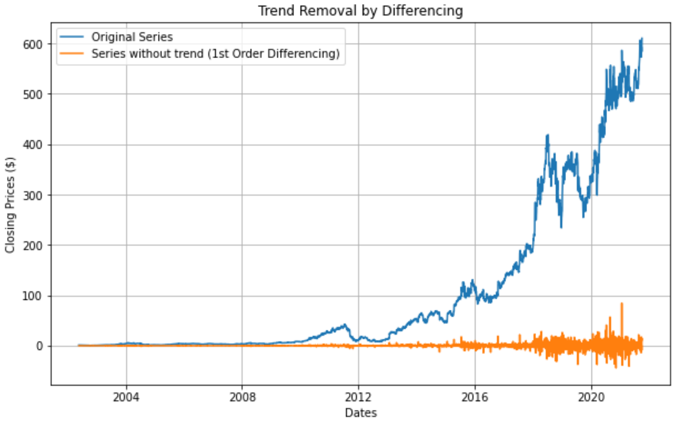

Fig. 8. Netflix's Stock Market since 2002

B. Decomposition Model

Using the polynomial fitting of the trend that we explained above, to see if the TS is an additive model, i.e., the difference between the TS and the polynomial fit should be near zero (meaning that the season and erratic components are independent of the trend), which was not the case (Figure 9). Then, we tried to see if the TS was a multiplicative model, i.e., the division between the TS and the polynomial fit should be near 1. In Figure 10, we can see 4 outliers that represent instances where the trend was near 0. When dividing the signal for the trend, if the denominator has small values, the division will be high. Because of this, the pseudo-additive model is the one that fits better since it combines the previous two, the first values are close to zero and the season, and erratic components seem to be dependent on the trend component. Figure 11 shows the obtained results.

C. Seasonality

Regarding the seasonality of our TS, we removed the trend from the original TS and plotted the Auto-Correlation Function (ACF). If this plot has regular peaks, that frequency is the seasonality of our TS, otherwise, it does not have any seasonality, like our case, as we can see in Figure 12.

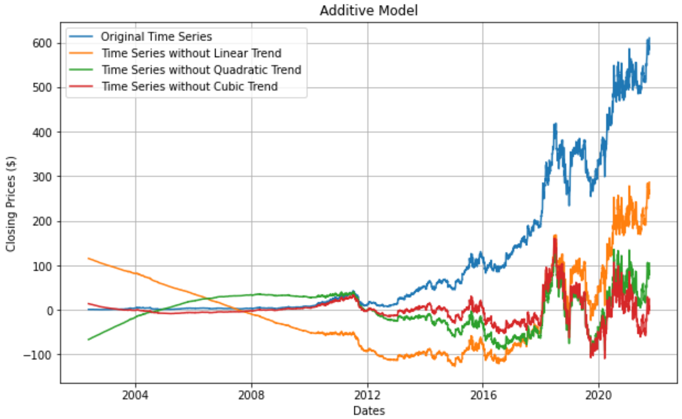

Fig. 9. Netflix's Stock Market since 2002

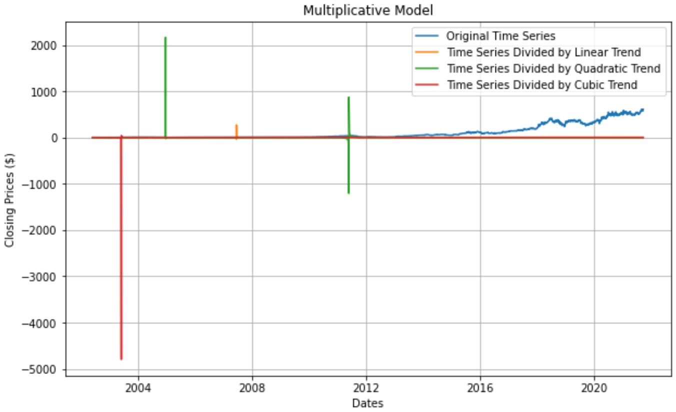

Fig. 10. Netflix's Stock Market since 2002

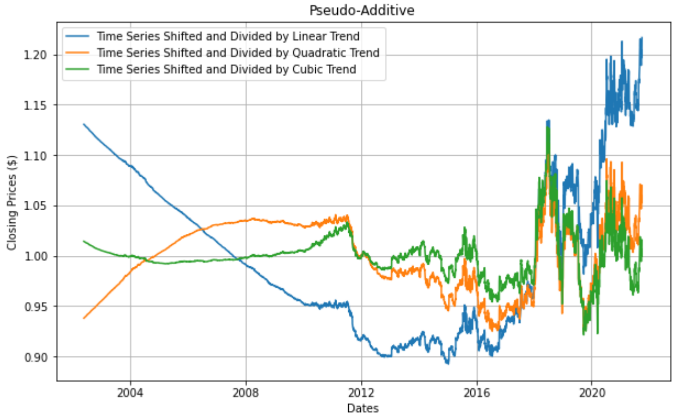

Fig. 11. Netflix's Stock Market since 2002

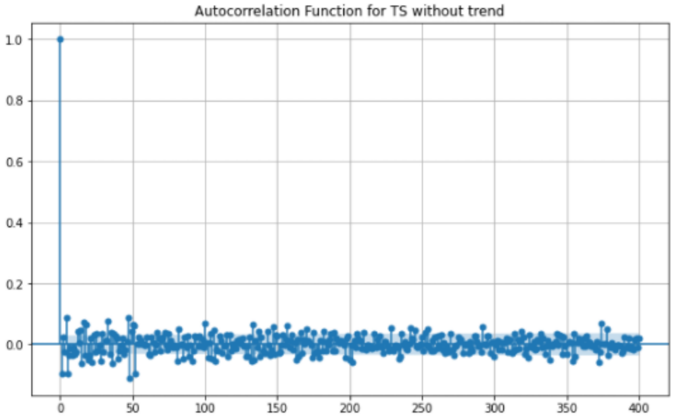

Fig. 12. Netflix's Stock Market since 2002

D. Stationarity

To find if our TS is stationary or not, we used the ADF method. When we tested for the original TS, with a 95% confidence level, we concluded that it was not stationary. After, we applied the first-order differencing and tried once again. This time, with the same 95% confidence level, the TS now was stationary. Knowing this information, we could fix the d value in the ARIMA model (d=1).

E. ARIMA

1) Best Parameters: To find the best parameters, we did a grid-search of the p and q values between [0, 1, ..., 9] (d=1, as explained before). At the same time, we tried to understand if there was a difference between some metrics (Akaike Information Criteria - AIC, Bayesian Information Criteria - BIC, Mean Squared Error - MSE). The Table I shows the results obtained. Although we cannot compare the results in the training set, we can compare them in the testing set. The metric that obtained the best result was the MSE with the result of 1618.165 of MSE in the testing set. From now on, we fixated the parameters of (9, 1, 9). To notice that in the plots, we can see that all the predictions were bad (Figure 13).

	Best Values	**Metric Value in Train**	**Metric Value in Test (MSE)**
AIC	(8, 1, 9)	24724.407	1707.954
BIC	(6, 1, 6)	24814.315	1629.670
MSE	(9, 1, 9)	12.834	1618.165

TABLE I
RESULTS OF ARIMA

2) Semi-Fixed Window Method: In this method, we predict one point at a time, but we have a variable window that increases with those predictions (that window grows in size after each prediction). If we compare this method with the best result of the task of finding the best parameters, we got a decrease of performance of MSE of 22.779. The Figure 14 shows the predictions with the best parameters and this new method.

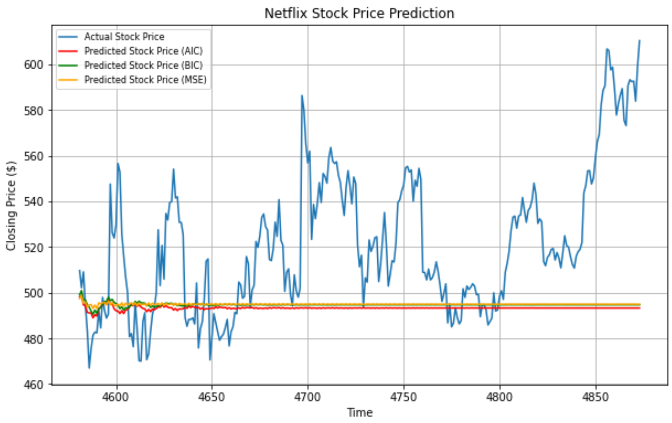

Fig. 13. Netflix’s Stock Market Prediction

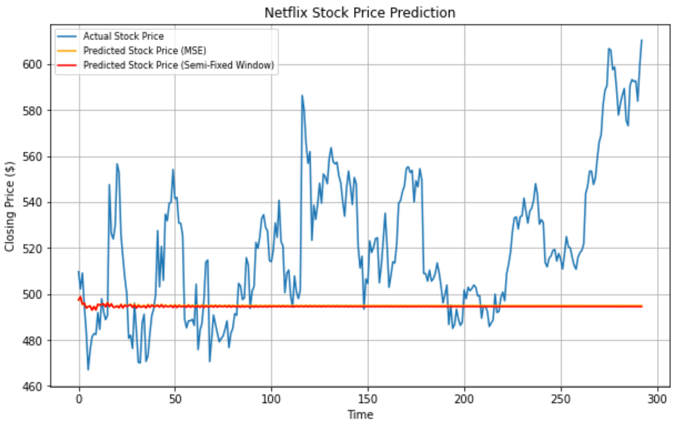

Fig. 14. Netflix’s Stock Market Prediction

3) Rolling Window Method: The Rolling Window Method uses a fixed-sized window that ”rolls” through the TS (in our case, the number of points of the training set): it predicts x points, roll the window by the same x points, and appends the real value to the window. This process repeats until we get to the end of the TS. Table II shows the results obtained. Generally, we can see an up-trend (the bigger the number of points predicted, the higher the error value - MSE in our case) (Figure 15).

Points Predicted	MSE Value
1	145.449
5	358.871
22	1114.158
88	1072.905
252	1326.063

TABLE II
ARIMA - Rolling Window

F. Holt’s Exponential Smoothing (DES)

To find the best parameters, we did a grid-search of the α and β values between [0.10, 0.11, ..., 0.99]. At the same time, we tried to understand if there was a difference between some metrics (Akaike Information Criteria - AIC, Bayesian Information Criteria - BIC, Sum of Squared Errors - SSE). Table III shows the results obtained. Although we cannot compare the results in the training set, we can compare them in the testing set. All the metrics obtained the same best parameters, and their best MSE value in the testing set is 11009.15. To notice that in the plots, we can see that all the predictions were bad (Figure 16).

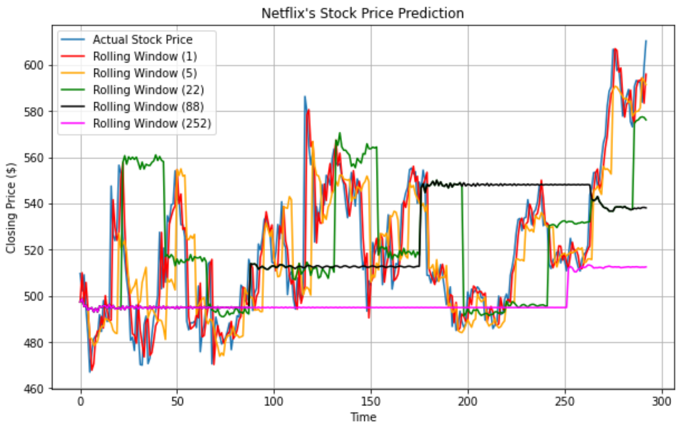

Fig. 15. Netflix’s Stock Market Prediction

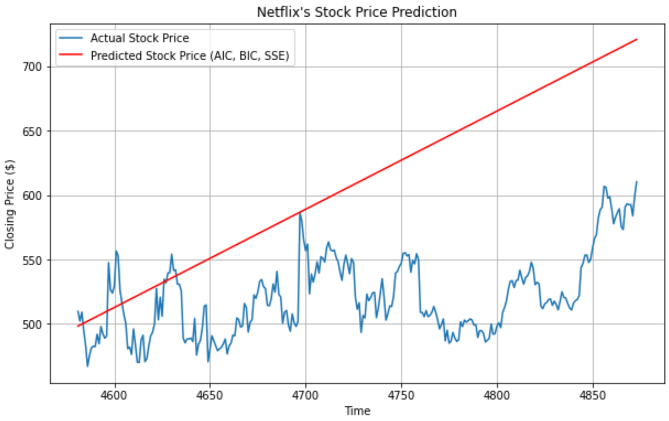

Fig. 16. Netflix’s Stock Market Prediction

	Best Parameters	Metric Value in Train	Metric Value in Test (MSE)
AIC	(0.89, 0.1)	12162.621	11009.153
BIC	(0.89, 0.1)	12188.340	11009.153
SSE	(0.89, 0.1)	65051.902	11009.153

TABLE III
DES

V. Conclusion

In summary, making predictions is hard, especially stock market prices. This work is far from finished but is on a good track. Since they depend on too many external factors (not just purely from the past), simple models such as ARIMA and DES cannot make reasonable predictions. From our study, the best predictor was the ARIMA Rolling Window (1), which means this method achieves good results when we are predicting short-term values. The models that take into account the past values and external variables, such as a Long-Short Term Memory or an Artificial Neural Network can achieve better results, or models that take into account the volability, such as ARCH, GARCH or ARMA-GARCH. One thing is for sure, the first people to achieve great results, will be rich.

References

[1] Ananthi, M., Vijayakumar, K. “Stock market analysis using candlestick regression and market trend prediction (CKRM)”. J Ambient Intell Human Comput 12, 4819-4826 (2021)

[2] George S. Atsalakis, Kimon P. Valavanis, Surveying stock market forecasting techniques - Part II: Soft computing methods, Expert Systems with Applications, Volume 36, Issue 3, Part 2, 2009, Pages 5932-5941

[3] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, ”Stock Price Prediction Using the ARIMA Model,” 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.