

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/353076805>

# Sarima-arch versus genetic programming in stock price prediction

Article · January 2021

DOI: 10.14744/sigma.2021.00001

---

CITATION

1

---

READS

76



## Research Article

# Sarima-arch versus genetic programming in stock price prediction

Gulder KEMALBAY<sup>1,\*</sup>, Ozlem BERAK KORKMAZOGLU<sup>1</sup>

<sup>1</sup>Department of Statistics, Yıldız Technical University, Istanbul, Turkey

## ARTICLE INFO

### Article history

Received: 04 Feb 2020

Accepted: 14 Jan 2021

### Key words:

Genetic Programming, XU100,  
SARIMA, ARCH, Stock Price  
Prediction

## ABSTRACT

In financial time series, one of the most challenging problems is predicting stock prices since the data generally exhibit deviation from the assumptions of stationary and homoscedasticity. For homogenous non-stationary time series, the Autoregressive Integrated Moving Average (ARIMA) model is the most commonly used linear class including some transformation such as differencing and variance stabilizing process. However, stock market data is often nonlinear, which indicates that more advanced methods are necessary. Genetic Programming (GP) is one of the evolutionary computational methods that could capture both linear and nonlinear patterns in time series data. The present study aims to build a machine learning tool using GP for prediction The Istanbul Stock Exchange National 100 (XU100) index and compare the obtained results with conventional seasonal ARIMA(SARIMA) and ARCH models. In order to achieve this goal, it was first modeled with the SARIMA model after appropriate transformations were made to the stock price series and the diagnostic control result showed that the residual of the SARIMA model have the heteroscedasticity problem. Then, the ARCH model was applied to SARIMA residuals to eliminate this effect and an integrated SARIMA-ARCH model is obtained. Since it is possible and capable to model nonlinear and non-stationary time series using GP without any pre-assumptions, we proposed GP to predict the stock price series. The function set of GP consists of not only arithmetic but also trigonometric functions. To the best of our knowledge, this study is the first to predict XU100 stock price data using GP. In this experiment, the data set consists of the daily closing prices of the XU100 index over 775 days from the beginning of 2017 until the end of January 2020. The experimental results obtained show that the accuracy metrics used in the study are lower in the proposed GP model compared to other models. These results reveal that the GP method provides better predictive results for the financial time series data of the XU100 index than traditional methods.

**Cite this article as:** Kemalbay G, Berak Korkmazoglu O. Sarima-arch versus genetic programming in stock price prediction. Sigma J Eng Nat Sci 2021;39(2):110–122.

### \*Corresponding author.

\*E-mail address: [kemalbey@yildiz.edu.tr](mailto:kemalbey@yildiz.edu.tr), [berak@yildiz.edu.tr](mailto:berak@yildiz.edu.tr)

*This paper was recommended for publication in revised form by  
Section Editor Ferkan Yılmaz*



Published by Yıldız Technical University Press, İstanbul, Turkey

Copyright 2021, Yıldız Technical University. This is an open access article under the CC BY-NC license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## INTRODUCTION

Prediction of financial time series is a complex task since the data is often non-stationary and usually exhibit nonlinear patterns. Traditionally, Autoregressive Integrated Moving Average (ARIMA) models are the most widely used linear methods that predict the behavior of time series based on both linear combinations of historical observations called autoregressive (AR) components and error terms called moving average (MA) components [1]. The procedure of fitting an ARIMA model is known as the Box-Jenkins method involves three steps: model identification, parameter estimation, and diagnostic checking [2]. ARIMA methods are applicable to non-stationary time series which can be stabilized by various transformations. If the data display evidence of non-stationary (e.g., having trend or seasonality), the first step should be to remove the trend and seasonal components. One of the simplest and most effective ways to stabilize the variance over time is to apply power transformations such as taking logarithm, square root or cube root of the time series. The other useful transformation technique is differencing the data for eliminating the trend and seasonality. For removing the seasonality, if it is necessary one can take seasonal differencing or use moving average method. If there is a seasonality pattern in time series, following the Box and Jenkins' pragmatic approach, the generalization of ARIMA model called seasonal ARIMA (SARIMA) can be applied for seasonal and non-stationary data. In ARIMA modeling, the error process is assumed to be independent and identically normally distributed, and homoscedastic (i.e. error terms have a constant variance). When the data is non-normal or homoscedasticity assumption is violated, the Box-Cox transformation can be applied since the transformed time series may approximately follow a time process with normal error terms. After the data is transformed into stationary, the next step is the model identification usually based on correlogram and the parameter estimation. Once a model is identified and estimated, the last and the most important step is to verify whether the selected model is adequate by checking the residuals. Diagnosis in the ARIMA model involves checking for residuals between predicted and actual series and determining if they are independent, not serially correlated, normally distributed, and homoscedastic. If the candidate model is not adequate, then the model identification and parameter estimation steps should be repeated until a suitable model is found [3]. Although ARIMA methods are widely used for short-term forecasting, there are some drawbacks of these models. First, many attempts are required until the best model is achieved for a specific data set. Second, the model order identification involves a trial and error approach that is the choice of the best ARIMA model is based on subjective evaluation. Therefore, the performance of the chosen ARIMA model may require an expert analysis. Also, one of its major limitations is that it is applicable to linear time

series. When the linearity assumption is not satisfied, the accuracy performance of prediction results may reduce [4, 5]. Furthermore, when the time series exhibit non-constant variance in error terms called heteroscedasticity problem, this problem sometimes cannot be solved by transforming the data. Then, ARIMA is not appropriate method for heteroscedastic time series forecasting. The autoregressive conditional heteroscedasticity (ARCH) model and its extensions are one of the appropriate time series forecasting methods to deal with the nonlinearity and heteroscedasticity problem. Although several proposed methods focus mostly on predicting linear and stationary time series, a considerable number of studies have emerged in the literature for nonlinear and non-stationary processes. In his fundamental book [6], Priestly made tremendous contributions to this area. In the literature, these time series forecasting methods may be based on neural networks, support vector machines, hidden Markov models, or Bayesian models. See, e.g., [7–15], among others. A comprehensive review study for nonlinear time series can be found in [16].

An alternative approach for time series prediction is derived from the use of evolutionary algorithms. In particular, Genetic Programming (GP) which is a metaheuristic optimization method introduced by Koza [17] is generally used for automatic programming or finding out mathematical functions. Moreover, GP has been efficiently used in time series prediction for capturing both linear and nonlinear relationships. For instance; Kaboudan, proposed a trading strategy based on predictions of stock prices using GP and introduced a new metric to measure the probability of a specific time series is GP-predictable [18]. Duan and Povinelli, extended the Kaboudan's metric to estimate the predictability of time series and presented stock price forecasting using GP [19]. Klůčik et al., used GP as a symbolic regression for forecasting time series of industrial production index and showed that GP outperformed traditional ARIMA models [20]. Lee and Tong, proposed a hybrid method based on ARIMA and GP for nonlinear time series forecasting and the hybrid method outperforms the other forecast methods [21]. Recently, Claveria and Tonna performed symbolic regression via GP to forecast economic growth [22].

The aim of the present paper is to build a predictive model for nonlinear time series of the daily closing stock price value of the Istanbul Stock Exchange National 100 (XU100) index using GP and to compare the prediction performance with traditional time series models. To the best of our knowledge, it is the first study focused on predicting the XU100 stock price index with GP. By using an integrated SARIMA-ARCH as a benchmark model, we aim to uncover the advantages of GP, such as time saving and powerful predictive modeling. The rest of the paper is organized as follows: The next section gives a brief overview of the prediction methods used in this study. Then, the data analysis and obtained experimental results are discussed and a comparison between prediction models is provided. The study is concluded in the last section.

## PREDICTION MODELS

In this part of the study, ARIMA, SARIMA, ARCH, and GP methods are briefly explained in the following subsections.

### ARIMA Model

Autoregressive Moving Average (ARMA) model is one of the most common classes for modeling stationary time series. A stationary time series  $\{Y_t\}$  is an ARMA( $p, q$ ) process if for every  $t$

$$Y_t - \phi_1 Y_{t-1} - \dots - \phi_p Y_{t-p} = e_t + \theta_1 e_{t-1} + \dots + \theta_q e_{t-q} \quad (1)$$

where  $\{e_t\}$  is a sequence of serially uncorrelated error terms with zero mean and finite variance (shortly, it is said to be white-noise);  $\phi(\cdot)$  and  $\theta(\cdot)$  are the  $p^{\text{th}}$  and  $q^{\text{th}}$  degree polynomials sharing no common factors, respectively. Using backshift operator  $B$  (which is defined as  $B^j Y_t = Y_{t-j}$ ,  $B^j e_t = e_{t-j}$ ,  $j = 0, \dots, \pm 1, \dots$ ), ARMA( $p, q$ ) process is represented as follows:

$$\phi(B)Y_t = \theta(B)e_t \quad (2)$$

where  $\phi(B) = 1 - \sum_{i=1}^p \phi_i B^i$  and  $\theta(B) = 1 + \sum_{i=1}^q \theta_i B^i$ .

ARIMA model which is reproduced by alteration of an ARMA model can be applied for homogeneous non-stationary time series. Frequently, the non-stationary data can be transformed into stationary data by taking proper degree of difference. ARIMA model is defined by three parameters  $p, d$ , and  $q$  which represent the order of autoregressive, differencing and moving average parameters, respectively. Then, ARIMA( $p, d, q$ ) process is expressed using backshift operator  $B$  as follows:

$$\phi(B)\Delta^d Y_t = \theta(B)e_t \quad (3)$$

where  $\Delta^d = (1 - B)^d$  indicates  $d$  nonnegative order of difference where  $\Delta$  is difference operator and  $\{e_t\}$  is white-noise. The ARIMA( $p, d, q$ ) process reduces into ARMA( $p, q$ ) if and only if  $d$  is equal to zero. In other words,  $\{Y_t\}$  is an ARIMA ( $p, d, q$ ), ( $d > 0$ ) process if  $X_t = (1 - B)^d Y_t$  is an ARMA( $p, q$ ) process. For more details about ARIMA models we refer to [23], among others.

### SARIMA Model

Since the ARIMA model given in the equation (3) is for non-stationary and non-seasonal time series data, Box and Jenkins [2] proposed a generalization to ARIMA model to deal with seasonality which is called SARIMA model. Suppose that  $d$  and  $D$  are nonnegative integers.  $\{Y_t\}$  is a seasonal ARIMA( $p, d, q$ )  $\times$  ( $P, D, Q$ ) [ $s$ ] process with period  $s$  if the differenced series  $X_t = (1 - B)^d (1 - B^s)^D Y_t$  is an ARMA process defined as follows:

$$\phi(B)\Phi(B^s)X_t = \theta(B)\Theta(B^s)e_t \quad (4)$$

where  $\{e_t\}$  is white-noise;  $\phi(B) = 1 - \sum_{i=1}^p \phi_i B^i$ ,  $\Phi(B^s) = 1 - \sum_{i=1}^P \phi_i B^{si}$ ,  $\theta(B) = 1 + \sum_{i=1}^q \theta_i B^i$ , and  $\Theta(B^s) = 1 + \sum_{i=1}^Q \theta_i B^{si}$ .

SARIMA model is defined by parameters  $p, d, q, P, D$ , and  $Q$  which represent the order of autoregressive, differencing, moving average, seasonal autoregressive, seasonal differencing, and seasonal moving average parameters, respectively [23].

### ARCH Model

An ARIMA model for the time series  $\{Y_t\}$  assumes that the conditional variance  $h_t$  of  $\{Y_t\}$  given  $\{Y_s, s < t\}$  is independent from  $t$  and from  $\{Y_s, s < t\}$  that is conditional variance is constant through time. However, return series of financial assets often exhibit heteroscedasticity due to the volatility clustering property. This property means that large changes in return series tend to be followed by large changes, and small changes in return series tend to cluster together. One of the useful ways to deal with heteroscedasticity problem is to use autoregressive conditional heteroscedasticity (ARCH) models proposed by Engle [24] and its extensions. If  $P_t$  is the closing price of a particular stock index at time  $t$ , then the return of the financial time series is defined as  $\log(P_t) - \log(P_{t-1}) = \Delta \log P_t$  and usually denoted by  $r_t$ . Let  $h(t) = \sigma_{t|t-1}^2$  denotes the conditional variance or conditional volatility of  $r_t$ , given returns through time  $t - 1$ . Then, ARCH(1) model for the  $\{r_t\}$  is given as follows:

$$r_t = \sigma_{t|t-1} e_t \quad (5)$$

$$h(t) = \alpha_0 + \alpha_1 r_{t-1}^2 \quad (6)$$

where  $\{e_t\} \sim iid N(0,1)$ ,  $\{e_t\}$  is independent of  $r_{t-1}$ ,  $i = 1, 2, \dots$ , and  $\alpha_0 > 0$ ,  $\alpha_i \geq 0$  are unknown parameters. The ARCH( $q$ ) model is the extension of the equation (6) and given as follows:

$$h(t) = \alpha_0 + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \dots + \alpha_q r_{t-q}^2 \quad (7)$$

where  $\alpha_0 > 0$ ,  $\alpha_i \geq 0$ ,  $i = 1, \dots, q$ , and  $q$  is a positive integer and referred to as ARCH order.

The generalized ARCH (GARCH) model is introduced by Bollerslev [25] and GARCH ( $p, q$ ) model is the generalization of the equation (7) and defined as follows:

$$h(t) = \alpha_0 + \beta_1 h_{t-1} + \beta_2 h_{t-2} + \dots + \beta_p h_{t-p} + \alpha_1 r_{t-1}^2 + \alpha_2 r_{t-2}^2 + \dots + \alpha_q r_{t-q}^2 \quad (8)$$

where  $\alpha_0 > 0$ ,  $\beta_i \geq 0$ ,  $i = 1, \dots, p$ ,  $\alpha_i \geq 0$ ,  $i = 1, \dots, q$ ; and  $p$  and  $q$  are positive integers and referred to as GARCH order and ARCH order, respectively [23].

### GP Model

Genetic Programming, which is originally developed by Koza [17] as an extension of the genetic algorithm, is an evolutionary computation method based on ideas of biological evolution to handle a complex problem. GP is generally used in problems such as discovering mathematical functions or automatic programming. Therefore, it is also referred as symbolic regression. The structure of the Genetic Programming consists of the following steps:

- Step1: Define the terminal set which are the input variables of the problem and constants.
- Step2: Define the set of primal functions usually include arithmetic operators or other mathematical functions.
- Step3: Randomly create an initial population of individual programs consists of existing function and terminal set.
- Step4: Select a fitness function to measure the accuracy of prediction.
- Step5: Determine the parameters for controlling the algorithm.
- Step6: Determine the stopping condition.
- Step7: Iteratively execute the following sub-steps i–iv called generation until the stopping condition is met.
  - i) Run each program in the population and figure out its fitness using the problem's fitness function.
  - ii) Select one or two individual computer program from the population based on fitness function to participate in the four genetic operations given in the next step.
  - iii) Create new individuals for the population by using the following genetic operations:
    - a) Reproduction: Copy the selected individuals from the current generation to the next population.
    - b) Crossover: By recombining randomly chosen parts from two selected individuals, create new individuals for the next population
    - c) Mutation: By randomly mutating a randomly chosen part of one selected individuals, create one new individual for the next population
    - d) Inversion: Choose an inversion operation from the operations and create one new individual for the next population by using the chosen inversion operation to one selected individual.
  - iv) When the stopping criterion is satisfied, the best program (individual) in the population produced during the algorithm. The result may be an approximate solution to the problem.

For detailed information about Genetic Programming, we refer to well-known book of Koza [17], among others.

### Performance Evaluation

In the present study, the Mean Squared Error (MSE), the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), the Mean Percentage Error (MPE), the Mean

Absolute Percentage Error (MAPE), the symmetric Mean Absolute Percentage Error (sMAPE), the Mean Error (ME), and The Theil's U-statistics are used as metrics to measure and compare the prediction accuracy of methods. These measures are defined as follows:

$$MSE = \frac{1}{n} \sum_{i=1}^n e_t^2 \quad (9)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n e_t^2} \quad (10)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |e_t| \quad (11)$$

$$MPE = \frac{1}{n} \sum_{i=1}^n \left( \frac{e_t}{y_t} \right) \times 100\% \quad (12)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{e_t}{y_t} \right| \times 100\% \quad (13)$$

$$sMAPE = 100 \times \frac{1}{n} \sum_{i=1}^n \frac{|e_t|}{(|y_t| + |\hat{y}_t|) / 2} \quad (14)$$

$$ME = \frac{1}{n} \sum_{i=1}^n e_t \quad (15)$$

$$Theil's U = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n e_t^2}}{\sqrt{\frac{1}{n} \sum_{i=1}^n y_t^2} \sqrt{\frac{1}{n} \sum_{i=1}^n \hat{y}_t^2}} \quad (16)$$

where  $y_t$  is actual value,  $\hat{y}_t$  is predicted value,  $e_t = y_t - \hat{y}_t$  prediction error and  $n$  is the sample size.

### EXPERIMENTAL RESULTS

The data set of the present study consists of daily closing price the XU100 stock index over 775 days, from the beginning of 2017 until the end of January 2020 and it is obtained from the website <https://tr.investing.com/>. It is a usual practice to use the first  $n$  observations as a training period and to test it in data separated from the training period. We split the available data consisting of total 775 observations in two parts of 70%–30%. Data from the date 02-Jan-17 to 22-Feb-19 are used as the training period (542 observations) and the test period (233 observations) is from the date 25-Feb-19 to 31-Jan-20. The predictive models are fitted on training data and model performances are measured and compared on the test period. All computations are performed via the R programming version of 3.6.2.

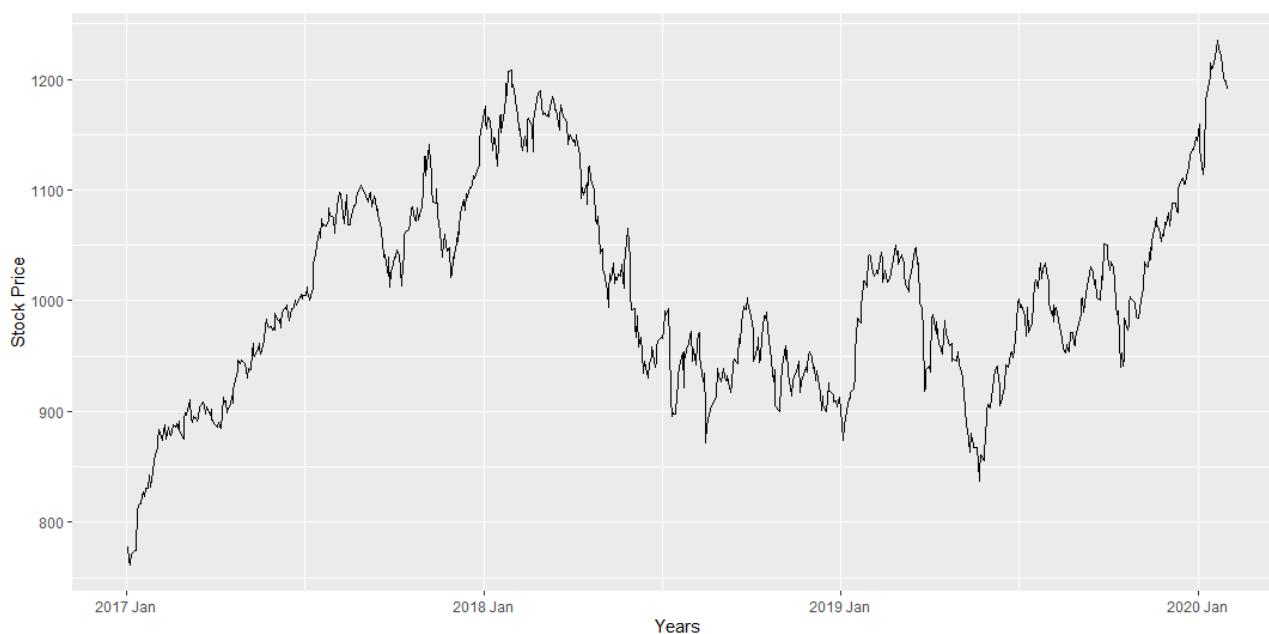
Time series graphs or called time-plots are useful to observe how the series exhibits patterns over time or to

predict, for example, the stock market index is trending up or down. The behaviour of the daily closing stock price index of the XU100 is investigated by the time-plot given in Figure 1.

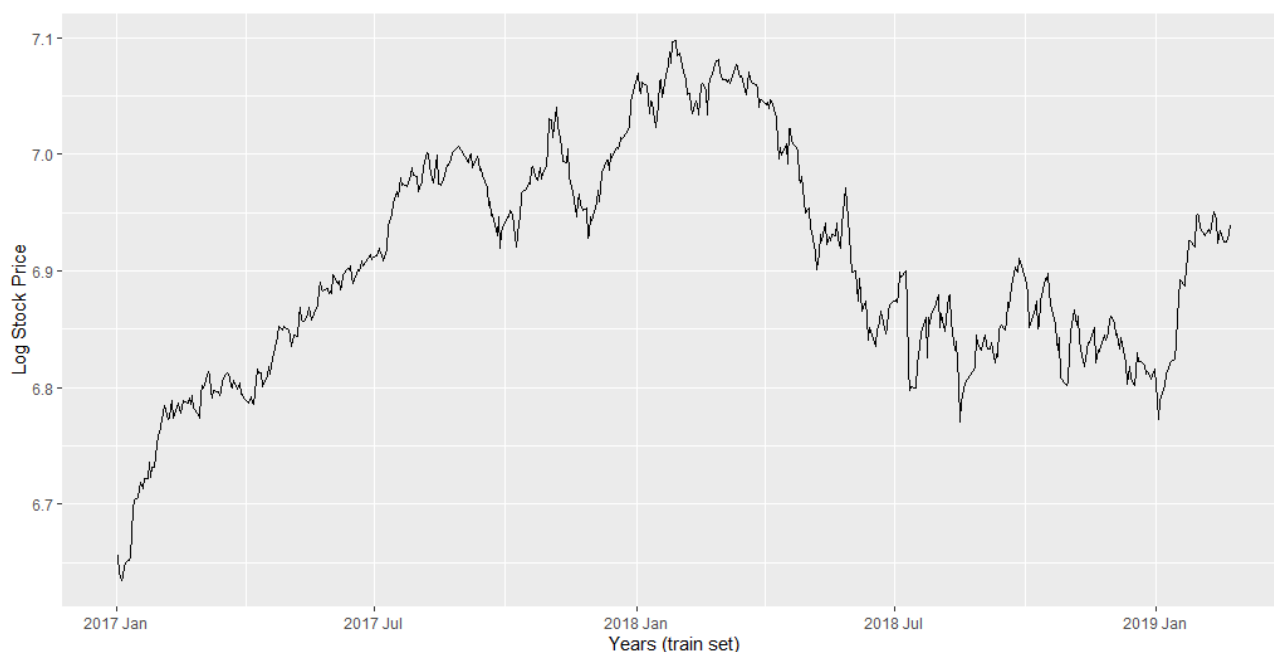
From Figure 1, some distinguishable patterns such that upward and downward trends are observed. It indicates that the series does not have a stable mean and variance that is the original pattern of the XU100 series is not stationary. For the initial step, we apply log transformation to the stock price data to try to stabilize the variance over time. Suppose that

the closing price of XU100 index on trading day  $t$  is denoted by  $P_t$ . Then, natural logarithm of closing price is denoted by  $\{Y_t\}$ , where  $Y_t = \log P_t$ . Then, the time-plot of log-transformed data on training period is given in Figure 2.

From Figure 2, the log stock price data have more constant fluctuations, but still have non-constant variance and exhibit trend component. Below in Figure 3, to confirm the indication of non-stationary, we examine the sample autocorrelation function (ACF) and the sample partial autocorrelation function (PACF), and Augmented Dickey-Fuller



**Figure 1.** The time series graph of daily XU100 closing stock price index.



**Figure 2.** The time series graph of daily XU100 closing log stock price index on training period.



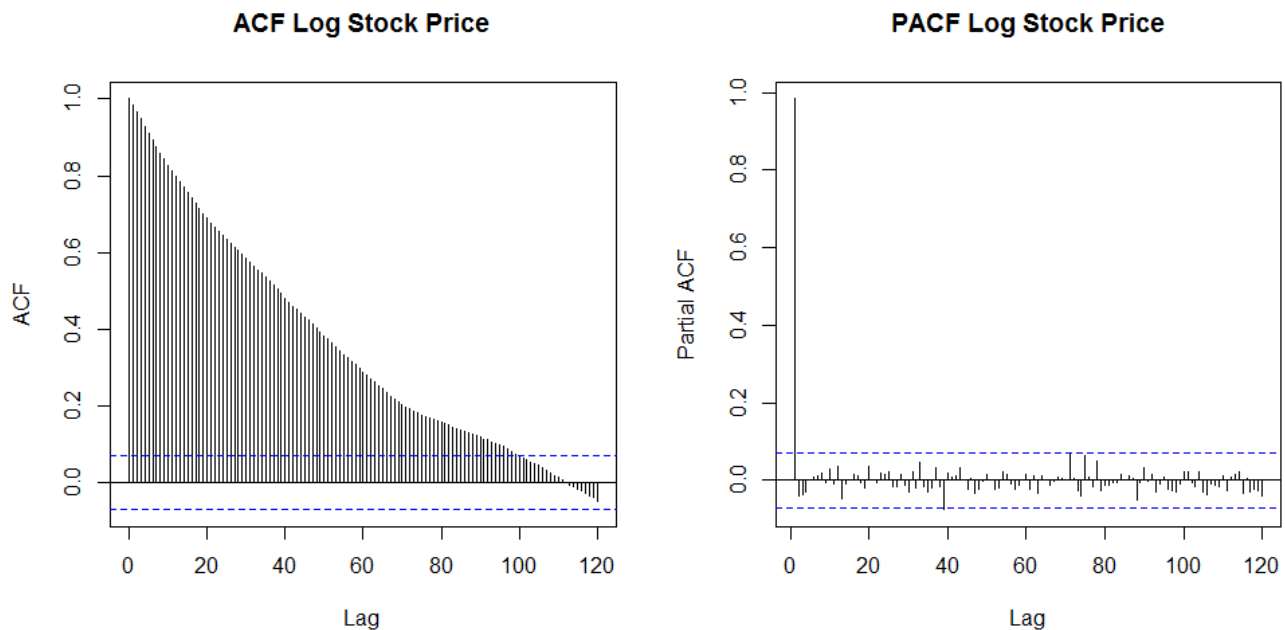


Figure 3. ACF and PACF graphs of log stock price series.

Table 1. Augmented Dickey-Fuller test for the level of log stock price series

$Y_t = \log P_t$	Level								
	neither intercept nor trend			an intercept and no trend			both intercept and trend		
Model	%1	%5	%10	%1	%5	%10	%1	%5	%10
Critical Values	-2.58	-1.95	-1.62	-3.43	-2.86	-2.57	-3.96	-3.41	-3.12
DF Statistics	1.1917			-2.5731			-2.5354		

(ADF) test results are given in Table 1 for the level of log stock price series.

According to Figure 3, the ACF decays extremely slowly which strongly points out the time series of XU100 is non-stationary. This result is also supported by the results of ADF test given in Table 1. The number of lag in the test equation is determined as 5 using Akaike Information Criteria (AIC) and the test equation including both intercept and trend is given as

$$\Delta Y_t = \alpha + \beta t + \delta Y_{t-1} + \sum_{j=2}^5 \delta_j \Delta Y_{t-j} + e_t$$

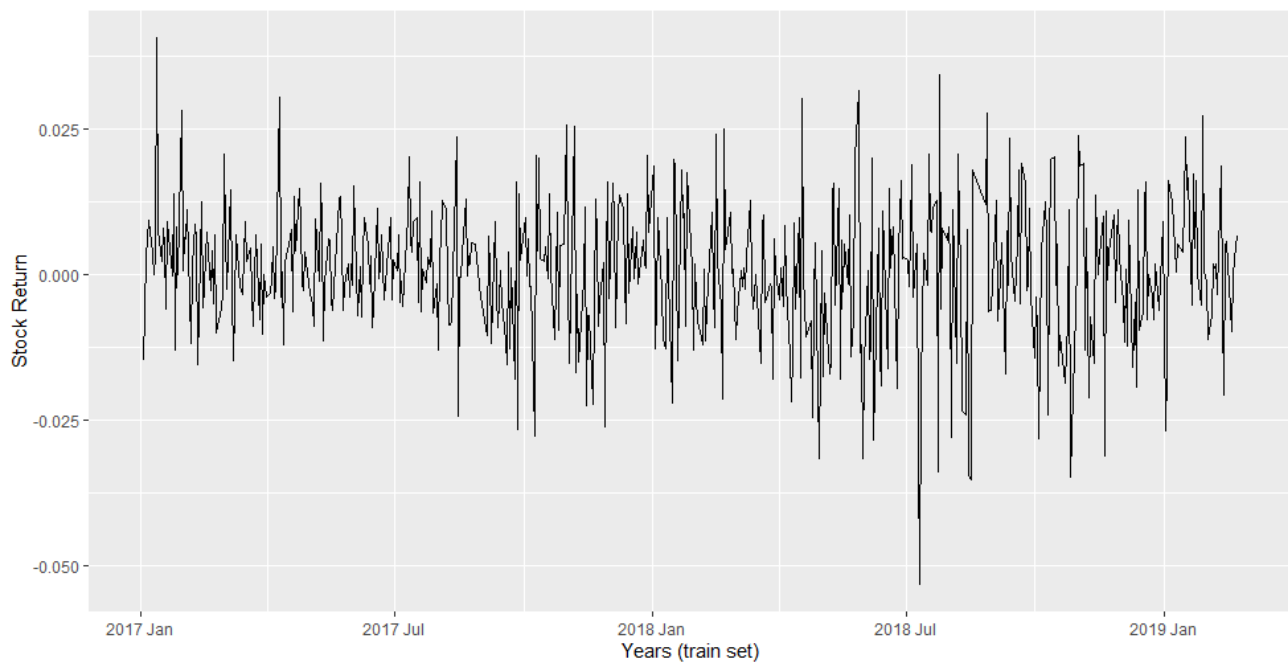
where the hypothesis are  $H_0: \delta = 0$  and  $H_1: \delta < 0$ . The DF test statistics of zero mean (neither intercept nor trend), single mean (an intercept and no trend), and trend (both intercept and trend) models are greater than the critical values for 1%, 5% and 10% significance levels. As a result, the null hypothesis that the  $Y_t = \log P_t$  series contains a unit root is failed to reject, i.e.,  $Y_t = \log P_t$  series in level is not stationary. Therefore, the 1<sup>st</sup> differencing transformation is performed to  $Y_t = \log P_t$  series as follows:  $\Delta \log P_t =$

$\log(P_t) - \log(P_{t-1})$ . The first difference of logged stock price series is called stock return series and denoted by  $r_t$ . After the 1<sup>st</sup> differencing, XU100 index series becomes stationary as shown in Figure 4.

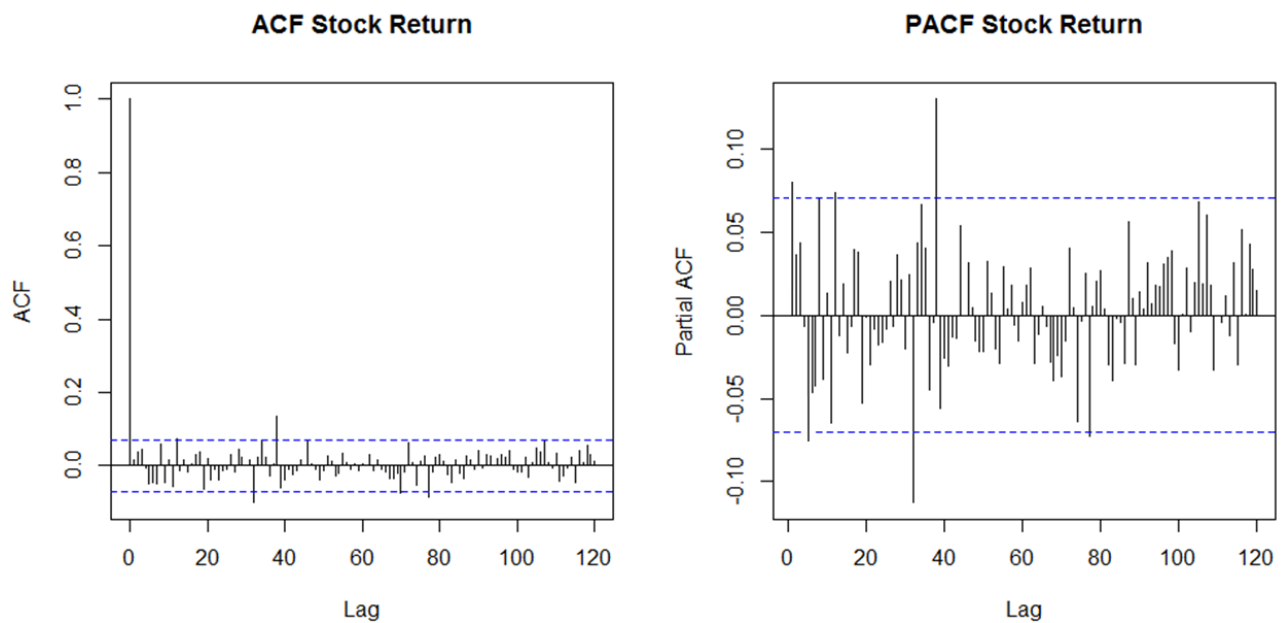
From Figure 4, the stock return series of XU100 index appears to randomly oscillate around zero which means that there is rather weak autocorrelation and this result is confirmed by the sample ACF graph displaying that the autocorrelation coefficients are close to zero given in Figure 5. However, there is evidence that the stock return series exhibit heteroscedasticity because the volatility clustering can be observed in Figure 4. We will address this problem in the diagnostic check of model residuals.

Moreover, we implement ADF test for the first difference of log stock price series, i.e., stock return series. The results are presented in Table 2.

According to Table 2, The DF test statistics of zero mean (neither intercept nor trend); single mean (an intercept and no trend), and trend (both intercept and trend) models are less than the critical values for 1%, 5% and 10% significance levels. Therefore, the null hypothesis that the  $r_t = \Delta \log P_t$  series contains a unit root is rejected, i.e.,  $Y_t = \log P_t$  series



**Figure 4.** The time series graph of daily stock return on training period.



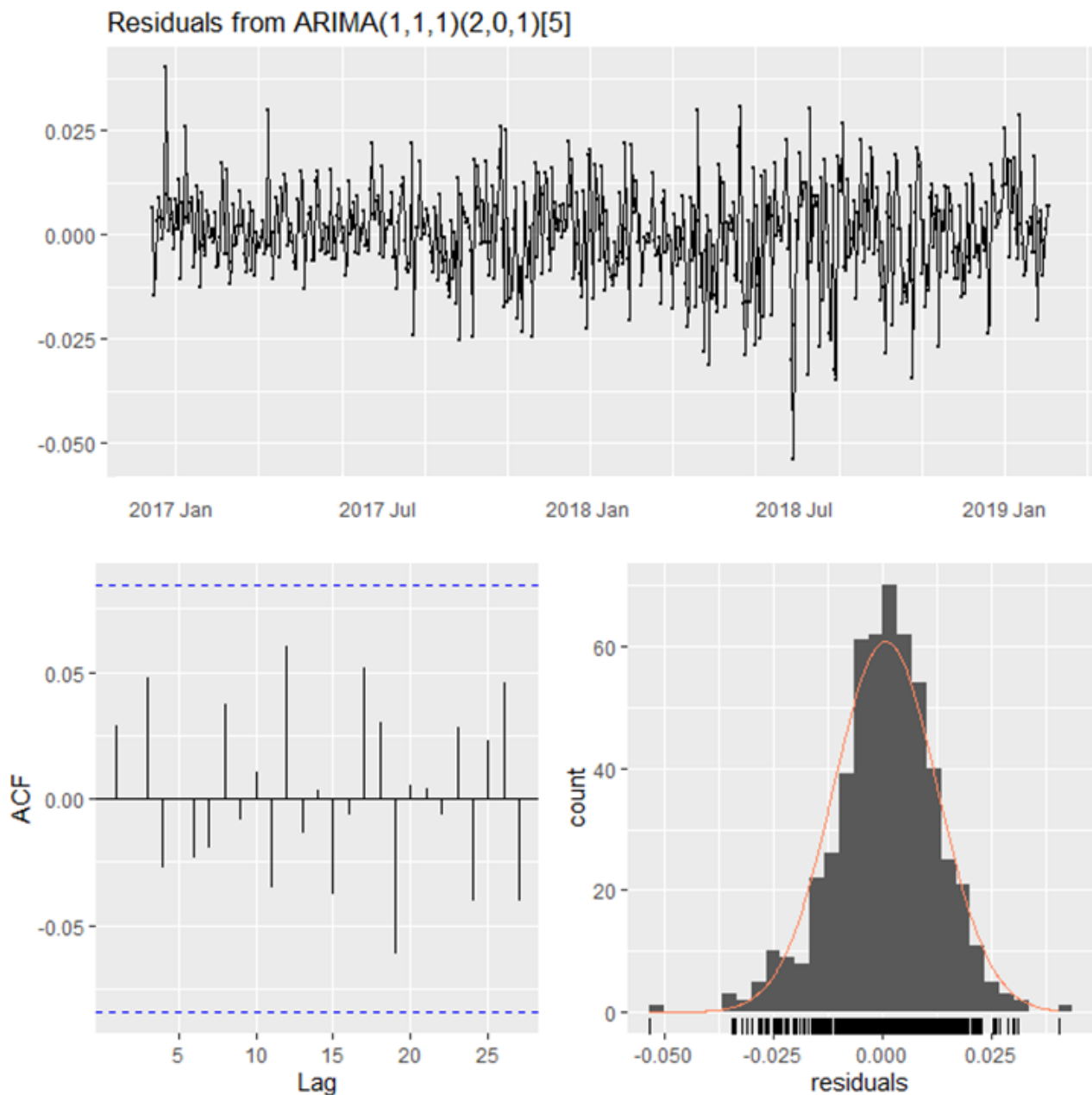
**Figure 5.** ACF and PACF graphs of stock return series.

**Table 2.** Augmented Dickey-Fuller test for the first difference of log stock price series

$r_t = \Delta \log P_t$	First Difference								
	neither intercept nor trend			an intercept and no trend			both intercept and trend		
Critical Values	%1 -2.58	%5 -1.95	%10 -1.62	%1 -3.43	%5 -2.86	%10 -2.57	%1 -3.96	%5 -3.41	%10 -3.12
DF Statistics	-18.647			-18.6888			-18.6837		







**Figure 6.** The residual analysis for SARIMA model.

The residual analysis given in Figure 6 shows that there are not significant spikes in the ACF graph and residuals are approximately normally distributed. However, the variance of residuals does not seem to be constant over time. The graphical results are also confirmed as follows.

In Table 6, we observe that the p-values corresponding to Ljung-Box Q statistics at different lags are greater than the significance level 5%. Therefore, the null hypothesis that the residuals are independently distributed is failed to reject, that is there is no autocorrelation in the residuals of SARIMA model.

It is noticed from Table 7 that the p-values corresponding to the ARCH-LM statistics are less than the all significance levels. Thus, the null hypothesis that there is no ARCH effect is rejected. To handle with the heteroscedasticity problem observed in the residuals, we fit ARCH models to the residuals series of obtained SARIMA model, i.e.,  $ARIMA(1,1,1)(2,0,1)$  [5] and the fitted  $ARCH(1)$  model is given in Table 8. We would also like to point out that  $ARCH(q)$  and  $GARCH(p,q)$  models with order greater than 1 are also implemented but their parameters found insignificant.

**Table 8.** ARCH(1) model results

Coefficients	Estimate	Standard Error	p-value
$\alpha_0$	1.177e-04	8.438e-06	0.000
$\alpha_1$	1.724e-01	4.982e-02	0.000
Ljung-Box Test for Squared Residuals	$\chi^2$	df	p-Value
	0.095893	1	0.7568

According to Table 8, the fitted parameters of ARCH(1) are statistically significant, and the model is represented as

$$h(t) = 0.0001177 + 0.1724e^2_{t-1} + u_t$$

where  $\{u_t\}$  is the error term series. Table 9 gives the ARCH-LM test results for ARCH(1) model residuals.

It is clearly seen from Table 9 that the p-values corresponding to the ARCH-LM statistics are greater than the significance level 5%. Then, the null hypothesis that there is no ARCH effect in ARCH(1) residuals is failed to reject.

Consequently, the adequacy of the model is ensured and we integrate the ARCH(1) model to the ARIMA(1,1,1) (2,0,1)[5] model and the integrated model shortly called SARIMA-ARCH.

In Table 10, we compare the SARIMA and SARIMA-ARCH model performances on training period to emphasize the improvement in accuracy metrics, and we would

also like to remark that we do not use SARIMA model for the prediction as it does not provide the homoscedasticity assumption.

According to Table 10, it can be said that the proposed SARIMA-ARCH model provides slightly better predictive performance than the SARIMA model for XU100 stock return series. Then, we continue to model the XU100 stock return series with one of the evolutionary computational methods GP.

Now using the parameters given in Table 11, Genetic Programming is adopted to fit training data of the stock return series  $r_t$  using function set with arithmetic functions (+, −, ×, /) also including other mathematical functions sinus (sin), cosinus (cos), natural logarithm (log), exponential (exp), square root (sqrt). In order to make comparison, we use the same  $p$  order of autoregressive variables of SARIMA model for the terminal set and therefore the input variables are  $x_1 = r_{t-1}$ ,  $x_2 = r_{t-5}$  and  $x_3 = r_{t-10}$ .

The log and sqrt functions are avoided against negative arguments and / operator is avoided against division by 0. The fitness function is selected as MAE. Based on the fitness function, for reproduction the selection of individuals is done with the tournament selection algorithm. Crossover and mutation operators are applied to the selected individuals generating new individuals for the next generations. The size of population is 100. In Table 11, the parameter of GP used in this experiment is summarized.

Eventually, the Genetic Programming algorithm is stopped after 1000 evolution steps, fitness evaluation

**Table 9.** ARCH-LM test results for ARCH(1) model residuals

Lag	10	15	20	25	35	50	100
ARCH-LM Statistic	12.007	12.182	14.445	20.537	31.804	36.803	57.054
p-value	0.285	0.665	0.807	0.7181	0.6232	0.918	0.999

**Table 10.** SARIMA and SARIMA-ARCH model performance metrics for training set

Model	MSE	RMSE	MAE	MPE	MAPE	sMAPE	ME	Theil's U
SARIMA	0.00014	0.01193	0.00920	0.00819	0.13331	0.13331	0.00057	0.00025
SARIMA-ARCH	0.00014	0.01192	0.00919	0.00613	0.13314	0.13314	0.00043	0.00025

**Table 11.** GP configuration

Population size	100
Function set	(+, −, ×, /, sin, cos, log, exp, sqrt)
Terminal set	$r_{t-1}, r_{t-5}, r_{t-10}$
Fitness function	Mean Absolute Error
Crossover function	Random subtree crossover
Mutation rate	0.3
Crossover rate	0.7

number is 49950, and the best fitted approximated solution is given as follows:

$$f(x_1, x_2) = x_1 - \cos(-2.425 + x_1) / \sin(x_1) \\ / ((\sin(\cos(-2.425 + x_1)) / \sin(x_1)) \\ / ((\sin(x_2) + x_1 \times x_2) \times x_1)) + \\ x_1 \times (x_1 - \cos(-2.425 + x_1) / \sin(x_1)) \\ / ((\sin(x_1) + x_1 \times x_2) \times x_1))) \times x_1$$

The model performance of the best GP model on training period is represented in Table 12.

Finally, we compare the prediction performance of SARIMA-ARCH and GP models on test samples according to MSE, RMSE, MAE, MPE, MAPE, sMAPE, ME, and Theil's U metrics. The performance results are presented in Table 13.

From Table 13, it is obviously seen that GP has better predictive performance than SARIMA-ARCH model according to RMSE, MAE, MPE, MAPE, sMAPE, and ME metrics. For instance, GP fits the data with MAE of 0.00975 while SARIMA-ARCH model produce MAE of 0.00981.

The results show that GP can perform well on the data set XU100 stock return series.

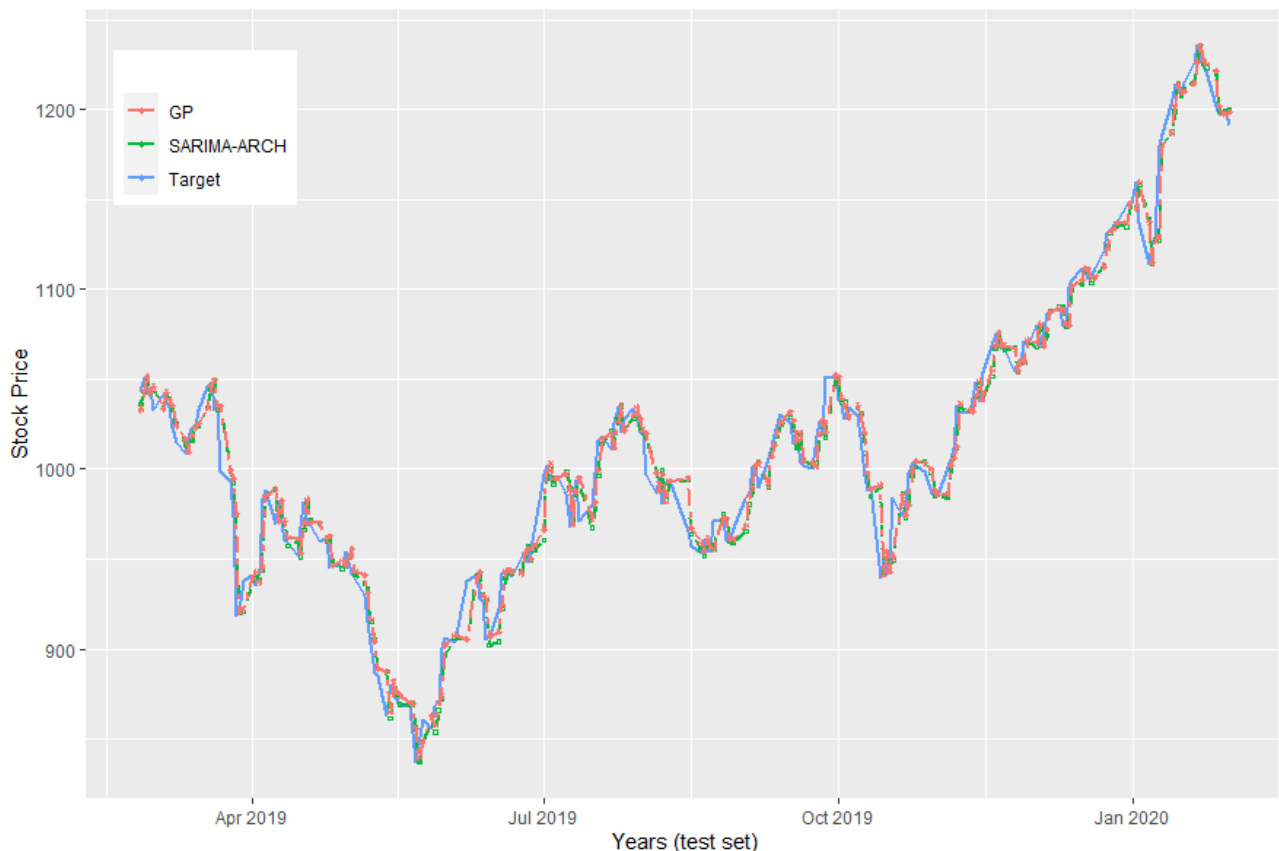
In Figure 7, the one-step-ahead predictions that are the test performance on the stock price data set are presented. The representation of Genetic Programming only includes AR and SAR terms, i.e., past values of stock return data, but the improved solutions are nonlinear due to the operators belong to the function set. The graphical and experimental results show that GP yields statistically lower prediction errors for the daily XU100 stock price index series relative to integrated SARIMA-ARCH model.

**Table 12.** GP model performance metrics for training set.

Model	MSE	RMSE	MAE	MPE	MAPE	sMAPE	ME	Theil's U
GP	0.00014	0.01204	0.00913	-0.01083	0.13222	0.13219	-0.00074	0.00025

**Table 13.** SARIMA-ARCH and GP model performance comparison for test set

Model	MSE	RMSE	MAE	MPE	MAPE	sMAPE	ME	Theil's U
SARIMA-ARCH	0.00018	0.01346	0.00981	0.00680	0.14215	0.14215	0.00049	0.00028
GP	0.00018	0.01323	0.00975	-0.00866	0.14134	0.14132	-0.00057	0.00028



**Figure 7.** The out-of-sample performances for SARIMA-ARCH and GP models.

## CONCLUSION

The purpose of this study is to propose one of the evolutionary optimization methods called Genetic Programming for nonlinear and non-stationary XU100 stock price series prediction and compare the results with the classical time series predictive models. ARIMA is the well-known prediction method based on some assumptions such as linearity, stationary, autocorrelation, normality, white-noise, and homoscedasticity. Since financial time series often exhibit deviation from stationary, and homoscedasticity assumptions, the prediction results of ARIMA and its extensions may not be accurate. One of the ways to deal with these problems is to model the residual series of ARIMA with ARCH models, and its extensions. Therefore, predicting financial time series with traditional methods need not only an expert analysis but also many attempts to achieve the best model with much more time.

For this purpose, the first difference of log stock price series, called the stock return series, was first modelled with seasonal ARIMA model. By following the Box and Jenkins' approach, the best model was chosen as ARIMA(1,1,1)(2,0,1)[5] and diagnostic check results showed that the variance of obtained SARIMA model residuals was not constant over time. We removed this effect applying ARCH(1) model to the residual series. Then, ARCH(1) model was integrated to the ARIMA(1,1,1)(2,0,1)[5] model, and the integrated model was named SARIMA-ARCH model. The adequacy of the model was ensured by checking the Ljung-Box autocorrelation test and the ARCH-LM heteroscedasticity test. To achieve our goal, Genetic Programming was adopted to fit the stock return series data using the function set with not only arithmetic but also trigonometric functions. In order to make comparison, the terminal set of GP was built based on the same order of autoregressive variables of SARIMA model. Finally, the predictive performances of SARIMA-ARCH model and GP model were measured and compared on the test period using the MSE, RMSE, MAE, MPE, MAPE, sMAPE, ME, and Theil's U metrics. According to the obtained results, GP provided reasonably good predictions for stock return time series. The in-sample and out-of-sample performances show that with lower performance metric criteria the GP model outperforms the SARIMA-ARCH model.

Comparing to traditional SARIMA-ARCH model, besides having better predictive performance, the major advantages of the GP model are that there is no pre-assumption about model, it is practical, time-saving, capturing both linear and nonlinear pattern in time series. As a result, it can be said that Genetic Programming could be successfully used for predictive modelling of stock prices and could yield significant benefits not only for financiers but also for researchers

## NOMENCLATURE

$p$	Order of autoregressive parameter
$d$	Order of differencing parameter
$q$	Order of moving average parameter
$P$	Order of seasonal autoregressive parameter
$D$	Order of seasonal differencing parameter
$Q$	Order of seasonal moving average parameter
$B$	Backshift operator
$P_t$	Closing price of stock index
$r_t$	Stock return series
$h_t$	Conditional variance

### Greek symbols

$\phi$	Autoregressive parameter
$\theta$	Moving average parameter
$\Phi$	Seasonal autoregressive parameter
$\Theta$	Seasonal moving average parameter
$\alpha$	Autoregressive conditional heteroscedasticity parameter
$\beta$	Generalized autoregressive conditional heteroscedasticity parameter
$\Delta$	Difference operator

### Subscripts

$t$	Refers to time
-----	----------------

## AUTHORSHIP CONTRIBUTIONS

Concept: G.K.; Design: G.K.; Supervision: G.K.; Materials: G.K.; Data: G.K., O.B.K.; Analysis: G.K.; Literature search: G.K., O.B.K.; Writing: G.K., O.B.K.; Critical Revision: G.K.

## DATA AVAILABILITY STATEMENT

The published publication includes all graphics and data collected or developed during the study.

## CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## ETHICS

There are no ethical issues with the publication of this manuscript.

## REFERENCES

- [1] Newbold, P. ARIMA model building and the time series analysis approach to forecasting. *J Forecast* 1983; 2: 23–35. <https://doi.org/10.1002/for.3980020104>.

- [2] Box GE, Jenkins GM. Time Series Analysis: Forecasting and Control. California: Holden-Day; 1976.
- [3] Hipel KW, McLeod AI. Time Series Modelling of Water Resources and Environmental Systems. Netherlands: Elsevier; 1994.
- [4] Hanke JE, Wichern DW. Business Forecasting. 9th ed. New Jersey; 2009.
- [5] Petrică AC, Stancu S, Tindeche A. Limitation of ARIMA models in financial and monetary economics. Theoret Appl Econ 2016; 23: 19–42.
- [6] Priestley MB. Non-linear and Non-stationary Time Series Analysis. London: Academic Press; 1988.
- [7] Guresen E, Kayakutlu G, Daim TU. Using artificial neural network models in stock market index prediction. Expert Syst Appl 2011; 38: 10389–97. <https://doi.org/10.1016/j.eswa.2011.02.068>.
- [8] Göçken M, Özçalıcı M, Boru A, Dosdoğru AT. Integrating metaheuristics and artificial neural networks for improved stock price prediction. Expert Syst Appl 2016; 44: 320–31. <https://doi.org/10.1016/j.eswa.2015.09.029>.
- [9] Oztekin A, Kizilaslan R, Freund S, Iseri A. A data analytic approach to forecasting daily stock returns in an emerging market. Eur J Oper Res 2016; 253: 697–710. <https://doi.org/10.1016/j.ejor.2016.02.056>.
- [10] Sarıca B, Eğrioğlu E, Aşıkil B. A new hybrid method for time series forecasting: AR–ANFIS. Neural Comput Appl 2018; 29: 749–60. <https://doi.org/10.1007/s00521-016-2475-5>.
- [11] Henrique BM, Sobreiro VA, Kimura H. Stock price prediction using support vector regression on daily and up to the minute prices. J Finance Data Sci 2018; 4: 183–201. <https://doi.org/10.1016/j.jfds.2018.04.003>.
- [12] Song Y, Lee JW, Lee J. A study on novel filtering and relationship between input-features and target-vectors in a deep learning model for stock price prediction. Appl Intell 2019; 49: 897–911. <https://doi.org/10.1007/s10489-018-1308-x>.
- [13] Göçken M, Özçalıcı M, Boru A, Dosdoğru AT. Stock price prediction using hybrid soft computing models incorporating parameter tuning and input variable selection. Neural Comput Appl 2019; 31: 577–92. <https://doi.org/10.1007/s00521-017-3089-2>.
- [14] Jiang XQ, Zhang LC. Stock price fluctuation prediction method based on time series analysis. Discrete Cont Dyn-S 2019; 12: 915–27. <https://doi.org/10.3934/dcdss.2019061>.
- [15] Yu P, Yan X. Stock price prediction based on deep neural networks. Neural Comput Appl 2020; 32: 1609–28. <https://doi.org/10.1007/s00521-019-04212-x>.
- [16] Cheng C, Sa-NGasoongsong A, Beyca O, Le T, Yang H, Kong Z, Bukkapatnam ST. Time series forecasting for nonlinear and non-stationary processes: A review and comparative study. IIE Trans 2015; 47: 1053–71. <https://doi.org/10.1080/0740817X.2014.999180>.
- [17] Koza JR. Genetic Programming: On the Programming of Computers by Means of Natural Selection. USA: MIT Press; 1992.
- [18] Kaboudan MA. Genetic programming prediction of stock prices. Comput Econ 2000; 16: 207–36. <https://doi.org/10.1023/A:1008768404046>.
- [19] Duan M, Povinelli RJ. Estimating Time Series Predictability Using Genetic Programming. Proc. Artificial Neural Networks in Engineering, 2001, p. 215–20.
- [20] Klůčik M, Juriova J, Klůčik M. Time Series Modeling with Genetic Programming Relative to ARIMA Models. Proc. Conference on New Techniques and Technologies for Statistics. NTTS2009, 2009, p. 17–27.
- [21] Lee YS, Tong LI. Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming. Knowl Based Syst 2011; 24: 66–72. <https://doi.org/10.1016/j.knsys.2010.07.006>.
- [22] Claveria O, Monte E, Torra S. Evolutionary computation for macroeconomic forecasting. Comput Econ 2019; 53: 833–49. <https://doi.org/10.1007/s10614-017-9767-4>.
- [23] Guresen E, Kayakutlu G, Daim TU. Using artificial neural network models in stock market index prediction. Expert Syst Appl 2011; 38: 10389–97. <https://doi.org/10.1016/j.eswa.2011.02.068>.
- [24] Engle RF. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica 1982: 987–1007. <https://doi.org/10.2307/1912773>.
- [25] Bollerslev T. Generalized autoregressive conditional heteroskedasticity. J Econom 1986; 31: 307–27. [https://doi.org/10.1016/0304-4076\(86\)90063-1](https://doi.org/10.1016/0304-4076(86)90063-1).