International Journal of Scientific Research and Management (IJSRM)
||Volume||10||Issue||09||Pages||EM-2022-3807-3815||2022||
Website: www.ijsrm.in ISSN (e): 2321-3418
DOI: 10.18535/ijsrm/v10i9.em01

Univariate Time Series Forecasting Using k-Nearest Neighbors Algorithm: A Case for GDP

[1]Georgios Rigopoulos

[1] Department of Economics, University of Athens, Athens, Greece

Abstract
k-Nearest Neighbors (k-NN) is a well-known algorithm, used for classification and regression. Its usage in time series forecasting is limited though, as demonstrated in relevant research, despite its simplicity and competitive accuracy. This work presents a method for time series forecasting, based on k Nearest Neighbors (k-NN) regression, which can be utilized for macroeconomic variable forecasting, like Gross Domestic Product (GDP). The approach focuses on one-step ahead forecast, and uses R package libraries for the implementation. The method is applied to forecast Greek Gross Domestic Product and the forecasting accuracy results are quite high and comparable to ARIMA approach. The work offers a competent approach for time series and GDP forecasting, which is comparable to traditional statistical approaches and can be further developed. Experimentation on diverse data sets can improve parameter tuning and aggregation approach, and can lead to improved accuracy.

Introduction

Forecasting is a valuable aid for quite diverse domains, either in public sector or industry. Research on forecasting methods, including development of new algorithms and methods, optimizing existing ones, is of real interest for both researchers in the field and beneficiaries [1]. In economics, the ability to forecast macroeconomic variables accurately, using time series data, plays a significant role for policy makers, either central banks, or states globally. Traditional time series analysis and forecasting methods, based on statistical approaches, have been developed at a large extent during the previous decades, because of computing capability developments [2]. In time series analysis, linear statistical models, like ARIMA, were dominating the forecasting domain for decades. However, due to some limitations in real life applications [4], nonlinear models were also developed, such as the bilinear model [5], or the autoregressive conditional heteroscedastic model (ARCH) [6]. Research in machine learning, during the past two decades, has however led to novel forecasting models, which are competitive to the traditional. New methods have been introduced, following a bottom-up approach, based on the underlying data rather than an explicit theoretical model [3]. Even if there is some controversy on utilization of machine learning for time series forecasting, machine learning has become a prominent and viable alternative for the creation of forecasting models, especially in time series. It competes traditional methods and, in some cases, outperforms [7]. Machine learning models are non-parametric and nonlinear, and they are characterized as data driven or black box models, as they use historical data to identify the stochastic dependency between past and future time series values. They are based on computational intelligence, and they are not so mature compared to statistical ones. However, research is very active and empirical findings support that utilization of algorithms like k Nearest Neighbors (k-NN) is very competitive [8]. k-NN is a popular algorithm for classification and regression and is based on the measurement of a point's similarity or distance to a training set, which contains target values or labels. Even if it is based on a conceptually simple idea, and is considered as not so novel any more, it is still used as benchmark for more complex algorithms [9].

Despite the fact that machine learning can possibly compete traditional statistical approaches, the application of simple algorithms, like k-NN, is very limited in time series forecasting and software packages do not embed such functionality. Given the limited works on k-NN use for time series forecasting, this work aims to introduce a method for time series forecasting with usage of k-NN. The novelty of our work lies in

the fact that it proposes an approach for GDP forecasting, it is not previously met, it is comparable to ARIMA, and it can be further developed, in terms of complexity, and compared to other statistical or machine learning methods.

In the paper, we present the method and how it can be applied to Gross Domestic Product (GDP) forecasting. The structure of the paper is as follows. Initially, some background on time series machine learning forecasting approaches and strategies is presented, and next the k-NN forecasting method is introduced. Then, an illustrative application on GDP forecasting follows, along with explanation and comparison to ARIMA approach. Finally, key findings and future directions are discussed in the conclusion.

Literature review on time series forecasting and machine learning

Time series analysis is a wide and rapidly developed research domain, which focuses on describing and summarizing features, identifying patterns and trends, forecasting future values and examining interactions between time series. One of the key aims in time series analysis is to identify future trends or forecast values in the future, that is to infer the stochastic dependency between values in the past and the future. Application domains are wide, including finance, economics, production and quality assurance among others. In economics, the key problems addressed in time series analysis is identifying causal effects and forecasting [10].

Basic types of traditional forecasting is based either in ordinary regression models, where time indices are used as the independent variables, or on autoregressive moving average models, where the independent variables are the past values and the past prediction errors. The Box and Jenkins method for ARIMA and exponential smoothing are some representative traditional statistical methods for time series forecasting [4]. In the past twenty years, there has been a rapid development of methods based on machine learning, which compete the traditional linear or nonlinear statistical based methods [7]. Even early studies support that methods like Artificial Neural Networks can outperform the traditional methods, like linear regression and ARIMA [11], [12]. But, also recent work on new methods, based on support vector machines, decision trees and nearest neighbors, prove to be quite competitive to traditional methods [13]. Some recent empirical evidence show that nonlinear machine learning models, combined with large data sets, can be extremely useful for economic forecasting [14], [15]. On top of relevant research in theoretical models, many forecasting competitions have been organized, providing thus the ability to use algorithms in data rich environments and compete for improving accuracy [16]. Empirical findings from the competitions have also led to some interesting scientific debates on the accuracy of machine learning methods for forecasting [17].

One important aspect of time series forecasting, which affects modelling, is the time horizon of the forecast. It can be either one-step ahead, or a multi-step, with multi-step being more challenging due to accumulation of prediction errors, reduced accuracy and increased uncertainty [18]. Another aspect that determines modelling is the theoretical interpretation of the forecasting problem. In statistical theory a time series sequence is considered as a random process realization, where a large number of independent degrees of freedom interact in a linear way, and this is the cause of randomness [19]. Another interpretation of the problem is based on dynamic systems theory, which considers that deterministic systems can generate a random process from a small number of degrees of freedom that interact in a nonlinear way [20]. This can lead to a deterministic chaotic behavior. In this case a time series can be interpreted as the observation of a dynamic system represented by a state function which evolves over time in a state space. However, the original state is not possible to be recovered, as the dynamics functions are not known, but the work is towards creating a state space that can be equivalent to the original. So, the reconstruction problem is based on the time series observations, and has been developed under the dynamical systems theory [21]. The reconstructed states can be used then to estimate the functional form of the series and can lead to a statistical nonlinear autoregressive formulation that can be further used for analysis and forecasting [3], [22].

Following the above, in a machine learning approach, the one-step ahead forecasting problem can be modelled as a supervised learning problem, where the model consists of a set of past observations, considered as input, and one or more output, which consists the forecast. In one-step forecasting past values are available and the problem can be considered as a general regression problem, where the forecasted value

is the value of the approximator function for the next time period. In the general machine learning forecasting setting, the training set is derived by the time series and is split into instances and the corresponding targets values [3]. Among the various machine learning methods, the local learning techniques [3], are considered as more capable to deal with learning in real data environment with increasing training samples, and require less assumptions on the process underlying the data. Nearest neighbor and lazy learning are the most representative methods of this category [13], [23].

The nearest neighbor approach is a well-known technique for local approximation and is based on the concept that the evolution of the current state will be alike to the nearest neighbor evolution. So, the method examines the nearest neighbors in the dataset, given a certain dimension that defines the pattern in past to search. The approach has been introduced by Lorenz in weather maps [24], and various extensions have been presented, including more neighbors [25] and higher order approximations [26]. Lazy learning on the other hand, follows a cross validation approach to optimally determine the number of neighbors and reducing the nonlinear problem to a sequence of local linear problems, one for each query, or forecast value [27].

For multi-step forecasting, there exist some strategies for machine learning methods adoption. The recursive strategy, which trains a one-step model and then it uses it recursively to return a multi-step forecast. The strategy has been used in real life problems with success, despite its limitations in error calculation sensitivity [28]. Another, strategy is the direct, which learns a number of models independently, and returns a multi-step forecast by concatenating the separate forecasts from the models. It has been applied in neural network models, with computational complexity a major drawback [28]. A combination of the recursive and direct strategies is the dirrec strategy, which computes the forecasts like the direct strategy for every time horizon using different models, and like the recursive strategy, it extends the input set, adding variables from the previous steps forecasts [29].

Proposed k-NN time series forecasting method

In this work, we focus on the utilization of k-NN for regression in univariate time series, so we will elaborate on the regression features of k-NN. In the basic regression modelling scenario, given a query value x, the associated, or predicted, value $\hat{y}$ is computed as the average of the associated target values of the k nearest neighbors:

$$\hat{y} = \frac{1}{k}\sum_{i=1}^{k} y_{o(i)} \qquad (1)$$

with $y_{o(i)} \in \mathbb{R}$ the associated target value of the i-th nearest neighbor.

Some accuracy improvement can be achieved by introducing weights for the neighbor values, where the closest ones receive increased importance. For example, a weight function which assigns weights inversely proportional to the distance, and promotes closest neighbors, can be defined as:

$$w_{o(i)} = \frac{1}{d(x, x_{o(i)})} \qquad (2)$$

where $i \in \{1,2,\dots,k\}$ and $d(x, x_{o(i)})$ is the distance between the value x and its neighbors.

So, in this case, the associated value to x is given by the following:

$$\hat{y} = \frac{\sum_{i=1}^{k} w_{o(i)} y_{o(i)}}{\sum_{i=1}^{k} w_{o(i)}} \qquad (3)$$

In the general case for regression modelling, k-NN, uses as input a dataset which comprises a set of training instances or data objects. Each training instance represents a point in a n-dimensional space, and comprises a

feature vector and a target vector. Each dimension represents a feature, so a training instance (i) can be represented as a vector of n features: $(f_1^i, f_2^i, \dots, f_n^i)$, which describes the object and an associated target vector $(t^1, t^2, \dots, t^m)$. Given the training dataset, for a new instance $(x_1, x_2, \dots, x_n)$, with known feature values and unknown target values, the target values of the new instance are computed from the k most similar or nearest instances from the training dataset. The k nearest neighbors are identified by using a similarity or distance metric to compare the feature vectors of the training data and the new instance. A widely used metric is the Euclidean distance. So, in this case the distance between the i-th training instance $(f_1^i, f_2^i, \dots, f_n^i)$ and the new instance $(x_1, x_2, \dots, x_n)$ can be computed as: $\sqrt{\sum_{r=1}^{n}(f_r^i - x_r)^2}$. This allows for the identification of the k nearest neighbors to the new instance. Considering that a new instance should follow its neighbors, we follow an approach which aggregates the k nearest neighbor target values in order to generate a prediction for the target value of the new instance. So, if we consider that the k nearest neighbors have the target vectors $t^1, t^2, \dots, t^k$, then we can take the average of them to compute the predicted target of the new instance as: $\hat{t} = \frac{1}{k}\sum_{i=1}^{k} t^k$. Table 1 presents a view of the dataset organization for k-NN input.

Table 1: Dataset organization for k-NN regression

	Feature 1	Feature 2		Feature n	Target
Instance 1	f_1^1	f_2^1		f_n^1	t^1
Instance 2	f_1^2	f_2^2		f_n^2	t^2
Instance m	f_1^m	f_2^m		f_n^m	t^m
New instance	x_1	x_2		x_n	$?\,\hat{t}$

In order to use k-NN regression method for univariate time series forecasting, we need to organize the time series data in a way that is appropriate input and sensible dataset for the algorithm, next we need to define the way to identify the k nearest neighbors, and finally select the appropriate distance metric to compute the distance between training and query instances. As soon as we focus on univariate time series, the forecast or explained variable is the same with the explanatory or predictor variables. So, the explanatory variables are defined as lagged values of the explained variable. Under this scenario, in the general case where we consider a multi-step ahead forecast, the target is a set of values from the time series, which is associated with a training instance, that comprises lagged values of the target. This approach indicates an autoregressive model. In case we focus on one step ahead forecast, then the target is a single value.

So, for the one step ahead case, given a time series $x = \{x_1, x_2, \dots, x_N\}$ with length N, the target of a training example is a value of the time series, and its features are lagged values of the target. The number of features, is defined as the number of the lagged values of the series and is a hyperparameter of the algorithm, that we need to define. The idea behind this approach is that, given a number of lagged values equal to p, for the target value x_t, with $t > p$, the associated feature values are equal to x_{t-p+i}, $i = 0,1, \dots, p-1$. So, each training instance is a lagged set of values of the target from the time series, of length p. Table 2, demonstrates the organization of the dataset for the general scenario.

Table 2: Dataset organization for k-NN in univariate time series regression

	Feature 1	Feature 2	…	Feature p	Target
Instance 1	x^{t-p}	x^{t-p+1}	…	x^{t-p+i}	x^t
Instance 2	x^{t-p+1}	x^{t-p+2}	…	$x^{t-p+i+1}$	t^{t+1}
…	…	…	…	…	…

Instance m	t^{t-p+m}	$t^{t-p+m+1}$	…	$t^{t-p+i+m}$	t^{t+m}
New instance m+1	$x^{t-p+m+1}$		…	$x^{t-p+i+m+1}$	$?\,\hat{t}^{t+m+1}$

In this approach, a new instance has the form shown in Table 2 (new instance m+1), where the feature values are known and the target value is unknown. This value can be forecasted using the k-NN regression model, given that we select appropriate distance metric and define the k nearest neighbors. The distance metric is used to calculate the distance between any two instances, with Euclidean distance being the most popular metric. The k nearest neighbors of the instance in query are the instances with the closest distance, based on the metric defined. The query instance has the feature values $x_{N+1-p+i},\ i = 0,1,\dots,p-1$. Following the above, we compute the target value of the x_{N+1} instance from the k neighbors, using an aggregation approach for the target values of the k neighbors. For the one step ahead forecast, a common approach is to take the average of the k neighbors target values, which represents the forecast value for the one step ahead case. Alternatively, the median of the k nearest target values can be used instead.

So, the proposed process for the one step ahead forecasting using k-NN comprises the following steps:

1. Parameter definition
    a. Define the parameters for the algorithm:
    k: number of neighbors
    p: the lags that will be used to as features for the model, which are the autoregressive explanatory variables.
    b. Define distance metric:
    The distance metric, that will be used to compute the distance between any two instances, is defined. It can be Euclidean or other.
    c. Define the aggregation operation:
    For the calculation of the target values for the query instance, an aggregation operation needs to be defined. It can be the average or other operator, which aggregates the target values of the k nearest neighbors.
    d. Define the query instance:
    This is the instance that we want to forecast.
2. Training phase
    a. Train the k-NN algorithm:
    Read the training time series and store it in a training set.
    b. Read the parameters k, p and store them.
    c. Organize values in appropriate input:
    The data need to be formatted in target and feature values given the lag number defined in the parameter p.
3. Forecast phase
    a. Find the k nearest neighbors:
    Use the distance metric to establish which are the k nearest neighbors to the query instance. The instance has the feature values $x_{N+1-p+i},\ i = 0,1,\dots,p-1$.
    b. Estimate the forecast:
    Use the aggregation operation, defined earlier, to estimate the target value for the query instance (x_{N+1} instance).

Application of k-NN regression method to GDP forecasting

Following relevant work with gross domestic product prediction using ARIMA model [30], we build a forecasting model based on the k-NN algorithm as proposed in the previous section. The motivation for this approach, is the limited works on GDP forecasting using k-NN, and on the other hand its simplicity and high accuracy level. We follow the steps for the one step ahead forecasting process, using k-NN as follows:

1. Parameter definition

a. Definition of the parameters for the algorithm: The number of neighbors is set to k=3. The number of lags that will be used to as features for the model, is set to p=15. The parameter values, are set after some preliminary experiments, and even they seem non justified, they are connected to the characteristics of the time series, like stationarity. There is not a universal approach to define them, but it is an iterative process to achieve increased accuracy levels.
b. Definition of the distance metric: The metric distance used in this case is the most commonly used distance, the Euclidean distance. Alternative metrics can be used.
c. Definition of the aggregation operation: The target value of the query instance will be the average of the neighbor target values. Other aggregation approaches can be used as well.
d. Define the query instance: This is the instance that we want to forecast. The purpose of this study is to achieve, with k-NN method, the apply a one-step forecast, and forecast the values of GDP time series on a time horizon $h = 1$ year. So, the query instance, is set to the next year after the end of the time series. In order to examine the accuracy, we repeat the process for three consecutive years and compare the forecasts with the actual values.

2. Training phase
   a. Train the k-NN algorithm: In the training phase, we need to define the dataset and ensure it is formatted appropriately as a univariate time series. To build the time series for the Greek GDP, we used publicly available data from the WorldBank database in constant USD2015 values for the period 1971-2020.The line plot below (Fig. 1) depicts the evolution of GDP through this period (in USD millions). It is obvious that there is an increasing trend until 2005, and a decreasing trend afterwards, showing a non-stationary process.

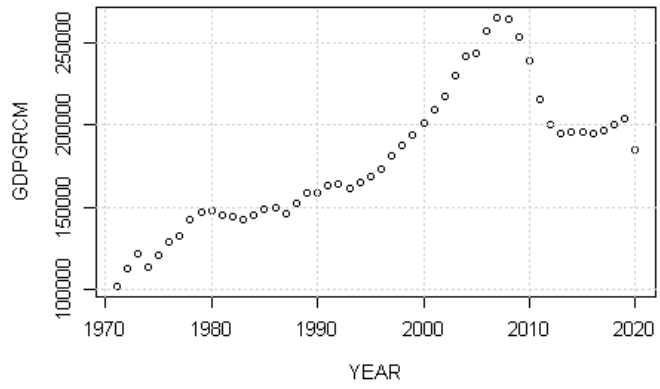

Figure 1: Greek GDP in constant millions USD2015 (1971-2020)

   b. Organize values in appropriate input: For the dataset processing, we used R package and especially the library tsfknn, which provides some functionality for k-NN modelling [31]. So, the dataset was loaded and stored the training set, as a set of instances with features and corresponding values, and a set of target values. The values of parameters k, p were also stored in the model.

3. Forecast phase
   a. Find the k nearest neighbors: We use the Euclidean distance metric to establish the k nearest neighbors to the query instance.
   b. Estimate the forecast: Using the R package and specifically tsfknn library, we applied the model, in one-step forecasts for the years, 2019, 2020, 2021. Results are summarized in Table 3, and are presented in the pictures below (Fig. 2, 3, 4), where the time series and the one-step ahead forecast is depicted.

Table 3. Forecasted values

Year	Forecast	Actual
2019	193857.7	203669.1874
2020	195775.3	185300.0005
2021	190062.2	200082.00

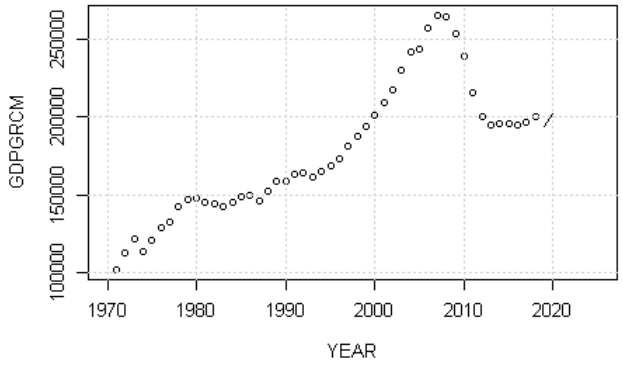

Figure 2: One-step forecast for year 2019

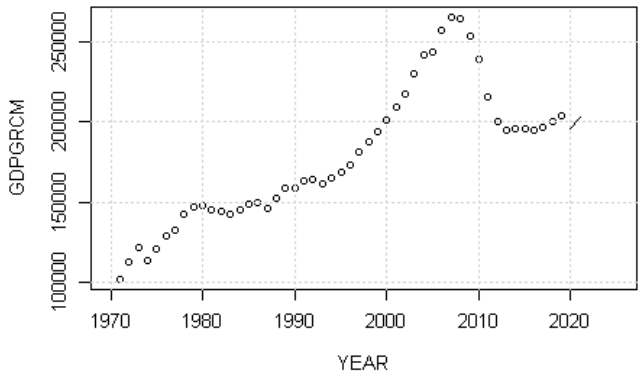

Figure 3: One-step forecast for year 2020

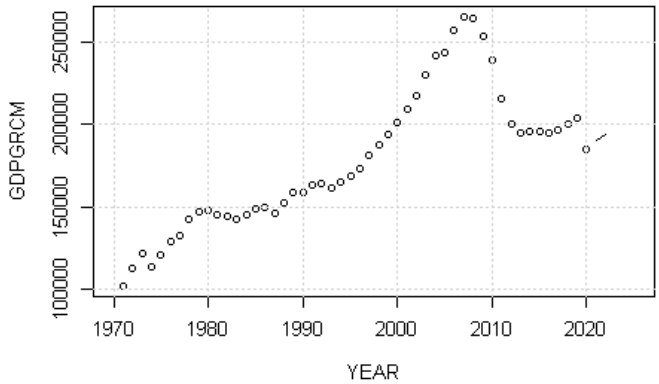

Figure 4: One-step forecast for year 2021

Results and discussion

In order to check the accuracy of the present model, we use a rolling forecast approach, which splits the dataset in training and test sets. The approach is based on the idea that a number of the last observations from the series are used as test set and the rest as training. The rolling approach is iterating the process for a number of repetitions equal to the number of the observations defined previously reducing the test set by one each time and until it reaches the end of the series. The global accuracy of the model is provided in Table 4 below, while the forecast accuracy for different time horizon predictions is depicted in Table 5. From the results, we can see that the accuracy falls substantially after the first step forecast.

Table 4. Global model accuracy

RMSE	MAE	MAPE
23316.767881	17541.819903	9.226776

Table 5. Time horizon (H in years) model accuracy

	H=1	H=2	H=3	H=4
RMSE	8000.319461	13983.528452	29927.12589	52941.60242
MAE	7361.829637	11806.560329	28804.79854	52941.60242
MAPE	3.767711	6.181035	15.04153	28.57075

In general, we can see that MAPE results for the one step ahead forecast are quite comparable to the results from ARIMA(1,2,1) model for the same dataset, as presented in previous work [30]. Also, forecasted values in both approaches are close to the actual ones. So, even if the k-NN model presented can be improved, it shows some decent level of accuracy for one-step ahead forecasts.

As a conclusion, the model presented is an initial approach to model GDP using k-NN method, and can be improved in future in parameter tuning and distance metric choice. Also, further assessment with larger and more diverse datasets is needed, as well as comparison with established models, like ARIMA and GARCH. However, the findings from this work demonstrate the feasibility of the approach and its comparable accuracy to traditional statistical approaches.

Ethics approval and consent to participate

Not applicable.

Conflicts of Interest

The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References

1. Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
2. Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.
3. Bontempi, G., Ben Taieb, S., & Borgne, Y. A. L. (2012, July). Machine learning strategies for time series forecasting. In European business intelligence summer school (pp. 62-77). Springer, Berlin, Heidelberg.
4. De Gooijer, J.G., Hyndman, R.J.: 25 years of time series forecasting. International Journal of Forecasting 22(3), 443–473 (2006)
5. Poskitt, D.S., Tremayne, A.R.: The selection and use of linear and bilinear time series models. International Journal of Forecasting 2(1), 101–114 (1986)
6. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. Econometrica 50(4), 987–1007 (1982)

7. Ahmed, N.K., Atiya, A.F., El Gayar, N., El-Shishiny, H.: An empirical comparisonof machine learning models for time series forecasting. Econometric Reviews 29(5- 6) (2010)
8. Crone SF,Hibon M, NikolopoulosK(2011) Advances in forecastingwith neural networks? Empirical evidence from the NN3 competition on time series prediction. Int J Forecast 27(3):635–660
9. Martínez, F., Frías, M. P., Pérez, M. D., & Rivera, A. J. (2019). A methodology for applying k-nearest neighbor to time series forecasting. Artificial Intelligence Review, 52(3).
10. Palit, A.K., Popovic, D.: Computational Intelligence in Time Series Forecasting: Theory and Engineering Applications. Advances in Industrial Control. Springer- Verlag New York, Inc., Secaucus (2005)
11. Werbos, P.J.: Generalization of backpropagation with application to a recurrent gas market model. Neural Networks 1(4), 339–356 (1988)
12. Lapedes, A., Farber, R.: Nonlinear signal processing using neural networks: prediction and system modelling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, NM (1987)
13. Hastie, T., Tibshirani, R., Friedman, J.: *The elements of statistical learning: data mining, inference and prediction*, 2nd edn. Springer (2009)
14. Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2021). Machine learning advances for time series forecasting. Journal of Economic Surveys. 2021;1–36.
15. Pavlyshenko, B. M. (2019). Machine-learning models for sales time series forecasting. Data, 4(1), 15.
16. Lendasse, A. (ed.): ESTSP 2008: Proceedings. Multiprint Oy/Otamedia (2008) ISBN: 978-951-22-9544-9
17. Crone, S.F.: Mining the past to determine the future: Comments. International Journal of Forecasting 5(3), 456–460 (2009); Special Section: Time Series Monitoring
18. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A.: Methodology for long-term prediction of time series. Neurocomputing 70(16-18), 2861–2869 (2007)
19. Anderson, T.W.: *The statistical analysis of time series*. J. Wiley and Sons (1971)
20. Farmer, J.D., Sidorowich, J.J.: Predicting chaotic time series. Physical Review Letters 8(59), 845–848 (1987)
21. Takens, F.: Detecting strange attractors in fluid turbulence. In: Dynamical Systems and Turbulence. Springer, Berlin (1981)
22. Casdagli, M., Eubank, S., Farmer, J.D., Gibson, J.: State space reconstruction in the presence of noise. PHYD 51, 52–98 (1991)
23. Atkeson, C.G., Moore, A.W., Schaal, S.: Locally weighted learning. AIR 11(1-5), 11–73 (1997)
24. Lorenz, E.N.: Atmospheric predictability as revealed by naturally occurring analogues. Journal of the Atmospheric Sciences 26, 636–646 (1969)
25. Ikeguchi, T., Aihara, K.: Prediction of chaotic time series with noise. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences E78-A(10) (1995)
26. Priestley, M.B.: *Non-linear and Non-stationary time series analysis*. Academic Press (1988)
27. Birattari, M., Bontempi, G., Bersini, H.: Lazy learning meets the recursive least-squares algorithm. In: Kearns, M.S., Solla, S.A., Cohn, D.A. (eds.) NIPS 11, pp.375–381. MIT Press, Cambridge (1999)
28. Weigend, A.S., Gershenfeld, N.A.: Time Series Prediction: forecasting the future and understanding the past. Addison Wesley, Harlow (1994)
29. Sorjamaa, A., Lendasse, A.: Time series prediction using dirrec strategy. In: Verleysen, M. (ed.) European Symposium on Artificial Neural Networks, ESANN 2006, Bruges, Belgium, April 26-28, pp. 143–148 (2006)
30. Rigopoulos, G., GDP Modeling Using Autoregressive Integrated Moving Average (ARIMA): A Case for Greek GDP, International Journal of Business Marketing and Management (IJBMM), Volume 7 Issue 4, 2022, P.P. 66-75 ISSN: 2456-4559
31. R Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2017, URL https://www.R-project.org