Machine Learning Approach For Small Samples ARMA Models Identification

Prof. Dr. Gamal Alshawadfi Dr.Mohamed Agamy
Dr. Nader Metwally

Statistics Department , Faculty of Commerce

Al-Azhar University

Machine Learning Approach For Small Samples ARMA Models Identification

Prof. Dr. Gamal Alshawadfi Dr.Mohamed Agamy
Dr. Nader Metwally

Statistics Department , Faculty of Commerce

Al-Azhar University

Abstract

(2-1) This paper proposes an effective machine learning approach to identify small samples data generated from autoregressive moving-average ARMA(p,q) models. The theoretical and practical aspects of the proposed approach are introduced , and its validity was evaluated by the ratio of correct identification(CIR) .

For evaluating the validity of the proposed machine learning approach, a simulation study was achieved. 192000 small samples were generated from ARMA(p,q) models with different sample sizes(10,20,30) and different parameters sets through the stationarity and invertibility regions. The ratio of the correct identification is calculated and used for evaluating the proposed approach. The average of CIR for all samples was 99.3% which shows a good performance for the proposed approach. The results also showed that the automatic ARMA identification Is less sensitive to small samples additionally, The proposed approach is quicker , automatic and more accurate alternative. A Python program is written for doing automatic Identification using a machine learning attached in the appendix.

Keywords : Artificial Intelligence (AI) , Machine Learning(ML) , Box-Jenkins Identification · The Neural Network Architecture.

(1) Introduction

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. Machine learning is used in many fields such as planning and operation, inventory, scheduling, shortest route problems.

Tom() provided a formal definition of the algorithms studied in the machine learning field, he stated that: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

There are many classical methods for time series identification. The first group implements the autocorrelation (ACF) and partial autocorrelation functions (PACF) for subjective identification of ARMA models. The second group choose the ARMA model which optimizes a suitably chosen function of the data , which is called “penalty function methods ‘. This approach based on Akaike’s final prediction error (FPE) and the Akaike's information criterion (AIC) . The rest groups of methods for time series identification are : innovation regression , pattern identification and testing hypothesis methods . Additional information on these techniques, are found in Choi (1992) and Box et al (2016).

Why do we need automatic ARMA identification?

Although there is a variety of ARIMA model identification, it has many difficult with small samples, need a moderate or large sample to do good, it also need good

experience for identifying models. The data preparation and parameter tuning processes end up being really time consuming. Before implementing ARMA, we need to make the series stationary, and determine the values of p and q using ACF and PACF plots. For more details about Identification difficult , see Choi (1992) and Box et al. (2016).

The main purpose of this research is to develop an effective machine learning approach to identify small samples data generated from ARMA(p,q) models. The theoretical and practical aspects of the proposed approach will introduced , and its validity will evaluated.

For evaluating the validity of the proposed machine learning approach, a simulation study will be introduced. Small samples were generated from ARMA(p,q) models with different sample sizes(10,20,30) and different parameters sets through the stationarity and invertibility regions. The simulated samples will be used for training the machine and testing the performance of the proposed automatic identification approach . The ratio of the correct identification is calculated and used for evaluating the proposed approach.

Through this research , we will introduce a literature review in section 2 . Section 3 introduces the proposed machine learning approach for identifying ARMA(p,q) models . In section 4 a simulation study to prove the performance of the proposed approach will done. Summary and conclusions are shown in section 5. References, Appendix and an Arabic abstract are included.

(2) Literature Review

There are many methods for time series models identification. The most popular one is the Box-Jenkins

which implements the autocorrelation (ACF) and partial autocorrelation functions (PACF) for subjective identification of ARMA models, described in detail in Box et al. (2016).

However , there are various difficult in the previous methods of identification . For example , the sample ACF and PACF are not clean, which makes the model identification more difficult. In particular, mixed models can be particularly difficult to identify. Although experience is helpful, developing good models using these sample plots can involve much trial and error. For this reasons, in the last decades information-based criteria such as FPE (Final Prediction Error) and AIC (Aikake Information Criterion) and others have been preferred and used. But these techniques require a moderate or large sample sizes. Additional information on these techniques are found in Choi (1992).

Stadnytska et al . (2008) assessed MINIC, SCAN, and ESACF methods for ARMA model identification. On average, in at least 60% of the trials conducted, the procedures either correctly identified the simulated structures or selected nearly equivalent mathematical representations. SCAN was superior to the two other mixed-structure procedures.

Stadnytska et al. (2008) identified ARMA models using SCAN and ESACF . The results were 79% of the right SCAN detection and 80% for ESACF.

Although there are many papers in classical time series identification in the past three decades , there is a few papers in machine learning Identification of time series data. Recent introduction for machine learning are found in Alpaydin, (2020).

Some statisticians have adopted methods from machine learning, leading to a combined field that they call statistical learning. Arminger and Enache (1995) states the relation between statistical models and artificial neural networks. Michael (2014) discussed the relation between statistics and machine learning. Bzdok (2018) discussed the statistics concepts versus Machine Learning concepts . An introduction for statistical Learning was found in Gareth et al. (2013).

Tran and Reed (2004) proposed an Automatic ARIMA Time Series Modeling .

Bishop (2006) introduced discussed the role of machine learning in pattern recognition.

Lénárt, Balázs(2011) proposed a new model order algorithm based on the ANN for identifying the ARMA() models. The basic concept was to create a minimum eigenvalue (MEV) matrix obtained from the data covariance . Then the model order can be selected automatically without the need for prior knowledge of the model or any human involvement.

Tang and Adrian (2018) compared the performance of neural network architectures, trained on simulated time series, with likelihood -based methods, particularly the Akaike and Bayesian information standard. They considered that their neural networks can considerably outperform these likelihood -based approaches in terms of precision, order of magnitude and speed.

(3) The proposed machine learning approach

In this section we introduce an effective machine learning approach consists of four phases for identifying the ARMA(p,q) models. The identification includes the

choice of the unknown orders p and q assuming that the maximum value of these orders are P_0 and Q_0 respectively . As Box et al. (2016) mentioned (p. 11) , in most practical time series , the order of p and q does not increase 2 , so values for each one varied over the range 0 to 2 , We have 16 combinations of orders .The proposed approach involves four phases : modeling , training , testing and identifying. We will propose the four phases in the following subsections.

(3-1) Modeling

The ARMA(p,q) model is :

$$y_t = \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \dots\dots + \emptyset_p y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots\dots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad \dots$$

… (3- 1)

where : y_t is the observation at time t , ε_t is a random shock , $\emptyset_i 's \; and \; \theta_i 's$ are parameters.

p and q is the order of the model. If p=0 , we will have a pure MA process. If q = 0, we will have pure AR process. The aim is to introduce an automatic approach for identifying p and q.

Before applying the machine learning proposed methodology for identifying p and q , the data need to be normalized into the range [0,1]. The following equation is used to normalize the inputs to the range [0,1].

$$x_i = 0.8 \left(\frac{y_{max} - y_i}{y_{max} - y_{min}} \right) + 0.1 \quad \dots\dots (3 - 2)$$

Where:

x_i 's the resulted series after transformation. y_{max} and y_{min}

are the respective maximum and minimum values of all observations. The resulted vector $X = [x_1\ x_2\ x_3\ \dots\ \dots\ \dots\ \dots\ x_n]$ Is a transformation of the series Y.

Machine learning identification function can be written as a nonlinear regression model containing input of time series data Y or a transformation of it X , generated from ARMA(p,q) with unknown values p and q.

These models may be used to approximate an unknown stochastic relation

$$Z = v(X) + \varepsilon \qquad \dots\dots (3\text{-}3)$$

Where: $E(\varepsilon \mid X) = 0$, $v(X) = E(Z \mid X)$ is usually an unknown function and is to be approximated as loosely as possible by a function g(x,w).

The parameters space and the parameter array w both depend on choice of the approximation function g(x,w) . Since no priori assumptions are made regarding the functional form of v(x) , the neural model g(x,w) is a non-parametric estimator of the conditional density E(Z|X), as opposed to a parametric estimator where the functional form is assumed priori.

In general, the proposed approximation machine learning model for ARMA(p,q) identification may have the form :

$$z_i = g(x_i, w_i) + \varepsilon_i \qquad \dots\dots\dots (3\text{-}4)$$

Where: $z_i\ is\ a\ vector\ [p_1, p_2, \dots, p_k, q_1, q_2, \dots, q_r]$

of the matrix Z . The elements of the output vector z_i have the values 0 and 1 , i=1,2,…..,k . w_i is a set of free parameters , called weights , which is an element of

parameters space and is to be estimated from a training set (samples). ε_i represents error between the input and the output .

The function $g(x_i, w_i)$ is called " network architecture" . It is a composition of linear or nonlinear functions in the input series

$$X = [x_1 \; x_2 \; \dots \; \dots \; \dots \; \dots \; x_n]$$

As Box et al. (2016) mentioned (p. 11) , in most practical time series , the order of p and q does not increase 2. Hence, the output is a row vector from the following 16 by 4 matrix:

$$Z = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}'$$

… … (3.5)

Due to its capability , the neural network may be used to approximate continuous mapping functions. The following theorem , due to Kolmogorov , shows the existence of the mapping neural network.

Theorem(3 - 1)

Any continuous function $f(x_1, x_2, \dots \dots, x_n)$ of several variables defined on I^n , where $n \geq 2$ and I =[0,1] , can be represented in the form :

$$f(x) = \sum_{j=1}^{2n+1} b_j \sum_{i=1}^{n} w_{ij} \, x_i \qquad \dots \dots (3\text{-}6)$$

where b_j and w_{ij} are continuous functions of one variable and w_{ij} are monotonic functions that do not depend on f .

The theorem states that : " a feed forward neural network with

three layers of neurons (input , hidden and output units) can represent any continuous function exactly , see Fausett(1994) pp.328-329.

Usually ,the activation function is a linear combination of parameters W and inputs X . In practice , many types of transfer functions may be used , which include : normal distribution , logistic distribution or sigmoid function, hyperbolic tangent , indicator function and threshold function. In this research we will use the sigmoid function.

The sigmoid activation function are easy to differentiate and make computation of the gradient easy. The sigmoid function or a logistic distribution function has the form:

$$H_1 = \frac{1}{\left(\sum^{n} \quad + b \right)} \qquad \ldots\ldots (3.7)$$

Where: x is the input observations , or any transformation of it , e=2.718 is a constant , w called weights , and b is bias . The values of w and b should be initialized before the training phase.

The first derivative of H_1 is:

$$H_2 = H_1(1 - H_1) \qquad \ldots\ldots (3.8)$$

Where :

$$\frac{d}{dx} H_1 = \frac{e^{\left(\sum_{j=1}^{n} w_{ij}\ x_k + b_j \right)}}{\left[1 + e^{\left(\sum_{j=1}^{n} w_{ij}\ x_k + b_j \right)} \right]^2}$$

$$= \frac{1}{1 + e^{\left(\sum_{j=1}^{n} w_{ij}\ x_k + b_j \right)}} \left(1 - \frac{1}{1 + e^{\left(\sum_{j=1}^{n} w_{ij}\ x_k + b_j \right)}} \right)$$

$$= H_1(1 - H_1)$$

logistic regression function separates the classes by maximizing the gap between training examples from each class. The examples in the test data are when assigned a label based on which side of the gap they fall. The logistic regression algorithm assumes a nonlinear separability of classes, however in reality this assumption is realistic. Therefore, a regularization parameter **b** is introduced which weights the importance of mis identification and allows to fit a linear separating hyperplane with some of the examples being mis identified. The following function will give the machine learning estimate of p and q :

$$\hat{z}_i = X.[2.W_1 - 1].[2.W_2 - 1] \quad \ldots\ldots (3-9)$$

Where: $\hat{z}_i = [\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_k, \hat{q}_1, \hat{q}_2, \ldots, \hat{q}_r]$' represent the values of p and q (outputs) , i=1,2,…..,k, X is 1.n matrix of time series values. W_1 is n.k matrix of weights. W_2 is k.k matrix of weights

The number of input neurons in the hidden layers H_1 and H_2depends on the length of the time series and on the trend parameter. Since the length of the time series is n and the trend parameter is k so the number of inputs is: n.k and k.k respectively.

The number of neurons in the hidden layers is selected by experientially. The outputs of the hidden neurons are used as input to neurons in the output layer. Neurons in the output layer use the summation and activation functions to determine the outputs of the neural network. The final output layer contains k neurons represent the estimated values of p and q . It is a vector of k elements , every element is an the estimate of the original value 0 or 1 .

(3-2) Training

Training for learning the machine is the process of adjusting the weights W_1 and W_2 of an artificial neural network(ANN) connections in order to obtain the desired outputs z_i where we will use a supervised learning approach to estimate a vector of p and q according to equation (3.9) where $\hat{z}_i = X.[2.W_1 - 1].[2.W_2 - 1]$

After setting up the neural network model as shown in equations (3.9) , we generate 100 time series like the original dataset . We split the samples into a training set and a test set. The first 70% of samples are used for a supervised training, and the remaining 30% of samples are used as a hold-out test set (unsupervised network for test). Machine learning algorithms are often categorized as supervised or unsupervised. The unsupervised machine learning algorithms are used testing phase.

The training strategy is to calculate the errors for all training examples and use this sum to update the weights W_1 and W_2 .

The iterative process of the back-propagation training algorithm continues until the error function reaches a predetermined level (say 0.001) , or the number of iterations is satisfied. The error function is:

$$\text{MSE} = \frac{1}{N}\sum[\hat{z}_i - z_i]^2 \qquad \ldots\ldots (3\text{-}10)$$

Where MSE is the mean of sum of squares of the difference between the NN estimated vector $\hat{z}_i$ and the actual z_i .

After this, the best p and q for ARMA(p,q) model is being selected by the above-mentioned approach. The output is a vector $\hat{z}_i$ where $\hat{z}_i$ is ANN approximation to a column vector k.1 from the matrix(3 - 5) .

(3-3) Testing

Train/Test Split is a simple and reliable validation approach. A portion of the data was split before any model identified and it was used only once to validate the identified ARMA(p,q) model. The overall performance was then calculated as the ratio of the correct identification(CIR) from the following equation:

$$CIR = \frac{n_i}{N_i} \quad \dots\dots \textbf{(3-11)}$$

Where: n_i is the number of samples identified correctly in group i , N_i is the total number of samples in group i , i = 1,2,….,k .

(3-4) Identifying ARMA(p,q) Model

After obtaining a satisfied result in the testing set , say a correct identification with CIR = 95% or more , we will use the estimation of the trained hidden functions to identify the original series X where:

$$\hat{z}_i = X'_{1.n}.[2.W1_{n.k} - 1].[2.W2_{k.k} - 1] \quad \dots\dots (3-12)$$

Where $\hat{z}_i$, X , W_1 and W_2 are as defined in equation (3-9).

If the resulting phase of testing step is not satisfied , we will retrain the network with more generated time series samples. The results obtained through less than one minute in any computer , lab top or a mobile with Python program attached in the appendix.

In summary ,the proposed machine learning identification approach has four phases :

1- Modeling or designing the network architecture where we

use equation (3-2) to rescale the data between 0 , 1 interval , we select suitable identification function (3-9) , determine the number of epochs , error level , and select initial values for the weights $w_i$and biases b_i .

2- Training the network: spilt the data into two groups (say 70% for the training group and 30% for the testing group), and use the first group to train the network and estimate the parameters $w_i$and b_i of the hidden layers functions.

3- Checking the validity by using the second group of data with equation (3-11) . Equation (3-11) gives the ratio of the correct identification which will be between 0% and 100% . It should more than 50% . Otherwise repeat the training process with more samples to obtain satisfied identification process.

4- Identifying new time series: If step 3 is satisfied, use the proposed identification function in identifying new time series data into a suitable set of (3 - 5).

(4) A Simulation Study

To evaluate the performance of the proposed identification approach , 192000 samples are generated from different ARMA(p,q) models with small sample sizes 10,20 and 30 observations and different sets of parameters values through the invertibility and stationarity regions. Data was simulated by randomly drawing values from a Gaussian distribution with mean zero and standard deviation one.

The samples of data are generated from 16 ARMA(p,q) models with pi,qi=0,1 , i= 1,2 and used for the evaluation process . i. e. every model is used to generate 1000 samples of time series data which are then used to identify the series with the machine learning proposed approach. The selected

sets of parameters values through the region of invertibility and stationarity were shown in the following table.

Table(4-1): The selected values of parameters through the region of invertibility and stationarity used in the simulation study

Parameters Set	$\emptyset_1$	$\emptyset_2$	θ_1	θ_2
1	**0.3**	**0.5**	**0.5**	**0.3**
2	**-0.3**	**0.5**	**-0.5**	**0.3**
3	**0.7**	**-0.4**	**0.8**	**-0.5**
4	**-0.7**	**0.4**	**-0.8**	**0.5**

The four phases of the proposed machine learning identification mentioned in section(3) will applied here as following:

1- Designing the network architecture : Through this phase , we used equation (3-2) to rescale the data between (0 , 1) interval , select suitable identification function (3-9) , determine the number of epochs , the value of error level , and select initial values for the weights $w_i$and biases b_i .

2- Training the network: We spilt the generated data into two groups , a training group includes 130 000 samples and a testing group includes 62000 samples. We use the first group to train the network and estimate the parameters $w_i$and $b_i$of the hidden layers functions according to the identification function (3-9).

3- Checking the validity: To check the validity of the proposed identification approach, we use a popular statistic; the ratio of the correct identification(CIR) , where the CIR is as defined in equation (3-11). The identification is satisfied say the ratio CIR is more than 50% and close to 100%.

To achieve the simulation study , we wrote a Python program to generate the simulated samples, apply the proposed machine learning identification approach and calculate the correct identification ratio(CIR) . The results of the correct identification ratio(CIR) are shown in the following table .

Table(4-2): simulation study results
The moderate behavior of CIR through Identifying different ARMA(p,q) models

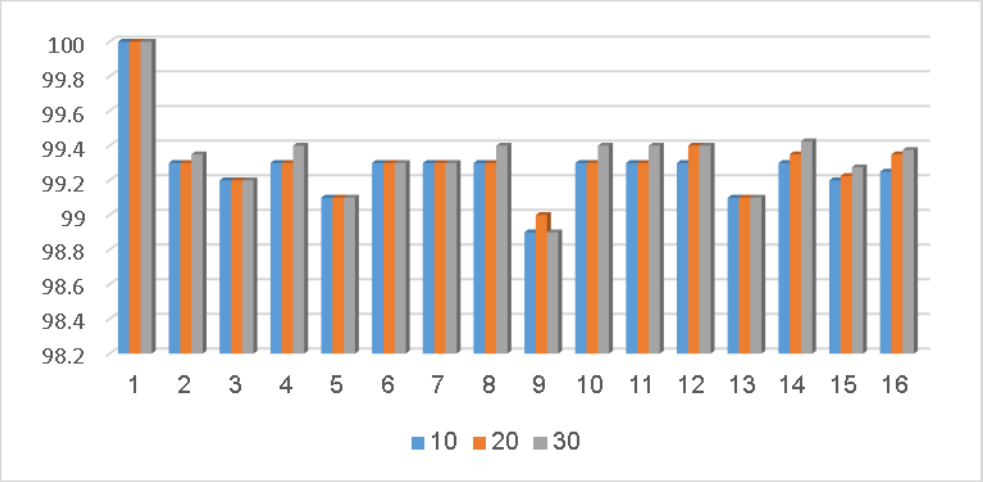
	p & q				n			Average
The true Model	**P1**	**P2**	**Q1**	**Q2**	**10**	**20**	**30**	
$y_t = \varepsilon_t$	0	0	0	0	100.0	100.0	100.0	100.0
AR(1)	1	0	0	0	99.275	99.325	99.35	99.317
$y_t = \emptyset_2\ y_{t-2} + \varepsilon_t$	0	1	0	0	99.175	99.225	99.2	99.2
AR(2)	1	1	0	0	99.275	99.325	99.375	99.325
MA(1)	0	0	1	0	99.075	99.1	99.1	99.092
ARMA(1,1)	1	0	1	0	99.275	99.275	99.3	99.283
$y_t = \emptyset_2\ y_{t-2} - \theta_1\ \varepsilon_{t-1} + \varepsilon_t$	0	1	1	0	99.25	99.275	99.325	99.283
ARMA(2,1)	1	1	1	0	99.325	99.325	99.4	99.35
$y_t = -\theta_2\ \varepsilon_{t-2} + \varepsilon_t$	0	0	0	1	98.925	99.0	98.975	98.967
$y_t = \emptyset_1 y_{t-1} - \theta_2\ \varepsilon_{t-2} + \varepsilon_t$	1	0	0	1	99.325	99.3	99.375	99.333
$y_t = \emptyset_2 y_{t-2} - \theta_2\ \varepsilon_{t-2} + \varepsilon_t$	0	1	0	1	99.225	99.275	99.35	99.283
$y_t = \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} - \theta_2\ \varepsilon_{t-2} + \varepsilon_t$	1	1	0	1	99.3	99.35	99.375	99.342
MA(2)	0	0	1	1	99.1	99.125	99.1	99.108
ARMA(1,2)	1	0	1	1	99.3	99.35	99.425	99.317
$y_t = \emptyset_2 y_{t-2} - \theta_1\ \varepsilon_{t-1} - \theta_2\ \varepsilon_{t-2} + \varepsilon_t$	0	1	1	1	99.2	99.225	99.275	99.2
ARMA(2,2)	1	1	1	1	99.25	99.35	99.375	99.325
	Average				99.267	99.302	99.331	99.29

The rows of table(4-2) shows the effect of the original model on the identification efficiency. We note that the correct identification ratio has a biggest value 99.425% at model: $y_t = \emptyset_1\ y_{t-1} + \emptyset_2\ y_{t-2} - \theta_2\ \varepsilon_{t-2} + \varepsilon_t$ and the lowest value 98.967% at the model: $y_t = -\theta_2\ \varepsilon_{t-2} + \varepsilon_t$

The columns of table(4-2) shows the effect of the sample size

on the identification efficiency. We note that the correct identification ratio increases with increasing the sample size from 98.925% at sample size 10 with model $y_t = -\theta_2\ \varepsilon_{t-2} + \varepsilon_t$ to 99.425% at sample size 30 for the model ARMA(1,2).

The last row represents the average ratio which increases from 99.267% at sample size 10 to 99.331% at sample size 30 . The total average for all samples is 99.29%. The following figure (4-1) shows the ratio for each model and sample size.



Figure(4-1)

the moderate CIR for Identifying different ARMA(p,q)

through a simulation study

According to figures (4,1) the machine learning identification approach tends to have the smallest deviations from the actual data(highest CIR) for all models . As suggested before, this may indicate that the machine learning identification approach is implicitly doing a better identification .

(5) Summary and Conclusion

This paper proposes an effective machine learning

approach to identify automatic ARMA models automatically . The proposed approach may be used in identifying a sample of time series data according to a proposed identification function . The identification function is obtained from a training process of the neural network . The selected identification function is the one which minimizes the difference between the actual and the expected numbers of parameters in ARMA(p,q) models. The validity of the proposed identification function is measured by the ratio of the correct identification (CIR).

The accuracy of the proposed ML approach increases with increasing sample size. The smallest CIR was 98.967% for model $y_t = -\theta_2\ \varepsilon_{t-2} + \varepsilon_t$. The biggest one was 99.35% for model ARMA(2,1) .In general , for all 16 models, CIR performance of the ML approach is 99.3%. This may suggest that the analysts may turned to ML technology as a quicker , automatic and more accurate alternative. ML are ideal for finding patterns in data. A Python program is written for doing automatic Identification using a machine learning attached in the appendix.

References

1) Alpaydin, Ethem (2020). Introduction to Machine Learning.: The MIT Press. ISBN 978-0-262-01243-0. London, U.K.

2) Arminger , G. and D. Enache (1995) Statistical Models and Artificial Neural Networks , Proceedings of the 19th annual conference of the Gesellschaft fur classification e.V., University of Basel , March 8-10 , 1995,H.-H. Bock .W.Polasek Editors , Springers , Germany.

3) Bishop, C. M. (2006), Pattern Recognition and Machine Learning, Springer, ISBN 978-0-387-31073-2

4) Box, G. E. P., G. M. Jenkins, and **G. C. Reinsel**(2016) Time Series

Analysis Forecasting and Control, 5th Edition, John Wiley & Sons, Hoboken, New Jersey, U.S.A.

5) Bzdok, Danilo; Altman, Naomi; Krzywinski, Martin (2018). "Statistics versus Machine Learning". Nature Methods. 15 (4): 233–234. doi:10.1038/nmeth.4642. PMC 6082636. PMID 30100822.

6) Choi, B (1992) ARMA Models Identification , Springer-Verlag, New York, U.S.A.

7) Fausett , Laurane (1994) Fundamentals of neural networks: architectures, algorithms, and applications.

8) **James**, G., D. Witten , T. Hastie and R. Tibshirani (2013) An Introduction to Statistical Learning with Applications in R ,Springer-Verlag New York.

9) Lénárt, B. (2011) "Automatic identification of ARIMA models with neural network", Periodica Polytechnica Transportation Engineering, 39(1), pp. 39-42.

10) Pankratz, A. (1983) Forecasting with Univariate Box-Jenkins Models. Wiley & Sons, Inc., New York.

11) Stadnytska, T., S. Braun, and J. Werner (2008) Model Identification of Integrated ARMA Processes, Multivariate Behavioral Research 43(1):1-28 , Taylor & Francis Group, LLC ISSN: 0027-3171 ,Germany.

12) Tran N, Reed D A(2004) Automatic ARIMA Time Series Modeling for Adaptive , I/O Prefetching, IEEE Transactions on parallel and distributed systems 15 (February 2004),no. 2.

APPENDIX

Python Program For Automatic Identification Of Time Series Data

```
# توصيف نماذج ARMA(p,q) باستخدام ماشين ليرننج
import numpy as np ; from random import random
N=1000; n=30 ; m= 0;nn=3
B1=[[ 0.3, 0.5 , 0.5 ,0.3 ], [ -0.3, 0.5 , -0.5 ,0.3 ],[ 0.7, -0.4 , 0.8 ,-0.5 ],
[-0.7, -0.4 , -0.8 ,-0.5 ]] ; y =[0, 0]; yy=[] ; P=[];R=np.zeros((16,3))
```

```
for B in B1:
  for q2 in (0,1):
   for q1 in (0,1):
    for p2 in (0,1):
     for p1  in (0,1):
      for j1 in range(N):
       pq= [p1 , p2, q1 , q2] ;   E= np.random.normal(size=n)
       E1= np.append(0,E[0:n-1]);   E2= np.append(0,E1[0:n-1])
       E3=E - q1*B[2]*E1  - q2*B[3]*E2
       for t in range (2,n):
        y.append(p1*B[0]*y[t-1] + p2*B[1]*y[t-2]+ E3[t])
       yy.append(y) ;   P.append(pq)
P=np.array(P)
def nonlin(x,deriv=False):
          if(deriv==True):
             return x*(1-x)
          return 1/(1+np.exp(-x))
np.random.seed(1342) ; syn1 = 2*np.random.random((4,4)) - 1
i=0;j=0
for w in yy:
  for n in (10,20,30):
        syn0 = 2*np.random.random((n,4))- 1
        z=w[0:n] ;   Q= P[i][:];  Q=np.array(Q);  l0 =np.array(z)
        l1 = nonlin(l0.dot(syn0)); l2 = nonlin(np.dot(l1,syn1))
        l2_error = Q - l2;   l2_delta = l2_error*nonlin(l2,deriv=False)
        l1_error = l2_delta.dot(syn1);  l1_delta = l1_error *
nonlin(l1,deriv=False);  syn1+= np.outer(l1,l2_delta)
        syn0+=np.outer(l0,l1_delta); l2=np.array(l2,dtype=float)
        r = (j%nn) ; k =( j//(nn*N))%16 ;s=(j//(nn*N*16))%3; u= j//(nn*N*16)
        if Q[0] -l2[0] < 0.5 and Q[1] -l2[1] < 0.5 and Q[2]- l2[2] < 0.5  and
```

```
Q[3]-l2[3] < 0.5:
          m=m+1
R[k][r]= R[k][r]  + 1
        else:
          m= m
        y=[0 , 0] ; j=j+1
  i=i+1
print('_____')
print(  ' Model   ' , 'ARMA(p,q)' )
print('_____')
print( '    ' ,' correct ratio')
print('n=10__n=20____n=30_____')
print((R  /(4*N))*100 )
print('_____')
rr=(R/(4*N))*100
print("Sum of rr(axis = 0) : ", np.sum(rr/16, axis = 0))
print("Sum of rr(axis = 1) : ", np.sum(rr/3, axis = 1))
print("\nSum of rr (keepdimension is True): \n", np.sum(rr/3, axis = 1,
keepdims = True)) ; print ((m/(192*N))*100)
```

استخدام اسلوب تعلم الالة في توصيف نماذج ARMA(p , q) في حالة العينات الصغيرة

أ.د/ جمال أحمد الشوادفي د / محمد عجمي د/ نادر متولي

كلية التجارة جامعة الازهر

ملخص البحث باللغة العربية

يعتبر توصيف نماذج ARMA(p , q) أحد أهم مراحل تحليل السلاسل الزمنية. والتوصيف يتضمن اختيار عدد مناسب لمعلمات النموذج . توجد طرق عديدة لتوصيف نماذج ARMA منها طرق تقوم على معاملات الارتباط الذاتي وتسمى Autocorrelation Methods ومن أهمها طريقة بوكس-جينكنز ، وطرق انحدار الأخطاء innovation regression methods ، وطرق دالة الجزاء penalty function methods ، وطرق نماذج التوصيف pattern identification methods ، وطرق اختبارات الفروض testing hypothesis methods . وبالرغم من وجود هذه الطرق العديدة إلا أنها تتضمن صعوبات في العمليات الحسابية وفي تفسير النتائج خاصة في حالة العينات الصغيرة .

هذا البحث يقدم طريقة جديدة ومؤثرة لتوصيف نماذج ARMA(p,q) باستخدام أسلوب تعلم الالة Machine learning . تتضمن الطريقة تصميم شبكة عصبية باستخدام دوال مناسبة ثم تدريبها واختبارها للتحقق من صلاحيتها ثم استخدامها في التعرف على النموذج المناسب للسلسلة الزمنية. ولتقييم أداء الطريقة المقترحة تم توليد 192000 سلسلة زمنية من نماذج ARMA(p,q) بمعلمات مختلفة واختبار مدى تعرف الشبكة على النموذج المناسب.

وقد أظهر البحث صلاحية الطريقة المقترحة في توصيف نماذج السلاسل الزمنية باستخدام أسلوب تعلم الالة Machine learning ، حيث تعرفت الالة بطريقة صحيحة على نماذج ARMA(p,q) بنسبة 99.3% ، وهى نسبة معقولة جدا في هذا المجال . بالاضافة إلى ذلك فإنه يمكن باستخدام برنامج صغير (مرفق في الملحق) توصيف السلسلة الزمنية دون الحاجة إلى خبرة كبيرة أو تفهم لأساليب تحليل رياضية وإحصائية معقدة.