PAPER • OPEN ACCESS

Self-Identification Deep Learning ARIMA

To cite this article: Paisit Khanarsa *et al* 2020 *J. Phys.: Conf. Ser.* **1564** 012004

View the article online for updates and enhancements.

Self-Identification Deep Learning ARIMA

Paisit Khanarsa[1], Arthorn Luangsodsai[2] and Krung Sinapiromsaran[3]

Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University, Bangkok, 10330, Thailand

E-mail: paisitkhanarsa@gmail.com[1], larthorn@chula.ac.th[2], Krung.S@chula.ac.th[3]

Abstract. The aspiration to predict the future values as close to the actual values as possible leads to the invention of time series models, the autoregressive integrated moving average (ARIMA) model which requires appropriate parameters of model identification, the ARIMA order, prior to fit coefficients of the models using the Box-Jenkins method. Statisticians for a decade identified the order via the sample autocorrelation function (ACF) and the sample partial autocorrelation function (PACF) which were very challenging for a human eye. To circumvent this issue, the recent model identification development uses a likelihood based-method that automatically generates orders and fits coefficients by varying the ARIMA order and pick the best one having the smallest Akaike information criterion (AIC) or Bayesian information criterion (BIC). The acquired ARIMA model may fail residual diagnostics. Consequently, this paper proposes the convolution neural network model, called the self-identification deep learning (SID) model, to automatically identify the ARIMA order via sample ACF/PACF. Accordingly, randomly simulated time series data with stationary and invertibility properties are generated by ARIMA model. Next, the time series data are converted into ACF/PACF graphs in order to feed into the SID model. The derived ARIMA order will be passed to determine the best fit coefficients of the ARIMA model via the Box-Jenkin methods for forecasting future values. The complete algorithm is called the self-identification deep learning ARIMA (SIDA) algorithm. The performance of identifying the ARIMA order from the SID model outperforms the likelihood based-method and ResNET50 which accepts the time series data directly in terms of precision, recall and F1-scores. Moreover, the SIDA algorithm applying to the real world dataset shows a better performance over the likelihood-based method via the mean absolute percentage error, the symmetric mean absolute percentage error, the mean absolute error and the root mean square error.

1. Introduction

The ambition to correctly forecast a future value of the time series and to comprehend the past drives the development of time series analysis models[1][2] which demonstrate the behavior of observed phenomena such as volatility of currency exchange rates[3][4][5][6] or price movement of stocks[7][8][9][10]. To achieve the better forecasting values, time series analysis is used to model the nature of the time series data. Over the past decade, new time series models are introduced from the autoregressive integrated moving average model (ARIMA) and the seasonal autoregressive integrated moving average model (SARIMA) by Box and Jenkins in 1970[11] to the self-exciting threshold autoregressive model (SETAR) by Tsay in 1989[12] to the autoregressive conditional heteroskedasticity model (ARCH) by Engle in 1983[13] and the generalized autoregressive conditional heteroskedasticity model (GARCH) by Bollerslev in 1990[14].

The autoregressive integrated moving average model is a basic stochastic model for forecasting time series using the Box-Jenkins model identification and estimation to build the best fitting ARIMA model for the time series data. It comprises of three components including the autoregressive (AR) component, the differencing component and the moving average (MA) component before applying the Box-Jenkins method for finding the best fit coefficients. Statisticians studying time series identify the ARIMA order via the sample autocorrelation function (ACF) and the sample partial autocorrelation function (PACF) by visualization. The first step in fitting an ARIMA model is the determination of the order of differencing. Commonly, the appropriate order of differencing is the lowest non-negative integer that fluctuates the time series around a constant mean and the sample ACF plot decays rapidly to zero, either from above or below. If its sample ACF is all positive and slowly decay then it needs the higher differencing order. For identifying the AR order, if the sample PACF plot displays a sharp drop and the sample ACF plot slightly decays, the lag at which the PACF drop indicates the order of AR terms. Likewise, for identifying the MA order, if the sample ACF plot displays a sharp drop and the sample PACF slightly decays, the lag at which the ACF drop indicates the order of MA terms.

To avert humans analysis, recent development uses a likelihood based-method to automatically generate the ARIMA order and the best ARIMA coefficients corresponding with the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) criteria by varying the ARIMA order for finding the best AIC by Akaike in 1987[15] or BIC by Akaike in 1977[16]. These criterias are the estimator of prediction error and quality of the model for a given data. However, its residuals may not comply with the white noise process. At present, both the Box-Jenkins method and the likelihood based-method have been used in many domains such as annual runoff series by Wang et al (2015)[17], monthly and weekly stock prices by Siami and Namin (2018)[18], wheat production in India by Bhola Nath et al. (2019)[19].

Various researchers attempted to develop a method for identifying the ARMA order and the ARIMA order. For example, Lee and Oh in 1996[20] used a neural network-driven decision tree classifier to evaluate the ARMA order and showed that most time series data in the real world have order within the ARMA(5, 5) model. Chenoweth et al. in 2000[21] used a time series of length 100 and 3,000 for evaluating order up to 2 and used the extended sample autocorrelation function (ESACF). After that, Al-Qawasmi et al. in 2010[22] used the time series of length 1,500 and applied the identification ARIMA order with a special covariance matrix of the MEV criterion as input.

In recent years, a deep learning model has received an increased amount of attention for solving several problems in the world including time series analysis. For example, Guo et al. in 2018[23] used a deep network framework of Deep Candlestick Predictor to predict the price movements by reading the candlestick charts. Brunel et al. in 2019[24] adapted the convolutional neural networks (CNN) to the time series data for the classification of Supernovae. Fawaz et al. in 2019[25] showed a method for an ensemble of 60 deep learning models which significantly improved upon the current state-of-the-art performance of CNNs for time series classification. In terms of predicting the future, Yang et al. in 2018[26] suggested a multi-indicator feature selection for the stock index prediction based on the CNN structure, named MI-CNN framework. Sandoval et al. in 2018[27] used CNN for building the innovative short-term forecasting method for financial time series using the limit order book data. Kim and Kim in 2019[28] proposed the long short-term memory-convolutional neural network model combining several features of the time series data, namely, stock time series and stock chart images, to predict stock prices of SPDR, S&P500, and ETF.

Aside from using deep learning in time series analysis and forecast, deep learning has also been applied to the identification of the ARMA order. Tang and Röllin in 2018[29] used CNN (ResNet50) to address the issue of the model identification for the ARMA model by comparing the performance of several neural network architectures with the likelihood-based method based on AIC. The results showed that CNN outperformed those likelihood based-methods in terms of accuracy and speed for the low-order ARMA model.

The issue of identifying the ARIMA order by the Box-Jenkins identification using sample ACF and sample PACF may give rise to different orders from different views. Although it can be avoided

using a likelihood-based method, the results of identifying the ARIMA order are still unsatisfactory due to unacceptable residual behavior. Therefore, this issue is resolved by applying ACF and PACF to the deep learning model called the self-identification deep learning (SID) model. Then the ARIMA order from SID model is used to fit coefficients of the time series analysis model via the Box-Jenkins method called the self-identification deep learning ARIMA (SIDA) algorithm. The forecast model is applied to the real world time series data to confirm the performance.

This paper consists of six parts. The first part is the introduction. The second part covers the concept of the convolution neural network. The third part reviews the stationary and invertibility of the time series data. The fourth part describes SIDA algorithm. The fifth part covers results from training the deep learning model and forecasting the real world time series data. The last section covers the conclusion and discussion.

2. Convolution Neural Network

A convolutional neural network[30] is one type of neural networks that successfully classify images in the ImageNet Large Scale Visual Recognition Challenge[31]. Currently, CNN is applied popularly to various fields after the success of AlexNet in 2012 by Krizhevsky[32]. Later, the modern architectures of convolutional neural networks are proposed including VGGNet from the University of Oxford. VGGNet had significantly improvement over ZFNet which is the winner in 2013 and AlexNet which is the winner in 2012. Therefore, this research will take advantage of CNN to learn the ARIMA order from ACF/PACF images of time series data.

3. Stationary and invertibility of the time series data

In this section, the important concept of time series analysis comprises of the autoregressive process, the moving average process, the autoregressive moving average process and the autoregressive integrated moving average process.

The AR process of time series analysis of p order can be expressed by

$$x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + \ldots + \phi_p x_{t-p} + \varepsilon_t$$

or it can be written as

$$\phi_p(B) x_t = \varepsilon_t$$

where $\phi_1, \phi_2, \ldots, \phi_p$ represent the real-valued coefficients of the AR process, $\phi_p(B) = \left(1 - \phi B - \ldots - \phi_p B^p\right)$ and $x_{t-1} = B x_t$. To obtain the AR process, the AR process must have the property of stationary which the roots of $\phi_p(B) = 0$ must be outside of the unit circle[1].

Next, the MA process of time series analysis of q order can be expressed by

$$x_t = \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}$$

or it can be written as

$$x_t = \theta_q(B) \varepsilon_t$$

where $\theta_1, \theta_2, \ldots, \theta_q$ represent the real-valued coefficients of the MA process and $\theta_q(B) = \left(1 - \theta_1 B - \ldots - \theta_q B^q\right)$.

To obtain the MA process, the MA process must have the property of invertibility which the roots of $\theta_q(B) = 0$ must be outside of the unit circle similar to the AR process[1]. The general form of the AR process and the MA process can be mixed as

$$\phi_p(B) x_t = \theta_q(B) \varepsilon_t$$

where

$$\phi_p(B) = \left(1 - \phi B - \ldots - \phi_p B^p\right)$$

and

$$\theta_q(B) = \left(1 - \theta_1 B - \ldots - \theta_q B^q\right) \quad .$$

To be stationary of the autoregressive moving average (ARMA) process, the root of $\phi_p(B) = 0$ must be outside of the unit circle. Likewise, for the invertibility of ARMA process, the roots of $\theta_q(B) = 0$ must be outside of the unit circle as well.

To confirm stationary and invertibility of the ARMA process, this process is extended to the ARIMA process by applying differencing to ARMA. The general ARIMA process can be expressed by

$$\phi_p(B) \, \nabla^d x_t = \theta_q(B) \, \varepsilon_t$$

where d is the number of differencing and $\nabla x_t = x_t = x_{t-1}$.

To identify the ARMA order, the concepts of ACF and PACF are applied[1]. The order of the AR term is identified by inspecting PACF. If PACF cuts off at lag k then the AR component should be fitted using order k. Likewise, the MA order is commonly associated with ACF. The lag at which ACF cuts off is the number of the MA order. For identifying the ARMA order, this research uses sample ACF and sample PACF which describe next.

Consider the observed time series $x_1, x_2, \ldots, x_n$, sample ACF $\hat{\rho}_k$ at lag k is defined as

$$\hat{\rho}_k = \frac{\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \qquad , k = 0, 1, 2, \ldots$$

where $\bar{x} = \frac{\sum_{t=1}^{n} x_t}{n}$.

Consider the observed time series $x_1, x_2, \ldots, x_n$ sample PACF $\hat{\phi}_{kk}$ at lag k is defined as

$$\widetilde{\phi}_{kk} = \frac{\hat{\rho}_{k+1} - \sum_{j=1}^{k}\widetilde{\phi}_{kj}\hat{\rho}_{k+1-j}}{1 - \sum_{j=1}^{k}\widetilde{\phi}_{kj}\hat{\rho}_j}$$

and

$$\widetilde{\phi}_{k+1,j} = \widetilde{\phi}_{kj} - \widetilde{\phi}_{k+1,k+1}\widetilde{\phi}_{k,k+1-j} \quad , j = 0, 1, 2, \ldots, k.$$

4. The self-identification deep learning ARIMA algorithm

This section will describe the method for synthesizing time series data and the deep learning model for building the self-identification deep learning (SID) model. The result of the ARIMA order will be used in the Box-Jenkin methods to find the best fit ARIMA model. This process is called the self-identification deep learning ARIMA (SIDA) algorithm.

4.1. Building the self-identification deep learning model

Before training the SID model, the time series data is simulated to identify orders p and q of the ARMA model. In this research, the simulated process for order p of the time series data randomly generates coefficients from the independent standard normal distribution and checks for stationary. If the property of stationarity is not satisfied, all coefficients would be randomly generated again. When the stationarity property is accepted, the process stops. Another simulated process of order q of the time series data is performed similarly but uses the test of invertibility instead of the test of stationary. Both training and testing data are generated using these processes by varying p and q from 0 to 5 where the number of the time series data in the training, validating and testing is collected 14400, 7200 and 720, respectively. In case of identifying d order, both training data and testing data are generated by varying d from 0 to 2 having 5,400 for training, 2,700 for validating and 540 for testing.

After calculating all lags of ACF and PACF, the ACF plot and the PACF plot are converted to black and white images as in Algorithm 1. Some examples of ACF and PACF images from our collection are shown in Table 1.

Algorithm 1: Pseudo code for generating images of ACF/PACF
Input: ACF/PACF, k = the maximum lags of ACF/PACF

Step 1: Define the image size as $k \times k$.
Step 2: Split ACF/PACF values to P if the ACF/PACF value is non-negative and to N otherwise.
Step 3: Generate the upper image (positive part)
      For lag i = 1 to k
           Plot point with the height of P[i]
Step 4: Generate the lower image (negative part) similar to step 3 using N instead of P
Step 5: Merge the upper image and lower image together
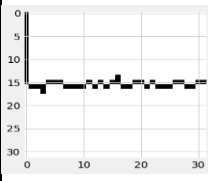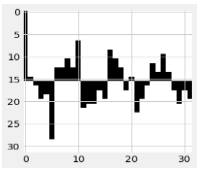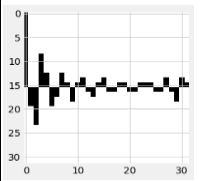
Table 1. Some examples of ACF images in training collections.

	ARMA(0,0)	ARMA(5,0)	ARMA(2,5)
ACF images			

Next, the architecture of the SID model is described. The input of the SID model comprises of 1, 2 or 3 channels of image sizes. The image is sent through convolutional layer of size 3×3. The convolutional stride is set to 1 pixel and the spatial padding of the convolutional layer is also set to 1 pixel for 3×3 convolutional layers. Then, max-pooling layers are applied. The window size of max-pooling is set to 2×2 with stride = 2. The last part of the architecture is followed by two fully connected layers which include 1024 nodes and 512 nodes respectively, and the activation function is a linear function. Finally, the final layer is the softmax layer. The SID model for training orders p, d, q of the ARIMA model is divided into three sub-models consisting of the model for training p order, the model for training q order and the model for training d order. The architecture of the SID model for each order are shown in Fig.1 and Fig.2.
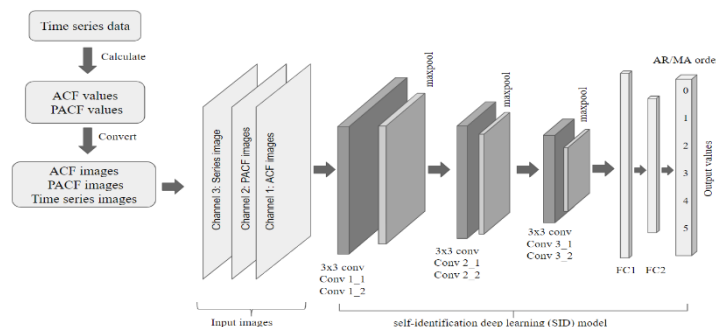

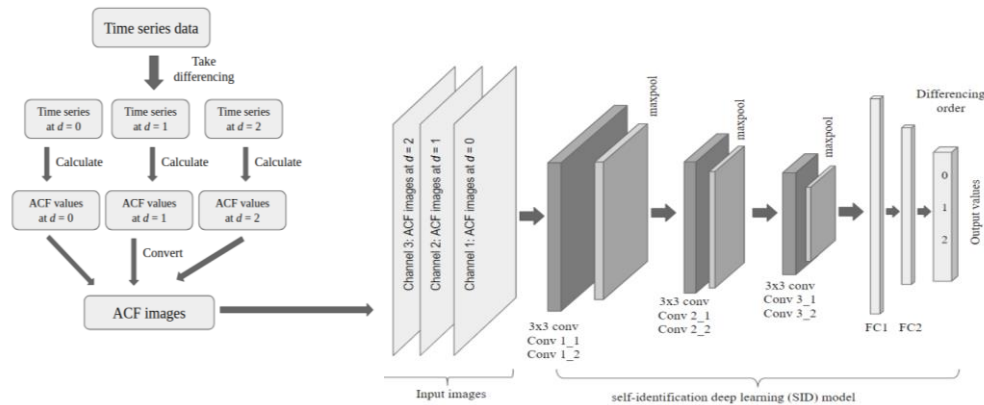**Fig.1.** The architecture of the SID model for training the AR/MA order.

Fig.2. The architecture of the SID mode for training differencing order.

4.2. The self-identification deep learning ARIMA (SIDA) algorithm

Finally, the output from the SID model is used to build the ARIMA model for forecasting future values. The entire step is called the self-identification deep learning ARIMA (SIDA) algorithm which is shown in Fig. 3.
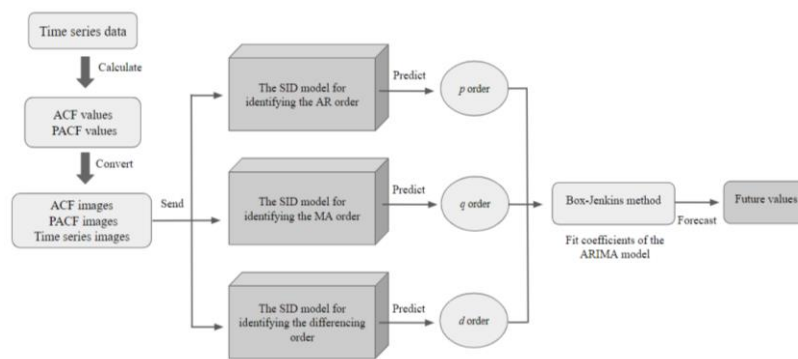

Fig.3. The algorithm of the self-identification deep learning ARIMA (SIDA).

5. Experimental results

The synthesized datasets are divided into three cases where the first case is called p underlying q which will fix the AR order p in $\{0, 1, 2, 3, 4, 5\}$ while varying the MA order from 0 to 5 repeating 300 times. The second case is called q underlying p which will fix the MA order q in $\{0, 1, 2, 3, 4, 5\}$ while varying the AR order from 0 to 5 repeating 300 times. The third case is called d underlying p and q which will fix the differencing order d in $\{0, 1, 2\}$ while varying both the AR order and the MA order from 0 to 5 repeating 100 times. Table 2 describes different models for case having different channels of inputs

5.1. Results of the SID model in the case of p underlying q

Fig 4. summarize the scores of all models corresponding to precision, recall and f1-score, respectively. From the results, model A1, A2 and A3 are the top-3 best scores for precision, recall and f1-score having scores close to 1. The performance of model A1, A2 and A3 is quite similar but model A1 is the best. Both model B and model C gives inferior performance.

Table 2. Description of each channel in the models.

Case: *p* underlying *q*		
Model	Model	channel(s)
Model A1	SID	1 channel: PACF images
Model A2	SID	2 channels: PACF and ACF images
Model A3	SID	3 channels: PACF, ACF and series images
Model B	ResNet50	Time series data
Model C	Likelihood method (AIC criteria)	Time series data

Case: *q* underlying *p*		
	Model	channel(s)
Model A1	SID	1 channel: ACF images
Model A2	SID	2 channels: ACF and PACF images
Model A3	SID	3 channels: PACF, ACF and series images
Model B	ResNet50	Time series data
Model C	Likelihood method (AIC criteria)	Time series data

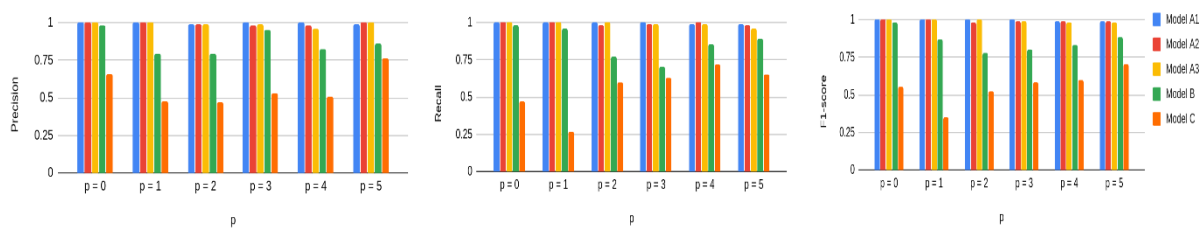| Case: *d* underlying *p* and *q*		
	Method	channels
Model A1	SID	3 channels: ACF images with *d* = 0, 1 and 2
Model C	Likelihood method (AIC criteria)	Time series data

Fig.4. Precision, Recall and F1-score of the models by considering *p* underlying *q*.

Table 3 displays the graphical analysis of deep learning models which are three SID models and the ResNet50 model. The experiments show that the accuracy values of all models are close to 1. Model A1, A2 and A3 are robust than model B. Moreover, model B demonstrates more fluctuate between training and validating datasets for large epochs.

Table 3. Graphical analysis of deep learning models (*p* underlying *q*).
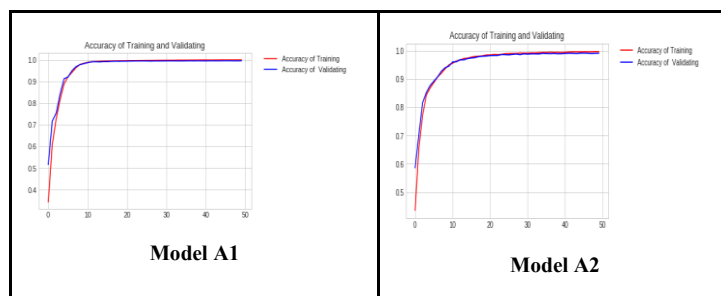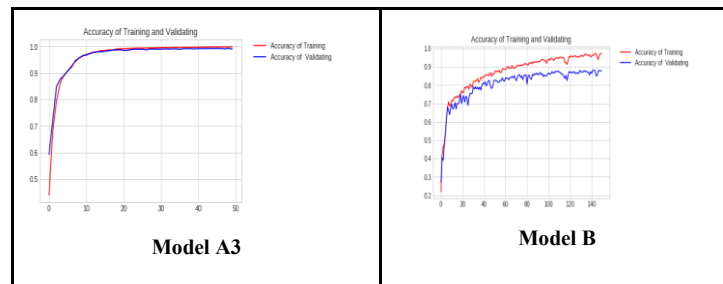
5.2. Results of the SID model in the case of q underlying p

From Fig 5, model A2 and model A3 gives the best precision, recall and f1-score. All scores of model A2 are highest with respect to other models. Model B has trouble predicting *q* having low precision, recall, F1-score. Model C is better than model B but it is inferior than model A1, model A2, model A3 for all measures.
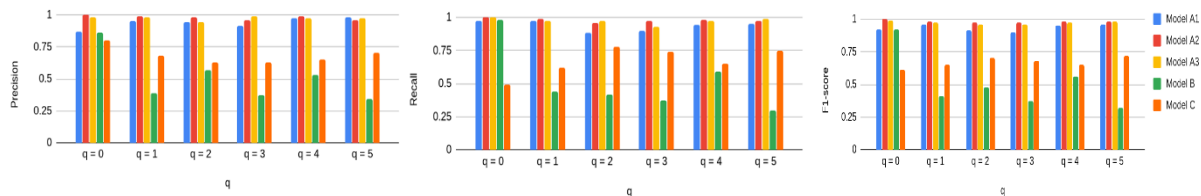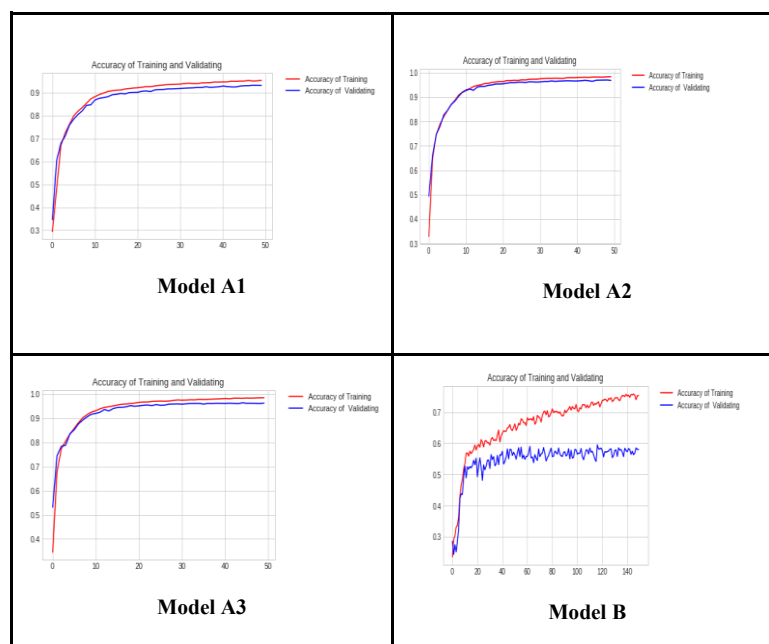

Fig.5. Precision, Recall and F1-score of the models by considering *q* underlying *p*.

For the case of *q* underlying *p*, the experiments of Table 4 show the similar results with the case of *p* underlying *q*.
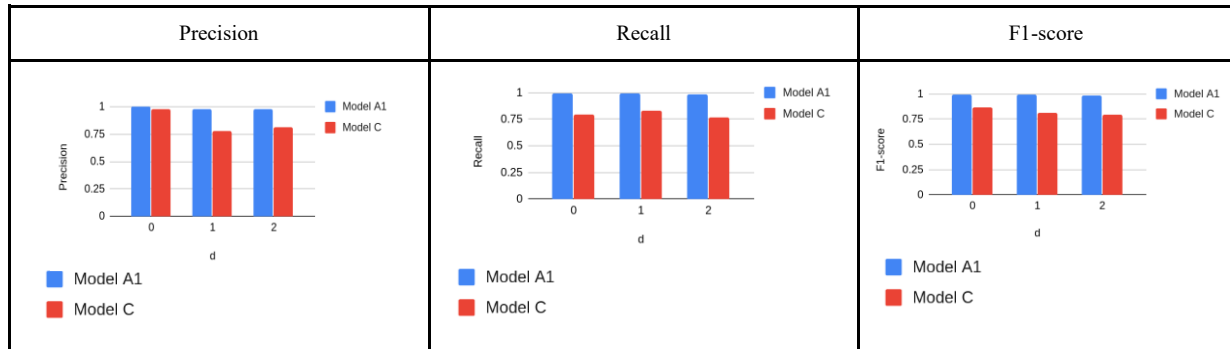
Table 4. Graphical analysis of deep learning models (*q* underlying *p*).

5.3. Results of the SID model in the case of d underlying p and q
Table 5 shows precision, recall and f1-score between model A1 and model C which it is clear that model A1 gives the best precision, recall and f1-score.

Table 5. Precision, Recall and F1-score of the models by considering *d* underlying *p* and *q*.

5.4. Results of the SIDA algorithm for forecasting the real world time series data.
The SIDA algorithm is applied to the non-seasonal real world time series data. There are 7 real world time series data which are available in "fpp2" package in R. Details of the real world time series data are shown in Table 6.

Table 6. Description of the real world dataset.

	Data	Quote Description from R	Source
1	ausair	"Total annual air passengers (in millions) including domestic and international aircraft passengers of air carriers registered in Australia. 1970-2016."	World Bank. https://data.is/x5KiEO
2	auscafe	"The total monthly expenditure on cafes, restaurants and takeaway food services in Australia ($billion). April 1982 - September 2017."	Australian Bureau of Statistics. Catalogue No. 8501.0
3	electsales	"Annual electricity sales for South Australia in GWh from 1989 to 2008. Electricity used for hot water has been excluded."	Australian Energy Market Operator.
4	gasoline	"Weekly data beginning 2 February 1991, ending 20 January 2017. Units are "million barrels per day"."	US Energy Information Administration.
5	guinearice	"Total annual rice production (million metric tons) for Guinea. 1970-2011."	World Bank. https://data.is/whKD8H
6	livestock	"Annual sheep livestock numbers in Asia (in million head)."	United Nations. https://data.is/GFxwQi
7	maxtemp	"Maximum annual temperatures (degrees Celsius) for Moorabbin Airport, Melbourne. 1971-2016."	Australian Bureau of Meteorology.

Table 7 demonstrates the evaluation measures applying with the real world time series datasets which are MAE, RMSE, MAPE and SMAPE. The SIDA algorithm outperforms the likelihood-based method four out of six datasets only one dataset exhibits the same performance. In order to confirm the performance of the SIDA algorithm, the residual test is performed using the Ljung-Box test.

**Table 7.** Evaluation Measures of the real word time series dataset.

		MAE	RMSE	MAPE	SMAPE

	Dataset	The SIDA algorithm	Likelihood method	The SIDA algorithm	Likelihood method	The SIDA algorithm	Likelihood method	The SIDA algorithm	Likelihood method
1	ausair	4.21	3.33	5.09	4.08	6	4.74	5.76	4.58
2	auscafe	0.13	0.19	0.17	0.23	3.67	5.55	3.73	5.35
3	electsales	0.2	0.27	0.22	0.29	2.49	3.35	2.53	3.42
4	gasoline	0.22	0.95	0.28	1.16	0.23	0.99	0.23	1
5	guinearice	0.01	0.01	0.02	0.02	0.22	0.22	0.22	0.22
6	livestock	12.08	23.22	13.84	25.42	2.72	5.21	2.68	5.37
7	maxtemp	32.97	2.17	35.69	2.45	77.42	5.07	134.21	5.24
	Winning cases	5	3	5	3	5	3	5	3

The residual from the suitable forecasting model should be white noise which is uncorrelated zero-mean with constant variance. Table 8 shows the p-values of the Ljung-Box test from the SIDA algorithm and the likelihood-based method. The results show that auscafe and maxtemp provide p-values above 0.01 and 0.05 indicating the residual errors are independent and corresponding to the white noise process. Two datasets which are ausair and electsales exhibits white noise residual from the SIDA algorithm but it fails for the likelihood-based method. Moreover, datasets of gasoline and guinearice, both methods fail the white noise test for residuals. It may be that these time series data may not be suitable for the ARIMA model.

Table 8. The p-values for the Ljung-Box test.

	Dataset	Ljung-Box test
1	ausair	
2	auscafe	

3	electsales	
4	gasoline	
5	guinearice	
6	livestock	
7	maxtemp	
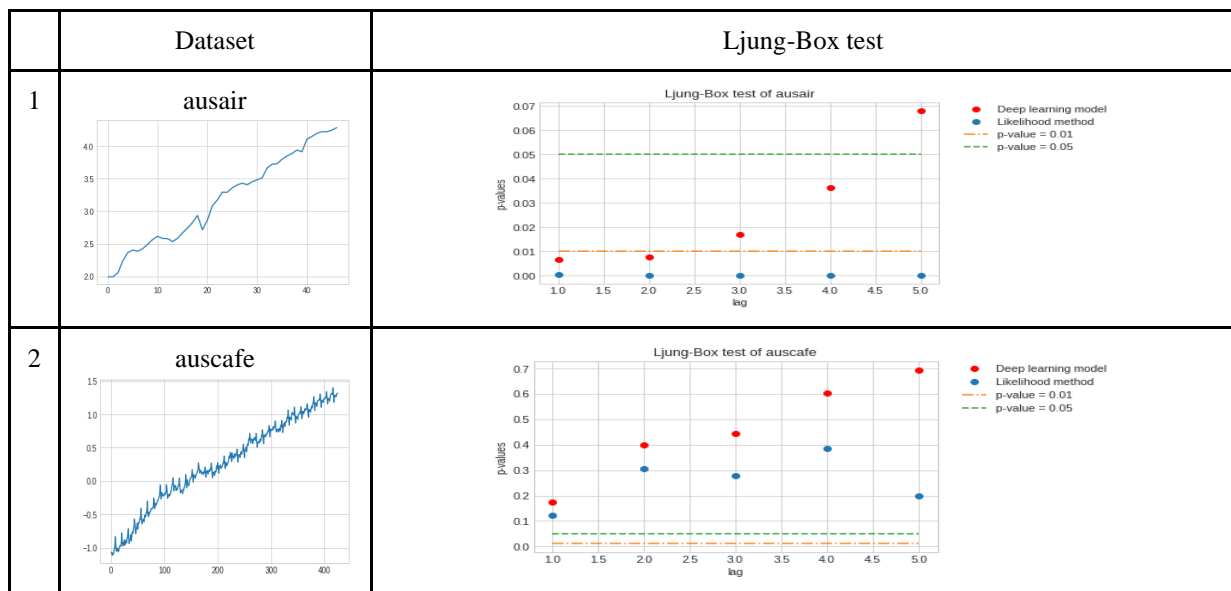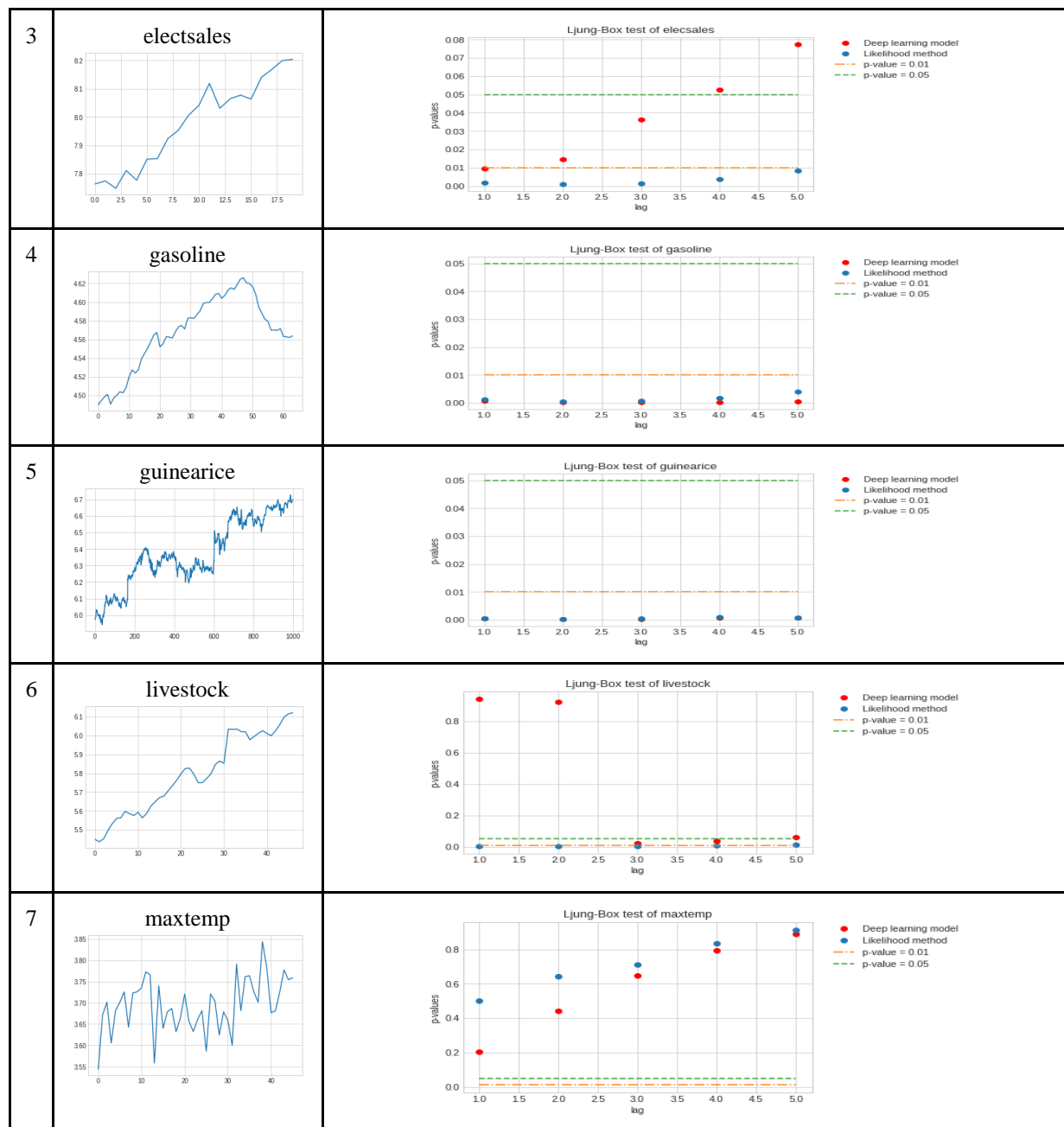
6. Conclusion and Discussion

In this research, the self-identification deep learning model is established by training ACF and PACF images as the input data to identify the ARIMA order and the result from the SID model will be fit using the Box-Jenkin methods for the best ARIMA model. The entire process is the self-identification deep learning ARIMA (SIDA) algorithm. The results of the SID model exhibit the suitable ARIMA order in the time series data whereas the likelihood-based method sometimes fails to give the appropriate ARIMA order. The outcome from this paper confirms that the use of sample ACF and sample PACF is still appropriate for model identification.

Moreover, the SIDA algorithm can apply to the real world datasets and shows the superior results over the likelihood-based method even though some datasets, residuals from both methods will fail the Ljung-Box test. This may cause by datasets that cannot be fitted by any ARIMA model.

These problems could be solved by changing the ARIMA model such as the SARIMA model to handle the seasonal component, the ARCH model to handle non-constant variance.

References

[1]   Wei, W. W. (2006). Time series analysis. In The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2.

[2]   Brockwell, P. J., & Davis, R. A. (2016). Introduction to time series and forecasting. springer.

[3]   Antoniou, A., & Holmes, P. (1995). Futures trading, information and spot price volatility: evidence for the FTSE-100 stock index futures contract using GARCH. Journal of Banking & Finance, 19(1), 117-129.

[4]   Fahimifard, S. M., Homayounifar, M., Sabouhi, M., & Moghaddamnia, A. R. (2009). Comparison of ANFIS, ANN, GARCH and ARIMA techniques to exchange rate forecasting. Journal of Applied Sciences, 9(20), 3641-3651.

[5]   Nwankwo, S. C. (2014). Autoregressive integrated moving average (ARIMA) model for exchange rate (Naira to Dollar). Academic Journal of Interdisciplinary Studies, 3(4), 429.

[6]   Nyoni, T. (2018). Modeling and Forecasting Naira/USD Exchange Rate In Nigeria: a Box-Jenkins ARIMA approach.

[7]   Kraft, J., & Kraft, A. (1977). Determinants of common stock prices: a time series analysis. The journal of finance, 32(2), 417-425.

[8]   Yaziz, S. R., Azizan, N. A., Zakaria, R., & Ahmad, M. H. (2013, December). The performance of hybrid ARIMA-GARCH modeling in forecasting gold price. In 20th International Congress on Modelling and Simulation, Adelaide (pp. 1-6).

[9]   Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. Journal of Applied Mathematics, 2014.

[10]   Yaziz, S. R., & Zakaria, R. (2019, November). ARIMA and Symmetric GARCH-type Models in Forecasting Malaysia Gold Price. In Journal of Physics: Conference Series (Vol. 1366, No. 1, p. 012126). IOP Publishing.

[11]   Box, G. E., & Jenkins, G. M. (1970). Time series analysis: Forecasting and control Holden-Day. San Francisco, 498.

[12]   Tsay, R. S. (1989). Testing and modeling threshold autoregressive processes. Journal of the American statistical association, 84(405), 231-240.

[13]   Engle, R. F. (1983). Estimates of the Variance of US Inflation Based upon the ARCH Model. Journal of Money, Credit and Banking, 15(3), 286-301.

[14]   Bollerslev, T. (1990). Modelling the coherence in short-run nominal exchange rates: a multivariate generalized ARCH model. The review of economics and statistics, 498-505.

[15]   Akaike, H. (1987). Factor analysis and AIC. In Selected papers of hirotugu akaike (pp. 371-386). Springer, New York, NY.

[16]   Neath, A. A., & Cavanaugh, J. E. (2012). The Bayesian information criterion: background, derivation, and applications. Wiley Interdisciplinary Reviews: Computational Statistics, 4(2), 199-203.

[17]   Wang, W. C., Chau, K. W., Xu, D. M., & Chen, X. Y. (2015). Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. Water Resources Management, 29(8), 2655-2675.

[18]   Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: ARIMA vs. LSTM. arXiv preprint arXiv:1803.06386.

[19]   Nath, B., Dhakre, D. S., & Bhattacharya, D. (2019). Forecasting wheat production in India: an ARIMA modelling approach. Journal of Pharmacognosy and Phytochemistry, 8(1), 2158-2165.

[20]   Lee, K. C., & Oh, S. B. (1996). An intelligent approach to time series identification by a neural network-driven decision tree classifier. Decision Support Systems, 17(3), 183-197.

[21]   Chenoweth, T., Hubata, R., & Louis, R. D. S. (2000). Automatic ARMA identification using

neural networks and the extended sample autocorrelation function: a reevaluation. Decision Support Systems, 29(1), 21-30.

[22] Al-Qawasmi, K. E., Al-Smadi, A. M., & Al-Hamami, A. (2010, July). Artificial neural network-based algorithm for ARMA model order estimation. In International Conference on Networked Digital Technologies (pp. 184-192). Springer, Berlin, Heidelberg.

[23] Guo, S. J., Hsu, F. C., & Hung, C. C. (2018, December). Deep candlestick predictor: A framework toward forecasting the price movement from candlestick charts. In 2018 9th International Symposium on Parallel Architectures, Algorithms and Programming (PAAP) (pp. 219-226). IEEE.

[24] Brunel, A., Pasquet, J., PASQUET, J., Rodriguez, N., Comby, F., Fouchez, D., & Chaumont, M. (2019). A CNN adapted to time series for the classification of Supernovae. Electronic Imaging, 2019(14), 90-1.

[25] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019, July). Deep neural network ensembles for time series classification. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-6). IEEE.

[26] Yang, H., Zhu, Y., & Huang, Q. (2018, December). A multi-indicator feature selection for CNN-driven stock index prediction. In International conference on neural information processing (pp. 35-46). Springer, Cham.

[27] Sandoval, J. (2018, October). CNN with Limit Order Book Data for Stock Price Prediction. In Proceedings of the Future Technologies Conference (FTC) 2018 (Vol. 1, p. 444). Springer.

[28] Kim, T., & Kim, H. Y. (2019). Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. PloS one, 14(2).

[29] Tang, W. H., & Röllin, A. (2018). Model identification for ARMA time series through convolutional neural networks. arXiv preprint arXiv:1804.04299.

[30] Aloysius, N., & Geetha, M. (2017, April). A review on deep convolutional neural networks. In 2017 International Conference on Communication and Signal Processing (ICCSP) (pp. 0588-0592). IEEE.

[31] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[32] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).