

Self-Identification ResNet-ARIMA Forecasting Model

PAISIT KHANARSA, ARTHORN LUANGSODSAI and KRUNG SINAPIROMSARAN
Department of Mathematics and Computer Science, Faculty of Science
Chulalongkorn University
Bangkok, 10330
THAILAND

Abstract: - The challenging endeavor of a time series forecast model is to predict the future time series data accurately. Traditionally, the fundamental forecasting model in time series analysis is the autoregressive integrated moving average model or the ARIMA model requiring a model identification of a three-component vector which are the autoregressive order, the differencing order, and the moving average order before fitting coefficients of the model via the Box-Jenkins method. A model identification is analyzed via the sample autocorrelation function and the sample partial autocorrelation function which are effective tools for identifying the ARMA order but it is quite difficult for analysts. Even though a likelihood based-method is presented to automate this process by varying the ARIMA order and choosing the best one with the smallest criteria, such as Akaike information criterion. Nevertheless the obtained ARIMA model may not pass the residual diagnostic test. This paper presents the residual neural network model, called the self-identification ResNet-ARIMA order model to automatically learn the ARIMA order from known ARIMA time series data via sample autocorrelation function, the sample partial autocorrelation function and differencing time series images. In this work, the training time series data are randomly simulated and checked for stationary and invertibility properties before they are used. The result order from the model is used to generate and fit the ARIMA model by the Box-Jenkins method for predicting future values. The whole process of the forecasting time series algorithm is called the self-identification ResNet-ARIMA algorithm. The performance of the residual neural network model is evaluated by Precision, Recall and F1-score and is compared with the likelihood based- method and ResNET50. In addition, the performance of the forecasting time series algorithm is applied to the real world datasets to ensure the reliability by mean absolute percentage error, symmetric mean absolute percentage error, mean absolute error and root mean square error and this algorithm is confirmed with the residual diagnostic checks by the Ljung-Box test. From the experimental results, the new methodologies of this research outperforms other models in terms of identifying the order and predicting the future values.

Key-Words: - ARIMA, ACF, PACF, differencing time series, ResNet, Ljung-Box

Received: October 16, 2019. Revised: April 9, 2020. Accepted: April 20, 2020. Published: May 13, 2020.

1 Introduction

The desire to predict a future value of the time series correctly and understanding the past values pushes the creation of time series analysis models that explain the behaviour of observed data in nature such as price products[1][2][3] or economic values[4][5]. An appropriate model must be able to explain the structure of the time series, and it would accurately predict future values.

At present, time series modelling is used to model the essential nature of the time series data to obtain the optimal forecast values. It begins with the basic models such as the autoregressive integrated moving average model or ARIMA and the seasonal autoregressive integrated moving average model or SARIMA by Box and Jenkins in 1970[6] for simple time series and drives to more complex models such as the autoregressive conditional heteroskedasticity model (ARCH) by Engle in 1983[7] and the generalized autoregressive conditional

heteroskedasticity model (GARCH) by Bollerslev in 1990[8].

The ARIMA model for forecasting time series is built on the Box-Jenkins model identification and estimation to obtain the best ARIMA model. The key components of the ARIMA model are the autoregressive or AR component, the differencing component and the moving average or MA component applying the Box-Jenkins method for finding the best fit coefficients. Statisticians who investigate time series define the ARIMA order using the sample autocorrelation function or ACF and the sample partial autocorrelation function or PACF via visualizing their plots. The characteristics of ACF lags and PACF lags leads to the identification order based on time series analysis[9].

Recently to avert human analysis, analysts use a likelihood-based method to automatically determine the ARIMA order and the best ARIMA coefficients according to the Akaike Information Criterion or AIC[10], the Bayesian Information Criterion or

BIC[11] and the Hannan Quinn Information Criterion or HQIC[12]. Although it is easy to use, its residuals may not satisfy a white noise property. At present, these methods are still used in many fields, such as stochastic wind power generation by Chen et al.[13] in 2009, water quality series by Faruk[14] in 2010 and daily and monthly average global solar radiation in Seoul, South Korea by Alsharif et al. in 2019[15].

Various researchers have proposed numerous methods to identify the suitable ARIMA order since it is a significant part of the creation of the ARIMA model. In 1996, Lee and Oh[16] used an ANN-driven decision tree classifier to identify the ARMA order and they concluded that most time series data uses order less than 6. Chenoweth et al. in 2000[17] used the extended sample autocorrelation function or ESACF to identify the orders of AR and MA at the same time. After that, Al-Qawasmi et al. in 2010[18] applied the identification ARIMA order with the concept of a special covariance matrix and used the MEV criterion for deciding the best ARIMA model.

In recent years, a deep learning model has been applied to solve several problems successfully including time series analysis. For example, Amini and Karabasoglu in 2016[19] used the improved ARIMA model by optimizing the integrated and auto-regressive order parameters to forecast the electric vehicle charging demand for stochastic power system operation. Guarnaccia et al. in 2017[20] used deterministic decomposition and seasonal ARIMA time series models to predict the airport noise at Nice international airport in France. Fukuoka et al. in 2018[21] used LSTM and 1D-CNN for predicting wind speed. Kim et al. in 2019[22] presented the long short-term memory-CNN model combining several features of the time series data comprising stock time series and stock chart images to predict stock prices of SPDR, S&P500, and ETF. Apart from the use of deep learning in analyzing and forecasting the time series, deep learning has also been applied to define the ARMA order. ResNet50 was presented by Tang and Röllin[23] to identify ARMA order by comparing the performance with the likelihood-based method based on AIC. Their findings showed that, for the low-order ARMA model, CNN outperformed the likelihood-based method in terms of accuracy and speed.

Researchers using sample ACF and sample PACF to identify the ARIMA order by the Box-Jenkins identification may generate different ARIMA models from different viewpoints. While the likelihood-based method will return a single

ARIMA model without the use of sample ACF and sample PACF. The result model may still be unsatisfactory due to the unacceptable residual behavior. Thus this issue is attacked by adapting ACF plots, PACF plots and differencing time series images as inputs from a collection of simulated stationary and invertibility time series data to train the deep learning model until it converges. The output model is called the self-identification ResNet-ARIMA order model or the SIRO model.

Real-World time series data from the fpp package in R will be used to forecast where their ARIMA orders are identified by the fully trained SIRO model. Then the Box-Jenkins method applies to these time series to construct the ARIMA model. The whole algorithm is called the self-identification ResNet-ARIMA algorithm or the SIRA algorithm.

The paper is composed of eight sections. The first section is the introduction. The second section describes the idea of convolutional neural networks. The third section is to describe the concept of time series model and stationary property. The fourth section explains the SIRA algorithm. The fifth section explains the measurements of the models. The sixth and the seventh section discusses findings from the experimental results. The last section covers the conclusion and future work.

2 Convolutional Neural Network concept

One type of neural network that successfully classifies images in the ImageNet Large Scale Visual Recognition Challenge[24] is a convolutional neural network or CNN. Presently, CNN is used popularly in various fields after the success of AlexNet by Krizhevsky in 2012[25]. Later, the modern architectures of convolutional neural networks are proposed including VGGNet[26] from the University of Oxford. VGGNet had significant improvement over ZFNet[26] which was the winner in 2013 and AlexNet which was the winner in 2012. Moreover, ResNet was introduced by Kaiming He et al[27] in 2015. The ResNet model uses the concept of skip connection in the convolutional layer and applies batch normalization to solve the vanishing gradient problem in the deep learning model.

Although there are many architectures about CNN, their basic structure of CNN is quite similar. The first concept starts with the convolutional layer to learn the features of the images by convolutional kernels or filter matrices. Then the concept of pooling is used for reducing the resolution of the images and extract the distinctive features of the

image. Finally, the fully connected layers are applied in the last layer according to the concept of the artificial neural network for calculating the outputs. The whole process is called deep learning. Then weights which are assigned in the deep learning model are trained by the concept of back-propagation to update the weights in the deep learning model depending on the training rounds, called epochs. The details of this process can be seen in [28].

For this research, the ResNet architecture is applied for solving the identifying ARIMA order problem by learning the ACF plot, the PACF plot and the differencing time series images. The details of the time series model, ACF, PACF and differencing time series are described in the next section.

3 Time Series Model and Stationary Time Series concept

The concept of time series model including stationary time series is an important part for understanding time series data suiting with the model. The basic 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 details of time series model and stationary time series are demonstrated as follows.

The autoregressive process or AR of time series analysis of p order can be explained by

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

or

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

where x_t and ε_t are the time series data and random error at time t . $\phi_1, \phi_2, \dots, \phi_p$ are the coefficients of the AR process and $\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$ which $x_{t-1} = Bx_t$.

To explain the stationary of the AR process, the root of $\phi_p(B) = 0$ where B is the variable of this polynomial must be outside of the unit circle[9].

Then, the moving average process or MA of q order can be explained by

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

or

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

where $\theta_1, \theta_2, \dots, \theta_q$ are the coefficients of the MA process and $\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q$. In the concept of MA process, the stationary is explained in the name of invertibility according that the root of $\theta_q(B) = 0$ where B is the variable of this polynomial must be outside of the unit circle[9].

Moreover, both processes are applied together, called the ARMA process.

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

To confirm stationary and invertibility of the ARMA process, the root of $\phi_p(B) = 0$ and $\theta_q(B) = 0$ must be outside of the unit circle.

Then, the concept of differencing is applied to deal with trend in the time series, called ARIMA process.

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

where d is the number of differencing for eliminating trends in the time series.

To determine the ARIMA order, the concept of ACF, PACF and differencing are applied. The first step, the differencing order is identified by considering the characteristic of differencing time series via ACF. If its sample ACF is all positive and slowly decay then it requires to take the differencing order. When finishing identifying the differencing order, the differencing time series is used to identify the AR order and the MA order by considering PACF for the AR order and ACF for the MA order. If PACF cuts off at lag k then the AR order should be fitted by order k . Similarly, the MA order can be identified in the same way as the AR order by using ACF instead of PACF. The details of calculating ACF and PACF are reviewed in [9].

This research uses these concepts to generate images as inputs to the ResNet model which the details of the proposed model are described in the next section.

4 Self-Identification ResNet-ARIMA

This section explains the process of the self-identification ResNet ARIMA algorithm or the SIRA algorithm. The first subsection covers the deep learning architecture and the training process of the SIRO model using simulated time series data. The SIRO model for identifying ARIMA order is splitted into two parts including the pq -SIRO model using for identifying the ARMA order and the d -

SIRO model using for identifying the differencing order. Then, the next subsection will explain the SIRA algorithm that uses an output of the SIRO model to generate the ARIMA forecasting model via the Box-jenkins method.

4.1 Constructing the self-identification ResNet-ARIMA order or SIRO model

In this research, the *pq*-SIRO model for identifying the ARMA order is built based on the ResNet architecture. The inputs of the *pq*-SIRO model consist of five channels of 50×50 black and white images. The first channel is an image of the ACF plot, the second channel is an image of the PACF plot and the rest are images of the time series image taking differencing from $d = 0$ to 2. The *pq*-SIRO model comprises of 14 convolutional layers applying the concept of a skip connection over two adjacent convolutional layers. The first six convolutional layers have 64 of 3×3 filter matrices with stride = 1 and the next eight convolutional layers have 128 of 3×3 filter matrices with stride = 1. The next layer is the max-pooling layer of filter of size 2×2 and stride is set to 2. It is followed by two fully connected layers having 1024 and 512 nodes, respectively. Finally, the final layer is softmax.

For the case of identifying the differencing order, the architecture of the *d*-SIRO model is similar to the case of identifying the ARMA order but the inputs are different. The *d*-SIRO model of this case consists of 3 channels of the ACF plots from time series images taking differencing from $d = 0$ to 2, respectively. The architectures of the *pq*-SIRO model and the *d*-SIRO model are shown in Fig.1 and Fig.2.

After designing the architectures of the *pq*-SIRO model and the *d*-SIRO, the parameters of the both models are shown in Table 1. It can be computed by (filter height x filter width x input channels + 1) x number of filters. The parameters are used to assign weights to the both models for training the models.

Table 1. The details of parameters in each layer of the SIRO model

Layer	Filter matrices size	Number of filter matrices	Number of parameters
Conv1	3x3	64	1792
Conv2-6	3x3	64	36928
Conv7-14	3x3	128	73856
FC1	1024	1	132096
FC2	512	1	524800
Total			1286464

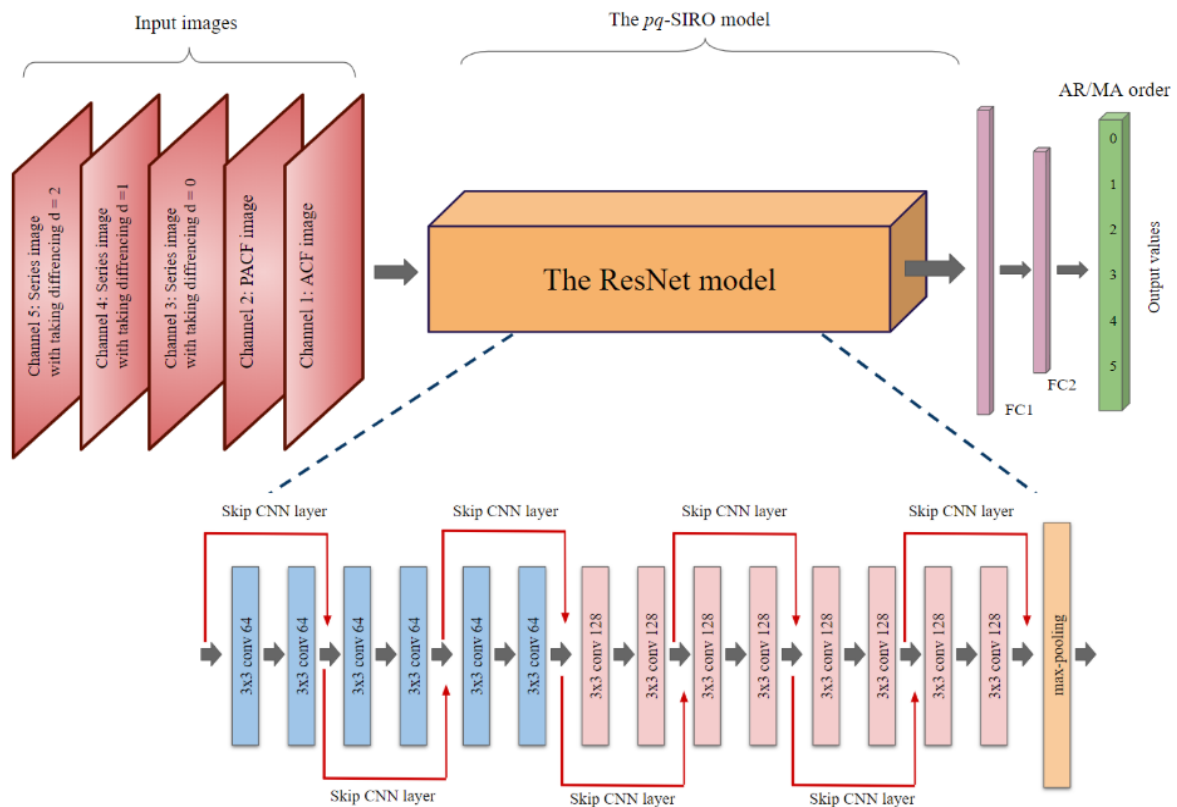


Fig.1. The architecture of the *pq*-SIRO model for training the AR and MA order.

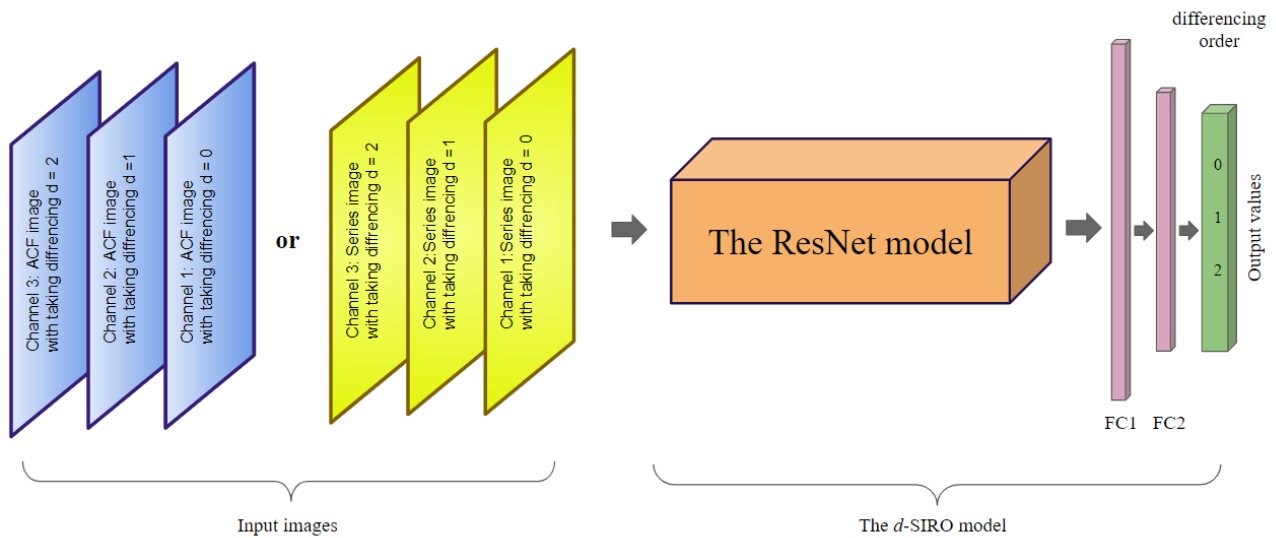


Fig.2. The architecture of the *d*-SIRO model for training the differencing order.

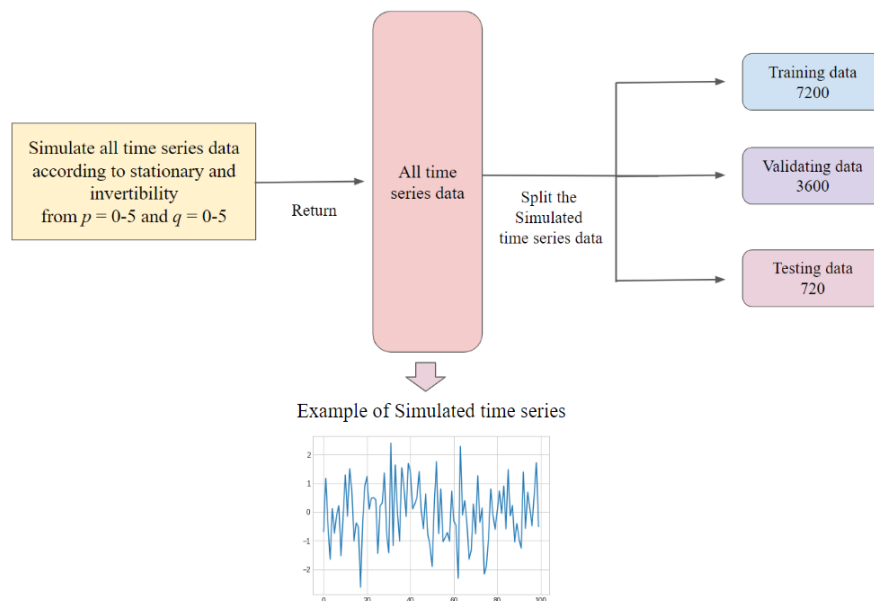


Fig.3. The process of simulated time series data of the ARMA order and the example of time series data.

The training process of both the *pq*-SIRO model and the *d*-SIRO model are the same. All simulating time series data are converted to their corresponding images which are split to the training data, the validating data and the testing data. The training data is used for learning and updating weights in each parameter of the deep learning model. The images of the training data are sent to the deep learning model in each epoch to learn the appropriate weights while the weights in the first epoch of learning are random. The validating data is used for evaluating the accuracy occurring in each epoch. For this research, the number of epochs for learning the SIRO model is set to be 50. The

weights in the deep learning model are updated until the accuracy of the training data and the validating data converge. After finishing training the deep learning model, the appropriate weights are used for the appropriate SIRO model and the testing data is used to measure the performance of the obtained SIRO model.

In this research, the simulated time series data are generated from the ARMA process in the section 3 by varying *p* from 0 to 5 and *q* from 0 to 5. Moreover, all coefficients of the process must be randomly generated to satisfy the stationary property and the invertibility property from the uniform distribution within range -1 to 1. If the

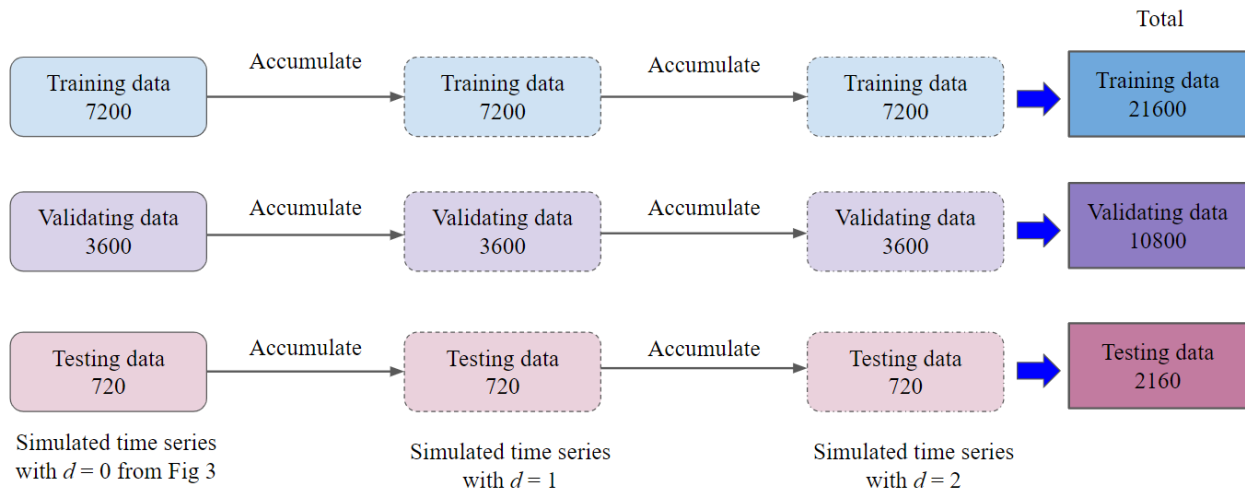


Fig.4. The process of simulated time series data of the differencing order

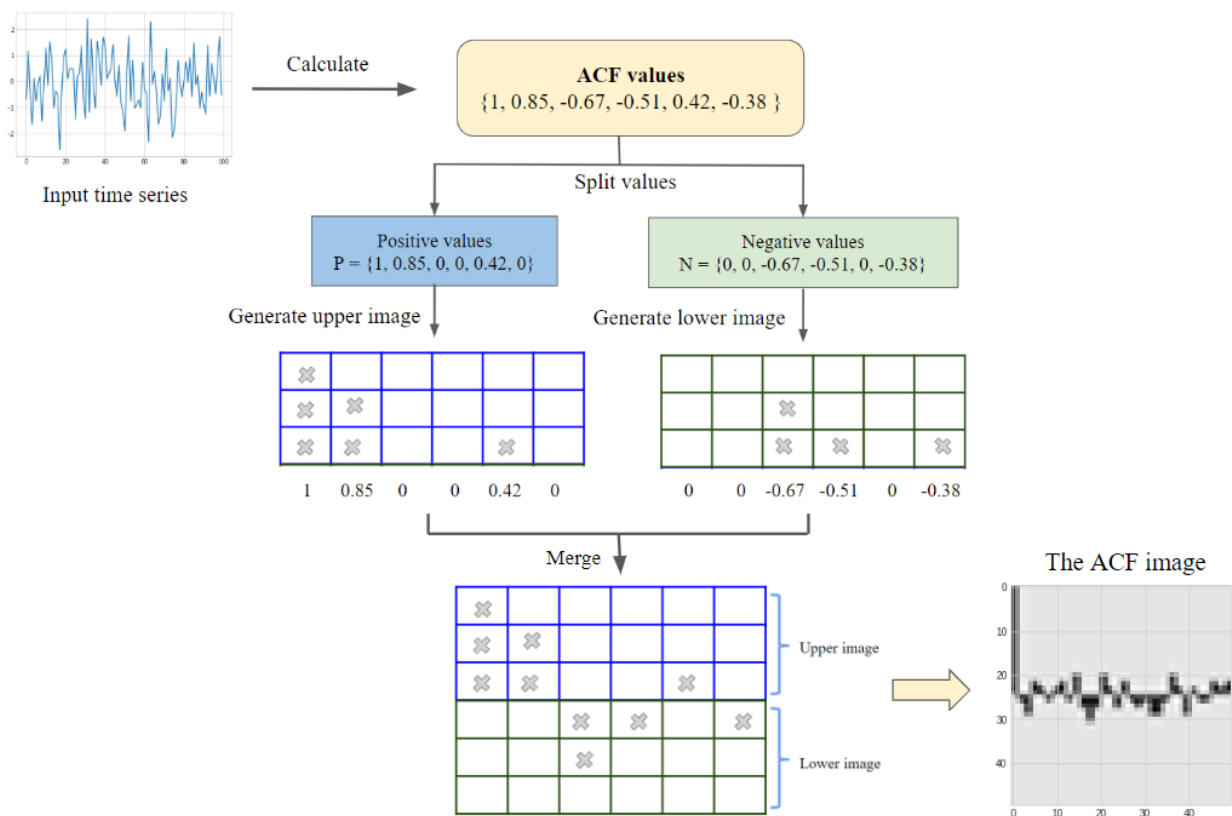


Fig.5. The example of generating the ACF image

stationary property or invertibility property is not satisfied, all coefficients will be regenerated. The number of the time series data in the training data, the validating data and the testing data are collected 7200, 3600 and 720, respectively. The process and the example of time series which are simulated are shown in Fig.3.

In the process of simulated time series for the differencing order, the simulated time series data from the ARMA process is accumulated by varying d from 0 to 2, respectively. Therefore, the number of the time series data in the training data, the validating data and the testing data are collected 21600, 10800 and 2160, respectively. The process are shown in Fig.4.

After the simulating time series data is generated, ACF and PACF are generated and are converted to the ACF image, the PACF image and the time series image which are black and white. The converting process is described in the algorithm A and B. The example of generating the ACF image is shown in Fig.5. For other images, PACF and time series can be done similarly.

Algorithm A: Generating images of ACF or PACF

Input: ACF or PACF, k = the maximum lags of ACF or 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 1 with the height of P[i] and 0 for otherwise
- 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

Algorithm B: Generating differencing time series images

Input: time series data, image size = k , number of differencing = d

- Step 1:** Define the image size as $k \times k$.
- Step 2:** Take differencing to time series data d time and define s = differencing time series
- Step 3:** Generate the image.
 Reshape 1 dimension of s with length $k \times 1$ to 2 dimension image with size $k \times k$
- Step 4:** Normalize image into range 0-1

The example of the ACF image, the PACF image and the differencing time series images are shown in Table 2.

4.2 The self-identification ResNet-ARIMA or SIRA algorithm

After obtaining the SIRO model to identify the ARIMA order from the images of ACF, PACF and differencing time series, the SIRO model is used to construct the forecasting time series algorithm to build the ARIMA model and forecast future values. This process is called the SIRA algorithm as shown in Fig 6.

The process of this algorithm starts by changing the input time series to the images of ACF, PACF and differencing time series to be able to send to the SIRO model. Then, the SIRO model predicts the AR order and the MA order from the pq -SIRO model and the differencing order from the d -SIRO model. Finally, the ARIMA order is used to fit all coefficients of the ARIMA model by the Box-Jenkins method. Then, the SIRA algorithm returns

Table 2. Some examples of images in training collections.

(p,d,q)	ACF images	PACF images	Series images by taking $d = 0$	Series images by taking $d = 1$	Series images by taking $d = 2$
ARIMA (0,0,0)					
ARIMA (1,0,3)					
Series images (1,1,2)					

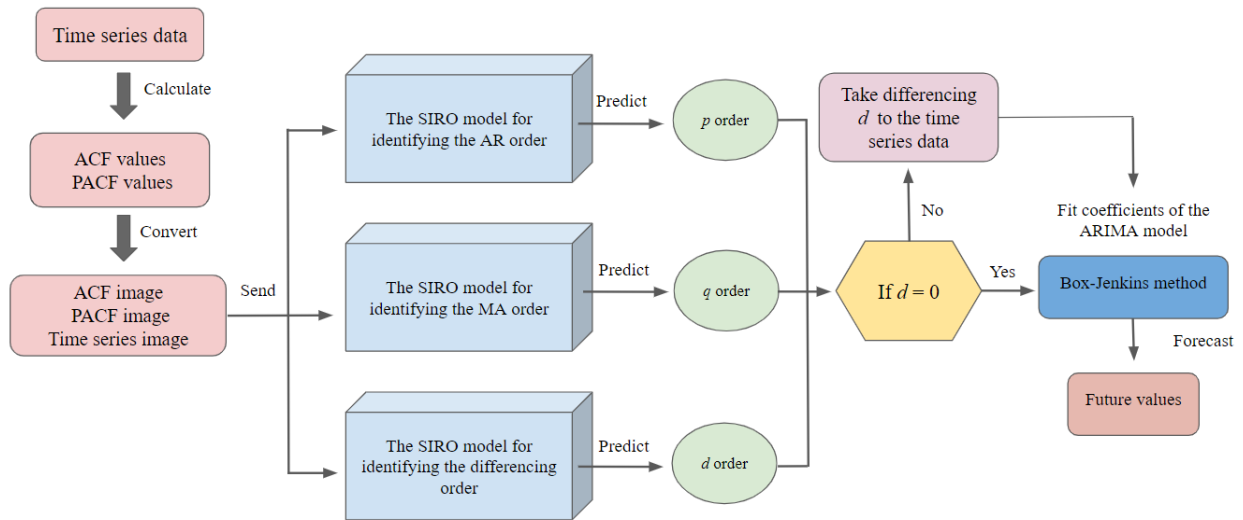


Fig.6. The algorithm self-identification ResNet-ARIMA or SIRA

the ARIMA model used to forecast the future values.

5 The model evaluation

This section explains the measurements used for testing the performance of the SIRO model and the SIRA algorithm. It consists of two sections. The measures of the SIRO model for classification problem are described in section 5.1 and the measures of the SIRA algorithm for forecasting the time series data are described in section 5.2

5.1 The classification measures of the SIRO model

The popular measurements for the classification problem comprises of Precision, Recall and F1-score which they are computed from the confusion matrix as follows in Table 3.

Table 3. The confusion matrix

		Predicted class	
		Class = Yes	Class = No
Actual class	Class = Yes	True Positive (TP)	False Negative (FN)
	Class = No	False Positive (FP)	True Negative (TN)

The confusion matrix is used to evaluate predictions with a model which Precision, Recall and F1-score can be computed as follows.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

5.2 The forecasting measures of the SIRA algorithm

The measurements for the time series forecasting used in this research comprises of MAE, RMSE, MAPE and SMAPE. The details are shown as follows.

The mean absolute error or MAE can be computed by

$$MAE = \sum_{i=1}^N \frac{|x_i - \hat{x}_i|}{N}$$

The root mean square error or RMSE can be computed by

$$RMSE = \sqrt{\sum_{i=1}^N \frac{(x_i - \hat{x}_i)^2}{N}}$$

The mean absolute percentage error or MAPE can be computed by

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - \hat{x}_i|}{|x_i|}$$

The symmetric mean absolute percentage error or SMAPE can be computed by

$$SMAPE = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - \hat{x}_i|}{\frac{|x_i| + |\hat{x}_i|}{2}}$$

where x_i and \hat{x}_i are the actual data and the forecasted data, respectively and N is the number of the forecasted data.

6 Performance of the SIRO model

The testing time series data consists of three cases where the first case is called p underlying q which the AR order in $\{0, 1, 2, 3, 4, 5\}$ is fixed by varying the MA order from 0 to 5. The second case is called q underlying p which the MA order q in $\{0, 1, 2, 3, 4, 5\}$ is fixed by varying the AR order from 0 to 5. The third case is called d underlying p and q which the differencing order d in $\{0, 1, 2\}$ is fixed by varying both the AR order and the MA order from 0 to 5. Different models for cases having different channels and type of inputs are demonstrated in Table 4.

The candidates of these experiments consist of likelihood based methods using AIC, BIC and HQIC and the previous deep learning model, proposed by Tang and Röllin[23] in 2018. For the deep learning model, the input data is the time series directly and the architecture of the model is ResNet50 using the concept of skip connection with the 50 convolutional layers.

6.1 Performance of the SIRO model in the case of p underlying q

The scores of all models including Precision, Recall and F1-score are shown in Fig 7. From this experiment, the models having the top-5 scores are S1, S2, S3, S4 and S5. All scores are very close to 1. Although the results of S1, S2, S3, S4 and S5 provide the similar scores of Precision, Recall and F1-score, S1 is the best model. For the worst case, L1, L2 and L3 show the similar results which are around 0.5 to 0.6 in Precision, Recall and F1-score. For the case of model R, it is quite better than L1, L2 and L3 but this model gives lower scores than S1, S2, S3, S4 and S5 when identifying higher order.

Next, the number of epochs for the deep learning models is determined via the accuracy plot from Table 5. The SIRO accuracy of all models are closer to 1 and more robust than model R. Besides,

Table 4. Description of models in each experiment.

	Model	Case: p underlying q	Case: q underlying p	Case: d underlying p and q
		Channel(s)	Channel(s)	Channel(s)
Model S1	SIRO	1 channel: PACF images	1 channel: ACF images	3 channels: ACF images with $d = 0, 1$ and 2
Model S2	SIRO	2 channels: PACF and ACF images	2 channels: PACF and ACF images	3 channels: series images with $d = 0, 1$ and 2
Model S3	SIRO	3 channels: PACF, ACF and series images with $d = 0$	3 channels: PACF, ACF and series images with $d = 0$	None
Model S4	SIRO	4 channels: PACF, ACF and series images with $d = 0, 1$	4 channels: PACF, ACF and series images with $d = 0, 1$	None
Model S5	SIRO	5 channels: PACF, ACF and series images with $d = 0, 1, 2$	5 channels: PACF, ACF and series images with $d = 0, 1, 2$	None
Model R	ResNet 50	Time series data	Time series data	None
Model L1	Likelihood method (AIC criteria)	Time series data	Time series data	Time series data
Model L2	Likelihood method (BIC criteria)	Time series data	Time series data	Time series data
Model L3	Likelihood method (HQIC criteria)	Time series data	Time series data	Time series data

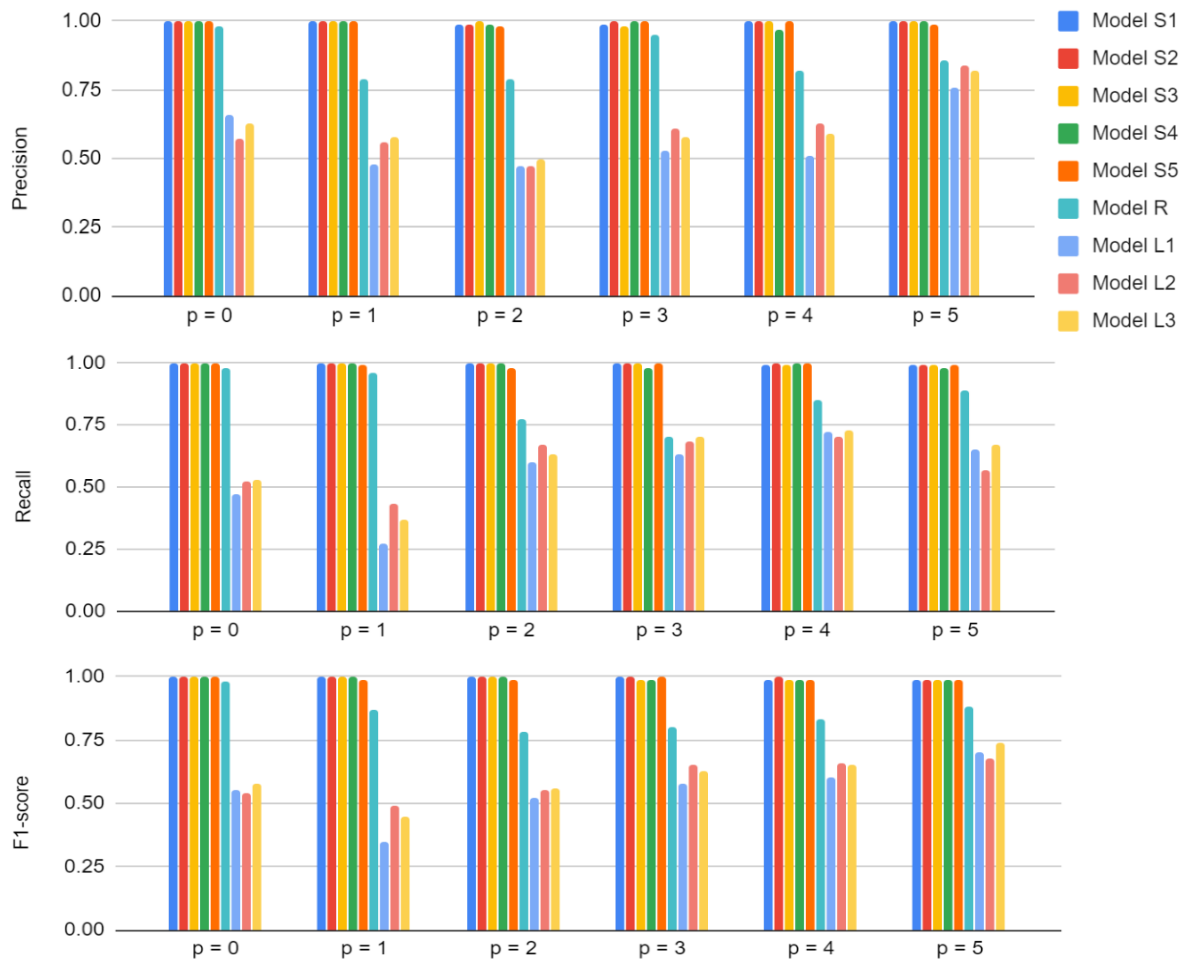
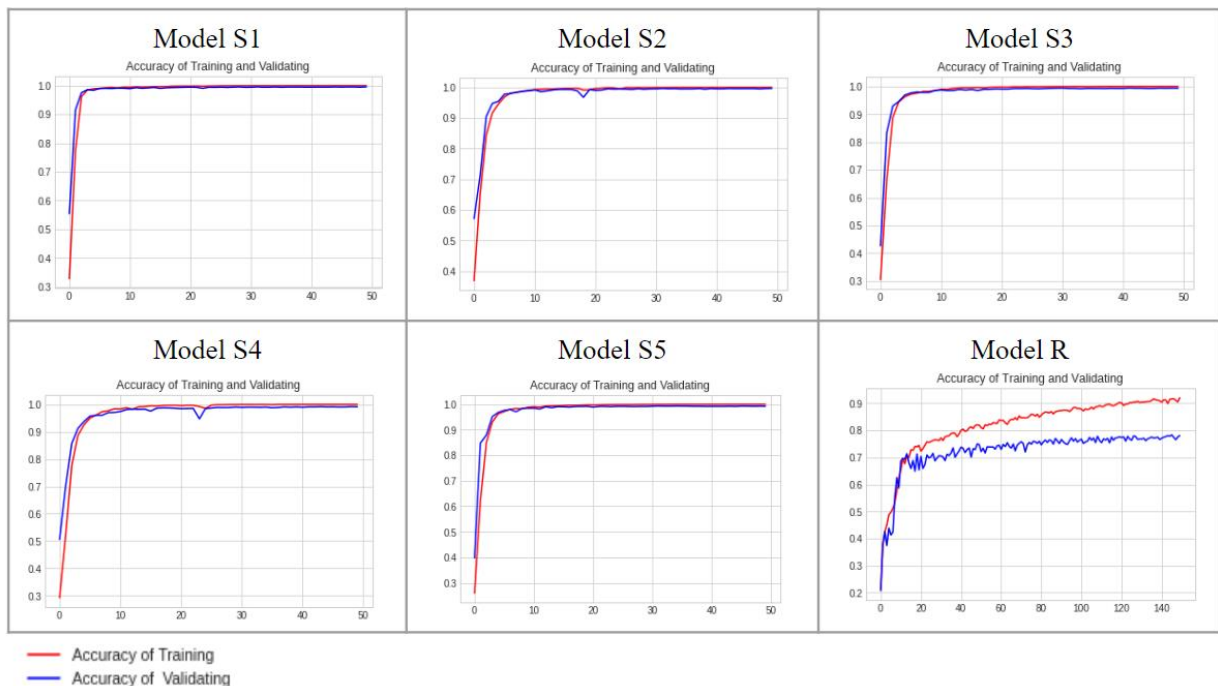


Fig.7. Precision, Recall and F1-score of the models by considering p underlying q .

Table 5. Graphical analysis in case of p underlying q .



model R fluctuates between training and validating datasets for a large number of epochs than the others.

6.2 Performance of the SIRO model in the case of q underlying p

For the case of identifying q order, the scores of Precision, Recall and F1-score are shown in Fig 8. The results show that model S1, S2 and S3 give the best scores of Precision, Recall and F1-score. For the result of model R, all scores have trouble predicting which are indicated by their scores near 0.5 in Precision, Recall and F1-score. For the likelihood-based method case, model L2 gives the best results and is better than model R, however it is still worse than model S1, S2, S3, S4 and S5.

6.3 Performance of the SIRO model in the case of d underlying p and q

For the case of identifying the differencing order in Fig 9, it is obvious that model S1 gives the best scores of Precision, Recall and F1-score in all cases. In the likelihood-based method case, model L3 has the best Precision when $d = 0$ and has the best Recall when $d = 2$.

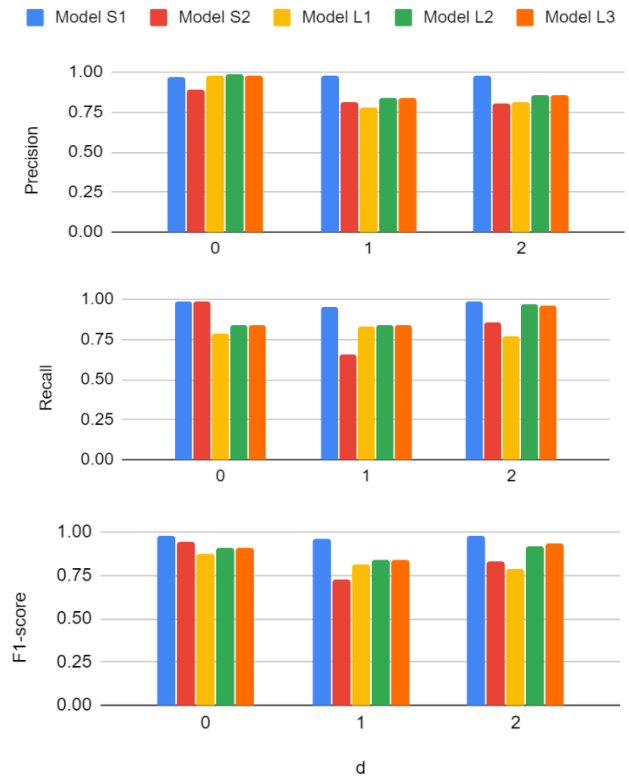


Fig.9. Precision, Recall and F1-score of the models by considering d underlying p and q .

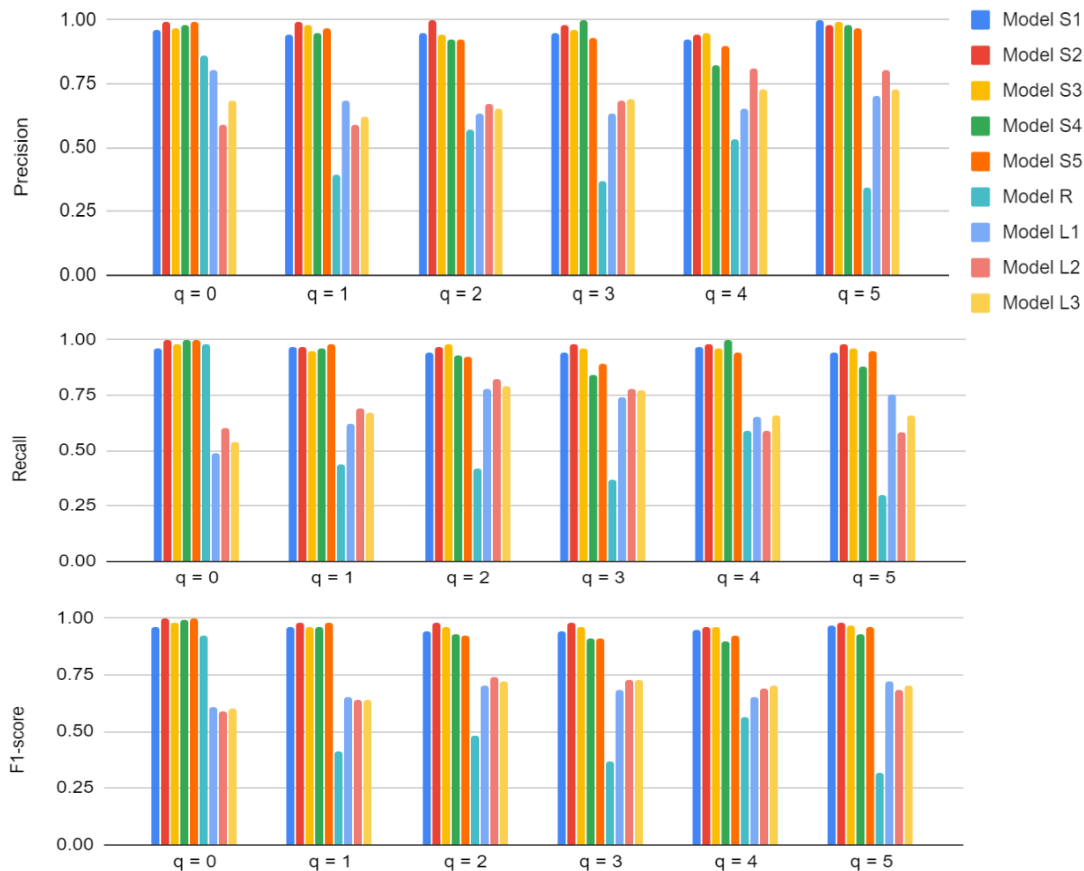


Fig.8. Precision, Recall and F1-score of the models by considering q underlying p .

Table 6. Description of the real world dataset.

Data	Quote Description from “fpp” package in R	Source
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
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
ausbeer	“Total quarterly beer production in Australia (in megalitres) from 1956:Q1 to 2010:Q2”	Australian Bureau of Statistics. Cat. 8301.0.55.001
austourists	“Quarterly visitor nights (in millions) spent by international tourists to Australia. 1999-2015.”	Tourism Research Australia.
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.
gasoline	“Weekly data beginning 2 February 1991, ending 20 January 2017. Units are “million barrels per day”.”	US Energy Information Administration.
guinearice	“Total annual rice production (million metric tons) for Guinea. 1970-2011.”	World Bank. https://data.is/whKD8H
livestock	“Annual sheep livestock numbers in Asia (in million head).”	United Nations. https://data.is/GFfwQi
maxtemp	“Maximum annual temperatures (degrees Celsius) for Moorabbin Airport, Melbourne. 1971-2016.”	Australian Bureau of Meteorology

Table 7. Evaluation measures of the real word time series dataset.

Dataset	MAE		RMSE	
	SIRA	Likelihood	SIRA	Likelihood
ausair	4.21	3.33	5.09	4.08
auscafe	0.13	0.19	0.17	0.23
ausbeer	0.024	0.028	0.029	0.035
austourists	0.035	0.117	0.046	0.147
electsales	0.2	0.27	0.22	0.29
gasoline	0.22	0.95	0.28	1.16
guineaice	0.01	0.01	0.02	0.02
livestock	12.08	23.22	13.84	25.42
maxtemp	32.97	2.17	35.69	2.45
Dataset	MAPE		SMAPE	
	SIRA	Likelihood	SIRA	Likelihood
ausair	6	4.74	5.76	4.58
auscafe	3.67	5.55	3.73	5.35
ausbeer	0.41	0.47	0.41	0.46
austourists	0.88	2.94	0.88	2.93
electsales	2.49	3.35	2.53	3.42
gasoline	0.23	0.99	0.23	1.02
guineaice	0.22	0.22	0.22	0.22
livestock	2.72	5.21	2.68	5.37
maxtemp	77.42	5.07	134.2	5.24

7 Performance of the SIRA algorithm for the real world time series data

To ensure the reliability of the SIRA algorithm, the real world time series data are applied to test the performance of the SIRA algorithm. These real world time series data are from the “fpp” package collecting from several fields in the world. All time series data are used in the example in the book "Forecasting: principles and practice" written by Rob J Hyndman and George Athanasopoulos[29] in 2018. The details and sources of the whole datasets are demonstrated in Table 6.

The evaluation measures applying with the real world datasets are shown in Table 7. The SIRA algorithm provides better performance than the likelihood-based method with AIC which the results give better accuracy six out of eight datasets and only one dataset, “guinearice”, shows the same performance.

The last experiment is to test the performance of forecasting by testing with the Ljung-Box test. The suitable ARIMA should exhibit the white noise characteristic of the residual which is uncorrelated zero-mean and constant variance. The results are shown in Table 8. Auscafe, ausbeer and maxtemp have *p*-values beyond 0.01 and 0.05 which suggest that residual errors are independent and conform to the white noise process whereas ausair, austourists, electsales and livestock datasets exhibit of white noise residual from the SIRA algorithm while the likelihood-based method fails. Unfortunately, the

white noise testing of Ljung-Box fails for gasoline and guinearice datasets which it may occur that the ARIMA process could not capture the characteristics of these series.

8 Conclusion and Future work

In this research, the deep learning model, called the self-identification ResNet-ARIMA order model or the SIRO model, is used to solve the issue of identifying ARIMA order by splitting into the pq -SIRO model and the d -SIRO model. The pq -SIRO model is constructed by training the characteristics of ACF, PACF and series images taking differencing d from 0 to 2 while the d -SIRO model is constructed by training using only ACF images taking differencing d from 0 to 2. Then the outputs from the pq -SIRO model and the d -SIRO model are used for the self-identification ResNet-ARIMA or SIRA algorithm by applying the Box-Jenkins method to build the ARIMA model and forecast future values. The details of discussions in this research are divided into the discussion of the SIRO model and the discussion SIRA algorithm.

8.1 Conclusion and discussion of the SIRO model

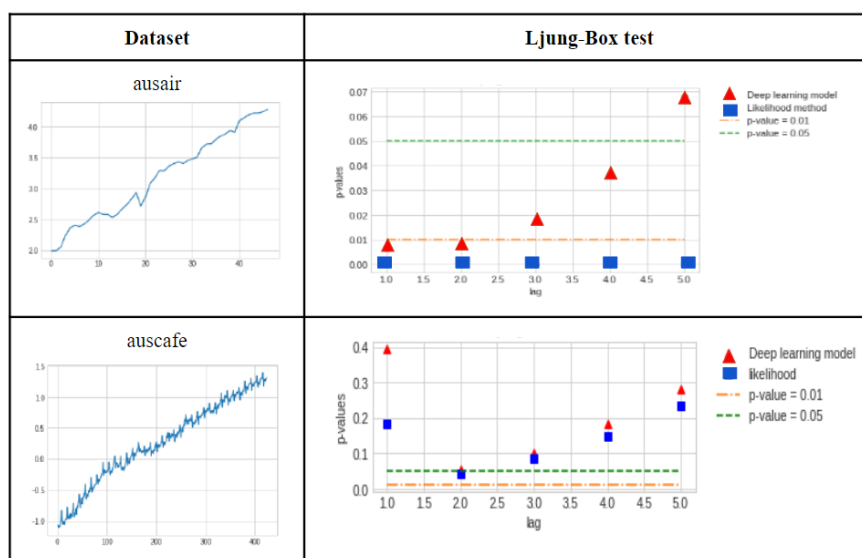
In the experimental results of the SIRO model from the testing data, it outperforms the likelihood-based method and the previous deep learning, ResNet50, in terms of Precision, Recall and F1-score which is better when a number of input channels increase. It is suggested that the SIRO model can extract the features in ACF, PACF or series images which are difficult for analysts to identify the ARIMA order.

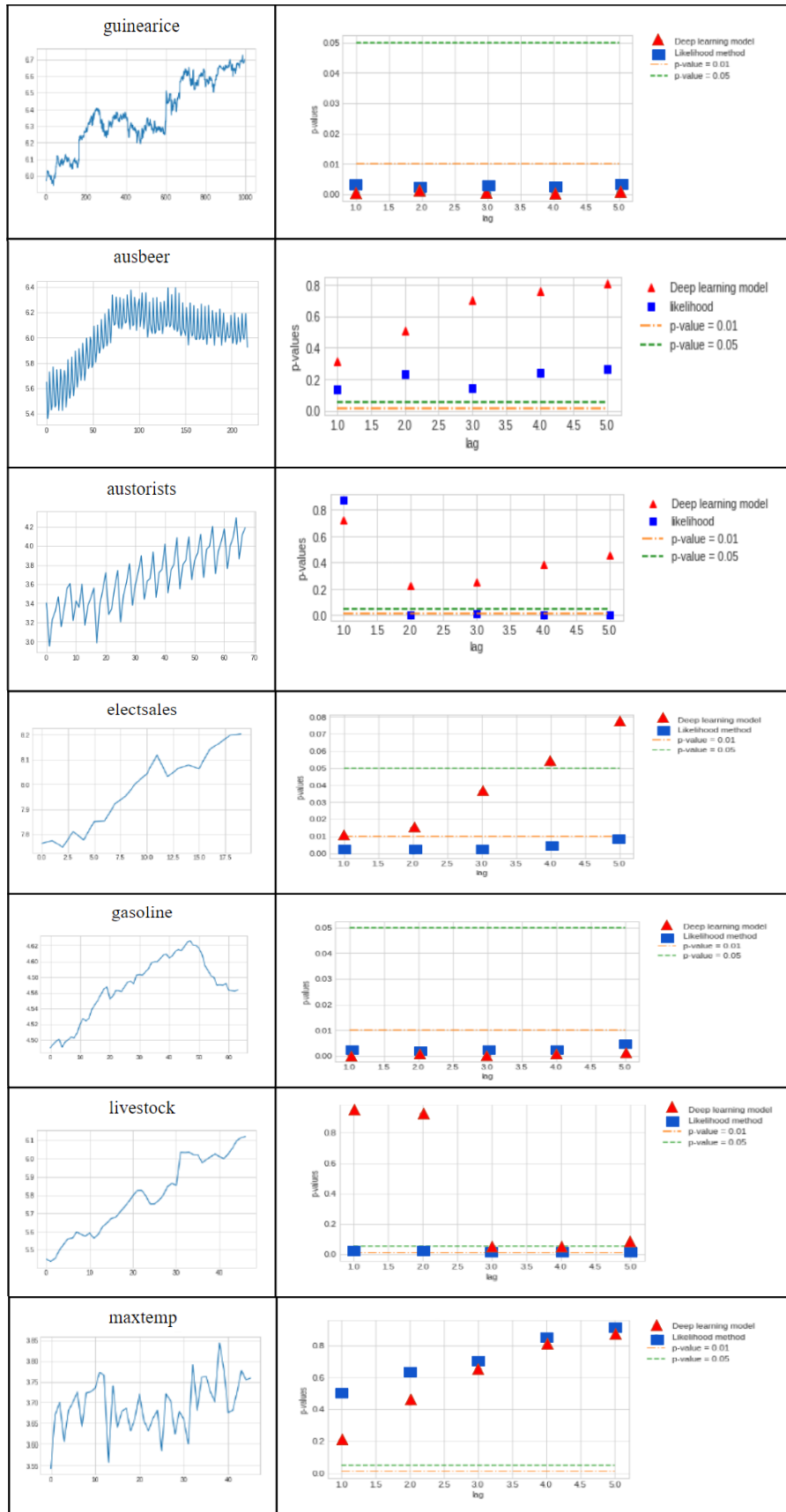
When comparing with the ResNet50 using the time series as inputs directly, the results show that changing the time series into ACF images, PACF images or differencing time series images can be used to predict orders better than using time series as direct input. The use of time series as direct input may not be able to predict orders effectively because the time series is disturbed by noises which the use of pure time series is difficult to find the true relationship of the time series data. Moreover, these results show that the identifying ARIMA order by using the tools of time series analysis is still more efficient than using the likelihood-based methods because ACF, PACF and differencing can exhibit the correlation within time series data whereas the likelihood-based methods choose the model by considering only minimum criteria which it does not show the correlations between values in the time series according to the time series analysis.

8.2 Conclusion and discussion of the SIRA algorithm

Finally, the SIRA algorithm is adapted with the real world datasets from the “fpp” package in R to forecast the future values and ensure the reliability of the SIRA algorithm. The results show that the SIRA algorithm can apply to the real world dataset like the likelihood-based method in terms of MAE, RMSE, MAPE and SMAPE and the Ljung-Box test, though there are some time series which both methods fail. For the future work, the inputs of the deep learning may be adjusted or the model for predicting the future values may be changed to apply with the time series having season or non-constant variance.

Table 6. The p -values for the Ljung-Box test.





References:

- [1] Faisal, F., Muhammad, P. M., & Tursoy, T. (2016). Impact of economic growth, foreign direct investment and financial development on stock prices in China: Empirical evidence from time series analysis. *International Journal of Economics and Financial Issues*, 6(4), 1998-2006.
- [2] Puchalsky, W., Ribeiro, G. T., da Veiga, C. P., Freire, R. Z., & dos Santos Coelho, L. (2018). Agribusiness time series forecasting using Wavelet neural networks and metaheuristic optimization: An analysis of the soybean sack price and perishable products demand. *International Journal of Production Economics*, 203, 174-189.
- [3] Sahu, P. K., & Kumar, R. (2013). Demand forecasting for sales of milk product (Paneer) in Chhattisgarh. *International Journal of Inventive Engineering and Sciences*, 1(9), 10-13.
- [4] Harvey, A. C. (2013). *Dynamic models for volatility and heavy tails: with applications to financial and economic time series* (Vol. 52). Cambridge University Press.
- [5] Schumacher, C. (2010). Factor forecasting using international targeted predictors: The case of German GDP. *Economics Letters*, 107(2), 95-98.
- [6] Box, G. E., & Jenkins, G. M. (1970). Time series analysis: *Forecasting and control*. Holden-Day. San Francisco, 498.
- [7] 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.
- [8] 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.
- [9] Wei, W. W. (2006). Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology*: Vol. 2.
- [10] Akaike, H. (1987). Factor analysis and AIC. In *Selected papers of hirotugu akaike*. Springer, New York, NY, pp. 371-386
- [11] Neath, A. A., & Cavanaugh, J. E. (2012). The Bayesian information criterion: background, derivation, and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 4(2), 199-203.
- [12] Karlsson, P. S., Behrenz, L., & Shukur, G. (2019). Performances of Model Selection Criteria When Variables are Ill Conditioned. *Computational Economics*, 54(1), 77-98.
- [13] Chen, P., Pedersen, T., Bak-Jensen, B., & Chen, Z. (2009). ARIMA-based time series model of stochastic wind power generation. *IEEE transactions on power systems*, 25(2), 667-676.
- [14] Faruk, D. Ö. (2010). A hybrid neural network and ARIMA model for water quality time series prediction. *Engineering applications of artificial intelligence*, 23(4), 586-594.
- [15] Alsharif, M. H., Younes, M. K., & Kim, J. (2019). Time series ARIMA model for prediction of daily and monthly average global solar radiation: the case study of Seoul, South Korea. *Symmetry*, 11(2), 240.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] Amini, M. H., Kargarian, A., & Karabasoglu, O. (2016). ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electric Power Systems Research*, 140, 378-390.
- [20] Guarnaccia, C., Quartieri, J., & Tepedino, C. (2017, June). Deterministic decomposition and seasonal ARIMA time series models applied to airport noise forecasting. In *AIP Conference Proceedings* (Vol. 1836, No. 1, p. 020079). AIP Publishing LLC.
- [21] Fukuoka, R., Suzuki, H., Kitajima, T., Kuwahara, A., & Yasuno, T. (2018). Wind Speed Prediction Model Using LSTM and 1D-CNN. *Journal of Signal Processing*, 22(4), 207-210.
- [22] 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).
- [23] Tang, W. H., & Röllin, A. (2018). Model identification for ARMA time series through convolutional neural networks. *arXiv preprint arXiv:1804.04299*.

- [24] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *2009 IEEE conference on computer vision and pattern recognition*(2009): 248-255.
- [25] Aloysius, Neena, and M. Geetha. "A review on deep convolutional neural networks." *2017 International Conference on Communication and Signal Processing (ICCSP)*(2017): 0588-0592.
- [26] Modi, Arpit Sunilkumar. "Review Article on Deep Learning Approaches." *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*(2018): 1635-1639.
- [27] Aloysius, Neena, and M. Geetha. "A review on deep convolutional neural networks." *2017 International Conference on Communication and Signal Processing (ICCSP)*(2017): 0588-0592.
- [28] Sahiner, Berkman, et al. "Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images." *IEEE transactions on Medical Imaging* 15.5 (1996): 598-610.
- [29] Hyndman, Rob J., and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.