TẠP CHÍ KHOA HỌC
TRƯỜNG ĐẠI HỌC QUY NHƠN

Dự báo lượng mưa hàng tháng của Quy Nhơn bằng mô hình Sarima

Nguyễn Quốc Dương[1,*], Lê Phương Thảo[1], Đinh Thị Quỳnh Như[1], Cao Thị Ái Loan[2], Phùng Thị Hồng Diễm[2], Lê Thanh Bính[2]

[1]Khoa Sư phạm, Trường Đại học Quy Nhơn, Việt Nam
[2]Khoa Toán và Thống kê, Trường Đại học Quy Nhơn, Việt Nam

Ngày nhận bài: 07/02/2020; Ngày nhận đăng: 04/04/2020

TÓM TẮT

Mô hình SARIMA được sử dụng rộng rãi để phân tích dữ liệu chuỗi thời gian. Trong bài báo này, chúng tôi sẽ áp dụng phương pháp này với sự trợ giúp của phần mềm thống kê R để dự báo lượng mưa hàng tháng tại thành phố Quy Nhơn, tỉnh Bình Định, Việt Nam. Dữ liệu được thu thập từ tháng 1 năm 2000 đến tháng 12 năm 2018 để thiết lập mô hình và dự báo.

Keywords: *ARIMA, phương pháp Box-Jenkins, SARIMA, chuỗi thời gian, lượng mưa Quy Nhơn.*

[*] *Tác giả liên hệ chính.*
Email: nguyenquocduongqnu1999@gmail.com

JOURNAL OF SCIENCE
QUY NHON UNIVERSITY

Monthly rainfall forecast of Quy Nhon using SARIMA model

Nguyen Quoc Duong[1,*], Le Phuong Thao[1], Dinh Thi Quynh Nhu[1], Cao Thi Ai Loan[2], Phung Thi Hong Diem[2], Le Thanh Binh[2]

[1]*Faculty of Education, Quy Nhon University, Viet Nam*
[2]*Faculty of Mathematics and Statistics, Quy Nhon University, Viet Nam*

Received: 07/02/2020; Accepted: 04/04/2020

ABSTRACT

The SARIMA model is widely used to analyze time series data. In this paper, we will apply this method with the help of R statistical software to forecast monthly rainfall in Quy Nhon city, Binh Dinh Province, Viet Nam. Mean monthly rainfall from 2000 to 2018 were used for modelling and forecasting.

Keywords: *ARIMA, Box-Jenkins methodology, SARIMA, time series, Quy Nhon rainfall.*

1. INTRODUCTION

Quy Nhon is a coastal city in Binh Dinh Province in central Vietnam. Here, weather is characterized by tropical monsoon climate with two distinct seasons, rainy season and dry season. Rainy season lasts from September to December, while dry season runs from January to August.

Changes in rainfall precipitation will be one of the most critical factors determining the overall impact of climate change. Therefore, its model-ling and forecasting are needed for water resources management, irrigation scheduling, agricultural management and reservoir operation, tourism. Prediction of rainfall is tough due to its non linear pattern and a large variation in intensity. Till today, numerous techniques have been used to forecast rainfall. Among them, Autoregressive Integrated Moving Average (ARIMA) modeling, introduced by Box and Jenkins is an effective method.[1] The Box-Jenkins Seasonal ARIMA (SARIMA) model has several advantages over other models, parti-cularly over exponential smoothing and neural network, due to its forecasting capability and richer information on time related changes.[2] ARIMA model considers the serial correlation, which is the most important characteristic of time series data, and also provides a systematic option to identify a better model. Another advantage of ARIMA model is that the model uses less parameter to describe a time series. Therefore, we use the SARIMA model to predict rainfall in Quy Nhon.

Besides mathematical, software tools today also play an important role in forecasting. There are many software tools for highly effective data analysis such as SPSS, Eviews, Python, etc. In this study, we use R statistical software to analyze the rainfall data. The advantages of R programming are open source programming language, providing exemplary support for data organization, package arrays, quality plotting and graphing, highly compatible, platform independent reporting and machine learning activity.

[*]*Corresponding author.*
Email: *nguyenquocduongqnu1999@gmail.com*

The rainfall data in Quy Nhon are obtained from General Statistics Office of Viet Nam (https://www.gso.gov.vn), and cover monthly observations from 2000 to 2018. We will proceed to build the appropriate forecasting model and compare evaluation between actual data and forecast data.

2. METHODOLOGY

2.1. The seasonal ARIMA model

Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal comp onent. SARIMA(p, d, q)(P, D, Q)$_s$ model can be most succinctly expressed using the backward shift operator:

$$\Phi_P(B^s)\phi_P(B)(1-B^s)^D(1-B)^d y_t \\ = c + \Theta_Q(B^s)\theta_q(B)\varepsilon_t,$$

where Φ_p, ϕ_p, Θ_Q and θq are polynomials of orders P, p, Q, and q, respectively. The parameter p and q define the order of the AR and MA processes with its non-seasonal lags, respectively, and d defines the degree of differencing of the series with its non-seasonal lags. Likewise, the P and Q parameters represent the corresponding order of the seasonal AR and MA processes of the series with its seasonal lags, and D defines the degree of differencing of the series with its non-seasonal lags. In general, the model is non-stationary, although if $D = d = 0$ and the roots of the characteristic equation all exceed unity in absolute value, the resulting model would be stationary.

2.2. Auto-Correlation Function (ACF) and Partial Auto-Correlation Function (PACF)

ACF is an (complete) auto-correlation function which gives us values of auto-correlation of any series with its lagged values. We plot these values along with the confidence band and tada! We have an ACF plot. In simple terms, it describes how well the present value of the series is related with its past values. A time series can have components like trend, seasonality, cyclic and residual. ACF considers all these com-ponents while finding correlations hence it is a 'complete auto-correlation plot'.[3]

PACF is a partial auto-correlation function. Basically instead of finding correlations of present with lags like ACF, it finds correlation of the residuals (which remains after removing the effects which are already explained by the earlier lag(s)) with the next lag value hence 'partial' and not 'complete' as we remove already found variations before we find the next correlation. So if there is any hidden information in the residual which can be modeled by the next lag, we might get a good correlation and we will keep that next lag as a feature while modeling. Remember while modeling we do not want to keep too many features which are correlated as that can create multicollinearity issues. Hence we need to retain only the relevant features.[3]

2.3. Modelling procedure

When fitting an SARIMA model to a set of time series data, the following procedure provides a useful general approach.

❖ Step 1: Data preparation: Here, we prepare the data for the training and testing process of the model. This step includes splitting the series into training (in-sample) and testing (out-sample) partitions, creating new features (when applicable), and applying a transformation if needed (for example, log transformation, scaling, and so on).

❖ Step 2: Train the model: Here, we used the training partition to train a statistical model. The main goal of this step is to utilize the training set to train, tune, and estimate the model coefficients that minimize the selected error criteria. The fitted values and the model estimation of the training partition observations will be used later on to evaluate the overall performance of the model.

❖ Step 3: Test the model: Here, we utilize the trained model to forecast the corresponding observations of the testing partition. The main idea here is to evaluate the performance of the model with a new dataset (that the model did not see during the training process).

❖ Step 4: Model evaluation: Last but not least, after the model was trained and tested, it is time to evaluate the overall performance of the model on both the training and testing partitions.[4]

2.4. Tuning the SARIMA model

The tuning process of the SARIMA model follows the same logic as one of the ARIMA models. However, the complexity of the model increases as there are now six parameters to tune, that is, p, d, q, P, D and Q, as opposed to three with the ARIMA model. Luckily, the tuning of the P, D, and Q seasonal parameters follows the same logic as the ones of p, d, q, respectively, with the use of the ACF and PACF plots. The main difference between the tuning of these two groups of parameters (non-seasonal and seasonal) is that the non seasonal parameters are tuned with the non-seasonal lags, as we saw previously with the ARIMA model. On the other hand, the tuning of the seasonal parameters are tuned with the seasonal lags (for example, for monthly series with lags 12, 24, 36, and so on).[4,5]

2.4.1. Tuning the non-seasonal parameters

Applying the same logic that we used with the ARIMA model, tuning the non-seasonal parameters of the SARIMA model is based on the ACF and PACF plots:

❖ An AR(p) process should be used if the non-seasonal lags of the ACF plot are tailing off, while the corresponding lags of the PACF plots are cutting off on the p lag.

❖ Similarly, an MA(q) process should be used if the non-seasonal lags of the ACF plot are cutting off on the q lag and the corre-sponding lags of the PACF plots are tailing off.

❖ When both the ACF and PACF non-seasonal lags are tailing off, an ARMA model should be used.

❖ Differencing the series with the non-seasonal lags should be applied when the non-seasonal lags of the ACF plot are decaying in a linear manner.[4]

2.4.2. Tuning the seasonal parameters

Tuning the seasonal parameters of the SARIMA model with ACF and PACF follows the same guidelines as the ones we used for selecting the ARIMA parameters:

❖ We will use a seasonal autoregressive process with an order of P, or SAR(P), if the seasonal lags of the ACF plot are tailing off and the seasonal lags of the PACF plot are cutting off by the P seasonal lag.

❖ Similarly, we will apply a seasonal moving average process with an order of Q, or SMA(Q), if the seasonal lags of the ACF plot are cutting off by the Q seasonal lag and the seasonal lags of the PACF plot are tailing off.

❖ An ARMA model should be used whenever the seasonal lags of both the ACF and PACF plots are tailing off.

❖ Seasonal differencing should be applied if the correlation of the seasonal lags are decaying in a linear manner.[6]

2.5. A step-wise procedure for traversing the model space

Suppose that we have seasonal data, we consider ARIMA(p, d, q)(P, D, Q)$_s$ models, where p and q can take values from 0 to 3, and P and Q can take values from 0 to 1. When c = 0 there is a total of 288 possible models, and when c^{1} 0 there is a total of 192 possible models, giving 480 models altogether. If the values of p, d, q, P, D and Q are allowed to range more widely, the number of possible models increases rapidly. If d and D are known, we can select the orders p, q, P and Q via an information criterion such as the AIC:

$$AIC := -2\log(L) + 2(p + q + P + Q + k),$$

where k = 1 if C ≠ 0 and 0 otherwise, and L is the maximized likelihood of the model fitted to the differenced data $(1 - B^s)^D$ $(1 - B)^d$ y_t. The likelihood of the full model for y_t is not actually defined and so the value of the AIC for different levels of differencing are not comparable.[5]

Consequently, it is often not feasible to simply fit every potential model and choose the one with the lowest AIC. Instead, we need a way of traversing the space of models efficiently in order to arrive at the model with the lowest AIC value. We propose a step-wise algorithm as follows.

Step 1: We try four possible models to start with.

❖ ARIMA(2, d, 2) if $s = 1$ and ARIMA(2, d, 2)(1, D, 1) if $s > 1$;

❖ ARIMA(0, d, 0) if $s = 1$ and ARIMA(0, d, 0)(0, D, 0) if $s > 1$;

❖ ARIMA(1, d, 0) if $s = 1$ and ARIMA(1, d, 0)(1, D, 0) if $s > 1$;

❖ ARIMA(0, d, 1) if $s = 1$ and ARIMA(0, d, 1)(0, D, 1) if $s > 1$.

If $d + D \leq 1$, these models are fitted with $C \neq 0$. Otherwise, we set $c = 0$. Of these four models, we select the one with the smallest AIC value. This is called the 'current' model and is denoted by ARIMA-(p, d, q) if $s = 1$ or ARIMA(p, d, q)(P, D, Q)$_s$ if $s > 1$.

Step 2: We consider up to 13 variations on the current model:

❖ where one of p, q, P and Q is allowed to vary by ±1 from the current model;

❖ where p and q both vary by ±1 from the current model;

❖ where P and Q both vary by ±1 from the current model;

❖ where the constant c is included if the current model has $c = 0$ or excluded if the current model has $C \neq 0$.

Whenever a model with lower AIC is found, it becomes the new 'current' model and the procedure is repeated. This process finishes when we cannot find a model close to the current model with lower AIC.

There are several constraints on the fitted models to avoid problems with convergence or near unit roots. The constraints are outlined below:

❖ The values of p and q are not allowed to exceed specified upper bounds (with default values of 5 in each case).

❖ The values of P and Q are not allowed to exceed specified upper bounds (with default values of 2 in each case).

❖ We reject any model which is 'close' to non-invertible or non-causal. Specifi-cally, we compute the roots of $\phi(B)\Phi(B)$ and $\theta(B)\Theta(B)$. If either have a root that is smaller than 1.001 in absolute value, the model is rejected.

❖ If there are any errors arising in the non-linear optimization routine used for estimation, the model is rejected. The rationale here is that any model that is difficult to fit is probably not a good model for the data.

The algorithm is guaranteed to return a valid model because the model space is finite and at least one of the starting models will be accepted (the model with no AR or MA parameters). The selected model is used to produce forecasts.[5,6]

2.6. Forecast evaluation methods

Once you finalize the model tuning, it is time to test the ability of the model to predict observations that the model did not see before (as opposed to the fitted values that the model saw throughout the training process). The most common method for evaluating the forecast's success is to predict the actual values with the use of an error metric to quantify the forecast's overall accuracy. The selection of a specific error metric depends on the forecast accuracy's goals. This study only considers common error metric is as follow:

Root Mean Squared Error (RMSE): This is the root of the average squared distance of the actual and forecasted values:

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2},$$

where Y_t and $\hat{Y}_t$ are the actual value of the original series and predicted value from the proposed hybrid model, respectively. The smallest value of RMSE indicates the best model.[4]

3. RESULTS AND DISCUSSION

The data has been collected at the Quy Nhon station from 2000 to 2018. We have a monthly series with 228 observations (19 years) and the goal is to forecast the next two years (24 months). The corresponding command packages and libraries for model prediction are *forecast, readxl, tseries, TSstudio*.[6-8] Let us load the rainfall series from file *datats.xlsx*.

Let us plot the series with the ts_plot function and review the main characteristics of the series by the code below:

```
#plot time series data with
ts_plot function
ts_plot(datats, title=
"Monthly Rainfall in Quy
Nhon", Ytitle="Rainfall(mm)",
Xtitle="Year")
```

We attain the output as shown in Figure 1:

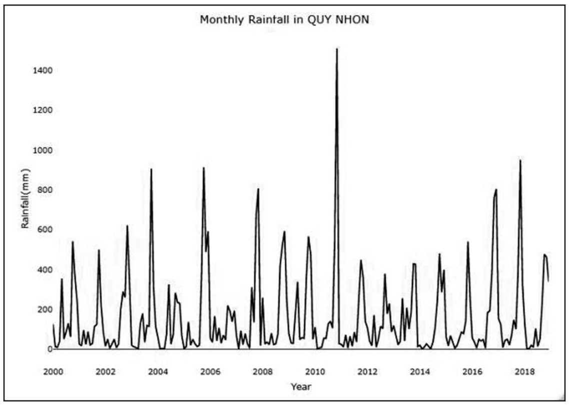

Figure 1. Time series plot of observed mean monthly rainfall from 2000 to 2018.

From Figure 1, the *datats* series has a strong seasonal pattern, therefore the SARIMA model is the most appropriate one for our data. In addition, the series is trending up, so we can already conclude that the series is not stationary and some differencing of the series is required. We would use the first 216 observations for training and test the performance using the last 12 observations. Creating partitions in R can be done manually with the *ts_split* function from the stats package. For instance, let is split the *mydata* series into partitions, leaving the last 12 observations of the series as the testing partition and the rest as training:

```
QN_rainfall<-ts_split(datats,
sample.out = 12)
train <- QN_rainfall$train
test <- QN_rainfall$test
```

In statistics, an *augmented Dickey–Fuller (ADF) test* the null hypothesis that a unit root is present in a time series sample. The alternative hypothesis is different depending on which version of the test is used, but is usually stationarity or trend stationarity. We obtain the following output:

```
adf.test(train, alternative =
"stationary")
##Dickey-Fuller = -8.0188,
Lag order = 5, p-value = 0.01
##alternative hypothesis:
stationary
```

Before we start the training process of the SARIMA model, we will conduct diagnostics in regards to the series correlation with the ACF and PACF functions. Since we are interested in viewing the relationship of the series with its seasonal lags, we will increase the number of lags to calculate.

```
par(mfrow=c(1,2))
acf(ts(train),main="ACF For
Rainfall", col="blue",lwd = 4)
pacf(ts(train),main="PACF For
Rainfall",col="coral",lwd = 4)
```

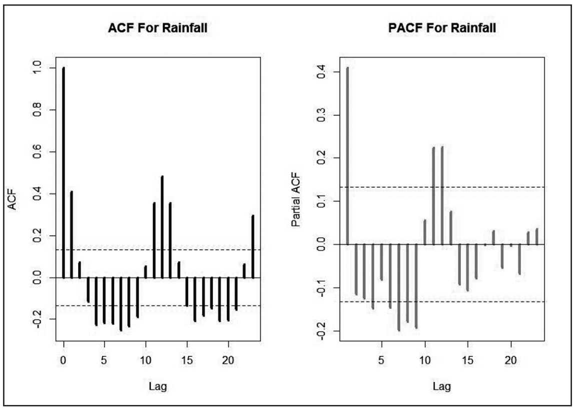

Figure 2. ACF and PACF plot of rainfall for Quy Nhon station.

The preceding ACF plot indicates that the series has a strong correlation with both the seasonal and non-seasonal lags. Furthermore, the linear decay of the seasonal lags indicates that the series is not stationary and that seasonal differencing is required. We will start with a seasonal differencing of the series and plot the output to identify whether the series is in a stationary state. The R commans and output are as follows:

```
ndiffs(train) #to determine d
(the number of seasonal
differences to use)
##0
nsdiffs(train)#to determine D
(the number of ordinary
differences to use)
##1
QN_rainfall_12 <- diff(train,
lag = 12, differences = 1)
ts_plot(QN_rainfall_12, title =
"Quy Nhon Monthly Rainfall -
First Seasonal
Difference",Ytitle = Rainfall
(First Difference)",Xtitle =
"Year")
```

By the *ts_plot* function, we get the output as Figure 3:

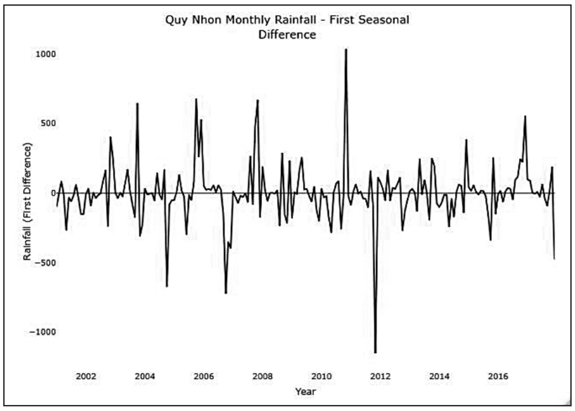

Figure 3. Plot of seasonal differencing of the series.

After taking the first order differencing, along with the first order seasonal differencing, the series seems to stabilize around the zero x axis line (or fairly close to being stable). After transforming the series into a stationary state, we can review the ACF and PACF functions again to identify the required process:

```
par(mfrow=c(1,2))
acf (ts(QN_rainfall_12), main=
"ACF For First Seasonal
Difference", col="blue", lwd = 4)
pacf(ts(QN_rainfall_12), main=
"ACF For First Seasonal
Difference", col="coral", lwd = 4)
```

It should note that this step is very important. We need to the observe from the ACF and PACF plots in order to choose simple models.

The output of the commans above is shown in Figure 4:

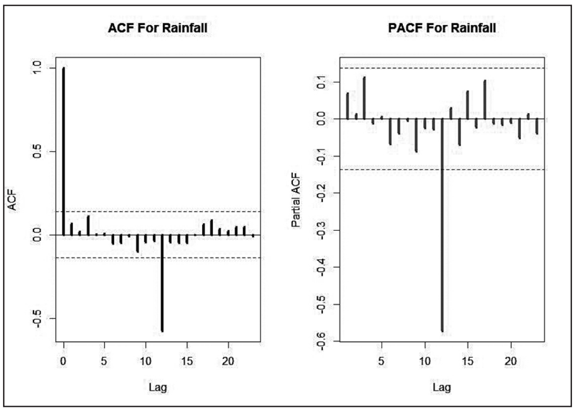

Figure 4. ACF and PACF plot of rainfall for Quy Nhon station.

The main observation from the preceding ACF and PACF plots is that both the nonseasonal and seasonal lags (in both plots) are tailing off. Hence, we can conclude that after we difference the series and transform them into a stationary state, we should apply an ARMA process for both the seasonal and non-seasonal components of the SARIMA model. Our aim now is to find an appropriate ARIMA model based on the ACF and PACF plot. The significant spike at lag 1 in the ACF suggests a non-seasonal MA(1) component, and the sign-sifycant spike at lag 12 in the ACF suggests a seasonal MA(1) component. Consequently, we begin with an SARIMA(1,0,0)(1,1,0)$_{12}$ model, indicating seasonal difference, and non-seasonal and seasonal MA(1) components. Based on the PACF plot, we can choose another simple model SARIMA(0,0,0)(0,1,0)$_{12}$. Of these two models, the best is the SARIMA model (i.e., it has the smallest AIC value).

```
md1<- Arima(train,
order=c(1,0,0),
seasonal=c(1,1,0))
summary(md1)
##AIC: 2683.2
md2<- Arima(train,
order=c(0,0,0),
seasonal=c(0,1,0))
summary(md2)
##AIC: 2762.83
```

Consequently, this initial analysis suggests that a possible model for these data is an SARIMA(1,0,0)(1,1,0)$_{12}$. We fit this model, along with some variations on it, compute the AIC values and test set evaluation shown as in Table 1.

Table 1. AIC and RMSE values for various SARIMA models applied to the monthly rainfall data.

Models	AIC	RMSE
SARIMA(1,0,0)(1,1,0)$_{12}$	2683.20	164.4443
SARIMA(0,0,0)(1,1,0)$_{12}$	2682.27	164.8799
SARIMA(2,0,0)(1,1,0)$_{12}$	2684.69	164.2065
SARIMA(1,0,1)(1,1,0)$_{12}$	2681.33	162.7866
SARIMA(1,0,0)(0,1,0)$_{12}$	2763.84	203.7255
SARIMA(1,0,0)(2,1,0)$_{12}$	2658.01	152.5503
SARIMA(1,0,0)(1,1,1)$_{12}$	2650.34	146.6646
SARIMA(2,0,1)(1,1,0)$_{12}$	2683.32	162.7785
SARIMA(1,0,0)(2,1,1)$_{12}$	2649.60	145.6458

Of these models, the best is the SARIMA-(1,0,0)(2,1,1)$_{12}$ model (which has the lowest RMSE value on the training set, and the best AIC value amongst models with only seasonal differencing). Before we finalize the forecast, let's evaluate the selected model's performance on the testing set. We will retrain the model using the settings of the selected model:

```
QNrainfall_best_md <-
Arima(train, order = c(1,0,0),
seasonal =list(order=
c(2,1,1)))
```

We then get the output as described in Table 2:

Table 2. Summary of SARIMA(1,0,0)(2,1,1)$_{12}$.

SARIMA(1,0,0)(2,1,1)$_{12}$				
SARIMA	*ar1*	*sar2*	*sar2*	*sma1*
Coefficients	0.067	-0.052	0.1	-0.881
s.e.	0.068	0.092	0.087	0.093
AIC = 2650.34				

Let us use the QNrainfall_best_md trained model to forecast the corresponding observations of the testing set:

```
QNrainfall_test_fc <- forecast(QNrainfall_
best_md, h = 12)
```

We will assess the model's performance by put

$$H = \frac{Actual\ Value - Forecast\ Value}{Actual\ Value}$$

and these values are calculated as in Table 3:

Table 3. Assess performances of SARIMA(1,0,0) - $(2,1,1)_{12}$ model.

Month	Forecast Value	Actual Value	H
January	*81.81485*	*128.6*	*0.36*
February	*32.32366*	*2.8*	*-10.54*
March	*34.07819*	*1.6*	*-20.29*
April	*46.95980*	*20.0*	*-1.34*
May	*82.40952*	*9.4*	*-7.76*
June	*44.63459*	*103.7*	*0.56*
July	*57.38414*	*14.0*	*-3.09*
August	*122.92861*	*51.1*	*-1.40*
September	*209.18078*	*235.5*	*0.11*
October	*420.49951*	*476.7*	*0.11*
November	*574.92124*	*462.0*	*-0.24*
December	*321.89014*	*337.9*	*0.04*

From Table 3, we see the predicted model has two values big deviations in February and March. This is unavoidable because climate change is very complex. However, the forecast value and actual value of the rainy season from September to December for very low error. Moreover, these predicted values are very suitable for the climate characteristics of Quy Nhon city.

Now, we will use the test_forecast function to get a more intuitive view of the model is performance on the training and testing partitions:

```
test_forecast(datats,forecast.
obj = QNrainfall_test_fc,test
= test)
```

We then have the output as shown in Figure 5:

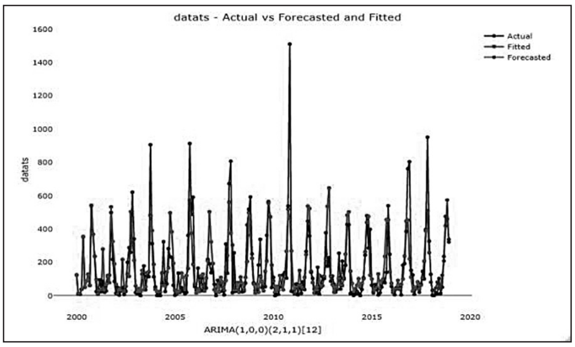

Figure 5. *Plot of datats – actual & forecasted an fited.*

Now that we have satisfied the preceding conditions, we can move on to the last step of the forecasting process and generate the final forecast with the selected model. We will start by retraining the selected model on all the series:

```
final_md <- Arima(datats,
order = c(1,0,0), seasonal =
list(order=c(2,1,1)))
```

Before we forecast the next 24 months, let is verify that the residuals of the model satisfy the model condition. We execute the code as follows:

```
checkresiduals(final_md)
```

The output is as Figure 6:

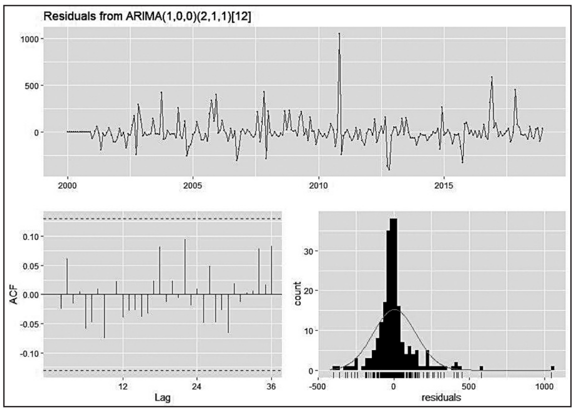

Figure 6. Residuals from the SARIMA(1,0,0)-$(2,1,1)_{12}$ model applied to monthly rainfall data.

The output of the Ljung-Box test suggested that the residuals of the model are white noise:

```
## Ljung-Box test
## data: Residuals from
ARIMA(1,0,0)(2,1,1)[12]
## Q* = 9.7273,
df = 20,
p-value= 0.9729
## Model df:4.
## Total lags used: 24
```

By looking at the preceding residuals plot, you can see that the residuals are white noise and normally distributed. Furthermore, the Ljung-Box test confirms that there is no autocorrelation left on the residuals with a p-value of 0.9729, we cannot reject the null hypothesis that the residuals are white noise. Thus, we now have a seasonal ARIMA model that passes the required checks and is ready for forecasting.

The main goal of the forecasting process is to minimize the level of uncertainty around the future values of the series. Although we cannot completely eliminate this uncertainty, we can quantify it and provide some range around the point estimate of the forecast. The confidence interval is a statistical approximation method that's used to express the range of possible values that contain the true value with some degree of confidence (or probability). We now use the *forecast* function to obtain the predicted values for the next 24 months of the data series.

```
QNrainfall_fc <-
forecast(final_md, h = 24)
QNrainfall_fc
```

Forecast package is written by Rob J Hyndman and is available from CRAN here. The R package forecast provides methods and tools for displaying and analysing univariate time series forecasts including exponential smoothing via state space models and automatic ARIMA modelling.

We obtain the result as described in Table 4:

Table 4. Forecasts of the mothly rainfall data using the SARIMA(1,0,0)(2,1,1)$_{12}$ model with 80% and 95% confidence intervals.

Point	Forecast	Lo 80	Hi 80	Lo 95	Hi 95
Jan 2019	91.76578	-96.60397	280.1355	-196.32090	379.8525
Feb 2019	42.27684	-146.48860	231.0423	-246.41498	330.9687
Mar 2019	33.84519	-154.92190	222.6123	-254.84917	322.5396
Apr 2019	43.94335	-144.82376	232.7105	-244.75103	332.6377
May 2019	83.59559	-105.17151	272.3627	-205.09878	372.2900
Jun 2019	46.60968	-142.15743	235.3768	-242.08470	335.3041
Jul 2019	61.58513	-127.18198	250.3522	-227.10925	350.2795
Aug 2019	116.15631	-72.61080	304.9234	-172.53807	404.8507
Sep 2019	198.88576	10.11865	387.6529	-89.80862	487.5801
Oct 2019	434.81604	246.04893	623.5831	146.12166	723.5104
Nov 2019	592.95521	404.18811	781.7223	304.26084	881.6496
Dec 2019	260.07814	71.31136	448.8449	-28.61573	548.7720
Jan 2020	89.25500	-99.61077	278.1208	-199.59027	378.1003
Feb 2020	27.27827	-161.58825	216.1448	-261.56814	316.1247
Mar 2020	31.56637	-157.30015	220.4329	-257.28005	320.4128
Apr 2020	40.21528	-148.65124	229.0818	-248.63113	329.0617
May 2020	75.65715	-113.20936	264.5237	-213.18926	364.5036
Jun 2020	58.20946	-130.65706	247.0760	-230.63695	347.0559
Jul 2020	53.28597	-135.58055	242.1525	-235.56044	342.1324
Aug 2020	102.82699	-86.03953	291.6935	-186.01943	391.6734
Sep 2020	214.99832	26.13180	403.8648	-73.84809	503.8447
Oct 2020	445.11325	256.24673	633.9798	156.26684	733.9597
Nov 2020	534.54059	345.67407	723.4071	245.69418	823.3870
Dec 2020	265.01471	76.14849	453.8809	-23.83125	553.8607

We can plot historical data with forecasts and confidence intervals by the *plot_forecast* function:

```
plot_forecast(QNrainfall_fc,title = "Quy Nhon Rainfall Forecast",
Ytitle = "Rainfall(mm)",Xtitle = "Year")
```

We then obtain the output as Figure 7:

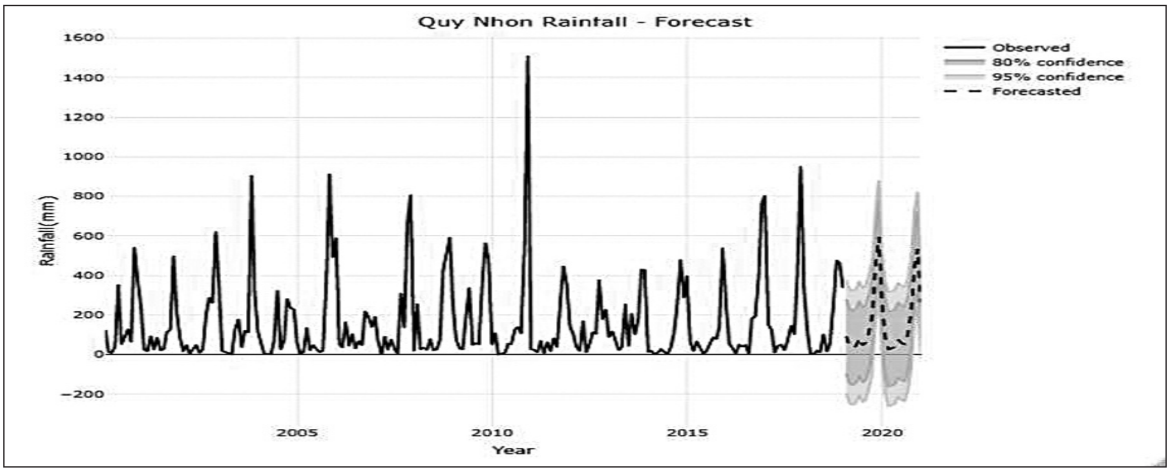

Figure 7. Forecasts of the mothly rainfall data using the ARIMA(1,0,0)(2,1,1)$_{12}$ model with 80% and 95% confidence intervals.

Note that the lower bounds are somewhat funny for rainfall. Is the earth going to sprinkle water back into the clouds? We explain this as follows: the simulation of the forecasts generating a family of forecasts for each period can simply be truncated at 0.0, which is meant to decouple the variance of the errors from the expected value of the model when one can safely ignore values lower than 0.0.

4. CONCLUSIONS

In this study, we used the SARIMA model for forecasting monthly rainfall data of Quy Nhon city. Based on seasonally differenced correlogram characteristics, different SARIMA models were evaluated; their parameters were optimized, and diagnostic check up of forecasts was made by using white noise and heter-oscedasticity tests. The best SARIMA model (corresponding to our data) was chosen based on smallest value of AIC and RMSE. A validation check was performed on residual series. Residuals was found white noise for SARIMA-(1,0,0)(2,1,1)$_{12}$ model. The predicted values from the model were compared with the actual values to determine prediction precision. We found that selected model predicted monthly rainfall with a reasonable accuracy. Therefore, year-long rainfall can be forecasted using these models. Moreover, this model can be applied in the study of the time series in similar fields at Quy Nhon or other cities.

Acknowledgement

This research is conducted within the framework of the student scientific research project for the academic year 2019 - 2020 under the project code S2019.564.01.

REFERENCES

1. G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control,* San Fran-cisco: Holden-Day, 1976.
2. A. K. Mishra and V. R. Desai. Drought forecasting using stochastic models, *Stochastic Environmental Research and Risk Assessment,* **2005**, *19*, 326-339.
3. J. Salvi. Significance of ACF and PACF Plots In Time Series Analysis, <https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8>, accessed 20/10/2019.
4. R. Krispin. *Hands-On Time Series Analysis with R,* Packt Publisher, 2019.
5. R. J. Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice, 2nd edition,* OTexts Publisher, 2018.

6. R. J. Hyndman and Y. Khandakar. Automatic Time Series Forecasting: The forecast Package for R*, Journal of Statistical Software*, **2008**, *27*(3), 1-22.

7. C. Chatfield. *The Analysis Of Time Series-An Introduction, 5th edition*, Chapman and Hall, 1996.

8. G. James, D. Witten, T. Hastie and R. Tibshirani. *An Introduction to Statistical Learning with Applications in R, 1st edition*, Springer Texts in Statistics, 2007.