Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Air Pressure Modeling with an ARIMA Time Series Model

M. Shams*[1] and M. A. Mirzaie†[1]

[1]Department of mathmatics, University of Kashan, Kashan, Iran

ABSTRACT

Air pressure is an important criterion for weather forecasting, and is also widely used in some branches of science. In this paper, we propose the ARIMA model for modeling air pressure at the Isfahan Airport meteorological station. In the next step, the model assumptions will be examined. Finally, we will show how well the model describes the data. For convenience, all **R** code used in the paper is included at the end of the paper in the Appendix section.

ARTICLE INFO

Article history:
Research paper
Received 02 July 2025
Accepted 16 July 2025
Available online 16 July 2025

Keyword: Akaike information criterion, time series model, forecasting, Augmented Dickey-Fuller test, Wald test.

AMS subject Classification: 62M10, 91B84, 62F10, 94A15.

1 Introduction

Weather forecasts are made by collecting statistical data about the current conditions of a given location using meteorology to predict how the weather will change in a given location. Weather forecasts have many applications in protecting people's lives and property, agriculture, and economics. In numerical weather forecasting, the state of the fluid must be sampled at a given time, and the future state of the fluid must be estimated using fluid dynamics and thermodynamic equations.

*Corresponding author: M. Shams. Email: mehdishams@kashanu.ac.ir

†Email: m.mirzaee@std.kashanu.ac.ir

Air pressure is the force exerted per unit area by the weight of the air above the surface on the Earth's surface. Since atmospheric pressure on Earth is highly variable, these changes are important in the study of weather and climate. Air pressure has a significant impact on weather patterns.

Changes in air pressure indicate approaching weather systems, and understanding these changes helps meteorologists predict future conditions. High-pressure systems are associated with clear, calm, and still air, while low-pressure systems tend to produce cloudy, windy, rainy, or snowy conditions. Because the flow of air from high-pressure areas to low-pressure areas creates wind, and the accumulation of air in low-pressure systems can lead to cloud formation and precipitation. Researchers use mathematical equations and statistical models to describe how pressure, temperature, density, and volume are related, and these equations are called the ideal gas laws.

Torricelli, following Galilei's research, first recognized that decreasing air pressure led to increasing clouds and precipitation, and increasing air pressure led to cloud dissolution and more sunlight. Shortly thereafter, the french physicist Descartes developed the paper scale so that longer series of observations could be recorded numerically.

Air pressure is an important measure for weather forecasting, and is also used in air transport, agriculture, NRM (Natural Resource Management), astronomical observation, geophysics, geodesy, etc.

Factors such as seasonality, economic fluctuations, unexpected events, and internal changes also affect the forecast. Classic time series models such as SES (Simple Exponential Smoothing), ARIMA (Autoregressive Integrated Moving Average), SARIMA (Seasonal ARIMA) and ARIMAX (ARIMA with explanatory variables) perform well for short-term forecasts, but are not recommended for long-term forecasts. Machine learning and deep learning-based algorithms are emerging approaches to predicting time series models. These approaches are based on artificial intelligence and move data analysis processes towards data-driven rather than model-driven. The accuracy of hourly air temperature forecasting is poor due to random variations and nonlinear relationships between temperature and other meteorological elements, such as air pressure and wind speed. To increase the accuracy, deep learning methods such as Support Vector Machines (SVM), Random Forests (RF), ANN (Artificial Neural Network), CNN (Convolutional Neural Networks), RNN (Recurrent Neural Networks), LSTM (Long Short-Term Memory) and BiLSTMs (Bidirectional LSTMs) are used. For example, suppose $\boldsymbol{x} = (x_1, \ldots, x_T)$ represents a sequence of length T, and h_t represents RNN memory at time step t, and RNN model updates its memory information using $h_t = \sigma(W_x x_t + W_h h_{t-1} + b_t)$ where σ is a nonlinear function (e.g., logistic sigmoid, a hyperbolic tangent function, ...), W_x and W_h are weight matrices that are used in deep learning model, and b_t is a constant bias. In LSTM models, the output of forget gate, i.e., f_t is a value between 0 and 1, where 0 means completely removing the learned value and 1 means keeping the entire value, and is expressed as follows $f_t = \sigma(W_{f_h}[h_{t-1}], W_{f_x}[x_t], b_f)$ where b_f is a constant bias value. Input gate consists of two layers: a sigmoid layer, i.e., $i_t = \sigma(W_{i_h}[h_{t-1}], W_{i_x}[x_t], b_i)$ (which decides what values should be updated) and a "tanh" layer, i.e., $\tilde{c}_t = \tanh(W_{c_h}[h_{t-1}], W_{c_x}[x_t], b_c)$ (which creates a vector of new candidate values that will be added to the LSTM memory).

The combination of these two layers is a suitable update for the LSTM memory, which is calculated as $c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$. We also use the following formulas for the output gate part:

$$o_t = \sigma(W_{o_h}[h_{t-1}], W_{o_x}[x_t], b_o),$$
$$h_t = o_t * \tanh(c_t)$$

where o_t is the output value, and $h_t \in [-1.1]$.

In [12], ARIMA, SARIMA, and LSTM models are used to predict profits for a time series model. The CNN method reduces the dimensionality of time series data, and the LSTM method records the long-term memory of temperature time series data. In [7], hourly air temperature prediction is performed based on the CNN-LSTM fusion method. In [8], temperature time series forecasting for a weather station in Ankara, Türkiye from January 2010 to March 2023 was performed using ARIMA and LSTM seasonal models. In [1], the models fitted with ARIMA and LSTM are compared in Mulkia Gulf real estate. In [2], time series modeling and forecasting of meteorological parameters on the West African coast are analyzed. In [9], using $ARIMA$ time series models, forecasting air quality and environmental data in the Salatiga region. In [10], using regression and $ARIMA$ time series models, the prediction of air quality index in Chennai has been analyzed.

In this paper, air pressure at the Isfahan Airport meteorological station modeled with the $ARIMA$ model. Observations were collected from 1402/1/1, 00:00:00 to 1402/12/29, 23:00 (from March 21, 2023 to March 19, 2024) at the Isfahan Airport meteorological station. Statistical comparisons and statistical fit were calculated for prediction purposes. The forecast was also reported in the next 24 hours. All model assumptions were controlled and the model follows its assumptions.

2 Autoregressive Integrated Moving Average

Consider a situation where we observe a random variable at different times and want to predict it in the future. For this purpose, we use a time series model like $\{X_t : t \geq 0\}$. Suppose $Z_t \sim N(0, \sigma^2)$ is the random component of the data. A random process $\{Z_t\}$, which is a sequence of uncorrelated variables, is also called white noise [4].

Let B is a backshift operator defined as $BX_t = X_{t-1}$. We define the first-order differencing operator as follows $\nabla X_t = X_t - X_{t-1} = (1 - B)X_t$. Also differences of order d are defined as $\nabla^d = (1 - B)^d$.

One of the time series models is the $ARIMA$ model. A stochastic processes $\{X_t : t \geq 0\}$ is said to be $ARIMA(p, d, q)$, if $\nabla^d X_t = (1 - B)^d X_t$ is $ARMA(p, q)$ model as

$$X_t = \delta + \sum\nolimits_{i=1}^{p} \phi_i X_{t-i} + \sum\nolimits_{j=1}^{q} \theta_j Z_{t-j} + Z_t$$

where δ, ϕ_i's and θ_j's are model parameters [3].

For $ARIMA$ model, ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) don't have closed forms, but we can graph these functions after differentiating them d times. After d times of differentiation, the model is converted to $ARMA(p, q)$ and the ACF and PACF of the model can be plotted. After fitting the model, the ACF and PACF of the data can be compared with the theoretical ACF and PACF of the model [4].

3 Time series modeling with air pressure data

The data are given in Table 1 (see Appendix 1).

id	DATE	Pressure (m bar)
1	1402/01/01 00:00:00	1009
2	1402/01/01 01:00:00	1008
3	1402/01/01 02:00:00	1008
4	1402/01/01 03:00:00	1007
5	1402/01/01 04:00:00	1008
⋮	⋮	⋮
8751	1402/12/29 23:00:00	1015

Table 1: Air pressure data

Our goal is to model this data based on the $ARIMA$ time series model and then forecast it. For this purpose, we need to answer the following two questions:

1. Is the variance of the data constant?

2. Is there a trend in the data?

We answer the first question about the stability of variance with the Box-Cox transformation [4]. The Box-Cox transformation is one of the most popular transformations for variance stabilization. The Box-Cox transformation is given as follows:

$$X^{(\lambda)} = \begin{cases} \frac{X^{\lambda}-1}{\lambda}, & \text{if } \lambda \neq 0 \\ \log X, & \text{if } \lambda = 0 \end{cases}$$

We know that the variance is constant ($\lambda = 1.24$ and rounded to $\lambda = 1$, see Appendix 2), and answer the second question with the ADF (Augmented Dickey-Fuller) [5] and Phillips–Perron tests [11]. The ADF test is designed to test the existence of a unit root in a time series model under the null hypothesis. This statistic is a negative quantity, and the more negative it is, the stronger the rejection of the unit root hypothesis. We reject the null hypothesis, meaning that the mean of the data is constant [(ADF statistic = -4.3692, order lag = 20, P-value < 0.01), (Phillips-Perron Unit Root Test: Dickey-Fuller Z(alpha) = -139.98, Truncation lag parameter = 12, P-value < 0.01)] (see Appendix 2 for more details). The Dickey–Fuller test involves fitting the regression model with ordinary least squares (OLS), say $y_t = \alpha + \rho y_{t-1} + \delta t + u_t$, but the Phillips-Perron test involves fitting the regression $y_t = \alpha + \rho y_{t-1} + \varepsilon_t$.

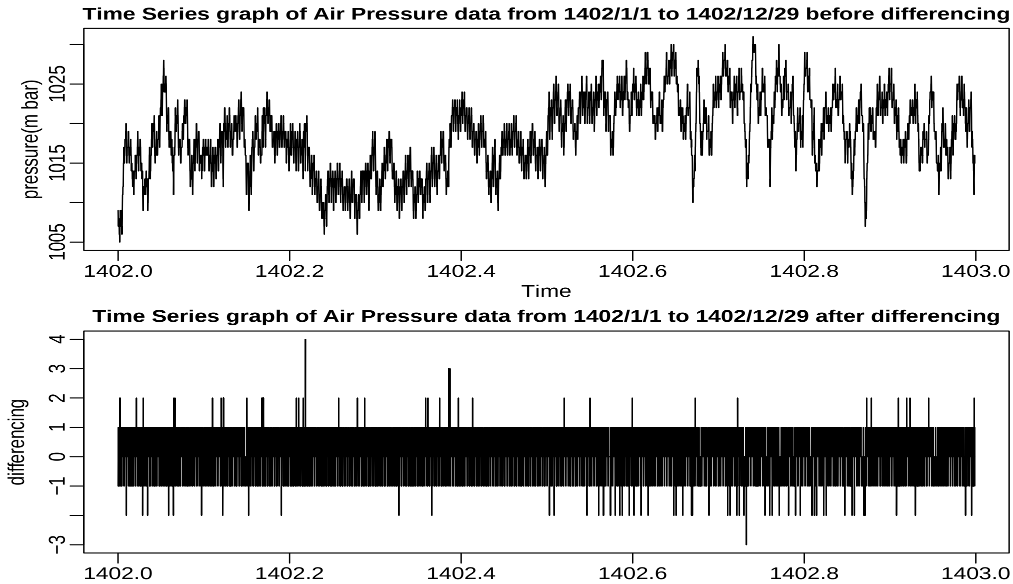

Figure 1: Time series graph of data before and after differencing

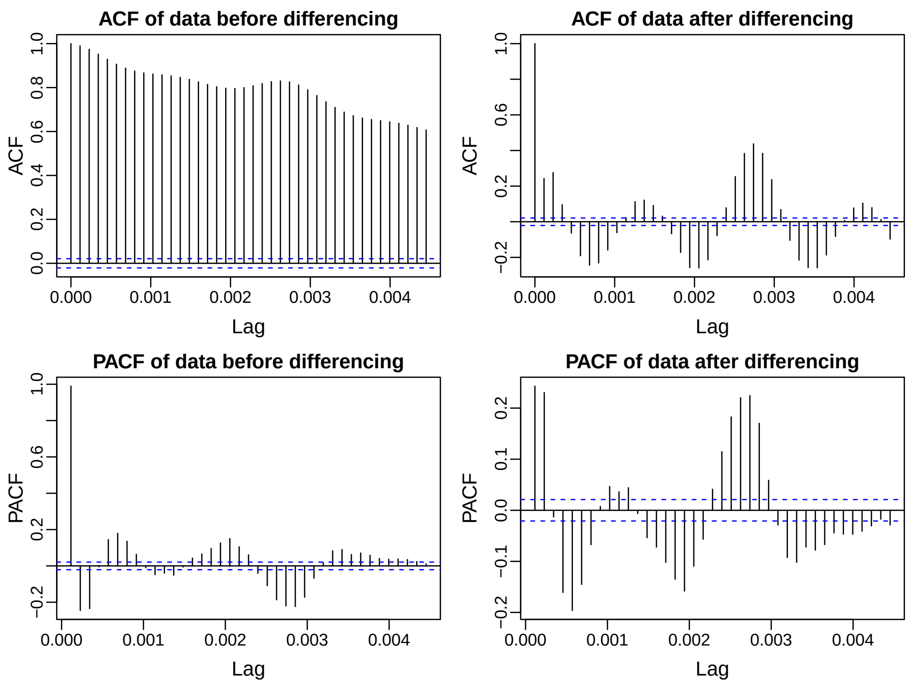

Figure 2: ACF and PACF of data before and after differencing

In Figure 1, (Figure 2), we see the time series graph (ACF and PACF of data) before and after differencing (see Appendices 3 and 4) and these two figures show that after one order of differentiation, the $ARMA$ model is a good fit to the differentiated time series data, and therefore the original time series data has $ARIMA(p, 1, q)$ model.
To select a suitable model, AIC (Akaike Information Criterion) is used to evaluate its performance and is equal to $\text{AIC}(k) = 2k - 2\ln(\hat{L})$ where k represents the number of model parameters and $\hat{L}$ is the maximized value of the likelihood function of the model [3]. In the **R** software, we can find the appropriate model based on AIC, i.e., $ARIMA(5, 1, 1)$. A summary of the specifications of this model is presented in Table 2 by the **R** software in Appendix 5.

	Ar1	**Ar2**	**Ar3**	**Ar4**	**Ar5**	**Ma1**
Coef	0.6137	0.1875	-0.0494	-0.1289	-0.1383	-0.4884
S.E.	0.0285	0.0133	0.0148	0.0124	0.0131	0.0275
Z-value	21.5311	14.0708	-3.3319	-10.3589	-10.5908	-17.7474
P-value	0.0000	0.0000	0.0009	0.0000	0.0000	0.0000
CI-lower	0.5579	0.1614	-0.0785	-0.1533	-0.1639	-0.5424
CI-upper	0.6696	0.2136	-0.0204	-0.1045	-0.1127	-0.4345

Table 2: Summary of the specifications of $ARIMA(5, 1, 1)$ model

In Table 2, point estimates and interval estimates (95%) of the coefficients and their standard errors, and the significance of the coefficients are also reported. To achieve this, we use the Wald test (Z-value) and the test statistic for the Wald test is equal to $\frac{\text{Coef}}{S.E} \sim AN(0, 1)$. The P-values in Table 2 for the Wald test indicate that all coefficients in the model are significant separatly.
In time series analysis, the Wald test can be used to test the significance of model coefficients. This test is particularly useful in regression models such as $ARIMA$ or dynamic regression models. The purpose of this test is to check whether one or more coefficients in a time series model are significant or not. The hypotheses are as follows:

$$H_0 : \text{The coefficient (or set of coefficients) is equal to zero (is insignificant)}$$
$$H_1 : \text{The coefficient (or coefficients) is opposite to zero (is significant)}$$

and the test statistic is as follows $W = \frac{\hat{\beta}-\beta_0}{SE(\hat{\beta})} \sim N(0, 1)$ where $\hat{\beta}$ is the coefficient estimate, $SE(\hat{\beta})$ is the standard deviation of the coefficient and k is the number of constraints (coefficients under test) [13]. On the other hand, for a set of parameters we can use multivariate testing. We know that MLE tends towards normal distribution. The variance of the MLE is calculated based on Fisher's information. In the **R** software we can see the covariance matrix of the MLE estimators as follows (see Appendix 5):

	AR_1	AR_2	AR_3	AR_4	AR_5	MA_1
AR_1	8×10^{-4}	-2×10^{-4}	-2×10^{-4}	0.0000	2×10^{-4}	-7×10^{-4}
AR_2	-2×10^{-4}	2×10^{-4}	0.0000	0.0000	0.0000	1×10^{-4}
AR_3	-2×10^{-4}	0.0000	2×10^{-4}	-1×10^{-4}	-1×10^{-4}	2×10^{-4}
AR_4	0.0000	0.0000	-1×10^{-4}	2×10^{-4}	-1×10^{-4}	0.0000
AR_5	2×10^{-4}	0.0000	-1×10^{-4}	-1×10^{-4}	2×10^{-4}	-2×10^{-4}
MA_1	-7×10^{-4}	1×10^{-4}	2×10^{-4}	0.0000	-2×10^{-4}	8×10^{-4}

Therefore, we can test the following hypotheses:

$$H_0 : \mathrm{AR}_1 = \mathrm{AR}_2 = \mathrm{AR}_3 = \mathrm{AR}_4 = \mathrm{AR}_5 = \mathrm{MA}_1 = 0$$
$$H_1 : \text{there is a parameter that is not equal to } 0$$

This test is similar to the F-test in regression analysis. To test this hypothesis, we use the Wald test ([13]) as follows $n\Big((\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^T I_n(\hat{\boldsymbol{\beta}})(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})\Big) \sim \chi^2_{(p)}$ where $\boldsymbol{\beta}$ is set of parameters, $\hat{\boldsymbol{\beta}}$ is a MLE for $\boldsymbol{\beta}$, $I_n(\boldsymbol{\beta})$ is a Fisher information matrix and p is the number of parameters. Here, under the null hypothesis, $\boldsymbol{\beta} = \mathbf{0}$, and so under the H_0, we have $n\hat{\boldsymbol{\beta}}^T I_n(\hat{\boldsymbol{\beta}})\hat{\boldsymbol{\beta}} \sim \chi^2_{(6)}$. Therefore, according to the codes provided in Appendix 5), the results of these tests are: (test statistic is 4738.5, $df = 6$, P-value < 0.0000).

σ^2	0.3612	M.E.	0.0008
log likelihood	-7955.86	RMSE	0.6007
AIC	15925.73	MAE	0.4643
BIC	15975.26	MPE	0.00006
ACF1	-0.0051	MAPE	0.04558

Table 3: Checking the accuracy of the $ARIMA(5,1,1)$ model

According to the model coefficient estimates in Table 2, the $ARIMA(5,1,1)$ time series model is as follows:

$$\nabla X_t = 0.6137\nabla X_{t-1} + 0.1875\nabla X_{t-2} - 0.0494\nabla X_{t-3} - 0.1289\nabla X_{t-4} - 0.1383\nabla X_{t-5} - 0.4884Z_{t-1} + Z_t$$

equivalently

$$X_t = 1.6137X_{t-1} - 0.4262X_{t-2} - 0.2369X_{t-3} - 0.0795X_{t-4} - 0.0095X_{t-5} + 0.1384X_{t-6} - 0.4881Z_{t-1} + Z_t.$$

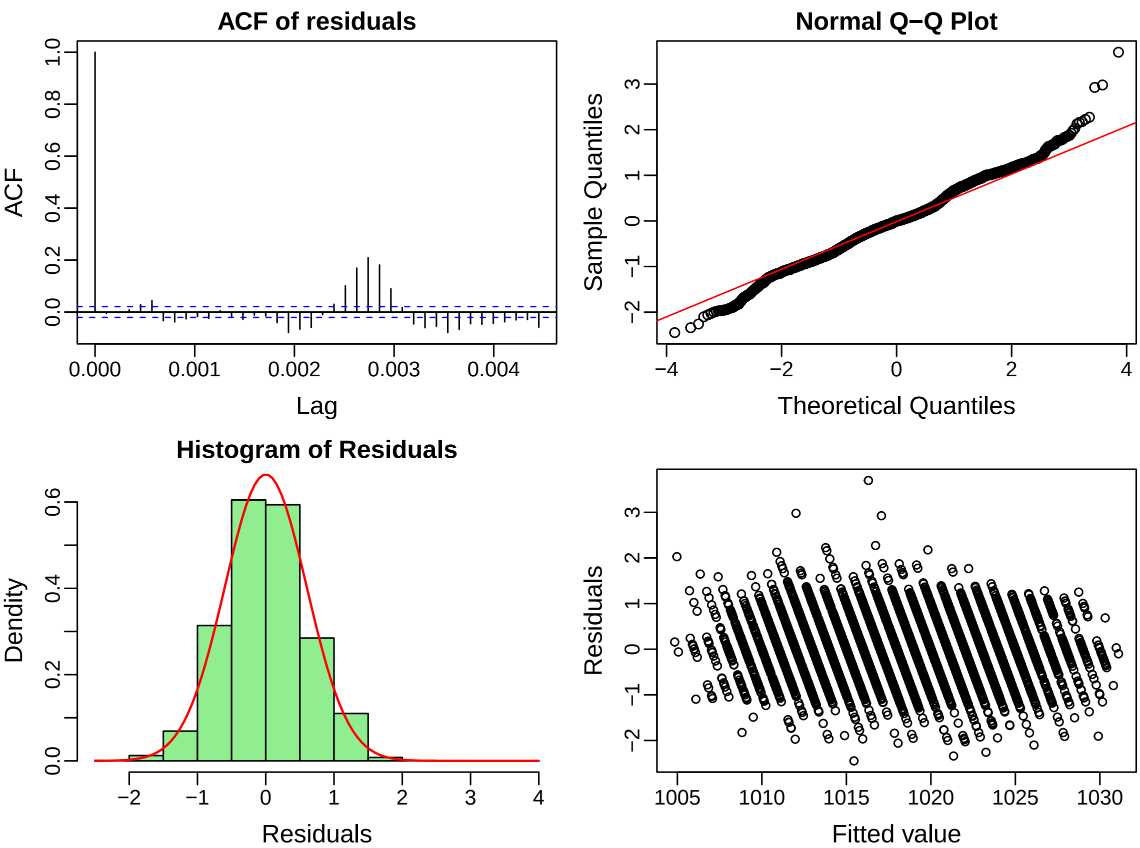

Figure 3: Checking model assumptions

The accuracy of the $ARIMA(5, 1, 1)$ model was calculated in Table 3 and the details of the codes are given in Appendix 5. In the Table 3, σ^2, i.e., the estimated variance of the residuals, M.E. (Mean Error), RMSE (Root Mean Square Error), MAE (Mean Absolute Error), BIC (Bayesian Information Criterion), MPE (Mean Percentage Error), MAPE (Mean Absolute Percentage Error) as well as the ACF1, i.e., the autocorrelation of residuals at lag 1 are given. The formulas for these statistics are:

$$M.E. = \frac{1}{n}\sum\nolimits_{i=1}^{n}(x_i - \hat{x}_i);$$

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}(x_i - \hat{x}_i)^2};$$

$$MAE = \frac{1}{n}\sum\nolimits_{i=1}^{n}|x_i - \hat{x}_i|;$$

$$MPE = \frac{1}{n}\sum\nolimits_{i=1}^{n}\frac{x_i - \hat{x}_i}{x_i} \times 100;$$

$$MAPE = \frac{1}{n}\sum\nolimits_{i=1}^{n}\left|\frac{x_i - \hat{x}_i}{x_i}\right| \times 100$$

where x_i is the observation and $\hat{x}_i$ is the value fitted by the model. From these measures, we see that the residuals are uncorrelated at the first lag (ACF1 $\simeq 0$), the model is good for prediction (MPE and MAPE $\simeq 0$) and the variance of the residuals, i.e., σ^2 is very small.

From Figure 3 (see Appendix 6), we can see the model obeying its assumptions, that is the residuals are uncorrelated (ACF of residuals), the residuals are normally distributed, i.e., $N(0, 0.3612)$ (histogram and Q-Q plot), variance is stable, the linearity assumption holds (scatter plot of ressiduals and fitted value).

In Figure 4 (see Appendix 7), a comparison is made between the ACF and PACF of the data after differencing with the theoretical ACF and PACF in the $ARMA(5, 1)$ model. In fact, we can see how close the fitted model and the observed data are to each other.

In Table 4, the prediction with model and standard errors in the next 24 hours is reported. We can also calculate the confidence interval for the prediction based on the Wald confidence interval [13] (see Appendix 8).

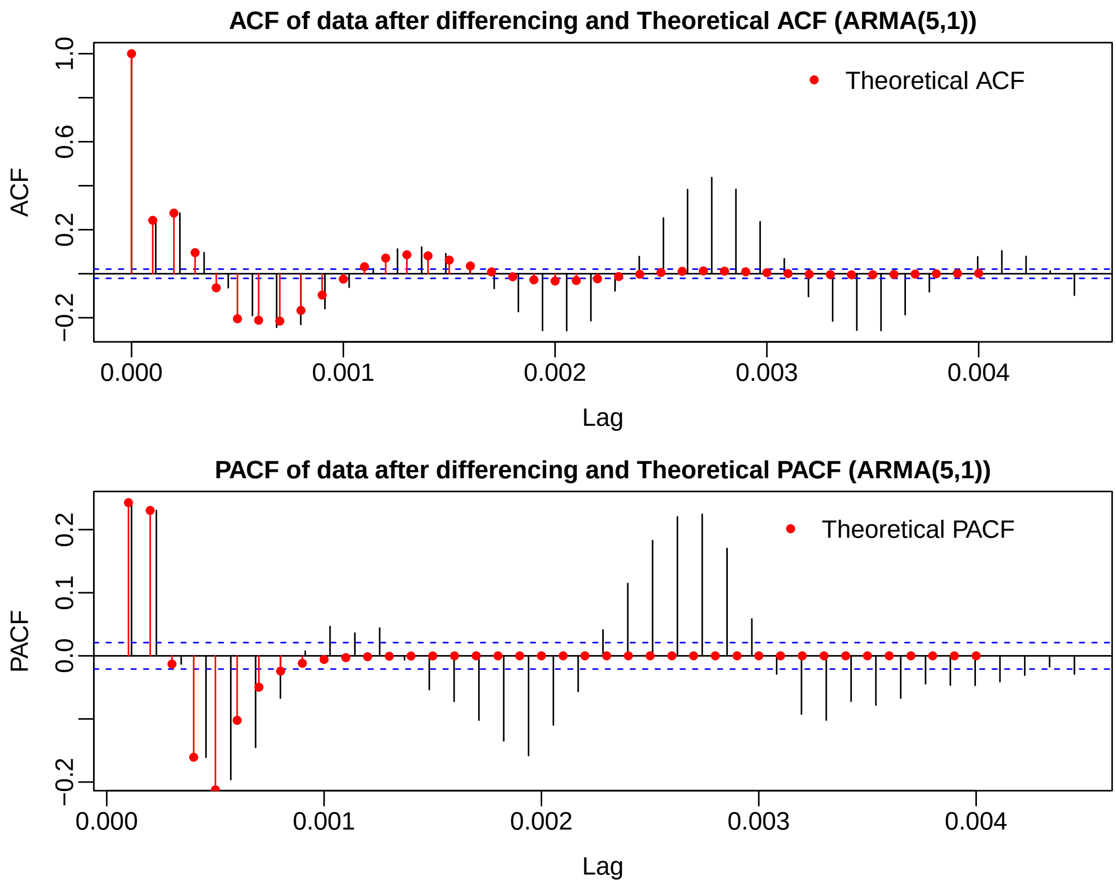

Figure 4: Comparison of ACF and PACF of data after differencing and theoretical ACF and PACF in the $ARMA(5, 1)$ model

hours	**1**	**2**	**3**	**4**	**5**	**6**
prediction	1014.667	1014.453	1014.397	1014.339	1014.347	1014.418
S.E	0.6010	0.9047	1.2313	1.5353	1.7878	1.9710
hours	**7**	**8**	**9**	**10**	**11**	**12**
prediction	1014.502	1014.581	1014.650	1014.692	1014.706	1014.698
S.E	2.1061	2.2010	2.2695	2.3231	2.3709	2.4188
hours	**13**	**14**	**15**	**16**	**17**	**18**
prediction	1014.673	1014.641	1014.609	1014.584	1014.569	1014.564
S.E	2.4711	2.5303	2.5965	2.6682	2.7429	2.8175
hours	**19**	**20**	**21**	**22**	**23**	**24**
prediction	1014.568	1014.577	1014.590	1014.602	1014.611	1014.617
S.E	2.8895	2.9575	3.0209	3.0801	3.1359	3.1893

Table 4: Prediction and standard error of forecasts in the next 24 hours

In Figure 5 (see Appendix 9), it can be seen how close the fitted value of model is to the observed data.

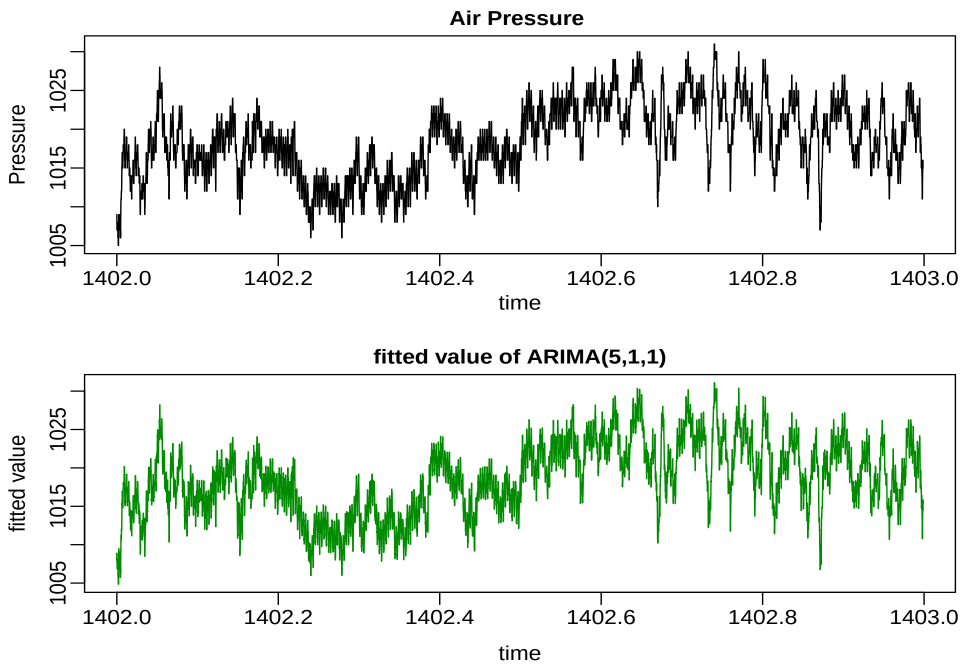

Figure 5: Time series plot of air pressure data and fitted value with $ARIMA(5,1,1)$

For the final assumptions on the proposed model, the stationarity of the AR part and the invertibility of the MA part are checked. In Figure 6 (see Appendix 9) the inverse root of AR and the inverse root of MA are plotted. All the roots are located in a circle with a unit radius, which means that the AR part is stationary and the MA part is invertible. This result tells us that the prediction with this model is good. For more advantages of this result, see [3].

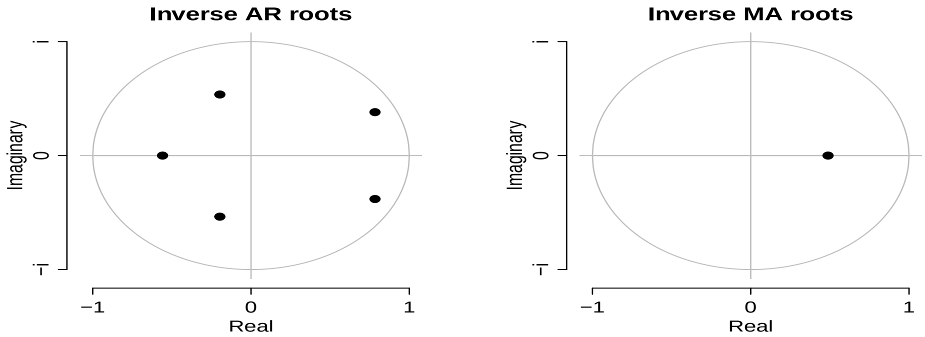

Figure 6: Inverse roots of AR and MA parts

4 Conclusions and Future Directions

Air pressure is an important measure for predicting rainfall and has many applications in various branches of science. In this paper, we discussed the applications of air pressure. One of our motivations for investigating air pressure is climate change. Isfahan is located near a meteorological drought. Over-extraction of groundwater is one of the reasons for the subsidence phenomenon. With air pressure, we can understand the duration of the subsidence phenomenon. The ARIMA is one of the traditional time series models that can model the behavior of such phenomena. We introduced some ARIMA symbols. Before modeling the data, the stationarity of the variance was checked with Box-Cox transformation and the stationarity of the mean was checked with ADF and Phillips-Perron tests. The power of Box-Cox transformation suggests 1, so we know that the data is stationary in variance. In the ADF and Phillips-Perron tests, the null hypothesis was rejected, meaning that there is no trend in the data and the mean. Due to the presence of noise in the data, we decided to derive from the observations. On the other hand, the software **R** suggests ARIMA(5,1,1) for this data. We estimated the parameters using the maximum likelihood method. For a set of parameters, we tested the Wald test for a set of parameters and observed that none of the parameters are equal to 0. Finally, we examined the assumptions of the ARIMA model. We found that the model follows its assumptions well. In fact, the proposed model is stationary and invertible.
One of the advantages of LSTM to ARIMA is that it performs better in long-term forecasting than ARIMA. But model ARIMA works better for short-term forecasts. We have some motivations to choose ARIMA model over modern models, for example LSTM. The advantages of ARIMA model over LSTM are as follows:

1- LSTM model is a black box model. There is no explanation for the prediction or parameter of black box model. Also, ARIMA model has simple explanation.

2- ARIMA model performs better on linear data than LSTM. From Figure 1, we can observe the linearity in the data trend. On the other hand, from Figure 3, we can see that the linearity assumption of the model is valid, hence we can say that linear model can obtain more information from the data.

3- In this paper, we used ARIMA model to show how the traditional model matches and competes with the new model.

4- The estimation of ARIMA model parameters is very simple and has no hyperparameters and there are only hyperparameters (p, d, q) in the model. The computations are very simple and few. In deep model, there are many hyperparameters and the initial values affect the result.

5- The noise range in the data is low. In such conditions, ARIMA model performs better.

6- For short-term forecasting, ARIMA model is a suitable model and for long-term forecasting, LSTM model is recommended.

7- The results of deep model are completely dependent on the architecture of the model network and the risk of overfitting is high in such models.

8- Last but not least, as Einstein famously advised, "A model should be as simple as possible, but not simpler than that [6]".

Appendices

This section provides suggested **R** codes and outputs. For each Appendix code, explanations are listed as comments.

Appendix 1

We imported the data, converted it to a time series in **R**, and printed observations 1 to 5 and the last observation 8751 as follows:

```
> ### Time series data for one year
> air_pre = ts(presure, start = c(1402, 1, 1), frequency = 24*365)
> ### data view
> air_pre[1:5] ; air_pre[8751]
[1] 1009 1008 1008 1007 1008
[1] 1015
```

Appendix 2

Before modeling with ARIMA, we need to check the stationarity of variance and stationarity of mean. For stationarity of variance, Box-Cox transformation is used, and for stationarity of mean, ADF test and Phillips-Perron tests are used.

```
> ########## stationarity tests
> ### stationarity of variance with Box-Cox transform
> forecast::BoxCox.lambda(air_pre)
[1] 1
> ########## stationarity of mean
> ### ADF test
> tseries::adf.test(air_pre)

 Augmented Dickey-Fuller Test

data:  air_pre
Dickey-Fuller = -4.3692, Lag order = 20, p-value = 0.01
alternative hypothesis: stationary

Warning message:
In tseries::adf.test(air_pre) : p-value smaller than printed p-value
> ### Phillips-Perron test
> tseries::pp.test(air_pre)

 Phillips-Perron Unit Root Test

data:  air_pre
Dickey-Fuller Z(alpha) = -139.98, Truncation lag parameter = 12,
p-value = 0.01
alternative hypothesis: stationary

Warning message:
In tseries::pp.test(air_pre) : p-value smaller than printed p-value
```

4.1 Appendix 3

A time series graph of air pressure was drawn before and after differentiation.

```
> ### ts plot
> ### plot 1
> par(mfrow = c(2, 1))
> plot.ts(air_pre, xlab = '', ylab = '', main = '')
> mtext(text = 'Time', side = 1, line = 1.5)
```

```
> mtext(text = 'pressure(m bar)', side = 2, line = 1.5)
> mtext(text = 'Time Series graph of air pressure data from 1402/1/1
     to 1402/12/29 before differencing', side = 3, line = 0.25, font = 2)
> ### plot 2
> plot.ts(diff(air_pre), xlab = '', ylab = '', main = '')
> mtext(text = 'Time', side = 1, line = 2)
> mtext(text = 'differencing', side = 2, line = 2)
> mtext(text = 'Time Series graph of air pressure data from 1402/1/1
     to 1402/12/29 after differencing', side = 3, line = 0.25, font = 2)
```

Appendix 4

The ACF and PACF graph of air pressure was drawn before and after differentiation.

```
> ### acf pacf
> ### plot 1
> par(mfrow = c(2, 2))
> acf(air_pre, main = '', xlab = '', ylab = '')
> mtext(text = 'Lag', side = 1, line = 2)
> mtext(text = 'ACF', side = 2, line = 1.5)
> mtext(text = 'ACF of data before differencing',
        side = 3, line = 0.25, font = 2)
> ### plot 2
> acf(diff(air_pre), main = '', xlab = '', ylab = '')
> mtext(text = 'Lag', side = 1, line = 2)
> mtext(text = 'ACF', side = 2, line = 2)
> mtext(text = 'ACF of data after differencing',
        side = 3, line = 0.25, font = 2)
> ### plot 3
> pacf(air_pre, main = '', xlab = '', ylab = '')
> mtext(text = 'Lag', side = 1, line = 2)
> mtext(text = 'PACF', side = 2, line = 1.5)
> mtext(text = 'PACF of data before differencing',
        side = 3, line = 0.25, font = 2)
> ### plot 4
> pacf(diff(air_pre), main = '', xlab = '', ylab = '')
> mtext(text = 'Lag', side = 1, line = 2)
> mtext(text = 'PACF', side = 2, line = 2)
> mtext(text = 'PACF of data after differencing',
        side = 3, line = 0.25, font = 2)
```

Appendix 5

The parameters of ARIMA model were estimated. Based on AIC, the **R** software selected the ARIMA(5,1,1) and the accuracy of the model was reported. In the second step, two tests were considered. In the first step, the set of parameters was tested. This test is similar to the F-test in regression. In the second step, the confidence interval and Wald test were reported.

```
> ### Time series model
> tmodel = forecast::auto.arima(air_pre, method = 'ML')
> summary(tmodel)
Series: air_pre
ARIMA(5,1,1)

Coefficients:
         ar1     ar2      ar3      ar4      ar5      ma1
      0.6137  0.1875  -0.0494  -0.1289  -0.1383  -0.4884
s.e.  0.0285  0.0133   0.0148   0.0124   0.0131   0.0275

sigma^2 = 0.3612:  log likelihood = -7955.86
AIC=15925.73   AICc=15925.74   BIC=15975.26

Training set error measures:
                       ME      RMSE       MAE          MPE       MAPE
Training set 0.0007634926 0.6007197 0.4642622 5.795284e-05 0.04558304
             MASE         ACF1
Training set          NaN      -0.005126946
> ### Confidence interval and testing hypothesis
> ### test for the set of parameters
> round(vcov(tmodel), 4)
       ar1    ar2    ar3    ar4    ar5    ma1
ar1  8e-04 -2e-04 -2e-04  0e+00  2e-04 -7e-04
ar2 -2e-04  2e-04  0e+00  0e+00  0e+00  1e-04
ar3 -2e-04  0e+00  2e-04 -1e-04 -1e-04  2e-04
ar4  0e+00  0e+00 -1e-04  2e-04 -1e-04  0e+00
ar5  2e-04  0e+00 -1e-04 -1e-04  2e-04 -2e-04
ma1 -7e-04  1e-04  2e-04  0e+00 -2e-04  8e-04
> t(tmodel$coef) %*% solve(vcov(tmodel)) %*% tmodel$coef
         [,1]
[1,] 4738.516
> car::linearHypothesis(tmodel, c("ar1 = 0", "ar2 = 0", "ar3 = 0",
                                  "ar4 = 0", "ar5 = 0", "ma1 = 0"))

Linear hypothesis test:
```

```
ar1 = 0
ar2 = 0
ar3 = 0
ar4 = 0
ar5 = 0
ma1 = 0

Model 1: restricted model
Model 2: tmodel

  Df  Chisq Pr(>Chisq)
1  6 4738.5  < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> ### CI and Wald test
> confint(tmodel) ; lmtest::coeftest(tmodel)
          2.5 %      97.5 %
ar1  0.55785174  0.66958468
ar2  0.16140681  0.21364933
ar3 -0.07852634 -0.02035826
ar4 -0.15327702 -0.10450355
ar5 -0.16393457 -0.11273363
ma1 -0.54236313 -0.43448324

z test of coefficients:

     Estimate Std. Error  z value  Pr(>|z|)
ar1  0.613718   0.028504  21.5311 < 2.2e-16 ***
ar2  0.187528   0.013327  14.0708 < 2.2e-16 ***
ar3 -0.049442   0.014839  -3.3319 0.0008626 ***
ar4 -0.128890   0.012442 -10.3589 < 2.2e-16 ***
ar5 -0.138334   0.013062 -10.5908 < 2.2e-16 ***
ma1 -0.488423   0.027521 -17.7474 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

> ### accuracy of model
> forecast::accuracy(tmodel)
                       ME      RMSE       MAE          MPE       MAPE
Training set 0.0007634926 0.6007197 0.4642622 5.795284e-05 0.04558304
             MASE          ACF1
Training set  NaN -0.005126946
```

Appendix 6

In this appendix, the analysis of residuals and the assumptions of the model with residuals were examined.

```
> ####### residuals analysis
> par(mfrow = c(2, 2))
> ### plot 1
> acf(tmodel$residuals, xlab = '', ylab = '', main = '')
> mtext(text = 'Lag', side = 1, line = 2)
> mtext(text = 'ACF', side = 2, line = 2)
> mtext(text = 'ACF of residuals', side = 3, line = 0.25, font = 2)
> ### plot 2
> qqnorm(tmodel$residuals, xlab = '', ylab = '', main = '')
> qqline(tmodel$residuals, col = 'red')
> mtext(text = 'Theoretical Quantiles', side = 1, line = 2)
> mtext(text = 'Sample Quantiles', side = 2, line = 2)
> mtext(text = 'Normal Q-Q Plot', side = 3, line = 0.25, font = 2)
> ### plot 3
> hist(tmodel$residuals, col = 'lightgreen', freq = F, ylim = c(0, 0.65),
      xlab = '', ylab = '', main = '')
> curve(dnorm(x, 0, sd(tmodel$residuals)), add = T,
       col = 'red', lwd = 1.5)
> mtext(text = 'Residuals', side = 1, line = 2)
> mtext(text = 'Density',side = 2, line = 2)
> mtext(text = 'Histogram of Residuals', side = 3, line = 0.25, font = 2)
> ### plot 4
> plot(tmodel$fitted, tmodel$residuals, xlab = '', ylab = '')
> mtext(text = 'Fitted value', side = 1, line = 2)
> mtext(text = 'Residuals', side = 2, line = 2)
```

Appendix 7

In this section, a comparison is made between the ACF and PACF of the data and the model.

```
> ### comparission of acf and pacf between data and theoric ARMA
> par(mfrow=c(2, 1))
> ### plot 1
> acf(diff(air_pre), xlab = '', ylab = '', main = '')
> points(seq(0.000, 0.004, 0.0001), ARMAacf(ar = c(0.6137, 0.1875,
        -0.0494, -0.1289, -0.1383), ma = c(-0.4884), lag.max = 40),
        type = 'h', col = 'red', pch = 20)
> points(seq(0.000, 0.004, 0.0001), ARMAacf(ar = c(0.6137, 0.1875,
```

```
         -0.0494, -0.1289, -0.1383), ma = c(-0.4884), lag.max = 40),
         col = ’red’, pch = 20)
> mtext(text = ’Lag’, side = 1, line = 2)
> mtext(text = ’ACF’, side = 2, line = 2)
> mtext(text = ’ACF of data after differencing and Theoretical
        ACF (ARMA(5,1))’, side = 3, line = 0.25, font = 2)
> legend(’topright’, legend = ’Theoretical ACF’, col = ’red’,
         pch = 20, bty = ’n’)
> ### plot 2
> pacf(diff(air_pre), xlab = ’’, ylab = ’’, main = ’’)
> points(seq(0.0001, 0.004, 0.0001), ARMAacf(ar = c(0.6137, 0.1875,
         -0.0494, -0.1289, -0.1383), ma = c(-0.4884), lag.max = 40,
          pacf = T), type = ’h’, col = ’red’, pch = 20)
> points(seq(0.0001, 0.004, 0.0001), ARMAacf(ar = c(0.6137, 0.1875,
         -0.0494, -0.1289, -0.1383), ma = c(-0.4884), lag.max = 40,
         pacf = T), col = ’red’, pch = 20)
> mtext(text = ’Lag’, side = 1, line = 2)
> mtext(text = ’PACF’, side = 2, line = 2)
> mtext(text = ’PACF of data after differencing and Theoretical
        PACF (ARMA(5,1))’, side = 3, line = 0.25, font = 2)
> legend(’topright’, legend = ’Theoretical PACF’, col = ’red’,
          pch = 20, bty = ’n’)
```

Appendix 8

In this appendix, the forecast for the next 24 hours or the next day and the prediction confidence interval were reported.

```
> #### prediction for next day
> ### output 1
> predict(tmodel, 24)
$pred
Time Series:
Start = c(1402, 8752)
End = c(1403, 15)
Frequency = 8760
 [1] 1014.667 1014.453 1014.397 1014.339 1014.347 1014.418 1014.502
 [8] 1014.581 1014.650 1014.692 1014.706 1014.698 1014.673 1014.641
[15] 1014.609 1014.584 1014.569 1014.564 1014.568 1014.577 1014.590
[22] 1014.602 1014.611 1014.617

$se
Time Series:
Start = c(1402, 8752)
```

```
End = c(1403, 15)
Frequency = 8760
 [1] 0.6009601 0.9046972 1.2312509 1.5352679 1.7877651 1.9710397
 [7] 2.1060982 2.2010398 2.2694667 2.3231239 2.3709345 2.4188069
[13] 2.4711381 2.5302600 2.5964710 2.6682449 2.7429095 2.8174824
[19] 2.8895058 2.9574750 3.0208968 3.0800876 3.1358672 3.1892574

> ### output 2
> p = forecast::forecast(tmodel, 24)
> p = as.data.frame(p) ; rownames(p) = c(1: 24) ;  p
   Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
1        1014.667 1013.897 1015.437 1013.489 1015.845
2        1014.453 1013.293 1015.612 1012.680 1016.226
3        1014.397 1012.819 1015.975 1011.984 1016.810
4        1014.339 1012.372 1016.307 1011.330 1017.349
5        1014.347 1012.056 1016.638 1010.843 1017.851
6        1014.418 1011.892 1016.943 1010.554 1018.281
7        1014.502 1011.803 1017.201 1010.374 1018.630
8        1014.581 1011.761 1017.402 1010.267 1018.895
9        1014.650 1011.741 1017.558 1010.201 1019.098
10       1014.692 1011.715 1017.669 1010.139 1019.245
11       1014.706 1011.668 1017.745 1010.059 1019.353
12       1014.698 1011.598 1017.798 1009.957 1019.439
13       1014.673 1011.506 1017.840 1009.830 1019.517
14       1014.641 1011.398 1017.884 1009.682 1019.600
15       1014.609 1011.282 1017.937 1009.520 1019.698
16       1014.584 1011.165 1018.004 1009.355 1019.814
17       1014.569 1011.054 1018.084 1009.193 1019.945
18       1014.564 1010.953 1018.175 1009.042 1020.086
19       1014.568 1010.865 1018.271 1008.904 1020.231
20       1014.577 1010.787 1018.367 1008.781 1020.374
21       1014.590 1010.718 1018.461 1008.669 1020.511
22       1014.602 1010.654 1018.549 1008.565 1020.639
23       1014.611 1010.592 1018.630 1008.465 1020.757
24       1014.617 1010.530 1018.704 1008.366 1020.868
```

Appendix 9

In this section, a comparison is made between the data and the fitted value of the model.

```
> ### plot 1
> par(mfrow=c(2, 1))
> plot(air_pre, xlab = '', ylab = '', main = '', type = 'l')
> mtext(text = 'time', side = 1, line = 1.5)
```

```
> mtext(text = 'Pressure', side = 2, line = 2)
> mtext(text = 'Air Pressure', side = 3, line = 0.25, font = 2)
> ### plot 2
> plot(fitted(tmodel), col = 'green4', xlab = '', ylab = '', main = '')
> mtext(text = 'time', side = 1, line = 2)
> mtext(text = 'fitted value', side = 2, line = 2)
> mtext(text = 'fitted value of ARIMA(5,1,1)', side = 3, line = 0.25,
      font = 2)
> ### Stationarity of AR part and invertibility of MA part
> plot(tmodel)
```

References

[1] Albeladi K., Zafar B. & Mueen A., Time series forecasting using LSTM and ARIMA, International Journal of Advanced Computer Science and Applications, 14(1), 313-320, 2023.

[2] Aweda F. O., Akinpelu J. A., Samson T. K., Sanni M. & Olatinwo B. S., Modeling and Forecasting Selected Meteorological Parameters for the Environmental Awareness in Sub Sahel West Africa Stations, Journal of the Nigerian Society of Physical Sciences, 4(3), 820, 2022.

[3] Brockwell, P. J. Davis R. A. & Calder M. V., Introduction to time series and forecasting, New York, Springer, (2002).

[4] Cryer J. D. & Chan K. S., Time series analysis: with applications in R, Springer Science and Business Media, (2008).

[5] Dickey D. A. & Fuller W. A., Distribution of the estimators for autoregressive time series with a unit root , Journal of the American statistical association, 74(366a), 427-431, 1979.

[6] Hastie T., Tibshirani R., & Friedman J., The Elements of Statistical Learning 2nd edition, Springer, (2009).

[7] Hou J., Wang Y., Zhou J. & Tian Q., Prediction of hourly air temperature based on CNN–LSTM, Geomatics, Natural Hazards and Risk, 13(1), 1962-1986 2022.

[8] Koçak H., Time Series Prediction of Temperature Using ARIMA and LSTM Models, Gazi Journal of Engineering Sciences, 9(3), 574-584, 2023.

[9] Kurniawan J. D., Trihandaru S. & Parhusip H. A., Performance of an AIOT-Particle Device for Air Quality and Environmental Data Prediction in Salatiga Area Using ARIMA Model, Jurnal Ilmiah Teknik Elektro Komputer dan Informatika, 10(2), 217-234, 2024.

[10] Mani G. & Viswanadhapalli J. K., Prediction and forecasting of air quality index in Chennai using regression and ARIMA time series models, Journal of Engineering Research, 10(2A), 179-194, 2022.

[11] Phillips P. C. B. & Perron P., Testing for a Unit Root in Time Series Regression, Biometrika, 75(2), 335-346, 1988.

[12] Sirisha U. M., Belavagi M. C. & Attigeri G., Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison, Ieee Access, 10, 124715-124727, 2022.

[13] Wald A., Tests of statistical hypotheses concerning several parameters when the number of observations is large, Transactions of the American Mathematical society, 54(3), 426-82, 1943.