A Hybrid Methodology for Short Term Temperature Forecasting

Wissam Abdallah, Nassib Abdallah, Jean-Marie Marion, Mohamad Oueidat, Pierre Chauvet

Scientific Research Publishing

A Hybrid Methodology for Short Term Temperature Forecasting

Wissam Abdallah[1], Nassib Abdallah[2], Jean-Marie Marion[3], Mohamad Oueidat[4], Pierre Chauvet[5]

[1]Angevin Research Laboratory in Systems Engineering, University of Angers, Angers, France
[2]Computer Science and Systems Laboratory, Polytech Marseille, Aix Marseille University, Aix-en-Provence, France
[3]Angevin Research Laboratory in Systems Engineering, Université Catholique de l'Ouest, Angers, France
[4]Department of Industrial Engineering and Maintenance, Faculty of Technology, Lebanese University, Beirut, Lebanon
[5]Angevin Research Laboratory in Systems Engineering, Université Catholique de l'Ouest, Angers, France
Email: wissam.abdallah@etud.univ-angers.fr, nassib.abdallah@univ-amu.fr, marion@uco.fr,
mohoueidat@yahoo.com, Pierre.chauvet@uco.fr

Abstract

Developing a reliable weather forecasting model is a complicated task, as it requires heavy IT resources as well as heavy investments beyond the financial capabilities of most countries. In Lebanon, the prediction model used by the civil aviation weather service at Rafic Hariri International Airport in Beirut (BRHIA) is the ARPEGE model, (0.5) developed by the weather service in France. Unfortunately, forecasts provided by ARPEGE have been erroneous and biased by several factors such as the chaotic character of the physical modeling equations of some atmospheric phenomena (advection, convection, etc.) and the nature of the Lebanese topography. In this paper, we proposed the time series method ARIMA (Auto Regressive Integrated Moving Average) to forecast the minimum daily temperature and compared its result with ARPEGE. As a result, ARIMA method shows better mean accuracy (91%) over the numerical model ARPEGE (68%), for the prediction of five days in January 2017. Moreover, back to five months ago, in order to validate the accuracy of the proposed model, a simulation has been applied on the first five days of August 2016. Results have shown that the time series ARIMA method has offered better mean accuracy (98%) over the numerical model ARPEGE (89%) for the prediction of five days of August 2016. This paper discusses a multiprocessing approach applied to ARIMA in order to enhance the efficiency of ARIMA in terms of complexity and resources.

Keywords

Time Series Analysis, ARIMA Auto Regressive Integrated Moving Average, Weather Forecasting Model, Multiprocessing

1. Introduction

World Meteorological Organization (WMO) called for integrating the global efforts needed to enhance the accuracy in weather forecasts [1]. Indeed, the development of efficient weather prediction models has always been particularly difficult and challenging for meteorological services [2] [3]. In fact, the prediction of temperature is highly recommended on many sectors such as agriculture, hydrology and meteorology. In the case of Lebanon, the temperature is considered as an important parameter in forecasting since it has substantial risks. Moreover, the temperature could produce the constitution of hazardous fog which produces a fatal risk during Lebanese people's movements. In addition, temperature can impact tourism sectors especially public and private institutions (hospitals, audiovisual media, hotels and "resorts", ski clubs ...). The prediction of the temperature can be achieved using many approaches and techniques such as numerical models, Learning Techniques (LT) and Time Series Analysis (TSA) [2] [4]. The proposed model ARIMA fit Moving Average (MA) model having an order 35 which is considered to be very high order, this order poses a high complexity and requires heavy computers resources and a long execution time; as a solution for this problem, a migration towards ARIMA parallel seems to be a very powerful proposed method in this framework.

In 2007, Mohsen Hayati *et al.* worked on the application of neural networks to study the design of short-term temperature forecasting (STTF) [5]. The result of the Multi-Layer Perceptron (MLP) network model used to forecast the temperature for one day ahead in the Kermanshah, Iran, shows that it has good performance and reasonable prediction accuracy compared to real temperature values.

In 2010, S. Santhosh Baboo *et al.* also worked on neural network-based algorithm for temperature prediction. The results are compared with real data issued from meteorological department. These results confirm that the model has a good potential to predict the temperature in the forecasting service [5].

Then, in 2012, Kumar Abhisheka *et al.* used Artificial Neural Network to forecast the temperature. The results show that by increasing the number of neurons from 20 to 50 and 80, and increasing the number of hidden layer from 5 to 10, the Mean Square Error decreased from 3.65 to 2.71 and the performance of the proposed model increased [6].

Moreover, in 2018, Thi-Thu-Hong *et al.* [7], discussed a comparative study of univariate Forecasting Methods for Meteoroligical Time Series, they have considered that the humidity is an important meteorological factor to study due to its direct effect on the temperature: The Single exponential smoothing method (SES), the Season Al-naive (SNAIVE), SARIMA, Feed-Forward Neural Network (FFNN), Dynamic Time Wraping Based Imputation (DTWBI), Bayesian Structural Times Series (BSTS). they conclude that the result issuing form method Seasonal SARIMA (92%) is more efficient than the result issuing from the other studied techniques: Feed-Forward Neural Network (FFNN) (91%), Dynamic Time Warping-based Imputation (DTWBI) (91%), Bayesian Structural Time Se-

ries (BSTS) (88%), Single exponential smoothing (SES) (87%), Seasonalnaive (Snaive) (86%), to predict the humidity where the real value is 93%.

This paper contains three sections: The first section presents the numerical prediction model ARPEGE; The second section discusses Auto Regressive Integrated Moving Average (ARIMA); The third section shows the comparative study between ARPEGE and two different approaches of ARIMA (sequential and multiprocessing) where as the justification of using such approaches would be discussed While the fourth section discusses the results obtained during our comparative study between different approaches, finally, the conclusion of our works and the presentation of the perspective of our study. This paper presents a short term prediction (5 days ahead) of the temperature in two different periods (e.g. summer and winter): 1) from 01/08/2016 to 05/08/2016; 2) from 01/01/2017 to 05/01/2017.

2. ARPEGE

ARPEGE is a numerical model developed by Meteo France and widely used in several countries. As other numerical prediction model, ARPEGE is based on a set of Navier-Stokes equations that describes the movements of fluids. ARPEGE model covers the whole globe with an average mesh of 16 km and Europe with a mesh of 7.5 km. Moreover, its horizontal resolution is about 7.5 km over France and 37 km in the antipodes. ARPEGE has 105 levels where the first level is 10 meters above the surface and can reach 11 km of altitude [8] [9] [10].

ARPEGE model is capable of guiding weather forecasters beyond the first few hours of many upcoming meteorological phenomena such as thunderstorm, snow, cold front, warm front etc. [9] [11] [12] [13] [14] [15]. However, ARPEGE's fundamental problems appear in its incapacity to solve the chaotic equations which model some of the atmospheric phenomena such as: tornado, turbulence, etc. and in the weakness of accuracy in the forecasting over a geographically defined area [9] [13] [16].

3. ARIMA

Recently, Time Series Analysis (TSA) becomes very useful in the prediction field as it is applied in several domains such as weather forecasting, economics, engineering, environment, medicine, etc. [8]. In weather forecasting, TSA helps in predicting future weather situations, by collecting historical data within regular time intervals [14] [15]. Known as series, this data is collected at specific intervals (hourly, daily, weekly, monthly, yearly, etc.). TSA offers certain advantages in terms of trusted results in weather forecasting and efficiency in detection both trends and seasonality within the series [8]. The ARIMA model follows a linear expression given by the following equation:

$$y_t = \Theta_0 + \phi_1 y_{t-1} + \cdots + \phi_p y_{t-p} + \varepsilon_t - \Theta_1 \varepsilon_{t-1} - \cdots - \Theta_q \varepsilon_{t-q}$$

where: y_t: is the observed value. ε_t: is the random error at time t. ϕ_j: are the coefficients respectively of the AR (Auto Regressive) model. Θ_j: are the coefficients

respectively of the MA (Moving Average) model. p: order of auto regressive. q: order of moving average [1].

On the other hand, while building an ARIMA model, the series must be stationary. Otherwise, stationary transformation is required [17].

4. Comparative Study between ARPEGE, ARIMA Sequential and ARIMA Parallel

In the proposed comparative methodology (**Figure 1**), ARPEGE is compared to two different approach of ARIMA (ARIMA SEQ. and ARIMA PARA.) in order to predict the temperature for five days ahead. The results of these three methods are compared to choose the best model in term of accuracy and resources consumption (RAM, CPU and execution time).

4.1. Numerical Weather Model ARPEGE

In Lebanon, a physical forecasting model used by civil aviation meteo service at BRHIA is ARPEGE.

The prognostic variables of the model for the atmospheric part are: the horizontal components of the wind, the temperature, the specific humidities of the water vapor and four categories of hydrometeors (liquid droplets, ice crystals, rain, snow) and the turbulent kinetic energy etc.
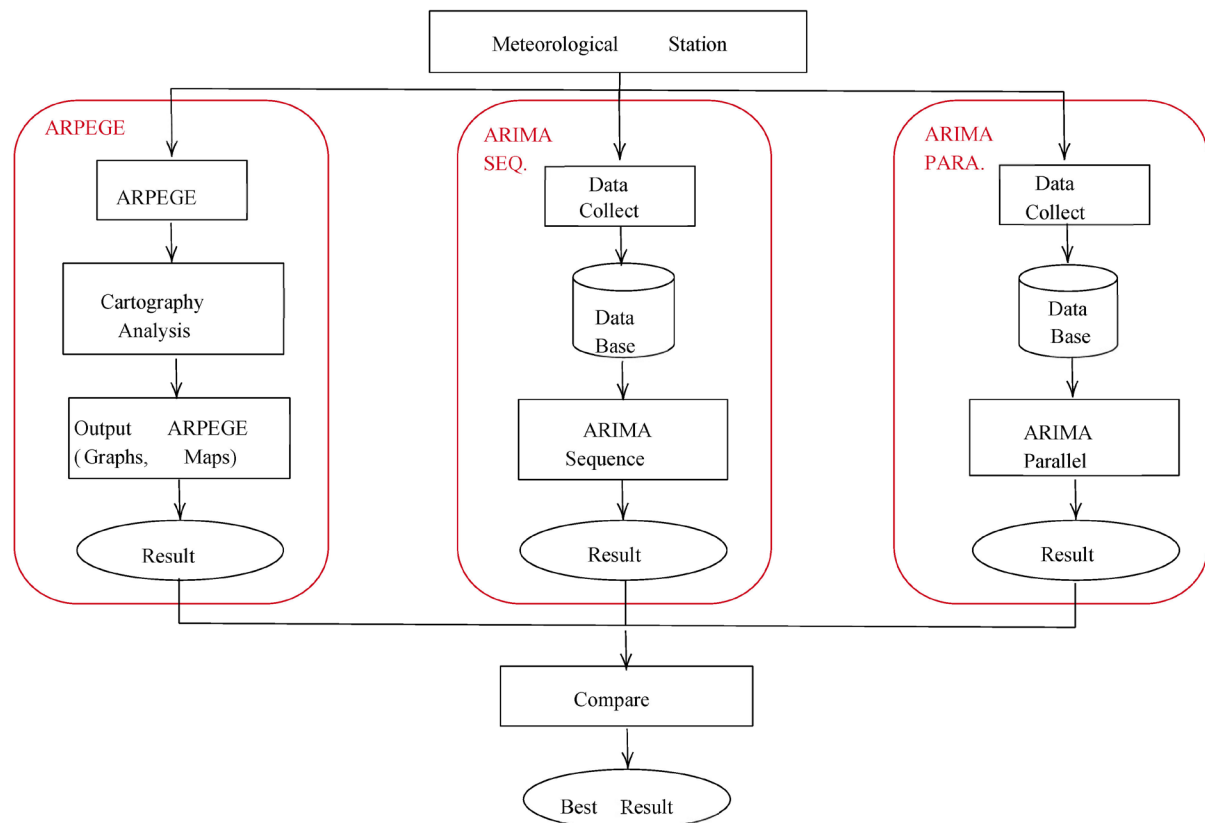

Figure 1. Comparison between predictive models to forecast short term temperature: ARPEGE, ARIMA sequential and ARIMA parallel.

In addition, the outputs of the ARPEGE model are graphs, vertical sections, time diagrams, weather maps etc.

A post-processing of data makes the recovered data as a possible process in term of graphs, maps of weather etc.

Unfortunately, in some cases, the forecasts provided by ARPEGE have been erroneous and biased due to several factors:

- character of the chaotic equations;
- errors resulting from the measurements of the initial state of the atmosphere;
- errors induced by the discretization of the atmosphere (horizontal and vertical dimensions of the mesh);
- error related to the spatio-temporal iterative process;
- problem related to the complication of modeling physical phenomena of the atmosphere like the convection.

These various sources of error enforce us to try to find an efficient and reliable alternative to reduce the minimum possible errors and the biases issuing form the ARPEGE outputs.

In the absence of satellite data, we propose to build a forecasting model based on a statistical approach to improve the quality of existing models such as time series analysis approach that provides the methodology ARIMA.

In order to validate the proposed methodology, we will focus on the case of Lebanon in collaboration with meteorological service in order to access a vast database related to Beirut meteorological station. These data that have been recorded over several years will be used to test the reliability of the proposed model.

4.2. ARIMA Sequential

The data of daily temperature of Beirut city for a period of 11 years from 01/01/2006 to 31/12/2016 were taken from the meteorological service at Beirut Rafic Hariri International Airport (BRHIA). ARIMA model takes the temperature time series data as an input. The three mandatory p, d and q parameters that have to be selected, represent the order of the ARIMA model. p: is the number of autoregressive terms, d is the number of non-seasonal differences and q is the number of moving average model coefficients. ARIMA model is considered to be a promoter approach in weather forecasting. It is a great statistical technique for modeling time series, temperature and other meteorological variables.

1) Data collecting

The meteorological service (Met-Service) of Beirut Rafic Hariri International Airport (BRHIA) has 20 climatological stations spread all over Lebanon. These stations are equipped with a multitude of sensors to measure several meteorological variables: temperature, humidity, precipitation, wind speed, wind direction and vapor pressure. Among these stations, 17 are automatic and autonomous weather observation stations, while 3 are manual observation stations that require the intervention of technicians to send data. The BRHIA climate department represents the data warehouse that collects all data from other meteorological stations via a Local Area Network (LAN); these data have been arc-

hived as text format. At the end of the day, it connects to the other meteorological stations in order to fetch and save daily temperature report in text files.

2) Data Base Oracle

Oracle database collects and organizes data from text files into tables; this mission was executed through PL/SQL procedural language packages and libraries. At the end of each day, a schedule job procedure connects to the data warehouse available in the climate department, opens each text temperature data file, reads values of temperature parameter to finally insert them into the database; these libraries also have the capability to generate daily CSV files.

3) ARIMA methodology

The realization of the ARIMA proposed model has been achieved after the execution of a sequence of processes illustrated in **Figure 2**.

a) Data processing

During this phase, null data is checked, in case there would be null data in the series, they would be replaced by the average values of the series itself related to a month where it has been found in a specific year. The data used in this article is: minimum daily temperature.

b) Seasonality verification

The result in **Figure 3** shows that there is a seasonality in the series. This seasonality is well seen while the periodicity is 360 days. So the process of differentiation ($X_t - X_{t-360}$) must be associated to remove the seasonality.
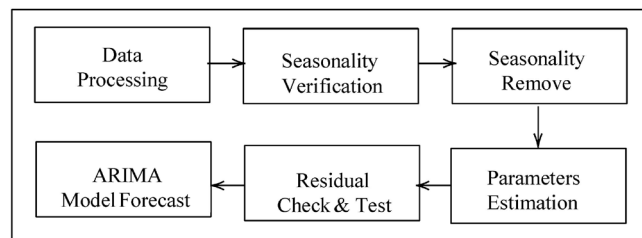
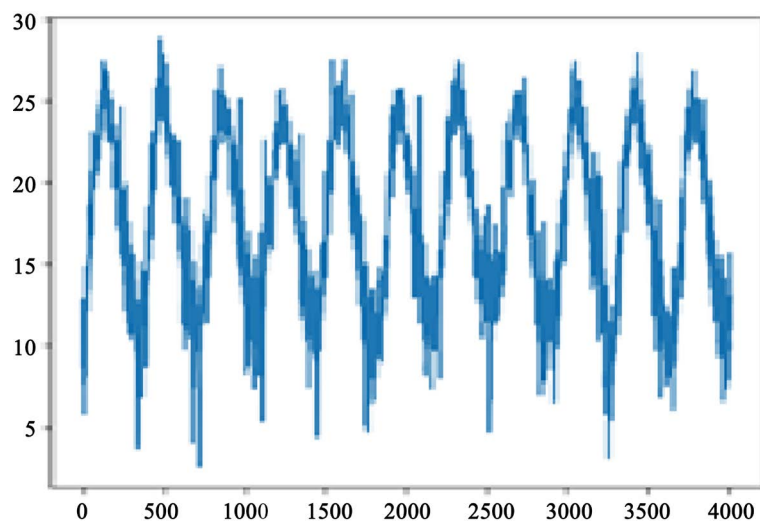**Figure 2.** Block diagram of methodology ARIMA model.

Figure 3. Representation of the seasonality in each 360 days in the temperature series.

c) Remove seasonality

This result in **Figure 4** shows that there is no more seasonality in the series because the periodicity is not seen anymore in the series.

d) Augmented Dickey Fuller (ADF) statistical test check if the given series is stationary in trend or not the ADF (Augmented Dickey Fuller) test is used [18].

Based on **Table 1**, we can consider that there is stationarity in trend (pvalues less than 0.05).

Choice of parameters p, d, q which satisfy the order of the ARIMA model **Figure 5**.

e) Parameters estimation

Choice of parameters p, d, q which satisfy the order of the ARIMA model:

- p: lag value where the PACF curve meets the zero axis, the series always jumps from the Quenouille interval.
- q: based on **Table 5** the value of the last lag where the ACF crosses very significantly the interval is 35, that means, at this point where q values is equal 35 the series remains within the Bartlett interval. The model MA(q) is considered to be our proposed model since there are fewer parameters to estimate than the AR(p).

f) Residuals Check and Test

After the selection of the ARIMA orders p = 0, d = 0 and q = 35, we cannot decide if they are the best orders parameter until we satisfy the two following criteria:

- check the ACF and PACF of the residuals issuing from the ARIMA (p, d, q) whether they are confined in the confidence interval;

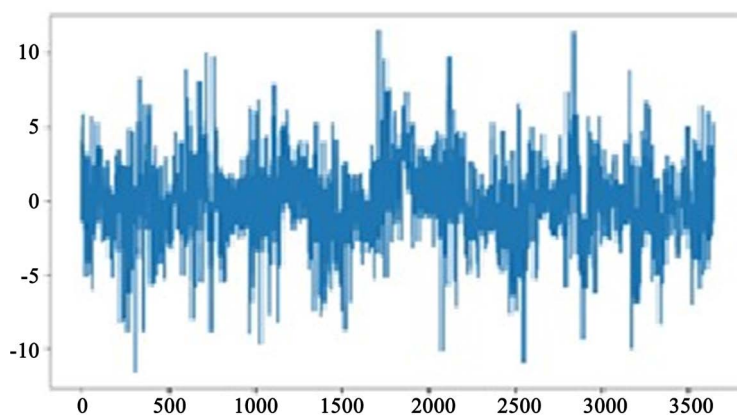**Figure 4.** Representation of the series after removing the seasonality.

Table 1. ADF statistical test.

ADF Statistic	−10.4917
p-Value	0.00000
Critical Values	
1%:	−3.432
5%:	−2.862
10%	−2.567

- apply the LjungBox statistical test to test if the residuals are white noise or not.

Based on **Figure 6**, ACF and PACF of the residuals issuing from the ARIMA (0, 0, 35) are confined in the confidence interval, that means the first criteria is satisfied.
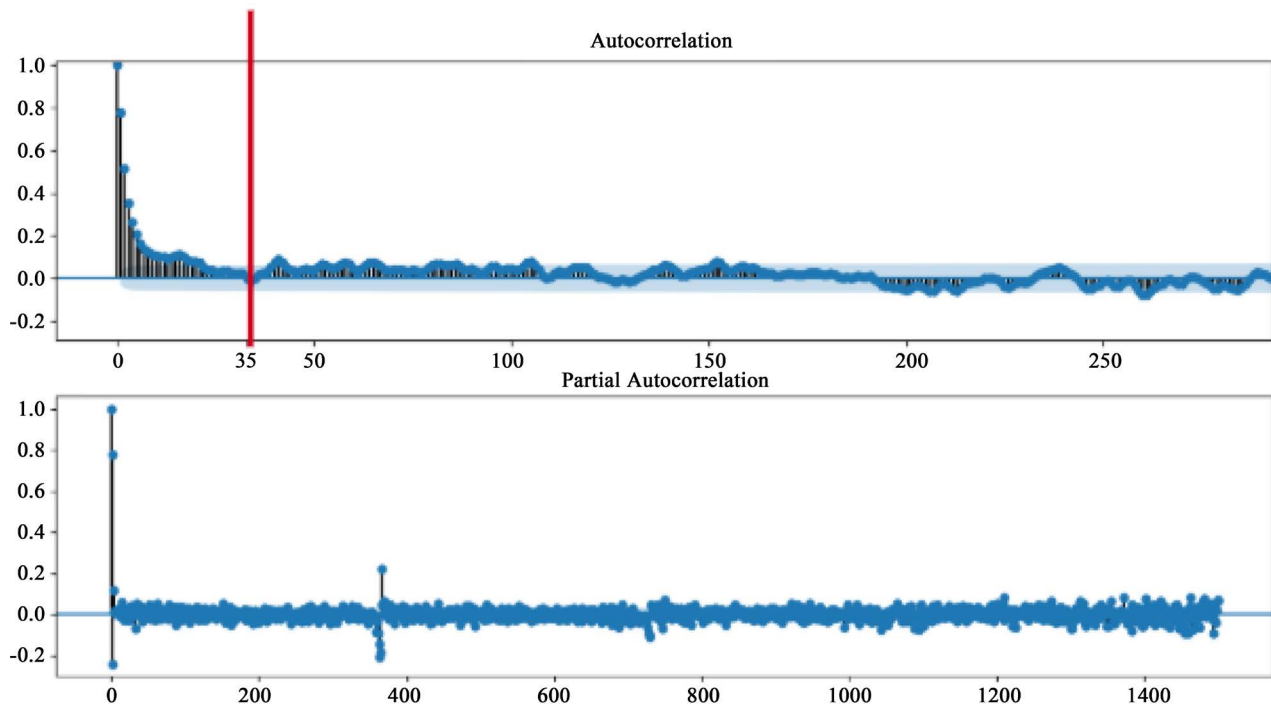

Figure 5. Plot ACF and PACF on which are based the selection of the ARIMA orders parameters q and p respectively.
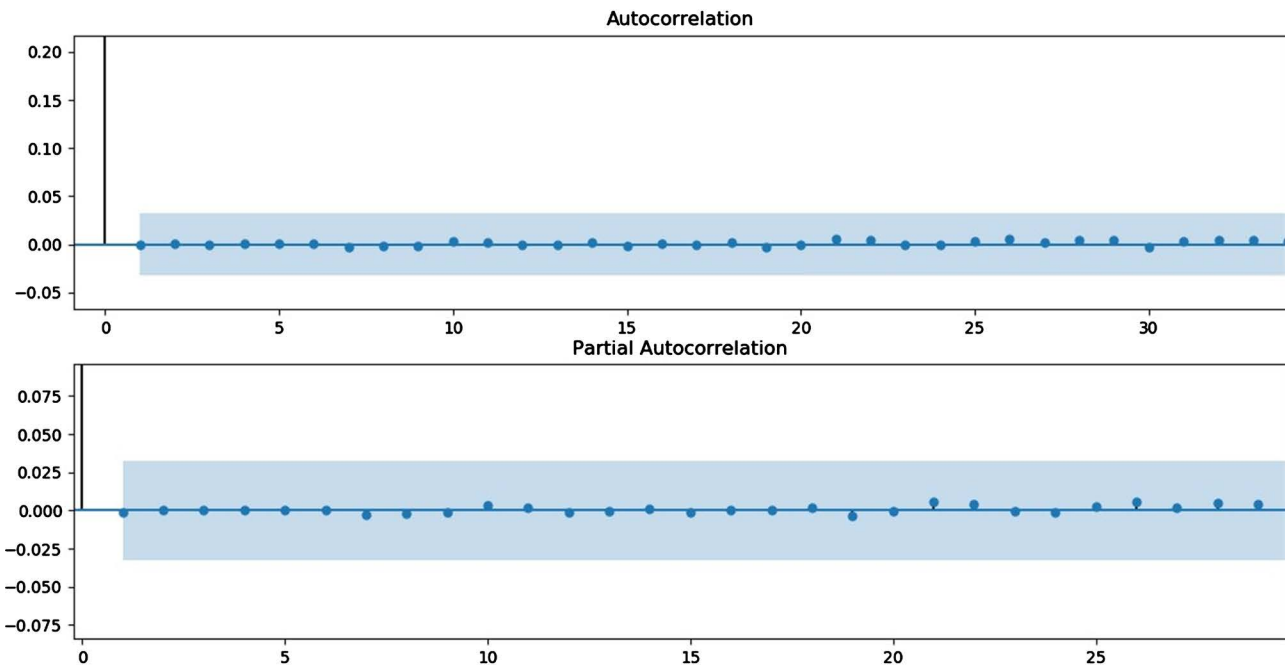

Figure 6. Plot ACF and PACF for the residuals issued form ARIMA model on which is based the decision whether they are white noise or not.

After the satisfaction of the criteria of ACF and PACF for the residuals, we proceed to check whether the residuals are white noise or not.

Ljung-Box statistical test should be applied to check if the residuals are considered as white noise [10]. Based on Table 2 that contains 35 values issued from Ljung-Box statistical test, all p values in the table are greater than 0.05 significant Value, this implies that residuals are white noise.

g) Model Ready for Prediction

The proposed ARIMA (0, 0, 35) model is ready to forecast and predict future values of daily temperature to prior five days of January 2017 as well as of August 2016 at Beirut meteorological station but with different coefficients.

In fact the model was built on all the data from 01/01/2006 to 31/12/2016 and can predict five days of January 2019 but to predict August 2016 the proposed model had been built on data from 01/01/2006 to 31/07/2016 is more usual.

4.3. ARIMA Parallel

The proposed model ARIMA sequence has the following order (0, 0, 35). The order of the model, in question, is defined by a Moving Average of high order q = 35 which poses a high complexity requires heavy resources of the computers to be able to execute this type of matrix computations, and a long execution time. To face this problem, the proposed solution is constituted of three parts:

PART I: Architecture proposed for system parallelism, Figure 7.

PART II: Sub-models ARIMA coefficients calculation, Figure 7.

PART III: RAM of my computer has been increased to reach 20 GB.

4.3.1. PART I: Architecture Proposed for System Parallelism

The proposed model is an ARIMA (0, 0, 35). The highest order 35 has shown complexity while creating the associated model coefficients. This diagram flow has been suggested as a solution for this complexity.

First step, start by dividing the highest order 35 into 7 lists, each one consists of 5 orders, that mean dividing the model ARIMA (0, 0, 35) into 7 sub-models: "ARIMA (0, 0, 1..7), ARIMA (0, 0, 8..14), ARIMA (0, 0, 15..21), ARIMA (0, 0, 22..28), ARIMA (0, 0, 29..35)". This step was followed by creating 7 threads in

Table 2. Ljung-Box statistical test is applied to confirm the result issued from Figure 6 that residues are considered as white noise.

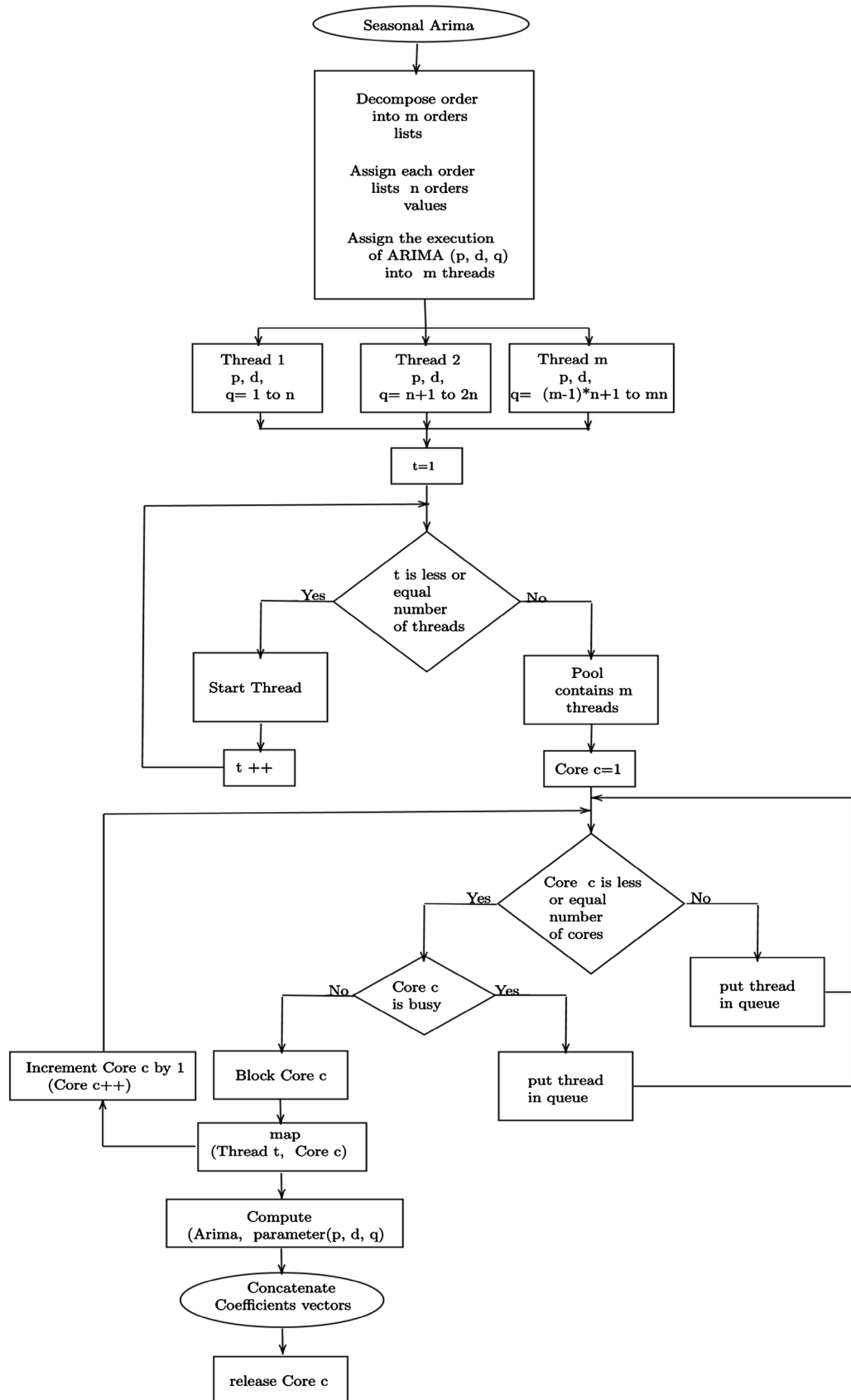| p-values (Ljung-Box statistical test) |
|:---:|
| 0.957 |
| 0.998 |
| 0.997 |
| 0.999 |
| 1 |
| 1 |
| ... |

Figure 7. Diagram flow chart represents the algorithm of ARIMA parallel execution.

order to map each sub-models into one of creating threads assigned to execute it; These created threads must be located in the Read Access Memory (RAM), for this reason a memory zone called Pool thread is allocated in the RAM which contains the threads. After associating each of the sub-models to the appropriate thread, distribute the job of each thread into a free core (not busy) implemented in the machine. The below algorithm represents the architecture proposed for system parallelism.

function Arima Parallel(*p*, *d*, *q*1: *order MA*, *n*1: *n consecutive values of q*)
q ← *list*[1 *to q*1];
n ← *n*1;

/*decompose order q into m lists where each one has n consecutive values of q*/
m ← *q/n*;
section ← [];
core c;
thread t;
buffer thread ← *[];*
pool memory ← *[];*
array coefficient ← *[];*
matrix coefficient ← *[];*

Begin
/*section contains m lists where each list has a length n*/
section ← *[(p, d, q1* ← *1 to n), (p, d, q2* ← *n+1 to 2n),..., (p, d, qm* ←*(m − n) ∗ n +*
1 to m ∗ n)];

t ← *1;*

while (t ≤ m) do
start thread t;
buffer thread ← *add to list(buffer thread[], thread t);*
t ← *t + 1;*
end while

/*pool memory is a memory zone, allocated for all threads, available to compute ARIMA (p, d, m list of order qn)*/
pool memory ← *allocate memory zone threads (buffer thread);*
c ← *1;*
j ← *1;*
s ← *1;*

while (c ≤ number of cores in the device) do
var thread ← *pool memory[j];*

if (mode thread (var thread) is busy)
queue ← var thread;

else
block thread (var thread);

/*parameter ARIMA= section[1]=(p, d, q1= 1 to n)*/
parameter ARIMA p d q ← section[s];

/*parameter ARIMA= section[2]=(p, d, q2= n+1 to 2n)*/
s ← s + 1;

/*assign the thread into the free processor in order to achieve the execution job*/
map thread(var thread, core c);

/*next core */
c ← c + 1;

/* … Sub-models ARIMA coefficient's calculation algorithm (PART II)... */

end if

end while

return matrix coefficient;
End function.

4.3.2. PART II: Sub-Models ARIMA Coefficient's Calculation

The second part of the diagram is dedicated to calculate the coefficients of each sub-models ARIMA and to join the calculated coefficients among them to get the final coefficients of the proposed model ARIMA (0, 0, 35); This process has been achieved based on Compute(ARIMA, parameter (p, d, q)) method. For example, in order to calculate the coefficients matrix of sub-model ARIMA (0, 0, 1..7), consider the following instructions:

- arima order = (0, 0, 1..7)
- model = ARIMA(series, order = arima order)
- coefficientsCalculate = calculateCoeff (start params = loop i in range (1 to 7))
- matrixCoefficients = CoefficientsCalculate.fittedvalues
- repeated to other submodels ARIMA.

First, the algorithm finds a free Core "C", then locks it after mapping a thread "t" to it. This process keeps repeating onto the other threads. When a thread finishes its job, it concatenates the generated coefficient vector with the

other threads' coefficient vector and release the core "C" (unblock). After finishing the job for each thread and joining their coefficient vectors among them, we get the coefficients of the proposed ARIMA (0, 0, 35) model. This scenario is represented by the following algorithm section:

```
/*execute and calculate the result of the function ARIMA(p, d q)*/
array coefficient ← compute(ARIMA, parameter_ARIMA_p _d_q)

/*concatenate all results issuing from each thread after its individual job is
done*/
matrix coefficient ← add to list(matrix coefficient, array coefficient);

/*release the core resource after terminating its executed job*/
release thread(var thread);

/*take another thread from the pool memory */
j ← j + 1;
```

Finally this algorithm is applied on our case as follow:
```
Begin
p order ← 0;
d order ← 0;
q order ← 35;
n lists of q orders ← 5;

Coefficient vector ← Arima P arallel(p, d, q, n lists of q orders);
End
```

5. Result

The values given by the numerical prediction model ARPEGE presented in Table 3 and Table 4 had been taken from the weather bulletin prepared by the forecasters in the meteorological service at BRHIA. In fact, ARPEGE archives maps and graphs related to the meteorological parameters back to 4 days passed a current date which is considered to be a critical time. After this critical time, maps and graphs are immediately removed from the storage. Forecasters, in the meteorological service, during the preparation of the daily weather bulletin, they check the graphs and maps of several meteorological parameters in order to write down the forecasted values given by ARPEGE on the weather bulletin. This process should be achieved before the deleting of the maps and graphs at the critical time, one of these parameters is the temperature

Table 3 shows the results of accuracy issued from ARIMA and ARPEGE prediction models for the first five days of January 2017. The real values for the temperature for the first day of January 2017 are 10.5, whereas the value forecasted by ARIMA is 10.47 and ARPEGE is 7.56. By comparison, we deduced that

Table 3. Results of accuracy issued from ARIMA and ARPEGE prediction models for the first five days of January 2017 (Beirut Meteorological Station).

Date	Real_Values Temperature	ARIMA Result	ARPEGE Result	Accuracy ARIMA	Accuracy ARPEGE
1/1/2017	10.5	10.47	7.56	99.7%	72%
2/1/2017	9.7	9.46	6.32	97.5%	65.1%
3/1/2017	10	9.72	6.14	97.2%	61.4%
4/1/2017	9.7	7.21	5.8	74.3%	59.7%

Table 4. Results of accuracy issued from ARIMA model based on all years from 01/01/2006 to 31/12/201 and ARPEGE prediction models for the first five days of August 2016 (Beirut Meteorological Station).

Date	Real Values Temperature	ARIMA Result	ARPEGE Result	Accuracy ARIMA	Accuracy ARPEGE
8/1/2016	26.7	26.49	24.5	99.2%	91.7%
8/2/2016	25.5	25.76	23.1	98.9%	90.5%
8/3/2016	24.1	23.8	22.1	98.7%	91.7%
8/4/2016	23.1	22.6	20.4	97.8%	88.3%
8/5/2016	25.5	24.7	21.8	96.8%	85.4%

the accuracy given by ARIMA is 99.9% which is better than the accuracy given by ARPEGE which is equal to 72%. For the second day of January 2017, we deduced that the accuracy given by ARIMA is 97.5% which is better than the accuracy given by ARPEGE which is equal to 65.1%. For the third day of January 2017, we deduced that the accuracy given by ARIMA is 97.2% which is better than the accuracy given by ARPEGE which is equal to 61.4%. For the fourth day of January 2017, we deduced that the accuracy given by ARIMA is 74.3% which is better than the accuracy given by ARPEGE which is equal to 59.7%. Finally for the fifth day of January 2017, we deduced that the accuracy given by ARIMA is 86.1% which is better than the accuracy given by ARPEGE which is equal to 80.6%.

Table 4 shows the results of accuracy issued from ARIMA and ARPEGE prediction models for the first five days of August 2016. The real value for the temperature for the first day of August 2016 is 26.7, whereas the value forecasted by ARIMA is 26.49 and ARPEGE is 24.5. By comparison, we deduced that the accuracy given by ARIMA is 99.2% which is better than the accuracy given by ARPEGE which is equal to 91.7%. For the second day of August 2016, we deduced that the accuracy given by ARIMA is 98.9% which is better than the accuracy given by ARPEGE which is equal to 90.5%. For the third day of August 2016, we deduced that the accuracy given by ARIMA is 98.7% which is better than the accuracy given by ARPEGE which is equal to 91.7%. For the fourth day of August 2016, we deduced that the accuracy given by ARIMA is 97.8% which is better than the accuracy given by ARPEGE which is equal to 88.3%. Finally for

the fifth day of January 2017, we deduced that the accuracy given by ARIMA is 96.81% which is superior to the accuracy given by ARPEGE which is equal to 85.4%.

Table 5 shows a comparison between the proposed prediction models ARIMA sequence, ARIMA parallel and ARPEGE. This comparison includes consumption hardware resource in addition to the operating system that supports them. We started by the execution time parameters, this table shows that ARIMA sequence model takes 36 hours to finish the execution, whereas ARIMA parallel model takes 10 hours as for ARPEGE the forecasts the duration is 102 hours. Regarding the CPU consumption ARIMA sequence model takes 100% of CPU capacity, whereas ARIMA parallel model takes 20% of CPU capacity as for ARPEGE 130 nodes containing 40 Broadwell CPU at 2.2 GHz during 52 mn. In other hand ARIMA sequence model needs 20 GB of RAM allocation, whereas ARIMA parallel model needs 15 GB of RAM allocation as for ARPEGE it uses 8.32 TB of RAM. About the operating system Microsoft Windows 10 Pro support ARIMA sequence and ARIMA parallel whereas Linux Redhat improved by the BULL team ARPEGE.

6. Conclusion

The temperature is considered an important parameter to forecast in the meteorological service in Lebanon since it influences many sectors in Lebanon such as economy, tourism, agriculture, etc. According to the results, the accuracy of predictions made for temperature by ARIMA model is better than that of the ARPEGE compared to it in two seasons: summer 2016 represented by the August month and winter season which is represented by January 2017, the Ljung-Box statistical test proves the power of the accuracy and assures that the parameters p, d and q fit well to the proposed ARIMA model. This was conducted through the test of the residuals that are considered as white noise. Furthermore,

Table 5. Result of resource consumption.

Date	ARIMA Sequence Order (0, 1, 35)	ARIMA Parallel Order (0, 1,)	ARPEGE Result
Execution Time	36 Hours	10 Hours	For the 00 UTC run, the duration is 102 hours
CPU Consume	100%	20%	130 nodes containing 40 Broad well CPU at 2.2 GHz
Operating System Platform	Microsoft Windows 10 Pro	Microsoft Windows 10 Pro	Linux Redhat improved by the BULL team
CPU Platform	CPU Intel(R) Core(TM) i7-6500U CPU@2.50 GHz 2.59 GHz	CPU Intel(R) Core(TM) i7-6500U CPU@2.50 GHz 2.59 GHz	HPC BULL B710 DLC
RAM Platform	20 GB	15 GB	8.32 TB of RAM

considering the hardware resources consumption, the result also shows that the ARIMA model takes 10 hours as execution time which is better than ARPEGE that takes 102 hours. Moreover, considering the consumption resource, ARIMA requires 20 GB space from the Read Access Memory (RAM) which is much better than the reservation required, made by ARPEGE during the execution which is equal to 8.32 TB (Terra Bytes). Finally, it is essential to mention that since this article shows some advantages of ARIMA on ARPEGE, the ARPEGE is considered to be one of the important numerical models that are widely adopted by many Arab's and Europ's meteorological departments. Also ARPEGE model may show more accuracy in the prediction in this department, but the type of geographical topography in Lebanon makes the mission of accuracy given by ARPEGE and many other numerical weather prediction models (GFS, ECMWF, etc.) a very complicated one.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Lopardo, G., Antonsen, I., Bell, S., Benyon, R., Boese, N., del Campo, D., Dobre, M., Drnovsek, J., Elkatmis, A., Georgin, E., Grudniewicz, E., Heinonen, M., Holstein-Rathlou, C., Johansson, J., Klason, P., Knorova, R., Melvad, C., Merrison, J., Migal, K., de Podesta, M., Saatho, H., Smorgon, D., Sparasci, F., Strnad, R., Szmyrka-Grzebyk, A., Merlone, A. and Vuillermoz, E. (2013) A New Challenge for Meteorological Measurements: The Meteomet Project—Metrology for Meteorology. *AIP Conference Proceedings*, **1552**, 1030.

[2] Karthicky, S., Malathi, D., Sudarsan, J., Arun, C. and Sheikh, F. (2016) Analysis of Data Mining Techniques for Weather Prediction. *Indian Journal of Science and Technology*, **9**, 270-290.

[3] Jadhawar, B.A. and Mandale, A. (2015) Weather Forecast Prediction: A Data Mining Application. *International Journal of Engineering Research and General Science*, **3**, 1279-1284.

[4] Ghosh, S.K. and Das, M. (2017) semBnet: A Semantic Bayesian Network for Multivariate Prediction of Meteorological Time Series Data. *Pattern Recognition Letters*, **93**, 192-201.

[5] Hayati, M. and Mohebi, Z. (2007) Application of Artificial Neural Networks for Temperature Forecasting. *International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering*, **1**, 654-658.

[6] Abhisheka, K., *et al.* (2012) Weather Forecasting Model Using Artificial Neural Network. *Procedia Technology*, **4**, 311-318.
https://doi.org/10.1016/j.protcy.2012.05.047

[7] Phan, T.-T.-H., *et al.* (2018) Comparative Study on Univariate Forecasting Methods for Meteorological Time Series. *26th European Signal Processing Conference* (*EUSIPCO*), Rome, 3-7 September 2018, pages.

[8] Deshpande, A. and Rodrigues, J. (2017) Prediction of Rainfall for All the States of India Using Auto-Regressive Integrated Moving Average Model and Multiple Li-

near Regression. *International Conference on Computing, Communication, Control and Automation* (*ICCUBEA*), Pune, 17-18 August 2017, 38-51.

[9] Radnoti, G., Horanyi, A. and Ihasz, I. (1996) ARPEGE/ALADIN: A Numerical Weather Prediction Model for Central-Europe with the Participation of the Hungarian Meteorological Service. *Idojaras*, **100**, 277-301.

[10] Lin, W.-S., Lin, C.-Y. and Wu, C.-H. (2018) Using Time Series Analysis to Predict Affecting Factors of Thallium Myocardial Perfusion Scan Service Usage Frequency. *IEEE International Conference on Applied System Innovation*, Chiba, 13-17 April 2018, 39-45.

[11] Kalnay, E. (2003) Atmospheric Modeling, Data Assimilation and Predictability. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511802270

[12] Masek, J., Smolikova, P., Yessad, K., Smith, C., Brozkova, R., Benard, P., Vivoda, J. and Geleyn, J.-F. (2009) Dynamical Kernel of the Aladin-NH Spectral Limited-Area Model: Revised Formulation and Sensitivity Experiments. *Quarterly Journal of the Royal Meteorological Society*, **136**, 155-169.

[13] Ricard, D., Leger, J., Brousseau, P. and Seity, Y. (2016) Improvement of the Forecast of Convective Activity from the AROME-France System. *Quarterly Journal of the Royal Meteorological Society*, **142**, 2231-2243. https://doi.org/10.1002/qj.2822

[14] Tiao, G.C. (2015) Time Series: ARIMA Methods. In: *International Encyclopedia of the Social Behavioral Sciences*, 2nd Edition, Elsevier, Amsterdam, 24. https://doi.org/10.1016/B978-0-08-097086-8.42182-3

[15] Lin, Y.H., Chang, D.-F. and Nyeu, F.-Y. (2019) Detecting the Issues of Population Aging by Using ARIMA Model. *ICIC Express Letters*, **10**, 39-45.

[16] Liu, Y., Masum, S. and Chiverton, J. (2017) Comparative Analysis of the Outcomes of Differing Time Series Forecasting Strategies. 13*th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, Guilin, 29-31 July 2017, 290-302.

[17] Manthalkar, R. and Bendrea, M. (2019) Time Series Decomposition and Predictive Analytics Using MapReduce Framework. *Expert Systems with Applications*, **116**, 108-120.

[18] Nugroho, A. and Simanjuntak, B.H. (2014) ARMA (Autoregressive Moving Average) Model for Prediction of Rainfall in Regency of Semarang—Central Java—Republic of Indonesia. *International Journal of Computer Science Issues*, **11**, 27-32.