On the Sulina Precipitation Data Analysis Using the ARMA models and a Neural Network Technique

ALINA BARBULESCU, ELENA PELICAN
Faculty of Mathematics and Computer Science
Ovidius University
124 Mamaia Blv., Constanta 900527
ROMANIA

Abstract: - Weather modification is a topic of substantial worldwide interest for all countries. Only a small number of studies is devoted to model the precipitation evolution in different regions of Romania, including the Black Sea coast [4]. This study concerns the data analysis of Sulina series, using an ARMA model, and a neural networks technique.

time series, ARMA model, break point, multi layer percepton.

1 Introduction

Dobrudja is situated in the South – East of Romania, between the Black Sea and the lower Danube River and Sulina station is situated at 13 km offshore.

The studied data represent monthly precipitation, collected at Sulina station, in the period 1965 – 2003 and represented in Fig.1.

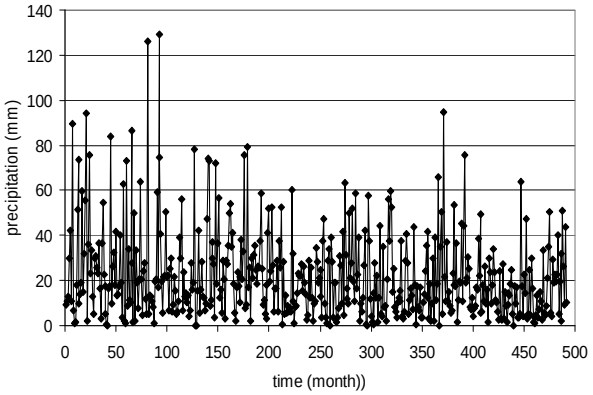

Fig.1. The data series

A time series model for the observed data (x_t) is a specification of the joint distributions of a sequence of random variables (X_t) of which (x_t) is postulated to be a realization.

Let (X_t) be a time series with the expectance $E(X_t) < \infty$. The mean function of (X_t) is defined by $\mu_X(t) = E(X_t)$ and the covariance function of (X_t) is

$\gamma_X(r,s) = Cov(X_r, X_s) = E[(X_r - \mu_X(r))(X_s - \mu_X(s))]$,

for all integers *r* and *s.*

The time series (X_t) is weakly stationary if $\mu_X(t)$ is independent of *t,* and $\gamma_X(t+h,t)$ is independent of *t* or each *h* ($h \in \mathrm{N}^*$).

Let (X_t) be a time series. The autocovariance function of (X_t) at lag h ($h \in \mathrm{N}^*$) is $\gamma_X(h) = Cov(X_{t+h}, X_t)$ and the autocorrelation function of (X_t) at lag *h* is $\rho_X(h) = \frac{\gamma_X(h)}{\gamma_X(0)}$.

Let $x_1, \ldots, x_n$ be observations of a time series. The empiric autocorrelation function, ACF, is:

$$\hat{\rho}(h) = \frac{\sum_{t=1}^{n-|h|}\left(x_t - \bar{x}\right)\left(x_{t+|h|} - \bar{x}\right)}{\sum_{t=1}^{n}\left(x_t - \bar{x}\right)^2}.$$

The random variables X and Y, with $E(X) < \infty$, $E(Y) < \infty$ are called uncorrelated if their covariance is zero, i.e. $E[(X - E(X))(Y - E(Y))] = 0$.

A sequence (X_t) of uncorrelated random variables, each with zero mean and the variance σ^2 is called a white noise. If the mean is not zero the noise is called coloured [2].

Let us consider the operators defined by:

$$B(X_t) = X_{t-1},$$

$$\Phi(B) = 1 - \varphi_1 B - \ldots - \varphi_p B^p, \varphi_p \neq 0,$$

$$\Theta(B) = 1 - \theta_1 B - \ldots - \theta_q B^q, \theta_q \neq 0,$$

$$\Delta^d(X_t) = (1-B)^d X_t.$$

The process (X_t) is said to be an integrated autoregressive moving average process, denoted ARIMA(*p, d, q*), if $\Phi(B)\Delta^d X_t = \Theta(B)\varepsilon_t$, where the absolute values of the roots of Φ and Θ are greater than 1 and (ε_t) is a white noise.

An ARIMA(*p, d, q*) process is called an autoregressive of p order process and is denoted by *AR*

(p), *if* $d = 0 = q$. An ARIMA(p, d, q) process is a moving average process and is denoted by MA(q) if $d = 0 = p$. The ARIMA(p, d, q) process is an autoregressive moving average process of p and q orders and is denoted by ARMA(p, q) if $d = 0$ [6].

2 Preliminary analysis

In order to develop different models for this series, a preliminary analysis is necessary.

After the application of Kolmogorov-Smirnov test [8], the hypothesis that the series is normally distributed is rejected.

After the same test applied to series obtained by a Box-Cox transformation:

$$Z_t = \frac{X_t^{\lambda} - 1}{\lambda},$$

with $\lambda = 0.34$, we accept the hypothesis that the series (Z_t) is normally distributed.

After the autocorrelogram (Fig.2) determination, the null hypothesis of randomness of the series is rejected at the confidence level of 95%.

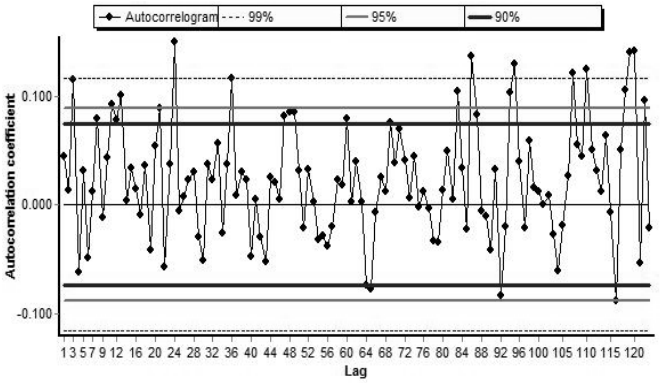

Fig.2. Autocorrelogram

The homoscedasticity (the same variance) hypothesis was rejected after the application of Bartlett test [8].

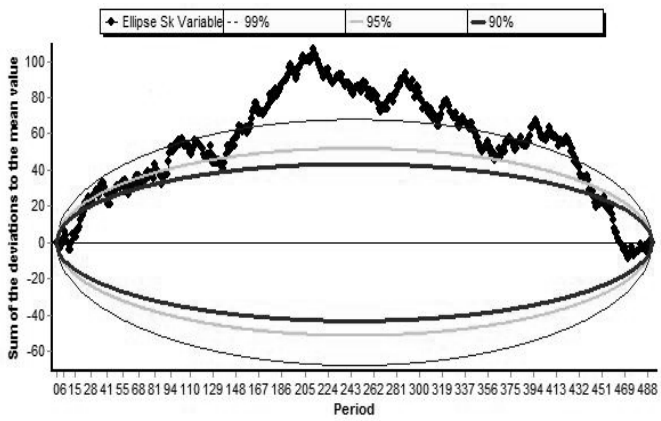

Fig. 3. Bois' ellipse

In order to determine the breaks Buishand [3], Pettitt [8], Lee and Heghinian tests [7] were applied. The null hypothesis to be tested was:

H_0: "There is no break in the time series".

The results of these tests were:

- the Buishand test and Bois' ellipse: H_0 accepted at confidence level of 99% and rejected at 95% (Fig.3);
- Pettitt's test: H_0 rejected and the break point position is 212 (Fig.4);
- the Lee and Heghinian test: H_0 rejected and the break point position is 212.

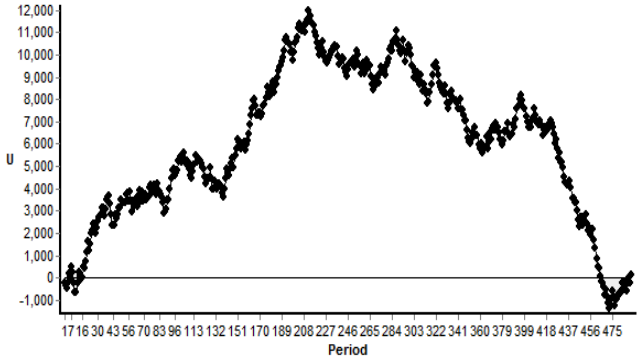

Fig.4. Pettitt's test

3 ARMA Model

In what follows we analyse the subseries of the data 1-212 and 213 – 492 and if they don't form Gaussian noises, we determine models for them.

Sub-series 1: *January* 1965 – *August* 1982

Since after the Box-Cox transformation the series was normally distributed, we analyze the data sub-series transformed.

We accept the hypothesis that the sub-series 1 is normally distributed, since the observed values (represented by circles) are superposed on the straight line represents the expected normal values, as it can be seen in Fig.5.

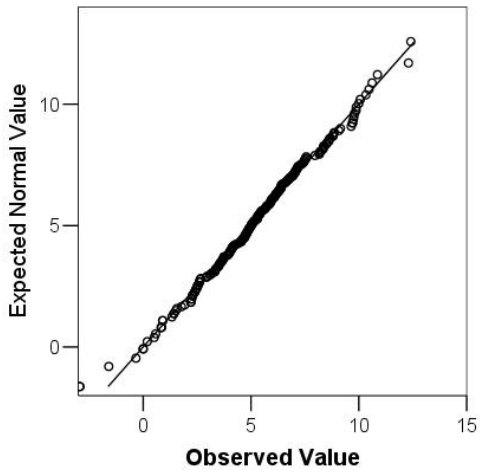

Fig.5 Q-Q plots of subseries 1

Analyzing the ACF (Fig.6), we accept the hypothesis that this subseries data is not correlated.

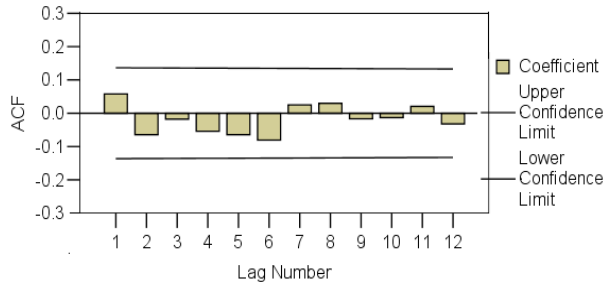

Fig.6. ACF of subseries 1

After applying Bartlett's test, we accept the hypothesis that the sub - series 1 is also homoskedastic.

So, the first sub - series forms a Guassian noise.

Sub-series 2: September 1982 – *December* 2005

We work with the sub-series obtained after the Box-Cox transformation with $\lambda = 0.34$ was applied.

Analyzing the Q-Q plot and ACF of this sub-series, we accept the hypothesis that the series is normaly distributed and correlated.

After subtracting the mean, the best model determined is an ARMA(1,1), with the equation:

$$Z_t = -0.7234 Z_{t-1} + \varepsilon_t + 0.90232 \varepsilon_{t-1},\ t \geq 2,$$

where (ε_t) is a white noise.

4 Neural Networks Paradigm

Recently, neural computing emerged as a practical technology, with many successful applications in different areas. Most of these applications make use of feed-forward network architectures such as the multi-layer perceptron and the radial basis function network. The sum-and-threshold model of a neuron used in such methods, arises naturally as the optimal discriminant function needed to distinguish two classes whose distributions are normal with equal covariance matrices. Similarly, the familiar logistic sigmoid is precisely the function needed to allow the output of a network to be interpreted as a probability, when the distribution of hidden unit activations is governed by a member of the exponential family.

4.1 Implementation Details

In our case, the steps to use multi layer neural networks (MLP) are the following:

I. *Design*

– Define the number of nodes in each layer (input, hidden, output);

– Define the transfer functions (e.g. *logsig, tansig, purelin*);

– Tell to Matlab what optimization (or training) routine to use.

Generally, we use either *traingdx*, which is gradient descent, or *trainlm* (for Levenburg-Marquardt, which is a combination of gradient descent and Newton's Method).

As a optional part, we can define the error function (Mean Square Error is the default), plot the progress of training, etc.

II. *Training*: made in order to optimize the error function. This process determines the "best" set of weights and bias for the data set.

III. *Testing*: made to see if the network has found a good balance between memorization (accuracy) and generalization.

The MLP architecture used in our application can be depicted as in Fig.7.

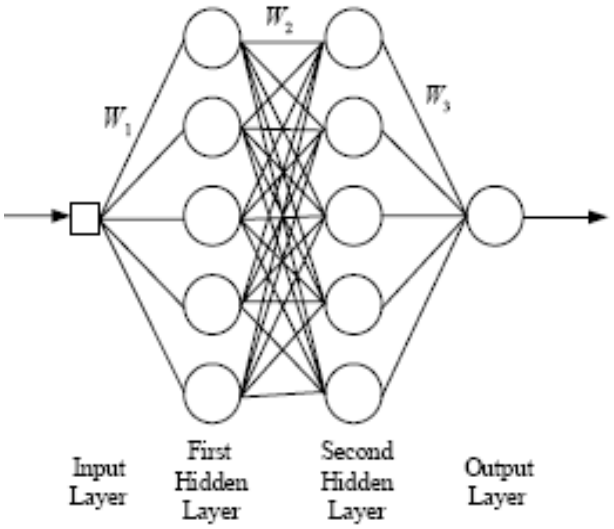

Fig.7. MLP Architecture

In order to create a feed - forward neural network (NN), we used the following

- Net=newff([-1 2; 0 5], [3 1], {'tansig', 'purelin'}, 'traingd');
- Neural Model – *tansig*, *logsig, purelin*;
- Training method – *traingd, traingdm:*

Training: [net, tr] =train(net,p,t) :

- Simulation: A=sim(net,q);

Also, the data set (for each sub-series) was splitted in three parts: a training set (first 65% of the monthly values), a validation set (the next 15% of the monthly values), and the test set (the last 20% of the monthly values), as reported in the literature.

A two-layer feed forward network was used, with an output layer and hidden layers. The hidden layer has 5 to 10 *tansig* or *purelin* neurons, and the output one has one *tansig* neuron.

As a training routine, we used the *trainlm* Matlab built-in function.

The obtained results are comparable with the ones obtained with ARIMA model, previously presented.

The data are given in the following table.

Table 1. The forecasts for the sub-series 1 and 2

Sub-series1	Sub-series 2
1.5229	1.5539
1.1228	2.0691
0.6711	0.4404
1.3874	0
3.4586	2.4459
-1.3422	0.8060
2.1166	1.4126
4.8978	0.2991
0.4969	0.2825
0.6001	0.4321
5.1108	5.3098
1.0841	1.4625
1.6649	0.3490
0.1420	1.8697
18.9295	18.1800
1.3487	3.9304
-2.0392	0.2410
2.2908	0.4321
- 1.6068	2.0442
-1.1874	0.3905
- 1.6971	2.5012
-1.6520	1.3545
2.4199	0.1080
3.7879	0.2825
1.6391	0.0831
-0.6066	0.3989
0.7163	1.1218
0.3356	0.3075
1.1615	0.6066
0.2065	1.2215
2.6522	0.8891
3.3685	0.2244
0.2777	1.8003
1.5487	0.4820
3.3814	0.3822
1.7423	0.7229
0.4001	1.7367
1.0777	2.9250
1.0647	4.1963
1.8714	0.4736
0.4001	0.3573
2.4328	2.4430
1.6649	1.5705
3.3878	1.8780
	1.6287
	1.8946
	3.3571
	0.4487
	1.6370
	2.6424
	0.1911
	4.2296
	2.2020
	0.7645
	3.6562
	0.8725

The Euclidean norms were respectively Euclidean norm 23.3435 and 22.9234.

5. Conclusions

ARMA models were determined for annual precipitation at Sulina station, from Dobrudja region. A neural network was also trained to predict the behavior of the precipitations for the same data. Tests showed that the networks trained in this way (with undecomposed noisy raw signals) relive the promising role of data clustering and multi layer percepton neural networks, in precipitations forecasting.

References

[1] J. Angstenberger, *Prediction of the S&P 500 Index with Neural Network, Neural Networks and Their Applications*. John Wiley & Sons Ltd. New York, NY., 1996.

[2] P. Brockwell, R. Davies, Introduction to time series, Springer, New York, 2002.

[3] T. A. Buishard, Tests for detecting a shift in the mean of hydrological time series, *Journal of Hydrology*, 73, 1984, pp. 51 – 69.

[4] V. V. Efimov, M. V. Shokurov and V. S. Barabanov, Statistical Modeling of Monthly Anomalies of Atmospheric Precipitations for the Region of the Ukraine and Black Sea, *Physical Oceanography*, vol.12, no.1, 2002, pp. 191 – 199.

[5] B. Cannas, A. Fanni, G. Sias, S. Tronci, M.K. Zedda, River Forecasting Using Neural Networks and Wavelet Analysis, *Geophysical Research Abstracts*, Vol.7, 08651, 2005.

[6] C. Gourieroux, A. Monfort, *Series temporelles et modeles dynamiques*, Economica, Paris, 1990.

[7] A.F.S. Lee, S. M. Heghinian, A Shift Of The Mean Level In A Sequence Of Independent Normal random Variables - A Bayesian Approach, *Technometrics*, 19 (4), 1977, pp. 503 – 506.

[8] D.J. Seskin, Handbook of parametric and nonparametric statistical procedures, Chapmann and Hall/CRC, Boca Raton, 2007.

Acknowledgements: This article was supported by grant PNII ID_262.