Analysis of a Gaussian Process and Feed-Forward Neural Networks based Filter for Forecasting Short Rainfall Time Series

C. Rodriguez Rivero, J. Pucheta, H. Patiño, J. Baumgartner, S. Laboret and V. Sauchelli

***Abstract*— In this paper, an analysis of kernel (GP) and feed-forward neural networks (FFNN) based filter to forecast short rainfall time series is presented. For the FFNN, the learning rule used to adjust the filter weights is based on the Levenberg-Marquardt method and Bayesian approach by the assumption of the prior distributions. In addition, a heuristic law is used to relate the time series roughness with the tuning process. The input patterns for both NN-based and kernel models are the values of rainfall time series after applying a time-delay operator. Hence, the NN's outputs will tend to approximate the current value of the time series. The time lagged inputs of the GP and their covariance functions are both determined via a multicriteria genetic algorithm, called NSGA-II. The optimization criteria are the quantity of inputs and the filter's performance on the known data which leads to Pareto optimal solutions. Both filters -FFNN and GP Kernel- are tested over a rainfall time series obtained from La Sevillana establishment. This work proposed a comparison of well-known filter referenced in early work where the contribution resides in the analysis of the best horizon of the forecasted rainfall time series proposed by Bayesian adjustment. The performance attained is shown by the forecast of the next 15 months values of rainfall time series from La Sevillana establishment located in (-31° 1'22.46"S, 62°40'9.57"O) Balnearia, Cordoba, Argentina.**

Keywords*— Artificial Neural Networks, Rainfall Forecast, Gaussian Process, Hurst's parameter, Bayesian inference.

I. Introduction

Since the last decade, there were a lot of approaches that face the problem of forecasting using different techniques in artificial neural networks (ANN). The problematic involved in predicting variables associated with the field production in agriculture activities is strongly related to water availability. Hence, in order to attain an expected production level at the end of the campaign, such issue may be accomplished with certain accuracy. So, natural phenomena prediction becomes a challenging topic, useful for control problems from agricultural activities. The availability of water turns out to be a crucial choice when the producer decides to plant. There are several approaches based on ANN that face the rainfall forecast problem for energy demand purposes [4], for water availability by taking an ensemble of measurement points [8]. Here, methods based on FFNN's and Kernels are related with the classical nonlinear auto regressive (NAR) filter. Those schemes of NN's and GP kernel are shown by Fig. 2 and Fig. 3, respectively. In the first one, the number of filter's parameters is puts in function of the roughness of the time series between the smoothness of the time series data and the forecasted that modifies the number of the filter parameters. In the second, the predictor filter is based on kernel methods [3], specifically Gaussian Process (GP) [12]. In this case, the target before tuning the filter is to find the optimal set of time lagged inputs and the covariance function. Considering the fact that each covariance function might require a different set of time lagged inputs, both tasks are solved simultaneously by a genetic algorithm [2]. This work introduces a comparison between three different method of adjustment of parametric and non-parametric filter used in the literature (heuristic and Bayesian approach) where the contribution resides in the analysis of the horizon of the forecasted rainfall time series (the choice of 15 values was about the farmer request) applying the well-known filter referenced in early work.

II. Overview of ANN Filters

A. FFNN and Kernel Approach overview

One of the motivations for this study follows the closed-loop control scheme [10] where the controller considers future conditions for the control law's design as shown Fig. 1. In that scheme the controller takes into account the actual state of the crop by a state observer and the monthly accumulative rainfall series. However, in this paper only the controller portion concerning with the prediction system is presented by using a benchmark time series.

This article faces the analysis and employment of two methods to predict time series values by using exact and inexact interpolation.

Therefore, the parametric filter based on ANN in the learning process handles the Levenberg-Marquardt rule and takes into account the long and short term stochastic dependence of passed values of the time series to adjust at each time-stage the number of patterns, the number of iterations (i_t), and the length (l_x) of the tapped-delay line, in function of the Hurst's parameter (H) associated to the time series considered as a path of an Fractional Brownian Motion (fBm). On the other hand, the proposed technique based on Bayesian inference considers the predictive distribution obtained by integrating the predictions of the model with respect to the posterior distribution of the model parameters.

Thus, according to the stochastic characteristics of each series, H can be greater or smaller than 0.5, which means that each series tends to present long or short term dependence, respectively.

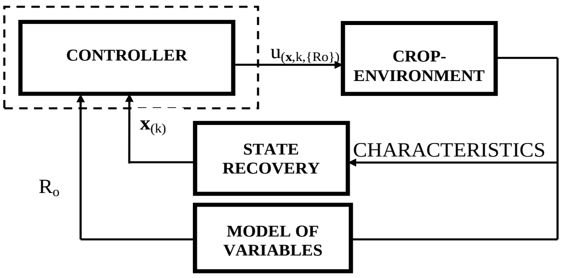

Fig. 1. The closed-loop scheme for crop development control, where a sequence of future weather conditions are considered.

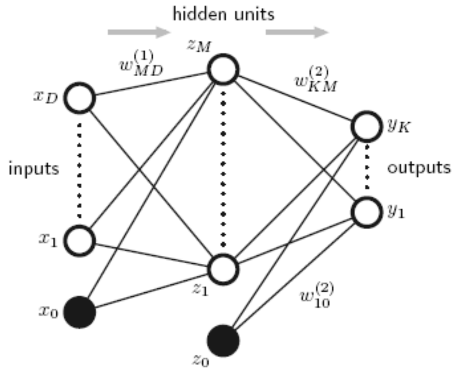

Fig. 2. The feed-forward neural network (FFNN) concept, which involves inexact interpolation.

Finally, the usage of a nonparametric filter whose time lagged inputs and the covariance function are optimized together by a multicriteria [20] , genetic algorithm leads to a maximum degree of freedom for the filter design because each covariance function might work well with a different set of time lagged inputs.

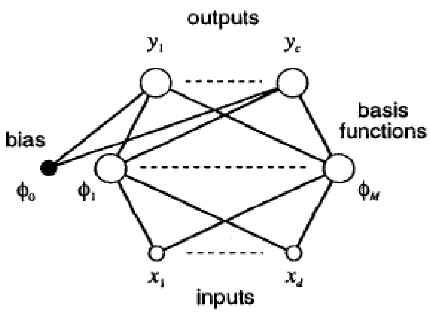

Fig. 3. The kernel neural network concept (GP), which involves exact interpolation.

In order to adjust the GP-Kernel parameters and to show that the algorithm chosen for nonparametric optimization leads to good combinations of a covariance function and a set of inputs, the rainfall time series are used to forecast the next 15 values given a historical data set of 64 values.

III. FORECASTING METHOD EMPLOYED

The main issue when forecasting a time series is how to retrieve the maximum of information from the available data. The prediction system is implemented using either a non-linear ARMA and a GP adaptive filter.

For the GP filter, depending on its covariance function a certain number of hyperparameters are available for tuning. Moreover, the inputs quantity of the GP filter needs to be determine, keeping in mind that the given time series might have long or short term dependences.

A. GP-based filter

The method to select the inputs and the covariance function of a nonparametric filter is performed as a Gaussian Process. Both problems – finding a covariance function and determining the inputs of the filter model – are discrete, because the covariance function is chosen out from a discrete set of possible functions and the inputs are defined by discrete time lags. To handle these two optimization problems a genetic, non-dominated sorting algorithm called NSGA-II [5] is used.

The NSGA-II is a multi-objective genetic algorithm that showed good results for various optimization problems [2]. It starts from a randomly chosen set of individuals whose fitness values are evaluated. Then the next generation of individuals is created via mutation and crossover operations out of the individuals with the best fitness values. Proceeding with this procedure for various generations leads to individuals with optimized fitness values. In this approach each individual represents one GP with a covariance function and certain time lags. To validate the results of this algorithm, more than one multistart can be executed. This means that the algorithm is started several times from a random population.

To apply the NSGA-II in this special case the fitness values, which serve as optimization criteria, have to be defined firstly. Thereby two goals should be kept in mind. On the one hand the filter model has to be as accurate as possible. On the other hand the number of inputs of the Gaussian Process should be as small as possible to avoid overfitting. The first fitness value describes the accuracy of the GP filter model. To validate the accuracy of a filter the given data is split into two parts. Then the filter is tuned with the first 79 values before it is evaluated on the last 15 points of the given time series. Keeping in mind that the time lags of the individuals are limited to 30, the first training point has the index 31. Otherwise it is impossible to create the input vector of the GP. Hence, 64 training points can be created out of the 79 data points.

B. FFNN based filter

In FFNN filters, the present value of the time series is used as the desired response for the adaptive filter, and the past values of the signal supply as input of the adaptive filter. Then, the adaptive filter output will be the one-step prediction signal. In Fig. 4 the block diagram of the nonlinear prediction scheme based on a NN filter is shown. Here, a prediction device is designed such that starting from a given sequence $\{x_n\}$ at time n corresponding to a time series it can be obtained the best prediction $\{x_e\}$ for the following 15 values sequence. Hence, it is proposed a predictor filter with an input vector lx, which is obtained by applying the delay operator, Z^{-1}, to the sequence $\{x_n\}$. Then, the filter output will generate x_e as the next value, that will be equal to the present value x_n. So, the prediction error at time k can be evaluated as

$$e(k) = x_n(k) - x_e(k). \qquad (1)$$

The coefficients of the FFNN filter is adjusted on-line in the learning process, by considering a criterion that modifies at each pass of the time series the number of patterns, the number of iterations and the length of the tapped-delay line, in function of the Hurst's value (H) calculated from the time series according to the stochastic behavior of the series, respectively [11].

IV. Algorithms used for training

A. *The FFNN Learning Process*

The FFNN's weights are tuned by means of the Levenberg-Marquardt rule, which considers the long and short term stochastic dependence of the time series measured by the Hurst's parameter H. The proposed learning approach consists on changing the number of patterns, the filter's length and the number of iterations in function of the parameter H for each corresponding time series. The learning process is performed using a batch model. In this case the weight updating is performed after the presentation of all training examples, which forms an epoch.

The pairs mentioned above (i_t, N_p) are modified taking into account the statistical dependence of the time series $\{x_n\}$, which is supposed to be an fBm. The dependence is evaluated by the Hurst's parameter H, which is computed using a wavelet-based method [1] [6]. Then, a heuristic adjustment for the pair (i_t, N_p) in function of H according to the membership functions is shown in Fig. 5.

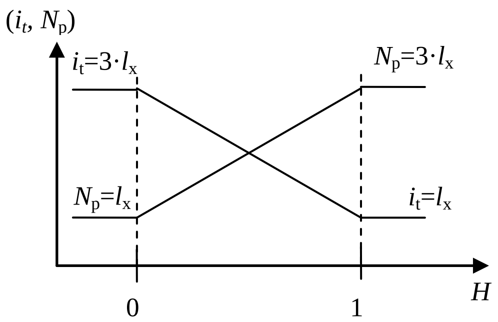

Fig. 4. Heuristic adjustment of the pair (it, Np) in terms of H after each epoch.

Finally, after each pass the number of inputs of the nonlinear filter is tuned —that is the length of tapped-delay line, according to the following heuristic criterion. After the training process is completed, both sequences —$\{x_n\}$ and $\{\{x_n\},\{x_e\}\}$, should have the same H parameter. If the error between $H(\{x_n\})$ and $H(\{\{x_n\},\{x_e\}\})$ is greater than a threshold, the value of l_x is increased (or decreased), according to $l_x \pm 1$. Explicitly,

$$l_x = l_x + 1 \cdot \text{sign}(\theta)$$

here, the threshold θ was set about 1%.

B. *Gaussian Process training*

Each individual that is evaluated by the NSGA-II consists of a certain time lags and a covariance function. Thus the training inputs of the GP have to be constructed from the given data according to the time lags of the individual before tuning the filter model quoted in [2].

In the case of a GP the model is tuned by varying the hyperparameters of the covariance function. Depending on the covariance function there are several hyperparameters that need to be adjusted to suit the training data. In other words one is interested in finding a maximum of the log marginal likelihood. Without going into detail the framework presented in [12] is used to optimize the hyperparameters. Once they are found, the training process is finished.

To evaluate a GP model one has to calculate the covariance matrix K and its inverse K^{-1}. For n given training points K has size $(n_x n)$. Their entries are the pairwise covariance of the training inputs which makes K a symmetric matrix. Supposing that the variables have a joint Gaussian distribution with zero mean, the mean prediction for an unknown input f* is given by

$$f^* = K(X^*, X) K(X, X)^{-1} f \qquad (2)$$

where X* is the unknown input, X are the training inputs and f are the training outputs. If the mean of the data is not zero, it can be transformed straightforward to fit the conditions.

A. *Bayesian approach training*

When a short or long series is being analyzed, it is important to make use of the simplest possible models. Specifically, the number of unknown parameters must be kept at a minimum.

The gamma distributions have been considered in the literature for this purpose. When a Bayesian analysis is conducted, inferences about the unknown parameters are derived from the posterior distribution. This is a probability model which describes the knowledge gained after observing a set of data. The application of the regression problem involving the correspond neural network function $y(x,w)$ and the data set consisting of N pairs, input vector lx and targets t_n (n=1,….,N)

Assuming Gaussian noise on the target, the likelihood function takes the form:

$$P(D/w,M) = \left(\frac{\beta}{2\pi}\right)^{N/2} \exp\left\{\frac{-\beta}{2}\sum_{n=1}^{N}\left\|y(x_n;w) - t_n\right\|^2\right\}, \qquad (3)$$

where β is a hyper-parameter representing the inverse of the noise variance. We consider in this work a single hidden layer of '*tanh*' units and a linear outputs units.

To complete the Bayesian approach for this work, prior information for the network is required. It is proposed to use, analogous to penalties terms, the following equation

$$P(w) = \left(2\pi w^2\right)^{-N/2} \exp\left(\frac{|w|^2}{2w^2}\right), \qquad (4)$$

assuming that the expected scale of the weights is given by *w* set by hand. This was carried out considering that the network function f(x_n+*1*,*w*) is approximately linear with respect to *w* in the vicinity of this mode, in fact, the predictive distribution for y_{n+1} will be another multivariate Gaussian.

V. Forecasting Results

The initial conditions for the FFNN learning algorithm and the GP filter were set in function of the input quantity. These initial conditions of the learning algorithm in FFNN and the the Bayesian approach were used for forecasting the time series, whose sizes have a length of 64 values each.

A. Forecast Performance measure

In order to test the proposed design procedure of the predictor, an experiment with time series obtained from rainfall time series of La Sevillana establishment and MG solution was performed.

The performance of the filter is evaluated using the mean Symmetric Mean Absolute Percent Error (SMAPE) proposed in the most of metric evaluation, defined by

$$SMAPE_S = \frac{1}{n}\sum_{t=1}^{n}\frac{|X_t - F_t|}{(X_t + F_t)/2}\cdot 100 \qquad (5)$$

where, t is the time observation, n is the test set size, s each time series, X_t and F_t are the actual and the forecast time series values at time t respectively. The SMAPE of each series s calculates the symmetric absolute error in percent between the actual X_t and its corresponding forecast F_t value, across all observations t of the test set of size n for each time series s.

B. Prediction Results for Rainfall Time Series

There are two classes of data sets: one is used for the algorithm in order to give the forecast, which consist of 64 values long. The other is used to compare either the forecast is acceptable or not where the last 15 values are used to validate the performance of the prediction system. Therefore, 64 values form the Data set, the Forecasted and the Real sets have 79 values.

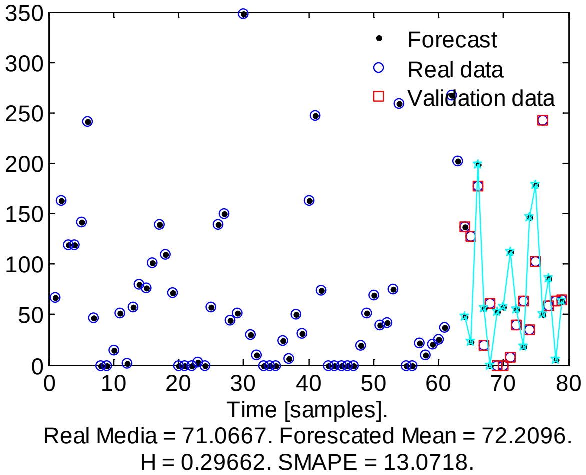

Fig. 6. FFNN predictor filter based on Bayesian approach for La Sevillana rainfall series.

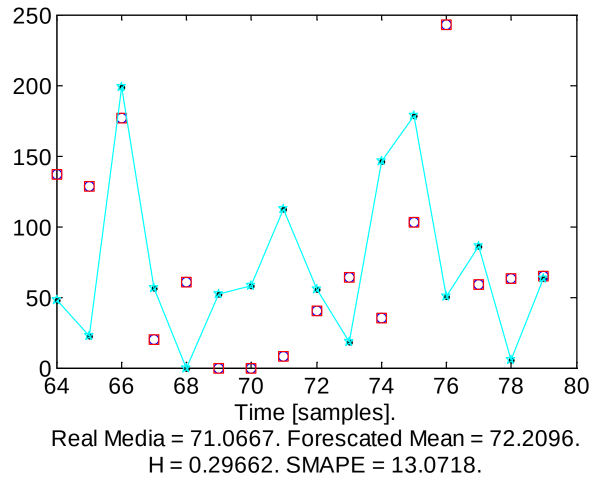

Fig. 7. Horizon of the forecast predictor filter based on Bayesian Approach.

The analysis performed by different predictor filter proposed in this work is computed using historical data of year 2004 to 2011 of La Sevillana establishment from Balenaria, Córdoba. The obtained results are shown in Fig. 6, Fig. 8 and Fig.10 for each case respectively. The horizon of the prediction is augmented in order for showing the comparison in Fig.7, Fig. 9 and Fig.11

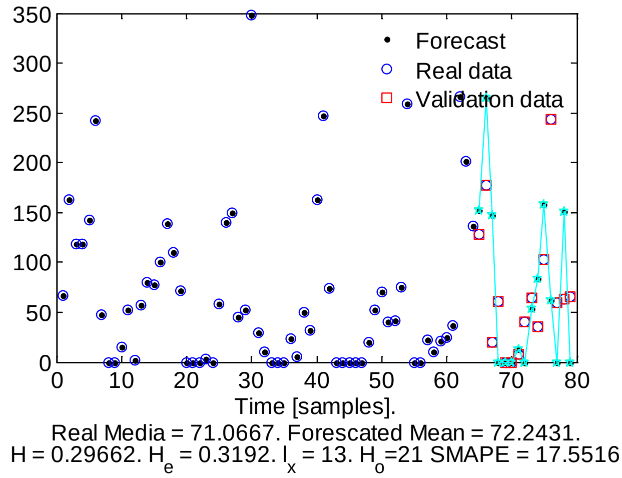

Fig. 8. FFNN predictor filter based on heuristic approach for La Sevillana rainfall series.

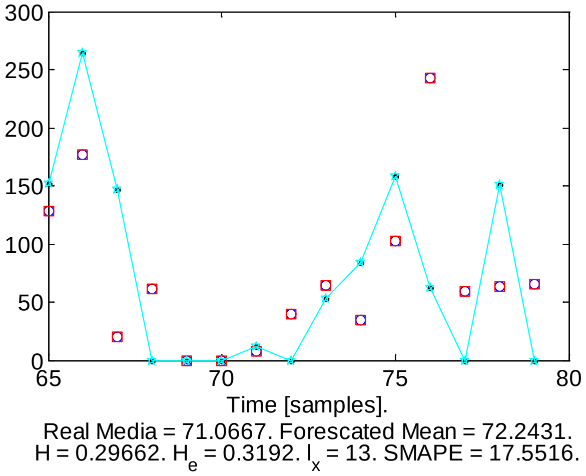

Fig. 9. Horizon of the forecast with FFNN predictor filter.

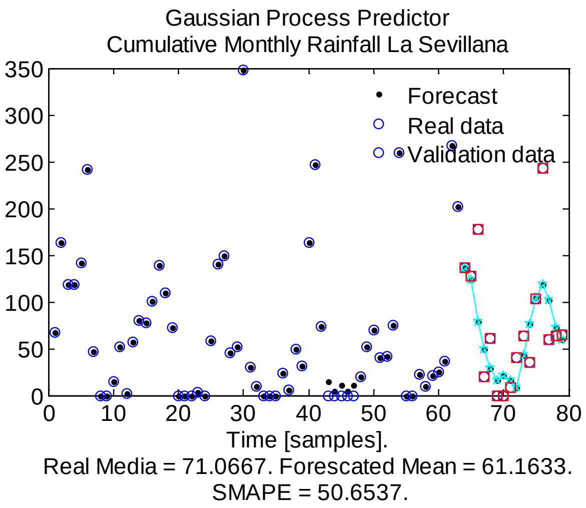

Fig. 10. Forecast with GP Kernel Filter for La Sevillana rainfall series.

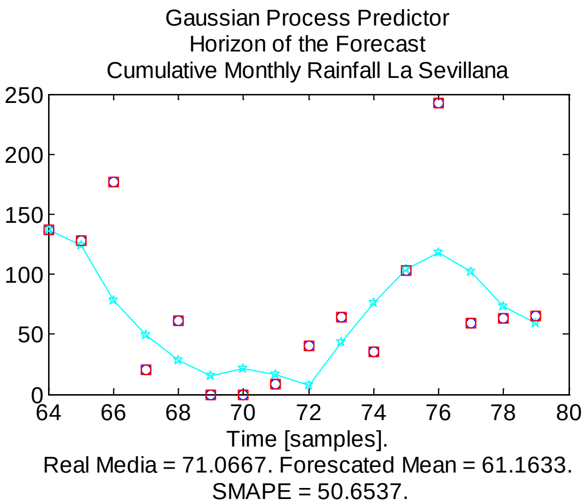

Fig. 11. Horizon of the Forecast with GP Kernel Filter.

C. Comparative Results

The performance of the NN-based predictor filter is evaluated through the SMAPE index, Eq. (8), across the time series from MG solutions.

TABLE I. RESULTS OBTAINED BY THE THREE ALGORITHMS.

Variable	Real Mean	Forecasted mean	SMAPE
Heuristic Approach FFNN	71.06	94.56	17.55
Bayesian Approach FFNN	71.06	72.20	13.07
GP Kernel Filter	71.06	61.16	50.65

The evolution of the SMAPE index for the proposed FFNN filter tuned by Bayesian approach, uses a learning algorithm with fixed parameters, and another labeled Modified FFNN filter and GP filter. The Modified FFNN filter uses the H parameter to adjust heuristically either structure of the net or parameters of the learning rule as detailed in [11]. On the other hand, the GP kernel filter uses the tuning algorithm described in [2]. It can be noted the improvement since the SMAPE index diminish from 50.65 (GP) to 17.55 (FFNN heuristic approach) and to 13.07 (FFNN Bayesian approach) which means an improvement of several times averaging over the short rainfall time series.

VI. Discussion

The analysis of the obtained results has been realized by comparing the performance of the proposed filters against GP filters, both based on NN. Although the difference between both FFNN's filters only resides in the adjustment algorithm, the coefficients that each filter has perform different behaviors. In the analyzed cases, the generation of 15 future values from 64 present values was made by each algorithm, the proposed tuned by roughness, the Bayesian approach and Gaussian process. The same initial parameters were used for each algorithm, although such parameters and filter's structure are changed by the proposed algorithm but they are not modified by the classic algorithm. In the proposed filter, the coefficients and the structure of the filter are tuned by considering their stochastic dependency. It can be noted that in each one of figures —Fig. 6 to Fig. 8— the computed value of the Hurst's parameter is denoted either H_e or H when it is obtained from the Forecasted time series or from the Data series, respectively, since the Real (future time series) are unknown. Index SMAPE is computed between the complete Real series (it includes the series Data) and the Forecasted one, as indicates the Ec. (8) for each filter. Note that the forecast's improvement is not over any given time series, which results from the use of a stochastic characteristic for generates a deterministic result, such as a prediction.

The filter proposed based on ANN were employed by FFNN, one of them, the tuning of the algorithm was made by Bayesian approach.

VII. CONCLUSION

In this work, an analysis of kernel (GP) and feed-forward neural networks (FFNN) based filter to forecast short rainfall time series was presented. The learning rule proposed to adjust the ANN weights is based on the Levenberg-Marquardt method and Bayesian approach, one modeling the number of NN parameter with a heurist law, the other filter is adjusted by considering the output as random variables whose posterior probability distribution is inferred from the data. On the other hand, the kernel filter that gives exact interpolation due to the nature of kernel NN, uses a genetic algorithm to determine the filter parameters where the optimization problems were solved by a genetic, non-dominated sorting algorithm called NSGA-II. Furthermore, in function of the long and short term stochastic dependence of the time series evaluated by the Hurst parameter H, an on-line heuristic adaptive law and Bayesian inference was proposed to update the FFNN's. The main result shows that the analysis of both filter resides in the best horizon the FFNN tuned by the heuristic technique performs against GP and FFNN based on Bayesian approach applied to time series forecasting when the observations are taken from a single point, due to similar roughness for both the original and the forecasted time series, evaluated by H and H_e respectively. Owing that the time series is short, presents short term stochastic dependence, there is a better approximation in order to predict short time series with the proposed FFNN tuned by roughness.

ACKNOWLEDGMENT

This work was supported by Universidad Nacional de Córdoba (UNC) - SECyT UNC , Universidad Nacional de San Juan, National Agency for Scientific and Technological Promotion (ANPCyT) - PFDT 003 RRRHH UNC and CONICET.

REFERENCES

[1] Abry, P.; P. Flandrin, M.S. Taqqu, D. Veitch., Self-similarity and long-range dependence through the wavelet lens. Theory and applications of long-range dependence, Birkhäuser, pp. 527-556. 2003.

[2] Baumgartner, J., Cristian Rodríguez Rivero, Julián Pucheta, "A Genetic Algorithm based design Approach for the Properties of a Gaussian Process for Time series Forecasting", AADECA 2010, XXIIº Congreso Argentino de Control Automático, Buenos Aires, Argentina, (2010).

[3] Bishop, C. (2006). Pattern Recognition and Machine Learning. Boston: Springer.

[4] Chow, T.W.S.; Leung, C.T. Neural network based short-term load forecasting using weather compensation. Power Systems, IEEE Transactions on, Vol.11, Iss.4, Nov 1996, Pp. 1736-1742.

[5] Deb K., Pratap A., Agarwal S. y Meyarivan T. (2002). A Fast and Elitist Multiobjective Genertic Algorithm. In: IEEE Transactions on evolutionary computation, Vol.35, No.2.

[6] Flandrin, P. Wavelet analysis and synthesis of fractional Brownian motion" IEEE Trans. on Information Theory, 38, pp. 910-917. 1992.

[7] L. Glass and M. C. Mackey. From Clocks to Chaos, The Rhythms of Life. Princeton University Press, Princeton, NJ, 1988.

[8] Liu, J.N.K.; Lee, R.S.T. Rainfall forecasting from multiple point sources using neural networks. In proc. of the International Conference on Systems, Man, and Cybernetics, Pp. 429-434 Vol. 3. 1999.

[9] Mandelbrot, B. B. The Fractal Geometry of Nature, Freeman, San Francisco, CA. 1983.

[10] Pucheta, J., Patiño, H., Schugurensky, C., Fullana, R., Kuchen, B. Optimal Control Based-Neurocontroller to Guide the Crop Growth under Perturbations. DCDIS A Supplement, Advances in Neural Networks, Watam Press, Vol. 14(S1), pp. 618—623. 2007.

[11] Pucheta, J., Herrera, M., Salas C., Patiño, H.D., and B. Kuchen. "A Neural Network-Based Approach for Forecasting Time Series from Mackey-Glass Equations". In proc. Of the XIII RPIC, Rosario, Argentina. (2009).

[12] Rasmussen, C. E. and Williams C. K. I. 2006. Gaussian Processes for Machine Learning. The MIT Press.

[13] MacKay, D. J. C. (1995). Probable networks and plausible predictions - a review of practical Bayesian methods for supervised neural networks. Network: Computation in Neural Systems, 6 (3), 469-505.

[14] R.M. Neal, in: Bayesian learning for neural networks, Lecture Notes in Statistics, Vol. 118, Springer, New York, 1996.

[15] C. Rodriguez Rivero, J. Pucheta, M. Herrera, V. Sauchelli and S. Laboret (2012). Time Series Forecasting Using Bayesian Method: Application to Cumulative Rainfall, IEEE Latin American Transactions in press, unpublished.

[16] J. Pucheta, C. Rodriguez Rivero, M. Herrera, C. Salas and V. Sauchelli. Rainfall Forecasting Using Sub sampling Non-parametric Methods, IEEE Latin American Transactions in press, unpublished.

[17] Pucheta, J., Patiño, H., Schugurensky, C., Fullana, R., Kuchen, B. Optimal Control Based-Neurocontroller to Guide the Crop Growth under Perturbations. Dynamics Of Continuous, Discrete And Impulsive Systems Special Volume Advances in Neural Networks-Theory and Applications. DCDIS A Supplement, Advances in Neural Networks, Watam Press, Vol. 14(S1), pp. 618—623. 2007.

[18] Pucheta, J., Patino, D. and Kuchen, B. "A Statistically Dependent Approach For The Monthly Rainfall Forecast from One Point Observations". In Book Series Title: IFIP Advances in Information and Communication Technology, Book Title: Computer and Computing Technologies in Agriculture II, Volume 2, IFIP International Federation for Information Processing Volume 294, Computer and Computing Technologies in Agriculture II, Volume 2, eds. D. Li, Z. Chunjiang, (Boston: Springer), ISBN: 978-1-4419-0210-8, pp. 787–798, Url: http://dx.doi.org/10.1007/978-1-4419-0211-5_1, Doi: 10.1007/978-1-4419-0211-5_1. (2009).

[19] J. Pucheta, M., C. Rodríguez Rivero, M. Herrera, C. Salas, D. Patiño and B. Kuchen. "A Feed-forward Neural Networks-Based Nonlinear Autoregressive Model for Forecasting Time Series". Revista Computación y Sistemas, Centro de Investigación en Computación-IPN, México D.F., México, Computación y Sistemas Vol. 14 No. 4, pp. 423-435 ISSN 1405-5546, 2011. http://www.cic.ipn.mx/sitioCIC/images/revista/vol14-4/art07.pdf

[20] Baumgartner, J., Cristian Rodríguez Rivero, Julián Pucheta, "A Genetic Algorithm based design Approach for the Properties of a Gaussian Process for Time series Forecasting", AADECA 2010, XXIIº Congreso Argentino de Control Automático, Buenos Aires, Argentina, (2010).