

HYBRID DIFFERENTIAL EVOLUTIONARY SYSTEM FOR TIME SERIES PREDICTION

RICARDO DE A. ARAÚJO*, GERMANO C. VASCONCELOS*, TIAGO A. E. FERREIRA*

*Center for Informatics – Federal University of Pernambuco
Av. Prof. Luiz Freire, s/n, CDU, 50732-970, Recife - PE - Brazil.

Emails: raa@cin.ufpe.br, gcv@cin.ufpe.br, taef@cin.ufpe.br

Abstract— This paper presents a Hybrid Differential Evolutionary System (HDES) for time series forecasting. It consists of an intelligent hybrid model composed of an Artificial Neural Network (ANN) combined with an Improved Differential Evolution (IDE). The IDE searches for the relevant time lags for a correct time series characterization, the number of processing units in the ANN hidden layer, the ANN training algorithm and the modeling of ANN. Initially, the proposed HDES chooses the most fitted forecasting model, thus it performs a behavioral statistical test in the attempt to adjust forecast time phase distortions that appear in some time series. An experimental analysis is conducted with the proposed HDES using three real world time series and five well-known performance measures are used to assess its performance. The obtained results are compared to MultiLayer Perceptron (MLP) networks and the previously introduced Time-delay Added Evolutionary Forecasting (TAEF) method.

Keywords— Artificial Neural Networks, Differential Evolution, Hybrid Systems, Time Series Forecasting.

1 Introduction

The development of models for time series forecasting is considered a rather difficult problem. In order to solve such problem, many efforts have been made to the development of models and techniques able to forecast the future based on present and past. The popular statistical technique of Box-Jenkins (Box et al., 1994) is considered one of the most common choices for the prediction of time series. However, since the Box-Jenkins models are linear and most real world applications involve nonlinear problems, it is very difficult the Box-Jenkins models to capture the generator phenomenon of the nonlinear time series and this introduces a limitation to the accuracy of the generated predictions.

In order to overcome this limitation, nonlinear statistical approaches have been proposed, such as the bilinear models (Rao and Gabr, 1984), the threshold autoregressive models (Ozaki, 1985), the exponential autoregressive models (Priestley, 1988), the general state dependent models (Rumelhart and McClelland, 1987), amongst others. However, these nonlinear statistical models have high mathematical complexity and, in the practical applications, similar performance to linear statistical models (Clements et al., 2004).

In this way, approaches based on Artificial Neural Networks (ANNs) have been successful applied for nonlinear modeling of time series (Zhang et al., 1998). In this context, a relevant work was presented by Ferreira (Ferreira, 2006; Ferreira et al., 2007). It consists of the Time-delay Added Evolutionary Forecasting (TAEF) method definition for time series forecasting, which performs an evolutionary search (using a Modified Genetic Algorithm (MGA) (Leung et al., 2003)) for the minimum necessary number of dimensions corresponding to the problem of determining the characteristic phase space of the time series, based on the Takens Theorem (Takens, 1980). The TAEF method (Ferreira, 2006; Ferreira et al., 2007) finds the most fitted predictor model for representing a time series, and then performs a behavioral statistical test in order to adjust forecast time phase distortions that may appear in the representation of some time series.

This paper presents a Hybrid Differential Evolutionary System (HDES) for time series forecasting. The proposed HDES is inspired on Takens The-

orem (Takens, 1980) and consists of an intelligent hybrid model composed of an Artificial Neural Network (ANN) combined with an Improved Differential Evolution (IDE) (Araújo et al., 2007) (using modified operators in order to accelerate its convergence speed). The IDE searches for the relevant time lags for a correct time series characterization, the number of processing units in the ANN hidden layer, the ANN training algorithm and the modeling of ANN. After training model, the proposed HDES, based on the TAEF method (Ferreira, 2006; Ferreira et al., 2007), chooses the most tuned prediction model for time series representation, thus it performs a behavioral statistical test (Ferreira, 2006; Ferreira et al., 2007) in the attempt to adjust forecast time phase distortions that appear in some time series.

Furthermore, an experimental analysis is conducted with the proposed HDES using a financial time series (Standard & Poor 500 (S&P500) Index) and two natural phenomena time series (Sunspot and Brightness of a Variable Star Series). Five well-known performance measurements are used to assess performance of the proposed method and the obtained results shown better performance of the proposed HDES when compared to MultiLayer Perceptron (MLP) networks and the previously introduced TAEF method (Ferreira, 2006; Ferreira et al., 2007).

2 The Time Series Forecasting Problem

A time series, in its simplest form, is a set of points generally time equidistant, defined by,

$$X_t = \{x_t \in \mathbb{R} \mid t = 1, 2, \dots, N\}, \quad (1)$$

where t is the temporal index and N is the number of observations. So, X_t will be seen as a set of temporal observations of a phenomenon, orderly sequenced and equally spaced.

The main objective when applying forecasting techniques to a given time series is to identify certain regular patterns present in the data set in order to create a model capable of generating the next temporal patterns. In this context, a crucial factor for a good forecasting performance is the correct choice of the time lags considered for the representation of the given time series. Such relationship structures among historical data constitute a d -dimensional phase space,

where d is the dimension capable of representing such relationship. Takens (Takens, 1980) proved that if d is sufficiently large, such built phase space is homeomorphic to the phase space which generated the time series.

In this way, it is concluded that a crucial problem in reconstructing the original state space is the correct choice of the variable d , or more specifically, the correct choice of the time lags.

3 The Improved Differential Evolution

In this section, it is introduced the Improved Differential Evolution (IDE) (Araújo et al., 2007). It uses specific DE operators (mutation and crossover) in order to explore the state space more efficiently and to enhance convergence speed. The IDE procedure is illustrated in Figure 1.

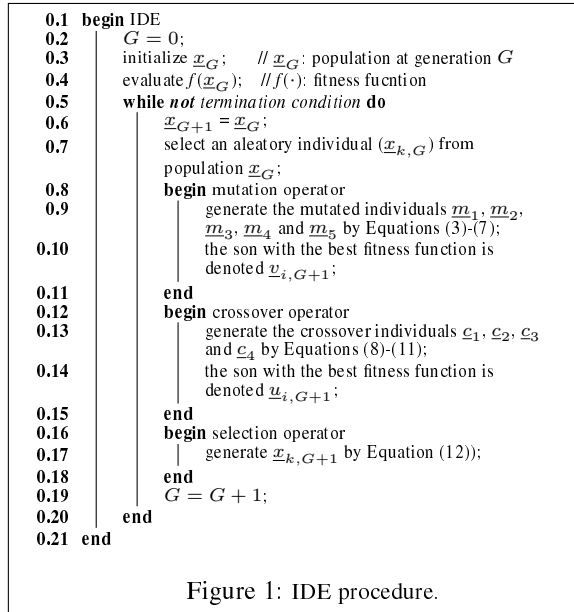


Figure 1: IDE procedure.

Each IDE individual is evaluated by a defined fitness function (or cost function). In this way, better chromosomes in IDE population will have high fitness function values. Hence, a possible fitness function definition is given by (Araújo et al., 2007)

$$fitness\ function = \frac{1}{1 + |min - f(\underline{x}_{S,G})|}, \quad (2)$$

where $f(\cdot)$ is an heuristic function and min denotes the minimum function value. It is worth to mention that the fitness function is dependent of the application objective (Leung et al., 2003).

The mutation generates five new individuals ($\underline{m}_j, j = 1, \dots, 5$), which are defined by the following equations (Araújo et al., 2007):

$$\underline{m}_1 = \underline{x}_{r1,G} + F(\underline{x}_{r2,G} - \underline{x}_{r3,G}), \quad (3)$$

$$\underline{m}_2 = \underline{x}_{best,G} + F(\underline{x}_{r1,G} - \underline{x}_{r2,G}), \quad (4)$$

$$\underline{m}_3 = \underline{x}_{r1,G} + F[(\underline{x}_{r2,G} - \underline{x}_{r3,G}) + (\underline{x}_{r4,G} - \underline{x}_{r5,G})], \quad (5)$$

$$\underline{m}_4 = \underline{x}_{best,G} + F[(\underline{x}_{r1,G} - \underline{x}_{r2,G}) + (\underline{x}_{r3,G} - \underline{x}_{r4,G})], \quad (6)$$

$$\underline{m}_5 = \underline{x}_{k,G} + \lambda(\underline{x}_{best,G} - \underline{x}_{k,G}) + F(\underline{x}_{r1,G} - \underline{x}_{r2,G}), \quad (7)$$

where $r1 \neq r2 \neq r3 \neq r4 \neq r5 \neq k$ are randomly chosen individuals indexes and $best$ denotes the best individual index. Term $F \in [0, 2]$ and $\lambda \in [0, 2]$ are a real-valued number which control the amplification of the difference vectors.

After the generation of the sons through mutation (Equations (3)-(7)), the son with the best evaluation (greater fitness value) will be chosen as the son generated by the mutation process and will be denoted by $\underline{v}_{i,G+1}$.

The crossover process to generate the vectors $\underline{c}_1, \underline{c}_2, \underline{c}_3$ and \underline{c}_4 is done with the use of four crossover operators, which are defined by the following equations (Araújo et al., 2007):

$$\underline{c}_1 = \frac{\underline{v}_{i,G+1} + \underline{x}_{k,G}}{2}, \quad (8)$$

$$\underline{c}_2 = \underline{pmax}(1 - w) + \underline{max}(\underline{v}_{i,G+1}, \underline{x}_{k,G})w, \quad (9)$$

$$\underline{c}_3 = \underline{pmin}(1 - w) + \underline{min}(\underline{v}_{i,G+1}, \underline{x}_{k,G})w, \quad (10)$$

$$\underline{c}_4 = \frac{(\underline{pmax} + \underline{pmin})(1 - w) + (\underline{v}_{i,G+1} + \underline{x}_{k,G})w}{2}, \quad (11)$$

where $w \in [0, 1]$ denotes the crossover weight, $\underline{max}(\underline{v}_{i,G+1}, \underline{x}_{k,G})$ and $\underline{min}(\underline{v}_{i,G+1}, \underline{x}_{k,G})$ denote the vector whose elements are the maximum and the minimum, respectively, between the gene values of $\underline{v}_{i,G+1}$ and $\underline{x}_{k,G}$. The terms \underline{pmax} and \underline{pmin} denote the maximum and minimum possible gene values, respectively.

After the generation of the sons through crossover (Equations (8)-(11)), the son with the best evaluation (greater fitness value) will be chosen as the son generated by the crossover process and will be denoted by $\underline{u}_{i,G+1}$.

The selection operator is responsible for the generation of the best sons (i.e. the vectors with the best evaluations). Thus, a greedy selection scheme is used, and is defined as:

$$\underline{x}_{k,G+1} = \begin{cases} \underline{u}_{i,G+1} & \text{if } f(\underline{u}_{i,G+1}) < f(\underline{x}_{k,G}) \\ \underline{x}_{k,G} & \text{otherwise} \end{cases}. \quad (12)$$

If, and only if, the trial vector $\underline{u}_{i,G+1}$ yields a better cost function value than $\underline{x}_{k,G}$, then $\underline{x}_{k,G+1}$ is set to $\underline{u}_{i,G+1}$; otherwise, the old value $\underline{x}_{k,G}$ is retained. However, it is worth to mention that only one individual ($\underline{x}_{k,G+1}$) is modified in IDE (this factor depends on the evaluation of the individual $\underline{u}_{i,G+1}$), while in the Standard DE (SDE) all the individuals are modified if its fitness is worse than the individuals generated by the crossover operator.

4 The Hybrid Differential Evolutionary System

The methodology proposed in this paper uses a differential evolutionary search mechanism in order to train and to adjust multilayer perceptrons applied to time series forecasting. It is based on the definition of the three main elements necessary for building an accurate forecasting system (Ferreira, 2006; Ferreira et al., 2007): (i) The underlying information necessary to predict the time series (the minimum number of time lags adequate for representing the time series),

(ii) The structure of the model capable of representing such underlying information for the purpose of prediction (the number of units in the ANN structure), and (iii) The appropriate algorithm for training the model. It is important to consider the minimum possible number of time lags in the correct representation of the series because the model must to be as parsimonious as possible.

According to this principle, the proposed system, referred to as Hybrid Differential Evolutionary System (HDES), consists of an intelligent hybrid model composed of an ANN (multilayer perceptron – MLP) and an IDE (Araújo et al., 2007), which determines the the following important parameters: (i) The minimum number of time lags to represent the time series: initially, a maximum number of lags (*MaxLags*) is defined by the user and the IDE can choose any number of specific lags (particular time lags capable of a fine tuned time series characterization) in the interval $[1, MaxLags]$ for each individual of the population, (ii) The number of units in the ANN hidden layer: the maximum number of hidden layer units (*NHiddenMax*) is determined by the user and the IDE chooses, for each candidate individual, the number of units in the hidden layer (in the interval $[1, NHiddenMax]$), and (iii) The training algorithm for the ANN: RPROP (Reidmiller and Braun, 1993), Levenberg-Marquardt (Hagan and Menhaj, 1994), Scaled Conjugate Gradient (Moller, 1993) and One Step Secant Conjugate Gradient (Battiti, 1992) are candidates for the best algorithm for training the ANN and the IDE defines one of these algorithms for each individual in the population. Figure 2 shows the proposed HDES scheme.

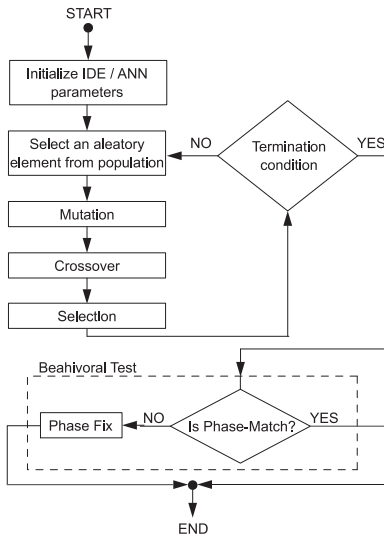


Figure 2: The proposed HDES.

After model training, when the IDE reaches a satisfactory solution, the proposed HDES uses the phase fix procedure from Ferreira (Ferreira, 2006; Ferreira et al., 2007), where a two step procedure is introduced, which tries to adjust time phase distortions that appear in the financial time series. Ferreira (Ferreira, 2006; Ferreira et al., 2007) shows that the representations of some series were developed by the model with a very close approximation between the actual and the predicted time series (“in-phase” matching), the predictions of other time series (finan-

cial time series like) were always presented with a one step shift (delay) with respect to the original data (“out-of-phase” matching).

In this way, the proposed HDES uses the statistical test applied by Ferreira (Ferreira, 2006; Ferreira et al., 2007) (t-test), where it is employed to verify if the network representation has reached an in-phase or out-of-phase matching. If this test accepts the in-phase matching hypothesis, the elected model is ready for practical use. Otherwise, the method performs a new procedure to adjust the relative phase between the prediction and the actual time series. According to Ferreira (Ferreira, 2006; Ferreira et al., 2007), the validation patterns are presented to the ANN and the output of these patterns are re-arranged to create new inputs that are both presented to the ANN and set as the output (prediction) target.

The approximation results for both the in-phase and out-of-phase models are measured and the best model (greater fitness function) is elected as the final model. It is worth to mention, that according to Ferreira (Ferreira, 2006; Ferreira et al., 2007), the phase fix procedure does not assume that the ANN is like a random walk model, but it is similarly to a random walk, that is, the $t + 1$ prediction is very close to the t value.

The IDE individuals are evaluated by the fitness function defined by,

$$fitness = \frac{POCID}{1 + MSE + MAPE + NMSE + ARV} \quad (13)$$

where *MSE*, *MAPE*, *NMSE*, *POCID* and *ARV* are the mean square error, the mean absolute percentage error, the normalized mean square error (or U of Theil Statistics), the prediction of change in direction and the average relative variance used to ANN performance evaluation, respectively, and were formally defined in (Ferreira, 2006; Ferreira et al., 2007).

The termination conditions for the IDE are,

1. The number of IDE iterations (*MaxGer*);
2. The increase in the validation error or generalization loss (*Gl*) (Prechelt, 1994): $Gl > 5\%$;
3. The decrease in the training error or process training (*Pt*) (Prechelt, 1994): $Pt \leq 10^{-6}$.

4.1 IDE Individuals Modeling

Each individual of the IDE population is an ANN (three-layer MLP). These individuals are represented by chromosomes that have the following parameters (ANN parameters): (i) W_{ij} : weights of connections between the input layer and the hidden layer, (ii) W_{jk} : weights of connections between the hidden layer and the output layer, (iii) b_j^1 : bias of the hidden layer, (iv) b_k^2 : bias of the output layer, (v) *NetMod*: ANN model, (vi) *NHidden*: the number of processing units in the ANN hidden layer, (vii) *NLags*: the number of relevant time lags, and (viii) *ANNTrain*: the ANN training algorithm.

Three distinct forms of modeling the ANN are proposed (*NetMod* = 1, 2, 3), where each is described in the following subsections.

4.1.1 First ANN model

The first architecture for modeling ANNs ($NetMod = 1$) is given by

$$y_k(t) = \sum_{j=1}^{n_h} W_{jk} Sig \left[\sum_{i=1}^{n_{in}} W_{ij} Z_i(t) + b_j^1 \right] + Sig(b_k^2), \quad (14)$$

where $Z_i(t)$ ($i = 1, 2, \dots, n_{in}$) are the ANN input values, n_{in} denotes the number of ANN input and n_h is the number of hidden units. Since the prediction horizon is one step ahead, only one output unit is necessary ($k = 1$).

4.1.2 Second ANN model

The second model ($NetMod = 2$) is given by:

$$y_k(t) = \sum_{j=1}^{n_h} W_{jk} Sig \left[\sum_{i=1}^{n_{in}} W_{ij} Z_i(t) + b_j^1 \right] + b_k^2. \quad (15)$$

4.1.3 Third ANN model

The third architecture ($NetMod = 3$) is given by:

$$y_k(t) = Sig \left\{ \sum_{j=1}^{n_h} W_{jk} Sig \left[\sum_{i=1}^{n_{in}} W_{ij} Z_i(t) + b_j^1 \right] + b_k^2 \right\}. \quad (16)$$

5 Experimental Results

A set of four real world time series was used as a test bed for evaluation of the proposed HDES: a financial time series (Standard & Poor 500 (S&P500) Index) and two natural phenomena time series (Sunspot and Brightness of a Variable Star Series). All series investigated were normalized to lie within the range $[0, 1]$ and divided in three sets according to Prechelt (Prechelt, 1994): training set (50% of the points), validation set (25% of the points) and test set (25% of the points).

The IDE parameters are the same for all experiments: the number of generations is 10^3 , $F = 0.5$, $\lambda = 0.95$ and $w = 0.9$. The IDE population is composed of 10 individuals, where each individual is an ANN with the maximum architecture: a 10-10-1 MultiLayer Perceptron (MLP) network, which denotes 10 units in the input layer, 10 units in the hidden layer and 1 unit in the output layer (prediction horizon of one step ahead).

Next, will be presented the simulation results involving the HDES model with and without the phase fix procedure (Ferreira, 2006; Ferreira et al., 2007), referred to as HDES out-of-phase model and HDES in-phase model, respectively. For each time series, was made ten experiments, where the experiment with the largest validation fitness function is chosen to represent the prediction model.

In order to establish a performance study, results previously published in the literature with the TAEF Method (Ferreira, 2006; Ferreira et al., 2007) on the same series and under the same conditions are employed for comparison of results. In addition, experiments with MultiLayer Perceptron (MLP) networks were used for comparison with the proposed HDES.

In all of the experiments, ten random initializations for each model (MLP) were carried out, where the experiment with the largest validation fitness function is chosen to represent the prediction model. The Levenberg-Marquardt Algorithm (Hagan and Menhaj, 1994) were employed for training the MLP network. For all the series, the best initialization was elected as the model to be beaten. The statistical behavioral test for phase fix was also applied to all the MLP models in order to guarantee a fair comparison between the models.

5.1 Standard & Poor 500 (S&P500) Index Series

The Standard & Poor 500 (S&P500) Stock Index is a pondered index of market values of the most negotiated stocks in the New York Stock Exchange (NYSE), American Stock Exchange (AMEX) and Nasdaq National Market System. The S&P500 series corresponds to the monthly records from January 1970 to August 2003, constituting a database of 369 points.

For the prediction of the S&P500 Index series (with 1 step ahead of prediction horizon), the proposed HDES automatically chose the lags 2 and 7 as the relevant lags for the time series representation, defined the second ANN model (Equation 15) with architecture 2-7-1, in which $NetMod = 2$, $NLags = 2$ and $NHidden = 7$, chose the Scaled Conjugate Gradient algorithm for ANN training and classified the model as "out-of-phase" matching. Table 1 shows the results (of the test set) for all performance measures for MLP network, TAEF method and the proposed HDES. Figure 3 shows the actual S&P500 Index values (solid line) and the predicted values generated by the HDES model out-of-phase (dashed line) for the 90 points of the test set.

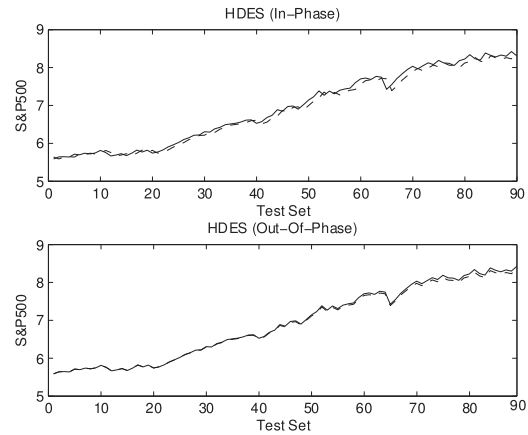


Figure 3: Prediction results for the S&P500 Index series (test set): actual values (solid line) and predicted values (dashed line).

5.2 Brightness of a Variable Star Series

The Brightness of a Variable Star series, or Star series, corresponds to daily observations in the same place and hour of an oscillating shine star, constituting a database of 600 points.

For the prediction of the Star series (with 1 step ahead of prediction horizon), the proposed HDES automatically chose the lags 1, 2, 3, 4, 5, 6, 7, 9 and 10 as

Table 1: Results for the S&P500 Index series.

	MLP		TAEF		HDES	
	In-Phase	Out-Of-Phase	In-Phase	Out-Of-Phase	In-Phase	Out-Of-Phase
MSE	0.0095	0.0096	7.4290e-4	8.0263e-4	1.4074e-4	1.8708e-5
MAPE	1.0100	1.0103	1.0431	1.0228	1.1229	0.3493
NMSE	0.9166	0.9179	7.2412	7.0883	1.3635	0.1745
ARV	7.2728e-3	7.2875e-3	0.0100	0.0012	1.0893e-2	1.4659e-3
POCID	51.11	50.98	50.54	100.00	50.00	98.87
Fitness	17.3644	17.3101	5.4373	10.9732	14.2961	64.8206

the relevant lags for the time series representation, defined the second ANN model (Equation 15) with architecture 9-10-1, in which $NetMod = 2$, $NLags = 9$ and $NHidden = 10$, chose the Levenberg Marquard algorithm for ANN training and classified the model as "in-phase" matching. Table 2 shows the results (of the test set) for all performance measures for MLP network, TAEF method and the proposed HDES. Figure 4 shows the actual Star values (solid line) and the predicted values generated by the HDES model (dashed line) for the last 100 points of the test set.

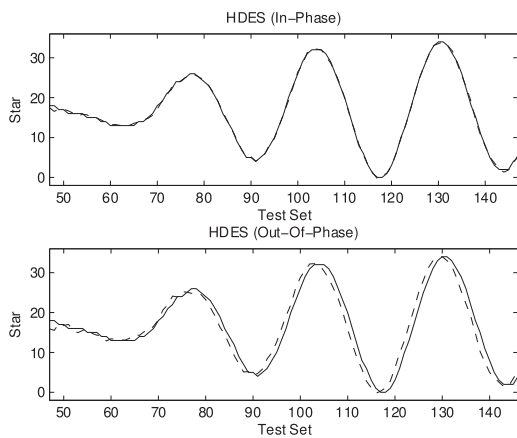


Figure 4: Prediction results for the Star series (test set): actual values (solid line) and predicted values (dashed line).

5.3 Sunspot Series

The selected Sunspot series consisted of the total annual measures of the sun spots from the years of 1700 to 1988, constituting a database of 289 points.

For the prediction of the Sunspot series (with 1 step ahead of prediction horizon), the proposed HDES automatically chose the lags 1, 2, 3, and 4 as the relevant lags for the time series representation, defined the second ANN model (Equation 15) with architecture 4-10-1, in which $NetMod = 2$, $NLags = 4$ and $NHidden = 10$, chose the Levenberg Marquard algorithm for ANN training and classified the model as "in-phase" matching. Table 3 shows the results (of the test set) for all performance measures for MLP network, TAEF method and the proposed HDES. Figure 5 shows the actual Sunspot values (solid line) and the predicted values generated by the HDES model (dashed line) for the 70 points of the test set.

6 Conclusion

In this paper a Hybrid Differential Evolutionary System (HDES) was presented for time series forecast-

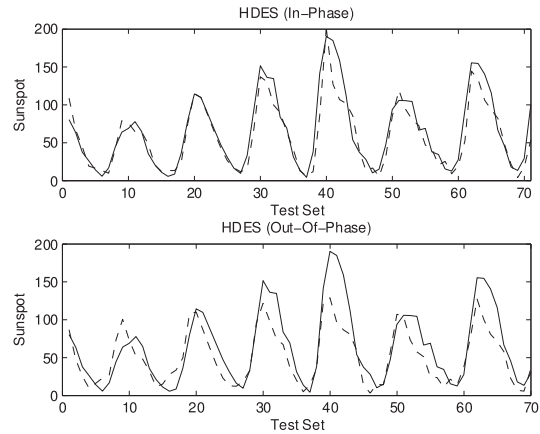


Figure 5: Prediction results for the Sunspot series (test set): actual values (solid line) and predicted values (dashed line).

ing. It consists of an intelligent hybrid model composed of an Artificial Neural Network (ANN) combined with an Improved Differential Evolution (IDE). The IDE searches for the relevant time lags for a correct time series characterization, the number of processing units in the ANN hidden layer, the ANN training algorithm and the modeling of ANN. Initially, the proposed HDES chooses the most tuned prediction model for time series representation, thus it performs a behavioral statistical test in the attempt to adjust forecast time phase distortions that appear in financial time series (Ferreira, 2006; Ferreira et al., 2007). Hence, the HDES was able to efficiently classify if the time series tends or not to a random walk like model (Mills, 2003), thus adjusting the model if necessary.

Five different metrics were used to measure the performance of the proposed HDES for time series forecasting. It was applied to three real world time series. The experimental results demonstrated slightly better performance, for natural phenomena time series, and better performance, for financial time series, of the proposed HDES model when compared to TAEF model (Ferreira, 2006; Ferreira et al., 2007). It was observed that the proposed HDES model obtained better performance than a random walk model (Mills, 2003) ($NMSE < 1$) and than a heads or tails experiment ($POCID > 50$) for all analyzed time series, overcoming the random walk dilemma for financial time series forecasting (where the predicted values were shifted one step ahead the original values). While the HDES model was able to adjust the time-phase delay, the MLP models were not capable to produce such correction behavior although the same procedure was applied to all the models. A feasible explanation for such phenomenon is that the phase fix procedure will de-

Table 2: Results for the Star series.

	MLP		TAEF		HDES	
	In-Phase	Out-Of-Phase	In-Phase	Out-Of-Phase	In-Phase	Out-Of-Phase
MSE	0.0234	0.8865	1.3852e-4	0.0087	1.3229e-4	3.8207e-3
MAPE	3.1000	27.7194	2.7834	26.7629	2.7698	17.0205
NMSE	0.0632	2.2185	0.0371	2.3149	0.0350	1.0204
ARV	3.3357e-3	0.7482	0.0020	0.6128	1.9438e-3	0.0564
POCID	72.97	66.19	77.39	73.79	77.55	69.86
Fitness	17.4154	2.0320	20.2451	2.4036	20.3710	3.6573

Table 3: Results for the Sunspot series.

	MLP		TAEF		HDES	
	In-Phase	Out-Of-Phase	In-Phase	Out-Of-Phase	In-Phase	Out-Of-Phase
MSE	0.9205	1.0163	0.0070	0.0307	9.5353e-3	0.0210
MAPE	37.4100	133.5613	30.0661	82.5523	26.8219	49.4800
NMSE	0.3443	1.3295	0.1763	1.2225	0.3300	0.8493
ARV	0.1418	0.3020	0.1233	0.4125	0.1469	0.3218
POCID	90.00	61.70	84.05	65.21	85.91	80.00
Fitness	2.2603	0.4529	2.6790	0.7652	3.0347	1.5482

pend on the complexity of the prediction model and on its ability to accurately define the best parameters to represent the time series.

The phase fix procedure proposed by Ferreira (Ferreira, 2006; Ferreira et al., 2007) was able to correct more efficiently the prediction phase of the proposed HDES model when compared to TAEF model (Ferreira, 2006; Ferreira et al., 2007). Further studies are being developed to better formalize and explain the properties of the HDES model and to determine possible limitations of the system with other financial time series with components such as trends, seasonalities, impulses, steps and other non-linearities.

References

- Araújo, R. A., Vasconcelos, G. C. and Ferreira, T. A. E. (2007). Improved differential evolution for single objective optimization problems, *VI Encontro Nacional de Inteligência Artificial (ENIA 2007)*, SBC.
- Battiti, R. (1992). One step secant conjugate gradient, *Neural Computation* **4**: 141–166.
- Box, G. E. P., Jenkins, G. M. and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*, third edn, Prentice Hall, New Jersey.
- Clements, M. P., Franses, P. H. and Swanson, N. R. (2004). Forecasting economic and financial time-series with non-linear models, *International Journal of Forecasting* **20**(2): 169–183.
- Ferreira, T. A. E. (2006). *A New Hybrid Intelligent Methodology for Time Series Forecasting*, PhD thesis, Federal University of Pernambuco.
- Ferreira, T. A. E., Vasconcelos, G. C. and Adeodato, P. J. L. (2007). Evolutionary design of morphological/rank/linear filters for time series forecasting, *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining*, Honolulu, Hawaii, USA.
- Hagan, M. and Menhaj, M. (1994). Training feedforward networks with the marquardt algorithm, *IEEE Transactions on Neural Networks* **5**(6): 989–993.
- Leung, F. H. F., Lam, H. K., Ling, S. H. and Tam, P. K. S. (2003). Tuning of the structure and parameters of the neural network using an improved genetic algorithm, *IEEE Transactions on Neural Networks* **14**(1): 79–88.
- Mills, T. C. (2003). *The Econometric Modelling of Financial Time Series*, Cambridge University Press, Cambridge.
- Moller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning, *Neural Networks* **6**: 525–533.
- Ozaki, T. (1985). *Nonlinear Time Series Models and Dynamical Systems*, Vol. 5 of *Hand Book of Statistics*, Noth-Holland, Amsterdam.
- Prechelt, L. (1994). Proben1: A set of neural network benchmark problems and benchmarking rules, *Technical Report 21/94*.
- Priestley, M. B. (1988). *Non-Linear and Non-Stationary Time Series Analysis*, Academic Press.
- Rao, T. S. and Gabr, M. M. (1984). *Introduction to Bispectral Analysis and Bilinear Time Series Models*, Vol. 24 of *Lecture Notes in Statistics*, Springer, Berlin.
- Reidmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm, *Proceedings of the IEEE Int. Conf. on Neural Networks (ICNN)*, San Francisco, pp. 586–591.
- Rumelhart, D. E. and McClelland, J. L. (1987). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, Vol. 1 & 2, MIT Press.
- Takens, F. (1980). Detecting strange attractor in turbulence, in A. Dold and B. Eckmann (eds), *Dynamical Systems and Turbulence*, Vol. 898 of *Lecture Notes in Mathematics*, Springer-Verlag, New York, pp. 366–381.
- Zhang, G., Patuwo, B. E. and Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting* **14**: 35–62.