A New Increasing Translation Invariant Morphological Method for Financial Time Series Forecasting

Ricardo de A. Araújo[1] and Glaucio G. de M. Melo[1]

[1]Information Technology Department, [gm][2] Intelligent Systems, Campinas, SP, Brasil

ricardo@gm2.com.br and glaucio@gm2.com.br

Abstract. *This paper presents a new method, referred to as Increasing Translation Invariant Morphological (ITIM), to overcome the random walk dilemma for financial time series forecasting. It consists of a hybrid intelligent model composed of a Modular Morphological Neural Network (MMNN) and a Modified Genetic Algorithm (MGA), which searches for the minimum number of time lags for a fine tuned time series representation, as well as by the initial weights, architecture and number of modules of the MMNN. Each element of the MGA population is trained via Back Propagation (BP) algorithm to further improve the parameters supplied by the MGA. The proposed method, after forecasting model adjustment, performs a behavioral statistical test and a phase fix procedure to adjust time phase distortions that appear in financial time series. An experimental analysis is conducted with the proposed method using two real world time series and five well-known performance measurements, demonstrating consistent better performance of this kind of morphological system.*

1. Introduction

Many efforts have been made to the development of models able to predict the future of a given phenomenon. Several linear and non linear statistical models were proposed for such [Box et al. 1994, Rao and Gabr 1984, Ozaki 1985, Priestley 1988, Rumelhart and McCleland 1987]. However, those statistical models usually involve high technical and mathematical complexities, limiting the development of an automatic forecast system [Clements et al. 2004]. In order to overcome the limitation of statistical models, approaches based on Neural Networks (NNs) have been successful proposed for nonlinear modeling of time series [Preminger and Franck 2007, Zhang 2007, Ferreira et al. 2008].

An important class of NNs are the Morphological Neural Networks (MNNs). Sousa [Sousa 2000] presented a particular MNN, referred to as Modular Morphological Neural Network (MMNN), based on the Matheron Decomposition Theorem [Matheron 1975]. In the morphological systems context, an interesting work was presented by Araújo et al. [Araújo et al. 2007], which consists of an evolutionary morphological approach definition for financial time series forecasting.

This paper proposes a new method, referred to as Increasing Translation Invariant Morphological (ITIM), to overcome the random walk dilemma for financial time series forecasting. It consists of a hybrid intelligent model composed of a Modular Morphological Neural Network (MMNN) [Sousa 2000] and a Modified Genetic Algorithm (MGA) [Leung et al. 2003]. The MGA is responsible to define the most fitted time lags for time series representation, based on Takens Theorem [Takens 1980], and the initial weights, architecture and number of modules of the MMNN. Each element of the MGA population is trained via Back Propagation (BP) algorithm [Sousa 2000] to further improve the parameters supplied by the MGA. Firstly, the proposed method chooses the most accurate prediction model, then it performs a behavioral statistical test and a phase fix procedure to adjust time phase distortions that appear in financial time series.

Furthermore, experimental results are presented for two real world time series: Dow Jones Industrial Average (DJIA) Index and Standard & Poor 500 (S&P500) Index. The results are discussed according to five well-known performance measurements: Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), U of Theil Statistic (THEIL), Prediction Of Change In Direction (POCID) and Average Relative Variance (ARV).

The experimental analysis of the proposed model demonstrates consistent better performance of this kind of morphological system when compared to results found with MultiLayer Perceptron (MLP) networks, the method proposed in Araújo et al.[Araújo et al. 2007] and the previously introduced Time-delay Added Evolutionary Forecasting (TAEF) method [Ferreira et al. 2008].

2. Fundamentals

This section presents the fundamentals and theoretical concepts necessary to comprehension of the proposed method.

2.1. The Time Series Prediction Problem

A time series is a sequence of observations about a given variable. This variable is observed in discrete or continuous time points, usually time equidistant. Thus, the analysis of this temporal behavior evolves the process or phenomenon description that generates such observations sequence.

In this way, a time series can be defined by,

$$X_t = \{x_t \in \mathbb{R} \mid t = 1, 2, \ldots, N\}, \tag{1}$$

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

The aim of forecasting techniques applied to a time series X_t is to provide a mechanism that allows, with certain accuracy, the forecasting of the future values of X_t, given by $X_{t+h},\ h = 1, 2, ...$, where h represents the prediction horizon of h step ahead. Nevertheless, in order to provide proper forecast performance, the most relevant factor to guarantee forecasting accuracy is the correct choice of time lags for representing a given time series [Ferreira et al. 2008].

2.2. MMNN Definition

Sousa [Sousa 2000] defined the MMNN for designing translation invariant operators that satisfy the MDT [Matheron 1975] for dilations as well as for erosions. Figure 1(a) presents the MMNN architecture for the Matheron Decomposition [Matheron 1975] by dilations.

The following equations define the MMNN architecture for the Matheron Decomposition [Matheron 1975] via dilations according to this approach.

$$v_k = \delta_k = max(\underline{x} + \underline{a}_k), \tag{2}$$

where $\underline{x}$ represents the MMNN input signal.

$$\text{MMNN Output: } Y = min(\underline{v}), \tag{3}$$

in which

$$\underline{v} = (v_1,\ v_2,\ \ldots\ , v_k). \tag{4}$$

The MMNN weights matrix, A, is defined by

$$A = (\underline{a}_1;\ \underline{a}_2;\ \ldots;\ \underline{a}_k), \tag{5}$$

in which $\underline{a}_k \ \in \mathcal{R}^k,\ k = 1, 2, \ldots,$ ND represents the MMNN weights (i.e., matrix rows composed by structuring elements $\underline{a}_k$). The Symbol $\wedge$ represents the minimum operator.

In a dual manner, the MMNN architecture for the Matheron Decomposition [Matheron 1975] via erosions is defined by substituting dilations by erosions and symbol $\wedge$ by $\vee$, where $\vee$ represents the maximum operation. Figure 1(b) presents the MMNN architecture for the Matheron Decomposition [Matheron 1975] by erosions.

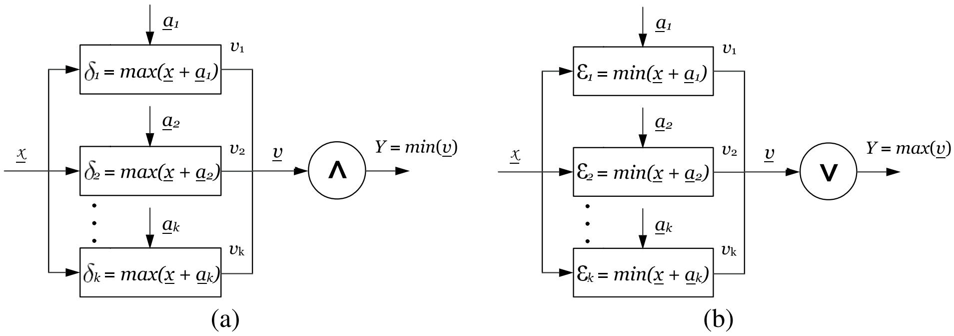


Figure 1. MMNN architectures for the Matheron decomposition.

2.3. MMNN Training Algorithm

Based on Back Propagation (BP) algorithm. Sousa [Sousa 2000] defined the MMNN training for Matheron Decomposition [Matheron 1975], which is formally defined by the following equations [Sousa 2000]:

$$A(n+1) = A(n) - \mu \nabla_A J(A),\ n = 0, 1, \ldots \tag{6}$$

in which A is the weight matrix, μ is the learning rate and $\nabla_A J(A)$ is the gradient matrix of a cost function $J(A)$ (to be minimized with respect to the weight matrix A). For a given training set,

$$\{(\underline{x}_m, d_m),\ m = 1, 2, \ldots, M\}, \tag{7}$$

where d_m is the desired output of a given input $\underline{x}_m$ and M is the number of patterns of training set, $J(A)$ is defined by

$$J(A) = \frac{1}{2} e_m^2, \tag{8}$$

where $e_m = d_m - y_m$ is the difference between the desired output and the actual output for the input $\underline{x}_m$, $m = 1, 2, \ldots, M$. The gradient presented in equation (6) is given by [Sousa 2000]

$$\frac{\partial J}{\partial \underline{a}_k} = -e \frac{\partial y}{\partial v_k} \frac{\partial v_k}{\partial \underline{a}_k},\ k = 1, 2, \ldots, \text{ND}. \tag{9}$$

According to Sousa [Sousa 2000], the partial derivatives in equation (9) are estimated by the methodology of Pessoa and Maragos [Pessoa and Maragos 1998] via rank indication vectors $\underline{c}$ and smooth impulse functions Q_σ. In matrix terms, the gradient may be defined by [Sousa 2000]

$$\nabla_A J(A) = -e \cdot \text{diag}(\underline{c}) \cdot C, \tag{10}$$

in which $C = (\underline{c}_1; \underline{c}_2; \ldots; \underline{c}_k)$. Term "·" represents the scalar product. Terms $\underline{c}$ and $\underline{c}_k$ are defined by Matheron Decomposition [Matheron 1975] via dilations by [Sousa 2000]

$$\underline{c} = \frac{Q_\sigma(min(\underline{v}) \cdot \underline{1} - \underline{v})}{Q_\sigma(min(\underline{v}) \cdot \underline{1} - \underline{v}) \cdot \underline{1}^T}; \tag{11}$$

$$\underline{c}_k = \frac{Q_\sigma(max(\underline{x} + \underline{a}_k) \cdot \underline{1} - \underline{x} - \underline{a}_k)}{Q_\sigma(max(\underline{x} + \underline{a}_k) \cdot \underline{1} - \underline{x} - \underline{a}_k) \cdot \underline{1}^T}, \tag{12}$$

where T denotes transposition and "·" represents scalar product.

In a dual way, terms $\underline{c}$ e $\underline{c}_k$ are defined by Matheron Decomposition [Matheron 1975] via erosions by [Sousa 2000]

$$\underline{c} = \frac{Q_\sigma(max(\underline{v}) \cdot \underline{1} - \underline{v})}{Q_\sigma(max(\underline{v}) \cdot \underline{1} - \underline{v}) \cdot \underline{1}^T}; \tag{13}$$

$$\underline{c}_k = -\frac{Q_\sigma(min(\underline{x} - \underline{a}_k) \cdot \underline{1} - \underline{x} + \underline{a}_k)}{Q_\sigma(min(\underline{x} - \underline{a}_k) \cdot \underline{1} - \underline{x} + \underline{a}_k) \cdot \underline{1}^T}. \tag{14}$$

3. The Proposed Approach

The approach proposed in this paper uses an evolutionary search mechanism in order to train and adjust the Modular Morphological Neural Network (MMNN) applied to financial time series forecasting (overcoming the random walk dilemma). It is based on the definition of the four main elements necessary for building an accurate forecasting system [Ferreira et al. 2008]:

- The underlying information necessary to predict the time series;
- The structure of the model capable of representing such underlying information for the purpose of prediction;
- The appropriate algorithm for training the model
- The behavior test to adjust time phase distortions

It is important to consider the minimum possible number of time lags in the representation of the series because the model must to be as parsimonious as possible.

Based on that definition, the proposed method, referred to as Increasing Translation Invariant Morphological (ITIM), consists of a hybrid intelligent morphological model composed of a MMNN [Sousa 2000] with a MGA [Leung et al. 2003], which searches for:

1. The minimum number of time lags to represent the series: initially, a maximum number of time lags ($MaxLags$) is pre-defined and then the MGA will search for the number of time lags in the range $[1, MaxLags]$ for each individual of the population;
2. the weights ($\underline{a}_k$), architecture (by dilations or by erosions – $\mathrm{MMNNArch}$) and number of modules of the MMNN ($\mathrm{NModules}$): initially, a maximum number of MMNN modules ($MaxMod$) is pre-defined and then the MGA chooses, for each candidate individual, the weigths, the most adequate MMNN architecture and the number of MMNN modules in the range $[1, MaxMod]$.

The MGA used is based on the work of Leung et al. [Leung et al. 2003], where special crossover and mutation operators are applied to accelerate the search convergence. The MGA procedure consists on the selection of a parent pair of chromosomes and then performing crossover and mutation operators (generating the offspring chromosomes – the new population) until the termination condition is reached; then the best individual in the population is selected as a solution to the problem.

The crossover operator is used for exchanging information from two parents (vectors $\underline{p}_1$ and $\underline{p}_2$) obtained in the selection process by a roulette wheel approach [Leung et al. 2003]. The recombination process to generate the offsprings (vectors $\underline{C}_1, \underline{C}_2, \underline{C}_3$ and $\underline{C}_4$) is done by four crossover operators, which are defined by the following equations [Leung et al. 2003]:

$$\underline{C}_1 = \frac{\underline{p}_1 + \underline{p}_2}{2}, \tag{15}$$

$$\underline{C}_2 = \underline{p_{max}}(1 - w) + max(\underline{p}_1, \underline{p}_2)w, \tag{16}$$

$$\underline{C}_3 = \underline{p_{min}}(1 - w) + min(\underline{p}_1, \underline{p}_2)w, \tag{17}$$

$$\underline{C}_4 = \frac{(\underline{p_{max}} + \underline{p_{min}})(1-w) + (\underline{p}_1 + \underline{p}_2)w}{2}, \tag{18}$$

where $w \in [0,1]$ denotes the crossover weight (the closer w is to 1, the greater is the direct contribution from parents), $max(\underline{p}_1, \underline{p}_2)$ and $min(\underline{p}_1, \underline{p}_2)$ denotes the vector whose elements are the maximum and the minimum, respectively, between the gene values of $\underline{p}_1$ and $\underline{p}_2$. The terms $\underline{p_{max}}$ and $\underline{p_{min}}$ denote a vector with the maximum and minimum possible gene values, respectively. After the offspring generation by crossover operators, the son with the best evaluation (greatest fitness value) will be chosen as the offspring generated by the crossover process and denoted $\underline{C}^{best}$.

After the crossover operator, $\underline{C}^{best}$ is selected to have a mutation process, where three new mutated offsprings are generated and defined by the following equation [Leung et al. 2003]:

$$\underline{M}_j = C_i^{best} + \gamma_i \Delta M_i, \; j = 1,2,3 \text{ and } i = 1,2,\ldots,\mathrm{NG}, \tag{19}$$

where γ_i can only take the values 0 or 1, ΔM_i are randomly generated numbers such that $p_{min} \leq C_i^{best} + \Delta M_i \leq p_{max}$ and NG denotes the number of genes in the chromosome.

The first mutated offspring ($\underline{M}_1$) is obtained according to (19) using only one term γ_i set to 1 (i is randomly selected within the range $[1, \mathrm{NG}]$) and the remaining terms γ_i are set to 0. The second mutated offspring ($\underline{M}_2$) is obtained according to (19) using some γ_i randomly chosen and set to 1 and the remaining terms γ_i are set to 0. The third mutated offspring ($\underline{M}_3$) is obtained according to (19) using all γ_i set to 1.

Then, each element of the MGA population is trained via Back Propagation (BP) algorithm [Sousa 2000] to further improve the parameters supplied by the MGA, that is, the BP is used, for each individual candidate, to perform a local search around the initial weights supplied by MGA. The main idea used here is to conjugate a local search method (BP) to a global search method (MGA). While the MGA makes possible the testing of varied solutions in different areas of the solution space, the BP acts on the initial solution to produce a fine-tuned forecasting model. Such process is able to seek the most compact MMNN, reducing computational cost and probability of model overfitting. Each MGA individual represents a MMNN, where its input is defined by the number of time lags and its output represents the prediction horizon of one step ahead.

Most works found in the literature have the fitness function (or objective function) based on just one performance measure, like Mean Square Error (MSE). However, Clements et al. [Clements and Hendry 1993], since 1993, shown that the MSE measure has some limitations to available and to compare the prediction model performance. Information about the prediction, as the absolute percentage error, the accuracy in the future direction prediction and the relative gain regarding naive prediction models (like random walk models and mean prediction) are not described using MSE measure.

In order to provide a more robust forecasting model, a multi-objective evaluation function is defined, which is a combination of five well-known performance measures: Prediction Of Change In Direction (POCID), Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE), Normalized Mean Square Error (NMSE) or U of Theil Statistic (THEIL) and Average Relative Variance (ARV), where all these measures are formally defined in [Ferreira et al. 2008]. The multi-objective evaluation function used here is given by

$$\text{Fitness Function} = \frac{\mathrm{POCID}}{1 + \mathrm{MSE} + \mathrm{MAPE} + \mathrm{THEIL} + \mathrm{ARV}}. \tag{20}$$

Whereas there are linear and nonlinear metrics in the such evaluation function and each one of these metrics can contribute of different forms for the evolution process, the Equation 20 was built of empirical form to have all information necessary to describe as well as possible the time series generator phenomenon.

After MMNN adjusting and training, the proposed method uses the phase fix procedure presented by Ferreira [Ferreira et al. 2008], where a two step procedure is introduced to adjust time phase distortions observed ("out-of-phase" matching) in financial time series. Ferreira [Ferreira et al. 2008] has shown that the representations of some time series (natural phenomena) were developed by the model with a very close approximation between the actual and the predicted time series (referred to as "in-phase" matching), whereas the predictions of other time series (mostly financial time series) were always presented with a one step delay regarding the original data (referred to as "out-of-phase" matching).

The proposed method uses the statistical test (t-test) to check if the MMNN model representation has reached an in-phase or out-of-phase matching. This is conducted by comparing the outputs of the prediction model with the actual series, making use only of the validation data set. This comparison is a simple hypothesis test, where the null hypothesis is that the prediction corresponds to in-phase matching and the alternative hypothesis is that the prediction is not correspond to in-phase matching (or correspond to out-of-phase matching).

If this test accepts the in-phase matching hypothesis, the elected model is ready for practical use. Otherwise, the proposed method performs a new procedure to adjust the relative phase between the prediction and the actual time series. The phase fix procedure has two steps: (i) the validation patterns are presented to the MMNN and the output of these patterns are re-arranged to create new inputs patterns (reconstructed patterns), and (ii) these reconstructed patterns are represented to the same MMNN and the output set as the prediction target. This procedure of phase adjustment considers that the MMNN is not a random walk model, it just shows a behavior characteristic of a random walk model: the $t+1$ prediction is taken as the t value (Random Walk Dilemma).

If the MMNN was like a random walk model, the phase adjust procedure would not work. Such phase fix was originally proposed by Ferreira [Ferreira et al. 2008], where he observed the fact that when Artificial Neural Network (ANN – Multilayer Perceptron like) is correctly adjusted, the one step shift distortion in the prediction can be softened.

The termination conditions for the MGA are:

1. Minimum value of fitness function: $fitness \geq 40$, where this value mean the accuracy to predict direction around 80% ($POCID \gtrsim 80\%$) and the sum of the other errors around one ($MSE + MAPE + THEIL + ARV \cong 1$);
2. The increase in the validation error or generalization loss (Gl) [Prechelt 1994]: $Gl > 5\%$;
3. The decrease in the training error process training (Pt) [Prechelt 1994]: $Pt \leq 10^{-6}$.

Each individual of the MGA population is a MMNN represented by the data structure with the following components (MMNN parameters):

- $\underline{a}_k$: weights (structuring elements) of the MMNN;
- $\mathrm{NModules}$: the number of modules in the MMNN structure (number of decompositions);
- $\mathrm{MMNNArch}$: a real-valued variable, where is used to determine if the architecture is by dilations ($\mathrm{MMNNArch} > 0$) or by erosions ($\mathrm{MMNNArch} \leq 0$);
- NLags: a vector, where each position has a real-valued codification, which is used to determine if a specific time lag will be used ($\mathrm{NLags}_i > 0$) or not ($\mathrm{NLags}_i \leq 0$).

4. Simulations and Experimental Results

A set of two real world financial time series (Dow Jones Industrial Average (DJIA) Index and Standard & Poor 500 Stock (S&P500) Index) were used as a test bed for evaluation of the proposed method. All time 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).

For all the experiments, the method parameters are: maximum number of MMNN modules ($MaxMod = 25$) and maximum number of time lags ($MaxLags = 10$). The

MGA parameters used in the proposed method are a maximum number of MGA generations, corresponding to 10^4, crossover weight $w = 0.9$ (used in the crossover operator), mutation probability equals to 0.1. Each element of the MGA population is then trained via the Back Propagation algorithm, using a smoothing parameter $\sigma = 0.05$ and a convergence factor $\mu = 0.01$. The termination conditions for the Back Propagation algorithm are the maximum number of epochs (10^4), the increase in the validation error or generalization loss ($Gl > 5\%$) and the decrease in the error of the process training ($Pt < 10^{-6}$).

Next, will be presented the simulation results involving the proposed ITIM model. In order to establish a performance study, results previously published in the literature with the TAEF Method [Ferreira et al. 2008] and the method proposed in Araújo et al.[Araújo et al. 2007] were examined in the same context and under the same experimental conditions. For each time series, it was made ten experiments, where the experiment with the best validation fitness function is chosen to represent the prediction model.

In addition, experiments with MultiLayer Perceptron (MLP) networks were used for comparison with the proposed method. According to Ferreira [Ferreira et al. 2008], in all the MLP experiments were tested three time lags windows, a windows with lag 1, a windows with lags 1 to 5 and a windows with lags 1 to 10, and five different numbers (1, 5, 10, 15 and 20) of hidden processing units being tested and one processing unit in output layer (one step ahead prediction). The Levenberg Marquardt Algorithm [Hagan and Menhaj 1994] was employed for training the MLP network for a maximum period of 10^3 epochs. The termination conditions for the MLP training are equal to the termination criteria for the MMNN training in the proposed method (Epochs, Gl and Pt). In all of the experiments, ten random initializations for each architecture were carried out, where the experiment with the best validation fitness function is chosen to represent the prediction model. The statistical behavioral test, for phase fix procedure, was also applied to all the MLP, Araújo et al. and TAEF models in order to guarantee a fair comparison among the models.

It is worth mentioning that the results with ARIMA models were not presented in our comparative analysis since Ferreira [Ferreira et al. 2008] has shown that MLP networks obtained results better than ARIMA models, for all financial time series used in this work. In this way, it is used only MLP networks in our comparative analysis.

4.1. Dow Jones Industrial Average (DJIA) Index Series

The Dow Jones Industrial Average Index (DJIA) series corresponds to daily observations from January 1st 1998 to August 26th 2003, constituting a database of 1420 points.

The parameters automatically defined by the proposed model for the prediction of the DJIA Index series were the lags 2, 7 and 8 as the most fitted lags for the time series representation, the MMNN model via dilations with 9 modules and the model as "out-of-phase" matching. Table 1 shows the results (with respect to the test set) for all the performance measures for the MLP, Araújo et al., TAEF and the proposed ITIM model.

Table 1. Results for the DJIA Index series.

Evaluation Metrics	MLP 5-10-1	Araújo et al.	TAEF	Proposed ITIM Model
MSE	8.2700e-2	8.3236e-4	2.6841e-5	9.8825e-6
MAPE	9.3700	9.6700	0.1993	1.4101e-2
THEIL	0.9878	0.9945	0.0318	1.1746e-2
ARV	3.3877e-2	3.4423e-2	0.0007	4.1246e-4
POCID	46.74	50.85	97.14	99.15
Fitness Function	4.0734	4.3462	78.8584	96.6121

According to Table 1, it is verified that the prediction of proposed ITIM model obtained a performance much better (in terms of evaluation function – 96.6121) than the MLP model (4.0734) and the Araújo et al. model (4.3462), and a slightly better result than the TAEF model (78.8584). Moreover, the obtained THEIL value (1.1746e-2) shows that the proposed ITIM model had a much better performance than a random walk like model [Mills 2003], the MLP model (0.9878) and the Araújo et al. model (0.9945), and a slightly better result than the TAEF model (0.0318). Again, the POCID

measure (99.15%) shows that the proposed ITIM model had a much better performance than a “coin-tossing” experiment, the MLP model (46.74%) and the Araújo et al. model (50.85%), and a slightly better performance than the TAEF model (97.14%).

Figure 2(a) shows the prediction results of DJIA Index for the last 10 points of the test set.

4.2. Standard & Poor 500 (S&P500) Index Series

The Standard & Poor 500 (S&P500) 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 used corresponds to the monthly records from January 1970 to August 2003, constituting a database of 369 points.

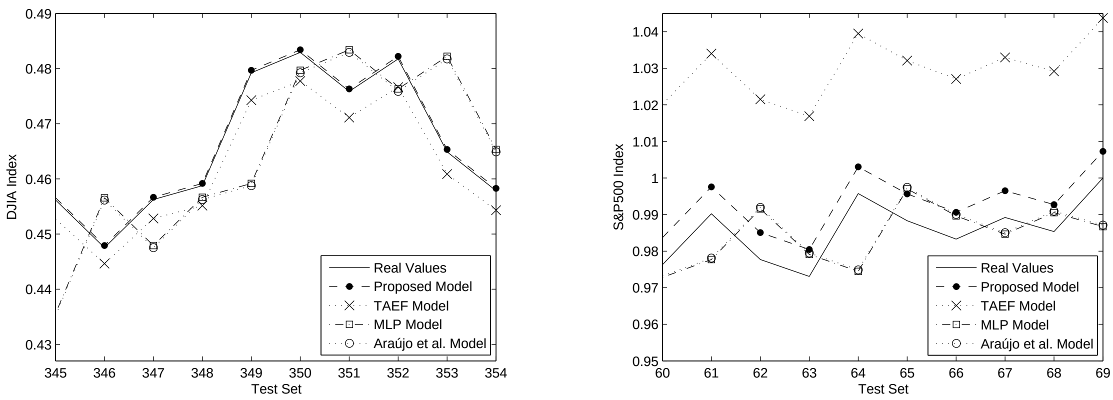
For the prediction of the S&P500 Index series, the proposed model automatically selected the lags 2, 4, 5, 8, 9 and 10 as the best lags for the time series representation, the MMNN model via dilations with 17 modules and the model as “out-of-phase” matching. Table 2 shows the results (with respect to the test set) for all the performance measures for the MLP, Araújo et al., TAEF and the proposed ITIM model.

Table 2. Results for the S&P500 Index series.

Evaluation Metrics	MLP 1-20-1	Araújo et al.	TAEF	Proposed ITIM Model
MSE	0.0095	9.7451e-5	8.0263e-4	7.1336e-5
MAPE	1.0100	0.9200	1.0228	7.8553e-3
THEIL	0.9166	0.9498	7.0883	0.6566
ARV	7.2728e-3	7.4749e-3	0.0012	5.5896e-4
POCID	51.11	81.31	100.00	100.00
Fitness Function	17.3644	28.2584	10.9732	60.0570

According to the Table 2, it is possible to note that the prediction of proposed ITIM model obtained performance much better (in terms of evaluation function – 60.0570) than the MLP model (17.3644), the Araújo et al. model (28.2584) and the TAEF model (10.9732). Again, the obtained THEIL value (0.6566) indicated that the proposed ITIM model had a much better performance than a random walk like model [Mills 2003], the MLP model (0.9166), the Araújo et al. model (0.9498) and the TAEF model (7.0883). According to the POCID measure (100.00%), it is possible to verify that the proposed ITIM model had a much better performance than a “coin-tossing” experiment, the MLP model (51.11%) and the Araújo et al. model (81.31%), and the same performance than the TAEF model (100%).

Figure 2(b) shows the prediction results of S&P500 Index for the last 10 points of the test set.



(a) (b)

Figure 2. Prediction Results for the analyzed financial time series.

In general, all generated ITIM models using the phase fix procedure to adjust time phase distortions shown forecasting performance much better than the MLP model and Araújo et al. model, and slightly better than the TAEF model. The proposed method was able to adjust the time phase distortions of all analyzed time series (the prediction generated by the out-of-phase matching hypothesis is not delayed with respect to the original data), while the MLP model and Araújo et al. model were not able to adjust the time phase. This corroborates with the assumption made by Ferreira [Ferreira et al. 2008], where he discusses that the success of the phase fix procedure is strongly dependent on an accurate adjustment of the prediction model parameters and on the model itself used for prediction.

5. Conclusion

This paper presented a new method, referred to as Increasing Translation Invariant Morphological (ITIM), to overcome the random walk dilemma for financial time series forecasting, which consists of a hybrid intelligent model composed of a Modular Morphological Neural Network (MMNN) and a Modified Genetic Algorithm (MGA). The proposed method searches for the minimum number of time lags for a correct time series representation and the best MMNN architecture in terms of the weights, the number of MMNN modules and the type of Matheron Decomposition (by dilations or by erosions) and its training algorithm. Also, it performs a behavioral statistical test and a phase fix procedure to adjust time phase distortions that appear in financial time series.

Five different metrics were used to measure the performance of the proposed method for financial time series forecasting, where a multi-objective empirical fitness function was built in order to improve the description of the time series phenomenon as well as possible. The five different evaluations measures used to compose this fitness function can have different contributions to final prediction result, where a more sophisticated analysis must be done to determine the optimal combination of such metrics.

The results were collected with two real world time series from the financial stock market with all their dependence on exogenous and uncontrollable variables (Dow Jones Industrial Average (DJIA) Index and Standard & Poor 500 (S&P500) Index). It is verified through lagplot analysis that it is possible to notice in financial time series indicative structures of some nonlinear relationship among the time lags even though they are superimposed by a dominant linear component.

It was observed that the proposed model obtained better results than MLP and Araújo et al. models for all the analyzed financial time series, overcoming the random walk dilemma, where the predictions of such time series are dislocated one step ahead with respect to the original data. In this way, the proposed model was able to adjust time phase distortions of all analyzed financial time series, while all MLP and Araújo et al. models do not obtained such behavior. Concerning TAEF method, the proposed model was able to adjust more efficiently the time phase distortions than TAEF model, obtaining slightly better results for all time series analyzed.

It is worth mentioning that the first time lag is never selected to predict any time series used in this work. However, a random walk structure is necessary to the phase fix procedure works, since the key of this procedure is the two step prediction (described by phase fix procedure) in order to adjust the one step time phase.

While the proposed model was able to adjust the time-phase delay, the MLP and Araújo et al. 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 depend on the information complexity contained in the time series data and the ability to accurately define the best prediction model parameters to estimate the real time series values, in other words, the success of the phase fix procedure is strongly dependent on an accurate adjustment of the prediction model parameters and on the model itself used for forecasting.

Future works will consider the development of further studies in order to formalize properties of the proposed model using the phase fix procedure. Also, other financial time series with components such as trends, seasonalities, impulses, steps and other nonlinearities are being used for the efficiency confirmation of the proposed method, as well as, further studies, in terms of risk and financial return, are being developed in order to determine the additional economical benefits, for an investor, with the use of the proposed method.

References

Araújo, R. A., Sousa, R. P., and Ferreira, T. A. E. (2007). An intelligent hybrid approach for designing increasing translation invariant morphological operators for time series forecasting. In *ISNN (2)*, volume 4492 PART II of *Lecture Notes in Computer Science*, pages 602–611. Springer-Verlag.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time Series Analysis: Forecasting and Control*. Prentice Hall, New Jersey, third edition.

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.

Clements, M. P. and Hendry, D. F. (1993). On the limitations of comparing mean square forecast errors. *Journal of Forecasting*, 12(8):617–637.

Ferreira, T. A. E., Vasconcelos, G. C., and Adeodato, P. J. L. (2008). A new intelligent system methodology for time series forecasting with artificial neural networks. In *Neural Processing Letters*, volume 28, pages 113–129.

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.

Matheron, G. (1975). *Random Sets and Integral Geometry*. Wiley, New York.

Mills, T. C. (2003). *The Econometric Modelling of Financial Time Series*. Cambridge University Press, Cambridge.

Ozaki, T. (1985). *Nonlinear Time Series Models and Dynamical Systems*, volume 5 of *Hand Book of Statistics*. North-Holland, Amsterdam.

Pessoa, L. F. C. and Maragos, P. (1998). MRL-filters: A general class of nonlinear systems and their optimal design for image processing. *IEEE Transactions on Image Processing*, 7:966–978.

Prechelt, L. (1994). Proben1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94.

Preminger, A. and Franck, R. (2007). Forecasting exchange rates: A robust regression approach. *International Journal of Forecasting*, 23(1):71–84.

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*, volume 24 of *Lecture Notes in Statistics*. Springer, Berlin.

Rumelhart, D. E. and McCleland, J. L. (1987). *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, volume 1 & 2. MIT Press.

Sousa, R. P. (2000). *Design of translation invariant operators via neural network training*. PhD thesis, UFPB, Campina Grande, Brazil.

Takens, F. (1980). Detecting strange attractor in turbulence. In Dold, A. and Eckmann, B., editors, *Dynamical Systems and Turbulence*, volume 898 of *Lecture Notes in Mathematics*, pages 366–381, New York. Springer-Verlag.

Zhang, G.P.; Kline, D. (Nov. 2007). Quarterly time-series forecasting with neural networks. *Neural Networks, IEEE Transactions on*, 18(6):1800–1814.