# A Quantum-Inspired Hybrid Methodology for Financial Time Series Prediction

Ricardo de A. Araújo, Adriano L. I. de Oliveira, *Senior Member, IEEE*, and Sergio C. B. Soares

***Abstract*— In this work a quantum-inspired hybrid methodology is proposed to overcome the random walk dilemma for financial time series prediction. It consists of a hybrid model composed of a Qubit Multilayer Perceptron (QuMLP) with a Quantum-Inspired Evolutionary Algorithm (QIEA), which searches for the best particular time lags able to characterize the time series phenomenon, as well as to evolve the complete QuMLP architecture and parameters. Each individual of the QIEA population is adjusted by the Complex Back-Propagation (CBP) algorithm to further improve the QuMLP parameters supplied by the QIEA. After the prediction model search procedure, it uses 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 methodology through four real world financial time series, and the obtained results are discussed and compared to results found with Multilayer Perceptron (MPL) networks and the previously introduced Morphological-Rank-Linear Time-lag Added Evolutionary Forecasting (MRLTAEF) method.**

## I. Introduction

Nowadays, prediction models have been widely used for decision making systems. Many efforts have been made to develop linear and nonlinear models able to predict a future behavior of a given phenomenon. Several linear and nonlinear statistical models were proposed for such [1]. However, those statistical models have some limitations, which prevents the development of an automatic prediction system [2].

Alternately, approaches based on Artificial Neural Networks (ANNs) [3] have been applied for the prediction of nonlinear time series [4]–[6]. However, in order to define a solution to a given problem, ANNs require the definition of a system parameters set. In addition to those, another crucial element is to determine the particular time lags that optimally characterize the time series generator phenomenon.

In this context, hybrid approaches have been investigated in the literature to solve the time series prediction problem [7]–[11]. Such approaches commonly use Evolutionary Algorithms (EAs) to evolve architecture and parameters of the prediction model. However, due to slower performance regarding time-consuming evaluation functions of the EAs, this implies in some practical limitations. In the attempt to improve the EAs performance, Cruz et al. [12] presented a Quantum-Inspired Evolutionary Algorithm (QIEA), which uses a novel real-valued representation, showing superior performance when compared to other well-established classical [13]–[16] and quantum-inspired [17]–[20] evolutionary algorithms.

In addition to those, research merging neural networks and quantum computing began in the late nineties. Quantum Artificial Neural Networks (QANN) represent a new paradigm based on classical neural computation and quantum computing [21]. Quantum-inspired computing [22] have been widely used to develop several quantum-inspired neural networks models (which is the branch of study characterized by some quantum mechanics concepts and it is performed into digital computers) like the Qubit Neural Networks Models [21], [23]–[26], showing its superior performance when compared to classical ANNs [3].

This paper introduces a quantum-inspired hybrid methodology to overcome the random walk dilemma (where the prediction generated by prediction models have one step delay regarding time series values) for financial time series prediction. It is inspired on Takens Theorem [27] and consists of a hybrid model composed of a Qubit MultiLayer Perceptron (QuMLP) [23] with a Quantum-Inspired Evolutionary Algorithm (QIEA) [12], which searches for the best time lags able to optimally represent the time series generator phenomenon, as well as to evolve the complete QuMLP architecture and parameters. Each individual of the QIEA population is trained and adjusted by the Complex Back-Propagation (CBP) [28] algorithm to further improve the QuMLP parameters supplied by the QIEA. After the prediction model search procedure, the proposed methodology uses a behavioral statistical test and a phase fix procedure to adjust time phase distortions that appear in financial time series.

Furthermore, an experimental analysis is conducted with the proposed methodology using four real world financial time series and the obtained results shows better performance of the proposed methodology when compared to Multilayer Perceptron (MLP) networks and the previously introduced Morphological-Rank-Linear Time-lag Added Evolutionary Forecasting (MRLTAEF) [9] method, which shown boosted performance enhancement when compared to classical prediction models.

## II. Fundamentals

### A. The Time Series Prediction Problem

A time series is a sequence of observations about a given phenomenon in discrete or continuous time space. Usually, a time series can be defined by

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

where $t$ is the temporal index and $N$ is the number of observations. The term $X_t$ will be seen as the set of temporal

Ricardo de A. Araújo is with the Information Technology Department, [gm]$^2$ Intelligent Systems, Brazil; Email: ricardo@gm2.com.br.

Adriano L. I. de Oliveira is with the Statistics and Informatics Department, Rural Federal University of Pernambuco, Brazil; Email: alio@cin.ufpe.br.

Sergio C. B. Soares is with the Informatics Center, Federal University of Pernambuco, Brazil; Email: scbs@cin.ufpe.br.

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

observations of a given phenomenon, orderly sequenced and equally spaced.

The aim of prediction techniques applied to a given time series ($X_t$) is to provide a mechanism that allows, with certain accuracy, the prediction of future values of $X_t$, given by $X_{t+k}$, $k = 1, 2, ...$, where $k$ represents the prediction horizon. These prediction techniques will try to identify certain regular patterns present in the data set, creating a model able to generate the next temporal patterns, where, in this context, a most relevant factor for an accurate prediction performance is the correct choice of the past window, or the time lags, considered for the representation of a given time series generator phenomenon.

### B. The Random Walk Dilemma

A naive prediction strategy is to define the last observation of a time series as the best prediction of its next future value ($X_{t+1} = X_t$). This kind of model is known as the random walk (RW) model [5], [29], which is defined by

$$X_t = X_{t-1} + r_t, \tag{2}$$

or

$$\Delta X_t = X_t - X_{t-1} = r_t, \tag{3}$$

where $X_t$ is the current observation, $X_{t-1}$ is the immediate observation before $X_t$, and $r_t$ is a noise term with a gaussian distribution of zero mean and standard deviation $\sigma$ ($r_t \approx N(0, \sigma)$). In other words, the rate of time series change ($\Delta X_t$) is a white noise.

The model above clearly implies that, as the information set consists of past time series data, the future data is unpredictable. On average, the value $X_t$ is indeed the best prediction of value $X_{t-1}$. This behavior is common in the finance and economics and is called random walk dilemma or random walk hypothesis [5], [29].

The computational cost for time series forecasting using the random walk dilemma is extremely low. Therefore, any other prediction method more costly than a random walk model should have a very superior performance than a random walk model. Otherwise its use is not interesting in practice.

However, if the time series phenomenon is driven by a law with strong similarity to a random walk model, any model applied to this time series phenomenon will tend to have the same performance as a random walk model.

Assuming that an accurate prediction model is used to build an estimated value of $X_t$, denoted by $\widehat{X_t}$, the expected value ($E[\cdot]$) of the difference between $\widehat{X_t}$ and $X_t$ must tend to zero,

$$E[\widehat{X_t} - X_t] \to 0. \tag{4}$$

If the time series generator phenomenon is supposed to have a strong random walk linear component and a very weak nonlinear component (denoted by $g(t)$), and assuming that $E[r_t] = 0$ and $E[r_t r_k] = 0$ ($\forall\ k \neq t$), the expected value of the difference between $\widehat{X_t}$ and $X_t$ (assuming that $X_t = X_{t-1} + g(t) + r_t$) will be

$$E[\widehat{X_t} - (X_{t-1} + g(t) + r_t)] \to 0, \tag{5}$$

$$E[\widehat{X_t}] - E[X_{t-1}] - E[g(t)] - E[r_t] \to 0, \tag{6}$$

$$E[\widehat{X_t}] - E[X_{t-1}] - E[g(t)] \to 0, \tag{7}$$

$$E[\widehat{X_t}] \to E[X_{t-1}] + E[g(t)]. \tag{8}$$

If $E[g(t)] \to 0$, then $E[X_{t-1}] + E[g(t)] \to E[X_{t-1}]$ and

$$E[\widehat{X_t}] \to E[X_{t-1}]. \tag{9}$$

Therefore, in these conditions, escaping the random walk dilemma is a hard task [5], [29]. Indications of this behavior (strong linear random walk component and a weak nonlinear component) can be observed from time series lagplots.

### C. The Qubit MultiLayer Perceptron

The Qubit MultiLayer Perceptron (QuMLP) used in this work is proposed by [23] and [25]. The QuMLP uses the Complex Back-Propagation (CBP) [28] to learn the relative phase information of neurons quantum states. The QuMLP differs from classical MLP in the sense that its neurons uses quantum mechanics concepts. Kouda et al. [25] and Mitrpanont and Srisuphab [23], presented a QuBit MultiLayer Perceptron (QuMLP) in the attempt to adjust the relative phase information of neurons quantum states, where all information that flows through the Qubit Neuron is complex encoded.

Therefore, for presenting real data to Qubit Neuron, all points are mapped into complex numbers through a phase $\theta_i \in (0, \frac{\pi}{2})$ in order to create the complex inputs that stores the phase information, given by

$$x_j = e^{i\theta_j}, \tag{10}$$

where $\theta_j = \frac{\pi}{2} X_j$, with $X_j$ representing the normalized real-valued data (into range $[0, 1]$).

*1) Qubit Neuron Model:* In the quantum computing field, term Qubit has been introduced as the counterpart of the bit in classical computers in order to describe the circuit state of quantum computation [25]. In this way, an interesting work presented, through the connection of the neuron states (classical neuron model) and quantum states, the QuBit-like neuron model, which uses the state $|1\rangle$ as firing neuron state and the state $|0\rangle$ as non-firing neuron state, being an arbitrary neuron state defined by the coherent superposition of two [23].

The Qubit Neuron performs two distinct steps: i) the weighted sum of input complex signals phase, and ii) the nonlinear activation of the quantum neuron. The first step is given by

$$u = \sum_{i=1}^{n} w_i x_i + b, \tag{11}$$

where $x_i \in \mathbb{C}$ ($i = 1, 2, \ldots, n$) represents the complex input signal and $w_i \in \mathbb{C}$ represents the neuron weights and term $b \in \mathbb{C}$, in analogy of classical neuron, denotes the neuron bias.

The second step is given by

$$y = f(u), \tag{12}$$

with

$$f(u) = sig(Re[u]) + i \cdot sig(Im[u]), \tag{13}$$

where $Re[\cdot]$ and $Im[\cdot]$ represents the real and imaginary part, respectively, of a given complex number, $i$ represents the imaginary unit and $y$ is the Qubit Neuron output. The term $sig(\cdot)$ is a sigmoidal function defined by

$$sig(x) = \frac{1}{1 + e^{-x}}. \tag{14}$$

Figure 1 illustrates the QuBit Neuron model scheme.

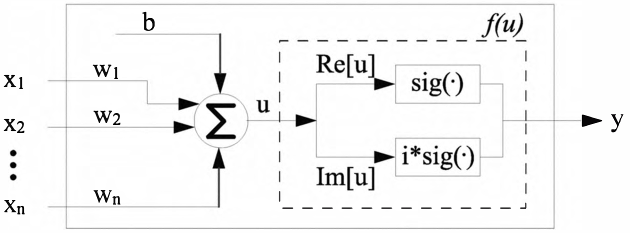

Fig. 1. The QuBit Neuron scheme.

The quantum state neuron output interpretation is based on the probability amplitude and according to the decoherence of the quantum state. Through quantum state mapping scheme using sigmoid function with range within 0 and 1, then the real world value of neuron output is given by

$$o = Im[y]^2 = sin^2(\theta), \tag{15}$$

where $\theta = tan^{-1}\frac{Im[y]}{Re[y]}$.

It is worth to mention that the function $f : \mathbb{C} \to \mathbb{C}$ is continuous and differentiable, which allows its usage on the training algorithm [23]. The superposition effects arise naturally as a result of the network dynamics due to the complex signals arithmetic which performs quantum state phase rotations.

Also, it is important to notice that the real output $o$ of the QuMLP is considered to be the inverse mapping of Equation 10 over the phase of the complex response (Equation 12) of the output layer neuron of QuMLP.

*2) Quantum Complex Back-Propagation:* The QuMLP is trained using the Complex Back-Propagation (CBP) [28] in order to adjust the complex number information. The CBP performs descent-gradient minimization of a cost function, defined by the sum of squared error, which is given by

$$E = \frac{1}{2}\sum_{j=1}^{P}\sum_{i=1}^{N}(d_i^j - y_i^j)^2, \tag{16}$$

where $P$ is the number of input patterns and $N$ the number of output neurons in QuMLP. Term $y_i^j$ represents the $i$-th output of QuMLP of the $j$-th input pattern. Term $d_i^j$ represents the $i$-th desired output of QuMLP of the $j$-th input pattern.

In this way, let $L$ be the last layer of QuMLP and $F$ its first layer, the complex weight adjustment of $L$ layer of the QuMLP is given by [28]

$$w_{i(t+1)}^{L} = w_{i(t)}^{L} + \Delta_i^L, \tag{17}$$

with

$$\Delta_i^L = \overline{y_{ji}^{L-1}} \cdot \mu \cdot \delta_{ji}^L, \tag{18}$$

and

$$\delta_{ji}^L = (d_j i - y_{ji}^L) \cdot f'(u_j^L), \tag{19}$$

where $\overline{y_{ji}^{L-1}}$ the complex conjugate of output $y_{ji}$ of neuron $i$ of layer $L-1$ of the $j$-th input pattern. Term $\mu$ represents the learning rate. Terms $d_j i$ and $y_{ji}^L$ represent the desired output and the output of $i$-th neuron for the $j$-th input pattern.

The complex weight adjustment of intermediate layers, denoted by $IL$, of the QuMLP is given by [28]

$$w_{i(t+1)}^{IL} = w_{i(t)}^{IL} + \Delta_i^{IL}, \tag{20}$$

with

$$\Delta_i^{IL} = \overline{y_{ji}^{IL-1}} \cdot \mu \cdot \delta_{ji}^{IL}, \tag{21}$$

and

$$\delta_{ji}^{IL} = f'(u_{ji}^{IL}) \cdot \sum_{k=1}^{N^{IL+1}} \delta_{jk}^{IL+1} w_{ki}^{IL+1}, \tag{22}$$

where $N^{IL+1}$ is number of neurons at layer $IL+1$.

### *D. The Quantum-Inspired Evolutionary Algorithm*

Quantum-Inspired Evolutionary Algorithms (QIEAs) are based on quantum bits concepts, referred to as Qubits, and also on the superposition of states of quantum computing [17], which can be expressed as a linear combination between states $|0\rangle$ and $|1\rangle$, where $|\cdot\rangle$ is a quantum state. A Qubit can be defined by [17]

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{23}$$

where $\alpha$, $\beta \in \mathbb{C}$ represents the probabilities amplitudes of states $|0\rangle$ and $|1\rangle$, respectively.

In this way, $|\alpha|^2$ and $|\beta|^2$ gives the probability that the Qubit will be found in states $|0\rangle$ and $|1\rangle$, respectively, where the normalization of the state to unity guarantees [17]

$$|\alpha|^2 + |\beta|^2 = 1. \tag{24}$$

Inspired by concept of quantum computing and based on [17], Cruz et al. [12] proposed a new QIEA using a novel real-valued representation, which is described in the next section. The QIEA procedure is described in Figure 2.

```
begin QIEA
    $t = 0$;                // $t$: actual iteration
    create quantum population;
    while not termination condition do
        $t = t + 1$;
        create the PDF's using quantum individuals;
        generate temporary classical population observing
        quantum population and using CDF's;
        if $t=l$ then
            classical population $\longleftarrow$ temporary classical
            population;
        else
            temporary classical population $\longleftarrow$ crossover
            operator between temporary classical population
            and classical population;
            evaluate temporary classical population;
            classical population $\longleftarrow$ $K$ best individuals from
            temporary classical population;
        end
        begin with the individuals of classical population
            quantum population $\longleftarrow$ apply translate operation
            to classical population;
            quantum population $\longleftarrow$ apply resize operation to
            quantum population;
        end
    end
end
```

Fig. 2. Quantum-inspired evolutionary algorithm procedure.

The QIEA quantum population at generation $G$ is defined by $\underline{QP}_G$, which is given by a superposition of states that are observed to generate classical individuals (possible solutions of the problem), where it is defined by,

$$\underline{QP}_G = (\underline{QP}_{1,G}, \underline{QP}_{2,G}, \ldots, \underline{QP}_{S,G}), \tag{25}$$

with

$$\underline{QP}_{i,G} = (QP_{i1}, QP_{i2}, \ldots, QP_{iN}), \tag{26}$$

and

$$QP_{ij} = (\rho_{ij}, \sigma_{ij}), \tag{27}$$

in which $\underline{QP}_G$ denotes the quantum population at generation $G$; $\underline{QP}_{i,G}$ denotes $i$-th quantum individual of the population $\underline{QP}_G$; $QP_{ij}$ denotes the $j$-th parameter (gene) of the $i$-th individual of the population; $S$ denotes the quantum population size (number of quantum individuals); $N$ denotes the number of individual parameters (genes or variables of the problem). Terms $\rho_{ij}$ and $\sigma_{ij} \in \mathbb{R}$ represents the center and the width of a square pulse, which is used in order to build the set of possible observable values over the problem domain [12]. The height ($h_{ij}$) of each pulse is defined using the quantum gene width ($\sigma_{ij}$) and the maximum number of quantum individuals ($N$) in quantum population, and it is given by [12]

$$h_{ij} = \frac{1/\sigma_{ij}}{N}. \tag{28}$$

In this classical individuals generating step the interference process among quantum individuals is performed to generate a Probability Density Function (PDF) [12]. The PDF consists of summing up the quantum individuals genes. In other words, the first gene of all quantum individuals are summed. All other genes of a given quantum individual do the same. The PDF is defined by [12]

$$PDF_j = \sum_{i=1}^{N} QP_{ij}, \tag{29}$$

where $QP_{ij}$ represents the square pulse with width $\sigma_{ij}$ and center $\rho_{ij}$ of the $j$-th gene of the $i$-th quantum individual.

According to Cruz et al. [12] these PDFs will be able to generate the classical individuals, which are a real-valued vector with same amount of quantum individuals genes, where such values are randomly selected using the PDFs as probability function. In order to perform such random selection, Cruz et al. [12] defined a Cumulative Distribute Function (CDF), which is given by

$$CDF_j(x) = \int_l^u PDF_j(x)dx, \tag{30}$$

where $u$ and $l$ represent the upper and lower bounds of PDF$_j$ function.

After, as all PDFs are built by a sum of square pulses, it is possible to calculate the PDF area by dividing the function curve in rectangles and by summing up its corresponding area, and then is possible to verify that CDFs can then be calculated using such PDFs based on these rectangles [12].

Through these CDFs, according to Cruz et al. [12], it is possible to generate a set of classical individuals by using such curves. Therefore, the classical population is created by an uniform choice of random numbers in the range $[0, 1]$ and by the identification of these points in CDF. This operation is mathematically defined by [12]

$$x = CDF^{-1}(r), \tag{31}$$

where $r$ is a random number in the range $[0, 1]$.

This procedure allows the build of the Temporary Classical Population ($TCP$), which is responsible to the classical individuals observed into quantum population. At the first QIEA generation, the Classical Population ($CP$), which is the best observations (in terms of fitness function) of quantum population, is a clone of the $TCP$. For next generations, the multi-point crossover operator [12] is applied in the classical population of the QIEA in order to generate better classical individuals, hence improving the quantum population.

After the $CP$ generation, it is necessary to update the quantum individuals in the quantum population. The first step, referred to as translate operation, is responsible to update the center ($\rho$) of each quantum genes. In this way, Cruz et al. [12] shown a simple procedure to do this, which replaces the mean of each gene values to the values of the genes from classical individuals. This step is formally defined by,

$$\rho_{ij} = CP_{ij}, \tag{32}$$

where $\rho_{ij}$ represents the center of $j$-th gene of the $i$-th quantum individual from quantum population, and $CP_{ij}$ denotes the $j$-th gene of the $i$-th classical individual from $CP$.

The second step, referred to as resize operation, is responsible to reducing or enlarging the width ($\sigma_{ij}$) of quantum gene [12]. This change should be made homogeneously for all quantum genes and for all quantum individuals [12]. Cruz et al. [12] used the $1/5th$ rule to determine if such width should be enlarged or reduced, which is given by [12]

$$\sigma_{ij} = \begin{cases} \sigma_{ij} \cdot \delta & \text{if } \varphi < 1/5 \\ \sigma_{ij}/\delta & \text{if } \varphi > 1/5 \\ \sigma_{ij} & \text{otherwise} \end{cases}, \tag{33}$$

where $\sigma_{ij}$ represents the width of $j$-th gene of the $i$-th quantum individual from quantum population, $\delta$ denotes a random number in interval $[0, 1]$, and $\varphi$ is the rate of how many classical individuals generated in new generation have their overall evaluation improved.

## III. The Proposed Methodology

The methodology presented in this paper uses a quantum-inspired evolutionary search mechanism in order to train and to adjust Qubit MultiLayer Perceptron (QuMLP) applied to financial time series prediction. It is based on the definition of the three main elements necessary for building an accurate prediction system [9]: (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 network 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 time series representation because the model must be as parsimonious as possible.

The proposed methodology consists of a quantum-inspired hybrid model composed of a QuMLP with a QIEA, which determines the following important parameters: (i) The minimum number of time lags to represent the time series: initially, a maximum number of lags (*MaxLags*) is predefined and the QIEA 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, and (ii) The number of units in the QuMLP hidden layer: the maximum number of hidden layer units (*NHiddenMax*) is predetermined and the QIEA chooses, for each candidate individual, the number of units in the hidden layer (in the interval [1,*NHiddenMax*]). The Complex Back-Propagation (CBP) [28] is then used to train each individual of the QIEA classical population, as it has been proved to be effective in speeding up the training while limiting its computational complexity.

These processes offer the effective capacity to seek the most compact QuMLP, reducing the computational cost and the probability of the overfitting problem. Each QIEA classical individual represents a three-layer QuMLP, where the first layer is defined by the number of time lags, the second layer is composed by the number of hidden processing units and the third layer is composed by one processing unit (prediction horizon of one step ahead). Figure 3 shows the proposed methodology scheme.

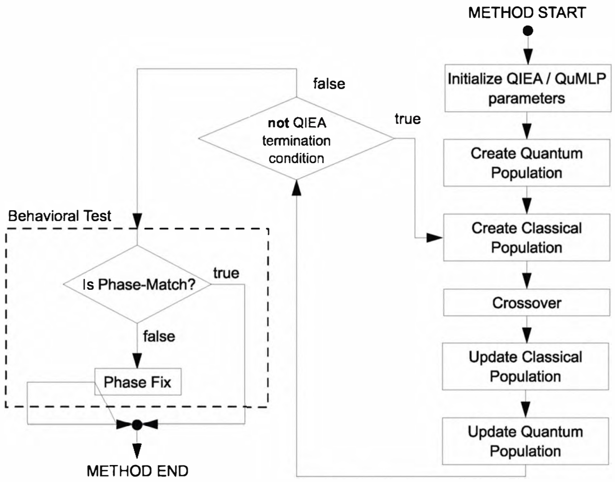
METHOD START
Initialize QIEA / QuMLP parameters
Create Quantum Population
Create Classical Population
Crossover
Update Classical Population
Update Quantum Population
not QIEA termination condition
true
false
Behavioral Test
Is Phase-Match?
true
false
Phase Fix
METHOD END

Fig. 3. The proposed methodology scheme.

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. [2], shown that the MSE measure has some limitations to evaluate the prediction model performance. In order to provide a more robust prediction model, it uses new a fitness function, 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) and Average Relative Variance (ARV), where all these measures were formally defined in [9]. The fitness function used here is given by

$$fitness = \frac{POCID}{1 + MSE + MAPE + NMSE + ARV}. \tag{34}$$

After model training (at the end of methodology iterations), it is used the phase fix procedure presented by Araújo and Ferreira [9], where a two step procedure is introduced to adjust time phase distortions observed in financial time series. Araújo and Ferreira [9] shows 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 methodology uses the statistical test (t-test) to check if the QuMLP 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 does 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 methodology performs a new procedure to adjust the relative phase between the prediction and the actual time series. The phase fix procedure has two steps (described in the Figure 4): (i) the validation patterns are presented to the QuMLP 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 QuMLP and the output set as the prediction target. This procedure of phase adjustment considers that the QuMLP does 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. If the QuMLP was like a random walk model, the phase adjustment procedure would not work. In this way, it is observed the fact that when QuMLP is correctly adjusted, the one step shift distortion in the prediction can be softened.

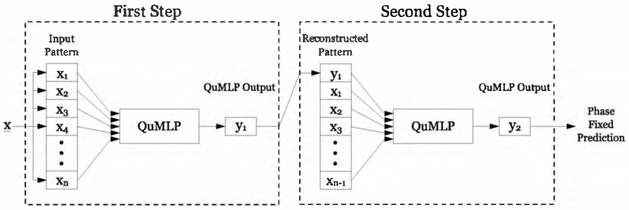

Fig. 4. Phase fix procedure.

The termination conditions for the QIEA 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 ($MSE+MAPE+NMSE+ARV \cong 1$);

2) The increase in the validation error or generalization loss ($Gl$) [30]: $Gl > 5\%$;
3) The decrease in the training error process training ($Pt$) [30]: $Pt \leq 10^{-6}$.

Each individual of the QIEA classical population is a QuMLP represented by the data structure with the following components (QuMLP parameters):

- $W_{ij}$: phase of weights of connections between the input layer and the hidden layer
- $W_{jk}$: phase of weights of connections between the hidden layer and the output layer;
- NHidden: the number of processing units in the QuMLP hidden layer;
- NLags: a vector, where each position has a real-valued codification, which is used to determine if a specific time lag will be used ($\text{NLags}_i \geq 0$) or not ($\text{NLags}_i < 0$).

## IV. Simulations and Experimental Results

A set of four real world financial time series was used as a test bed for evaluation of the proposed methodology: Applied Materials Inc Stock Prices, Exxon Mobil Corporation Stock Prices, General Electric Company Stock Prices and Hewlett-Packard Company Stock Prices. All time series investigated corresponds to the daily records of NASDAQ Stock Market from June 23th 2000 to June 22th 2007, constituting a database of 1,758 points. These time series were normalized to lie within the range $[0, 1]$ and divided in three sets according to Prechelt [30]: training set (50% of the data points – 879 points), validation set (25% of the data points – 440 points) and test set (25% of the data points – 439 points).

The QIEA parameters are the same for all experiments: the maximum number of generations is $10^3$, number quantum individuals equals to 10 and the crossover probability ($p_{cross} = 0.9$). The QIEA classical population is composed of 10 individuals, where each individual is a QuMLP with the maximum architecture: a 10-10-1 network, which denotes 10 units in the input layer ($MaxLags$), 10 units in the hidden layer ($MaxHidden$) and 1 unit in the output layer (prediction horizon of one step ahead). The QuMLP is trained by the CBP [28] algorithm for $10^4$ training epochs using a learning rate equals to 0.01. The termination conditions for the CBP 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 ($Pt < 10^{-6}$).

In order to establish a performance study, results previously published in the literature with the MRLTAEF method [9] 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 methodology. In all of the experiments, ten random initializations for each model were carried out, where the experiment with the more large validation fitness function is chosen to represent the prediction model. The Levenberg Marquardt Algorithm [31] 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 QuMLP training in the QIEA system (Epochs, $Gl$ and $Pt$ ). For all the series, the best MLP initialization was elected to represent the prediction model. The statistical behavioral test, for phase fix procedure, was also applied to all the MLP and MRLTAEF models in order to guarantee a fair comparison among the models.

The results (regarding test set with out-of-phase matching, that is, with adjusted time phase) for the Applied Materials Inc Stock Prices series with all the performance measures for MLP, MRLTAEF and proposed models are presented in Table I.

TABLE I

Prediction performance for the Applied Materials Inc Stock Prices series.

| | MLP | MRLTAEF | Proposed Model |
|---|---|---|---|
| MSE | 5.9457e-5 | 1.1428e-7 | 2.6685e-8 |
| MAPE | 2.7718e-2 | 1.6847e-3 | 5.4662e-4 |
| NMSE | 0.9892 | 2.8729e-3 | 4.4581e-4 |
| ARV | 5.1824e-2 | 1.5032e-4 | 2.6122e-5 |
| POCID | 54.00 | 97.48 | 99.18 |
| Fitness | 26.1021 | 97.0232 | 99.0791 |

The Figure 5 shows the actual Applied Materials Inc Stock Prices (solid line) and the predicted values generated by the proposed model for both in-phase and out-of-phase matching (dashed line) for the last one hundred points of the test set.

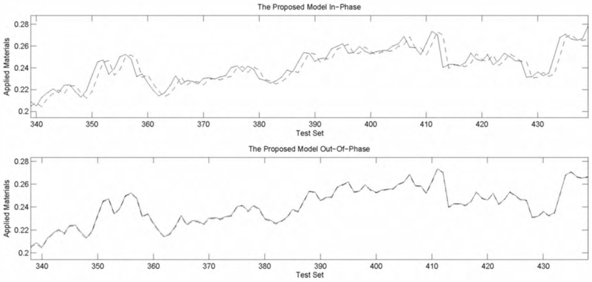

Fig. 5. Prediction results for the Applied Materials Inc Stock Prices series (test set): actual values (solid line) and predicted values (dashed line).

The obtained results with all the performance measures for the Exxon Mobil Corporation Stock Prices series, regarding test set with out-of-phase matching, that is, with adjusted time phase, for MLP, MRLTAEF and the proposed models are presented in Table II.

TABLE II

Prediction Performance for the Exxon Mobil Corporation Stock Prices series.

| | MLP | MRLTAEF | Proposed Model |
|---|---|---|---|
| MSE | 4.0748e-4 | 2.4791e-4 | 2.4368e-9 |
| MAPE | 2.3552e-2 | 1.1017e-2 | 7.2832e-5 |
| NMSE | 1.7801 | 0.6891 | 1.0730e-5 |
| ARV | 1.9928e-2 | 1.1438e-2 | 1.2032e-7 |
| POCID | 49.88 | 97.03 | 98.86 |
| Fitness | 17.6630 | 56.6829 | 98.8517 |

Figure 6 illustrates the actual Exxon Mobil Corporation Stock Prices (solid line) and the predicted values generated by the proposed model for both in-phase and out-of-phase matching (dashed line) for the last one hundred points of the test set for both cases: in-phase and out-of-phase matching models.

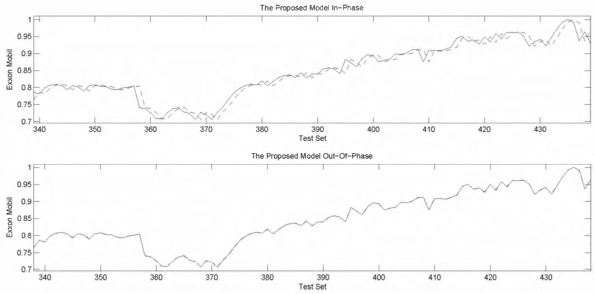

Fig. 6. Prediction results for the Exxon Mobil Corporation Stock Prices series (test set): actual values (solid line) and predicted values (dashed line).

Table III presents the results for the General Electric Company Stock Prices series (subject of the test set with out-of-phase matching, that is, with adjusted time phase) for all the performance measures for MLP, MRLTAEF and the proposed models.

TABLE III
PREDICTION PERFORMANCE FOR THE GENERAL ELECTRIC COMPANY STOCK PRICES SERIES.

| | MLP | MRLTAEF | Proposed Model |
|---|---|---|---|
| MSE | 6.0758e-5 | 5.8721e-7 | 4.0754e-8 |
| MAPE | 1.7509e-2 | 1.7725e-3 | 5.9669e-4 |
| NMSE | 1.0111 | 9.7126e-3 | 6.8711e-4 |
| ARV | 4.1323e-2 | 3.4379e-4 | 2.7958e-5 |
| POCID | 47.14 | 94.06 | 97.03 |
| Fitness | 22.7730 | 92.9603 | 96.9029 |

The Figure 7 exhibits the actual General Electric Company Stock Prices (solid line) and the predicted values generated by the proposed model for both in-phase and out-of-phase matching (dashed line) for the last one hundred points of the test set for the in-phase and out-of-phase matching models.

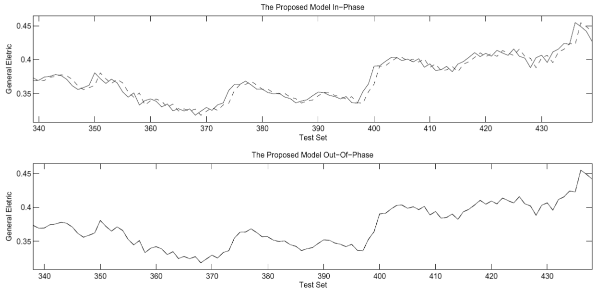

Fig. 7. Prediction results for the General Electric Company Stock Prices series (test set): actual values (solid line) and predicted values (dashed line).

In Table IV it is possible to show the results for the Hewlett Packard Company Stock Prices series (concerning to the test set with out-of-phase matching, that is, with adjusted time phase) with all the performance measures for MLP, MRLTAEF and the proposed models.

The Figure 8 shows the actual Hewlett Packard Company Stock Prices (solid line) and the predicted values generated by the proposed model for both in-phase and out-of-phase matching (dashed line) for the last one hundred points of the test set for in-phase and out-of-phase matching models.

TABLE IV
PREDICTION PERFORMANCE FOR THE HEWLETT PACKARD COMPANY STOCK PRICES SERIES.

| | MLP | MRLTAEF | Proposed Model |
|---|---|---|---|
| MSE | 8.2013e-5 | 1.0182e-6 | 2.2308e-7 |
| MAPE | 1.6384e-2 | 1.9735e-3 | 1.0573e-3 |
| NMSE | 1.0331 | 1.1243e-2 | 8.8790e-4 |
| ARV | 9.0904e-3 | 1.2859e-4 | 2.4762e-5 |
| POCID | 49.36 | 95.66 | 99.08 |
| Fitness | 23.9768 | 94.4001 | 98.8852 |

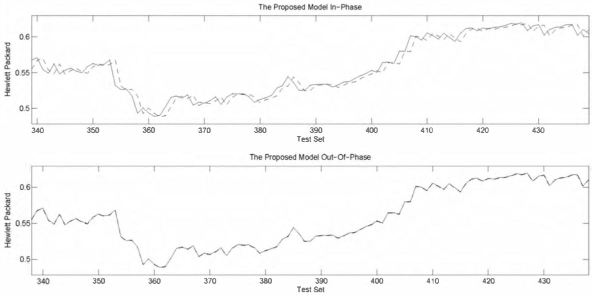

Fig. 8. Prediction results for the Hewlett Packard Company Stock Prices series (test set): actual values (solid line) and predicted values (dashed line).

In general, all generated prediction models using the phase fix procedure (out-of-phase matching) to adjust time phase distortions shown prediction performance much better than the MLP model, and slightly better than the MRLTAEF model. The proposed methodology 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 was not able to adjust the time phase. This corroborates with the assumption made by Araújo and Ferreira [9], where they 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.

## V. CONCLUSION

A quantum-inspired hybrid methodology was presented to overcome the random walk dilemma for financial time series prediction, where the predicted values of a given financial time series are shifted one step ahead of the real time series values. It performs a quantum-inspired evolutionary search for the minimum dimension necessary to determine the characteristic phase space that generates the financial time series phenomenon.

Five different metrics were used to measure the performance of the proposed methodology for financial time series prediction, where a fitness function was built in order to improve the description of the time series phenomenon as better 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 will must be done to determine the optimal combination of such metrics.

The results were collected with four real world time series from the financial stock market with all their dependence on exogenous and uncontrollable variables. The experimental results demonstrated a consistent much better performance of the proposed model when compared to the MLP networks, and better performance than the MRLTAEF method [9], which had previously shown boosted performance enhancement when compared to other results found in the literature.

It was also observed that the proposed model obtained a much better performance than a random walk model [29] for the analyzed financial time series, overcoming the random walk dilemma (where the predicted values are shifted one step ahead of the original values). While the proposed 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 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 prediction.

Finally, the results shown that the phase fix procedure [9] was able to correct more efficiently the time phase of the proposed model when compared to MRLTAEF model [9]. Further studies are being developed to better formalize and explain the properties of the proposed model and to determine possible limitations of the methodology with other financial time series with components such as trends, seasonalities, impulses, steps and other nonlinearities. Also, 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 methodology. Also, a particular study about the computing complexity and CPU time of the proposed model must be done in order to establish a complete cost-performance evaluation of the proposed model. According to this investigation, it will be possible to relate, in terms of cost, the necessary time to generate an optimal prediction model.

## Acknowledgment

This work was partially supported by the National Institute of Science and Technology for Software Engineering (INES), funded by CNPq and FACEPE, grants 573964/2008-4 and APQ-1037-1.03/08.

## References

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

[2] M. P. Clements, P. H. Franses, and N. R. Swanson. Forecasting economic and financial time-series with non-linear models. *International Journal of Forecasting*, 20(2):169–183, 2004.

[3] S. Haykin. *Neural networks: A comprehensive foundation.* Prentice Hall, New Jersey, 1998.

[4] G. Zhang, B. E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14:35–62, 1998.

[5] Renate Sitte and Joaquin Sitte. Neural networks approach to the random walk dilemma of financial time series. *Applied Intelligence*, 16(3):163–171, May 2002.

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

[7] J. Binner, G. Kendall, and A. Gazely. Evolving neural networks with evolutionary strategies: A new application to divisia money. *Advances in Econometrics*, 19:127–143, 2004.

[8] Mariano Matilla-García and Carlos Argüello. A hybrid approach based on neural networks and genetic algorithms to the study of profitability in the spanish stock market. *Applied Economics Letters*, 12(5):303–308, April 2005.

[9] R. A. Araújo and T. A. E. Ferreira. An intelligent hybrid morphological-rank-linear method for financial time series prediction. *Neurocomputing*, 72(10-12):2507–2524, 2009.

[10] R. de A. Araújo and T. A. E. Ferreira. A morphological-rank-linear evolutionary method for stock market prediction. *Information Sciences*, 2010. In Press.

[11] R. de A. Araújo. A hybrid intelligent morphological approach for stock market forecasting. *Neural Processing Letters*, 2010. In Press.

[12] André V. A. da Cruz, Marley M. B. R. Vellasco, and Marco A. C. Pacheco. Quantum-inspired evolutionary algorithm for numerical optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Vancouver, Canada, 2006.

[13] M. Gen and R. Cheng. *Genetic algorithms and engineering optimization.* John Wiley, New York, 2000.

[14] C. T. Chan, C. H. Chuek, D. P. Mital, and T. T. Huat. Time series modelling and forecasting using genetic algorithms. In L. C. Jain, editor, *First International Conference on Knowledge-Based Intelligent Electronic Systems*, pages 260–268, Adelaide, 1997.

[15] F. H. F. Leung, H. K. Lam, S. H. Ling, and P. K. S. Tam. Tuning of the structure and parameters of the neural network using an improved genetic algorithm. *IEEE Transactions on Neural Networks*, 14(1):79–88, January 2003.

[16] S. H. Ling, H. K. Lam, F. H. F. Leung, and Y. S. Lee. A genetic algorithm based neural-tuned neural network. In *The 29th Annual Conference of the IEEE Industrial Eletronics Society - IECON'03*, volume 3, pages 2423–2428, 2003.

[17] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. Evolutionary Computation*, 6(6):580–593, 2002.

[18] Kuk-Hyun Han and Jong-Hwan Kim. Quantum-inspired evolutionary algorithms with a new termination criterion, $h_{epsilon}$ gate, and two-phase scheme. *IEEE Trans. Evolutionary Computation*, 8(2):156–169, 2004.

[19] Jun-Su Jang, Kuk-Hyun Han, and Jong-Hwan Kim. Evolutionary algorithm-based face verification. *Pattern Recogn. Lett.*, 25(16):1857–1865, 2004.

[20] Jun-Su Jang, Kuk-Hyun Han, and Jong-Hwan Kim. Face detection using quantum-inspired evolutionary algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Portland, Oregon, 2004.

[21] Rigui Zhou, Ling Qin, and Nan Jiang. Quantum perceptron network. In *ICANN (1)*, pages 651–657, 2006.

[22] M. Moore and A. Narayanan. Quantum-inspired computing. *M. Moore and A. Narayanan, Quantum-Inspired Computing, Dept. Comput. Sci., Univ. Exeter, Exeter, U.K., 1995.*, 1995.

[23] J. L. Mitrpanont and A. Srisuphab. The realization of quantum complex-valued backpropagation neural network in pattern recognition problem. *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02*, 1:462– 466, 2002.

[24] Noriaki Kouda, Nobuyuki Matsui, and Haruhiko Nishimura. Image compression by layered quantum neural networks. *Neural Process. Lett.*, 16(1):67–80, 2002.

[25] Noriaki Kouda, Nobuyuki Matsui, and Haruhiko Nishimura. A multilayered feed-forward network based on qubit neuron model. *Systems and Computers in Japan*, 35(13):43–51, 2004.

[26] Noriaki Kouda, Nobuyuki Matsui, Haruhiko Nishimura, and Ferdinand Peper. An examination of qubit neural network in controlling an inverted pendulum. *Neural Process. Lett.*, 22(3):277–290, 2005.

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

[28] T. Nitta. Structure of learning in the complex numbered back-propagation network. *IEEE International Conference on Neural Networks*, 1:269–274, 1994.

[29] T. C. Mills. *The Econometric Modelling of Financial Time Series.* Cambridge University Press, Cambridge, 2003.

[30] Lutz Prechelt. Proben1: A set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, 1994.

[31] M. Hagan and M. Menhaj. Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6):989–993, November 1994.