University of Pretoria
Department of Economics Working Paper Series

Hybrid ARFIMA wavelet artificial neural network model for DJIA Index Forecasting

Heni Boubaker
International University of Rabat
Giorgio Canarella
University of Nevada
Rangan Gupta
University of Pretoria
Stephen M. Miller
University of Nevada

Working Paper: 2020-56
June 2020

Department of Economics
University of Pretoria
0002, Pretoria
South Africa
Tel: +27 12 420 2413

Hybrid ARFIMA Wavelet Artificial Neural Network Model for DJIA Index Forecasting

Heni Boubaker[a]

Giorgio Canarella[b]

Rangan Gupta[c]

and

Stephen M. Miller[b]

Abstract: This paper proposes a hybrid modelling approach for forecasting returns and volatilities of the stock market. The model, called ARFIMA-WLLWNN model, integrates the advantages of the ARFIMA model, the wavelet decomposition technique (namely, the discrete MODWT with Daubechies least asymmetric wavelet filter) and artificial neural network (namely, the LLWNN neural network). The model develops through a two-phase approach. In phase one, a wavelet decomposition improves the forecasting accuracy of the LLWNN neural network, resulting in the Wavelet Local Linear Wavelet Neural Network (WLLWNN) model. The Back Propagation (BP) and Particle Swarm Optimization (PSO) learning algorithms optimize the WLLWNN structure. In phase two, the residuals of an ARFIMA model of the conditional mean become the input to the WLLWNN model. The hybrid ARFIMA-WLLWNN model is evaluated using daily closing prices for the Dow Jones Industrial Average (DJIA) index over 01/01/2010 to 02/11/2020. The experimental results indicate that the PSO-optimized version of the hybrid ARFIMA-WLLWNN outperforms the LLWNN, WLLWNN, ARFIMA-LLWNN, and the ARFIMA-HYAPARCH models and provides more accurate out-of-sample forecasts over validation horizons of one, five and twenty-two days.

JEL Code: C45, C58, G17

Keywords: Wavelet decomposition, WLLWNN, Neural network, ARFIMA, HYGARCH

a International University of Rabat, BEAR LAB, Technopolis Rabat-Shore Rocade Rabat-Salé. Morocco.
b Department of Economics, Lee Business School, University of Nevada, Las Vegas; Las Vegas, Nevada
c Economics Department, University of Pretoria, Pretoria, 0002, South Africa.

Research highlights

- A wavelet decomposition improves the forecasting accuracy of the LLWNN model.
- A hybrid method forecasts the DJIA series.
- The hybrid model, called ARFIMA-WLLWNN, combines parametric ARFIMA model, Wavelet decomposition, and non-parametric LLWNN methods.
- Applying two learning algorithms, BP and PSO, obtains optimum WLLWNN structure to avoid overfitting.
- The PSO-optimized version of the hybrid ARFIMA-WLLWNN outperforms the LLWNN, WLLWNN, ARFIMA-LLWNN, and the ARFIMA-HYAPARCH models.

1. Introduction

Forecasting the stock market is a not a trivial task. On the contrary, it is one of the most challenging applications in economics and finance. The stock market is a complex, dynamic allocative mechanism (Baumol, 1965), characterized by nonlinear and complex dimensionalities (Guresen et al., 2001 ; Lee and Chiu, 2002), high data intensity and noise (Chang et al., 2009), high degree of uncertainty and hidden relationships (Khan et al., 2011; Tay and Cao, 2001). A dominant paradigm in economics and finance, the efficient market hypothesis (Fama, 1965) does not support stock market predictability. The efficient market hypothesis (EMH) associates with the ''random walk,'' a term rather loosely used in the economics and finance literature to characterize a price series in which the price change in t is unaffected by the price change in t-1 and exhibits no memory (Mandelbrot, 1971). The random-walk model conveys the idea that for unimpeded information flows, stock prices immediately reflect that information. Then, tomorrow's price change will reflect only tomorrow's news and will not reflect price changes today. But news is by definition unpredictable and, thus, resulting price changes must be unpredictable and random. As a result, prices fully reflect all known information, ''and even uninformed investors buying a diversified portfolio at the tableau of prices given by the market will obtain a rate of return as generous as that achieved by the experts'' (Malkiel, 2003, p. 59).

In spite of the compelling theoretical appeal of the EMH and the random-walk paradigm, a growing interest in the last few decades surrounds the development and application of stock market forecasting models with daily and intra daily data. This interest follows along many avenues, including long-memory models, artificial-neural-network (ANN) methods, and wavelet decomposition techniques.

Long memory in financial time series reflects the existence of fractional dynamics, i.e., persistent temporal dependencies in the data. This phenomenon, long-memory processes, exhibit hyperbolic decay in the autocovariance function, which is not absolutely summable in the time domain, and in the frequency domain by the high spectral density at low frequency. In contrast, short memory processes exhibit exponential decay of the autocovariance function, which is absolutely summable in the time domain, and constant spectral density in the frequency domain. Classical time-series models, namely ARIMA models, cannot capture the long memory phenomenon.

Long memory in the conditional mean of stock market returns has been investigated extensively (see, e.g., Gil-Alana, 2006; Henry, 2002; Aye et al., 2014; Lopez-Herrera et al., 2012; Bhardwaj and Swanson, 2006; Barkoulas and Baum (1996) Barkoulas et al., 2000 Bourbonnais and Maftei, 2012) using the autoregressive fractionally integrated moving average ARFIMA (p, d, q) model developed by Granger and Joyeux (1980) and Hosking (1981). ARFIMA models provide parsimonious accounting for long-range dependence, by the addition of a single parameter to classical ARMA models. Importantly, they allow the simultaneous modeling of short-term processes (by the combination of the p and q parameters), and long-range dependence through the d parameter, and as such the isolation of their respective effects. Finally, the ARFIMA parameters can be estimated using exact maximum likelihood, allowing the significance of the difference of d from 0 to be tested. As such, ARFIMA modelling can effectively detect the presence of long-range dependence in time series. Moreover, the estimation of d allows the quantification of the intensity of the long-range correlations within the series, as d relates to the spectral exponent β by the simple relation $\beta = 2d$. Long memory in stock market returns poses a serious challenge to the EMH as it implies significant autocorrelations between returns that, although they are distant in time, can help predict future returns. Long memory in stock market returns also means that the stock market does not immediately respond to new information, but reacts to it in a gradual manner over time

Modeling long memory in volatility has also recently attracted a great deal of attention from finance literature. GARCH models, used extensively in empirical analysis, do not account for long memory in volatility The property of long memory in the conditional variability of stock market returns has been investigated extensively using fractionally integrated (FI) GARCH models or FIGARCH (Baillie et al., 1996) and its follow-up extensions, including fractionally Integrated Asymmetric Power ARCH (FIAPARCH) models (Tse, 1998), Hyperbolic GARCH (HYGARCH) models (Davidson, 2004), and Hyperbolic APARCH (HYAPARCH) models (Dark, 2006, 2010; Schoffer, 2003). Long memory in volatility occurs when the effects of volatility shocks decay slowly. The presence of long memory in volatility creates serious problems with the application of standard GARCH models, as GARCH models are either at the I(0) integration level (or mean-reverting) or I(1) integrated level (or non-mean-reverting).

Long memory in return volatility (see, e.g., Bollerslev and Mikkelsen, 1996; Gurgul and Wojtowicz, 2006; Floros et al., 2007; Cavalcante and Assaf, 2005; Kang and Yoon, 2007; Killic, 2004; Kasman and Torun, 2007; Jefferis and Thupayagale, 2008; McMillan and Thupayagale, 2008, 2009; Lin and Fei, 2013; Kang et al., 2010; Disario et al., 2008; Sadique and Silvapulle, 2001; Conrad, 2007) implies that financial markets do not quickly forget volatility shocks. The FIGARCH model allows for fractional integration and estimates an intermediate process between the GARCH model and the Integrated GARCH (IGARCH) model. The FIGARCH model derives its short-run dynamics from the conventional GARCH parameters. In contrast to the GARCH model where shocks to the conditional variance dissipate quickly at an exponential rate, shocks in the FIGARCH domain exhibit hyperbolic decay. It also generalizes GARCH and the integrated GARCH (IGARCH) formulations, which prove unsuitable to capture this feature of the data. In fact, the GARCH model only accounts for short memory, while the IGARCH considers infinite memory, which is an unrealistic situation. By introducing a fractional differencing parameter d, the FIGARCH model allows long memory in the observed data for $0 < d < 1$ and accommodates both the GARCH ($d = 0$) and the IGARCH ($d = 1$) frameworks as special cases.

The literature on long memory in the conditional mean (Granger and Joyeux, 1980; Hosking, 1981) and conditional variance (Baillie et al, 1996) have evolved independently of each other. Long-memory phenomena, however, often appear in both the conditional mean and conditional variance simultaneously. Teyssiere (1997) introduced dual long-memory models, where the first-order conditional dependency structure employs the ARFIMA model, while the second-order conditional dependency structure employs the FIGARCH model. Teyssiere (1997) shows through Monte Carlo simulations that ignoring long memory in the conditional mean of a dual long memory process leads to significant biases in the estimation of the conditional volatility process. The joint ARFIMA-FIGARCH dual long-memory model incorporates two parameters jointly driving the long memory in returns and in volatility (see, e.g., Kang and Yoon, 2007; Kasman and Torun, 2007; Kormaz et al., 2009; Kasman et al., 2009; Tan and Khan,2010; Ural and Kucukozmen, 2011; Macheshchandra, 2012; Turkyilmaz and Balibey, 2014; Jefferis and Thupayagale, 2008 for stock market applications). In general, evidence of long memory in returns is mixed, and not uniform across stock markets. In contrast, evidence of long memory in volatility proves systematic.

The FIGARCH model suffers from some drawbacks, mainly because it is not covariance stationary and does not allow for asymmetric responses of volatility to positive and negative shocks. These issues motivated the development of new models, such as Fractionally Integrated Asymmetric Power ARCH (FIAPARCH) models (Tse, 1998), which allows for long memory and asymmetries in volatility, the Hyperbolic GARCH (HYGARCH) model (Davidson, 2004), where shock also decay hyperbolically and allow the existence of a finite variance, and its asymmetric version, the Hyperbolic APARCH (HYAPARCH) model (Dark, 2006, 2010 and Schoffer, 2003). Applications of these models to stock markets are, however, sparse. Dual memory evidence based on ARFIMA-HYGARCH models is provided, among others, by Conrad (2007), Kasman et al. (2009) and Chikki, Peguin-Feissolle, and Terraza (2013). Long memory evidence based on ARFIMA-FIAPARCH models is provided, among others, by Duppati et al. (2017) while long memory evidence based on ARFIMA-HYAPARCH models is provided by Ojeda Echeverri and Castano Velez (2014).

In the last two decades, the ability of artificial neural networks (ANNs) to forecast the stock market has received extensive investigation (see, e.g., Leung et al., 2000; Leight et al., 2002; Chen et al., 2003; Kim and Lee, 2004; Lee et al., 2007; Kara et al., 2011; Guresen et al., 2011; Moghaddam et al., 2016; Tsanga et al., 2007; Liao and Wang, 2010; Qiu et al., 2016). Artificial neural networks (ANNs) are a powerful tool in modern quantitative finance and have emerged as a powerful statistical modeling technique. Several distinguishing features of ANNs make them valuable and attractive in quantitative finance and forecasting, especially for nonlinear environments. ANN models effectively simulate and describe the dynamics of non-stationary time series due to their unique non-parametric, noise-tolerant and highly adaptive characteristics. First, ANNs have the capacity of performing complex nonlinear modeling without a priori knowledge of the underlying data generating processes and relationships. ANNs can reason, learn and generalize in an uncertain and imprecise environment. They can derive meaning from complicated and imprecise data, extract patterns and detect trends that are too complex or too hidden to be revealed by econometric models; can emulate certain performance characteristics of the biological functions of the human brain, and can learn from its environment and adapt in an interactive manner similar to biological counterparts (Ham and Kostanic, 2001). Second, ANNs are universal function approximations. They can flexibly map nonlinear functions and can approximate any continuous function with arbitrary desired accuracy (Hornik,1993; Hornik et al., 1987). Third, ANNs can learn and generalize from experience. After learning the data presented, ANNs can often correctly infer the unseen part of a population, even if the sample data contain highly noisy information. Neural networks can capture the underlying pattern or autocorrelation structure within a time series, even when the underlying law governing the system is unknown or too complex to describe. The neural network is trained from a mass of historical data and tries to discover hidden dependencies to use for prediction into the future.

Ever since McCulloch and Pitts (1943) pioneering work, an array of artificial neural network models, such as back-propagation neural network (Rumelhart, Hinton and Williams, 1986), radial basis function neural network (Lowe and Broomhead, 1988), wavelet neural network (Zhang and Benveniste, 1992; Daubechies, 1990), Kohonen neural network (Kohonen, 1990), and Hopfield neural network (Hopfield and Tank , 1985), have been proposed and investigated. Among all these methods, wavelet neural network (WNN) has shown its advantages in regression accuracy and fault-tolerant ability due to the adoption of a wavelet transform. In contrast to the traditional neural network, which employs common sigmoid activation functions, wavelet neural network (WNN) models employ nonlinear wavelet basis functions (called wavelets), which are localized in both the time and frequency domains (Gencay et al., 2002). In contrast to the standard time- series econometric models, which consider at most two (ad hoc) time scales, the short and long run, and rely on model parameters, the wavelet approach uses a model-free methodology. Unlike the Fourier transform that can only provide frequency information, wavelets can keep track of time and frequency information. When considering a Fourier transform of a signal, it is impossible to tell when a particular event took place. The Fourier transform loses information when it transforms the data to the frequency domain. Wavelet analysis, instead, allows the use of long time intervals, where we want more precise low-frequency information, and shorter time intervals, where we want high-frequency information (Reboredo and Rivera-Castro, 2014; Gülerce and Ünal, 2016). An additional appealing feature of wavelet modeling is that, unlike Fourier transform and standard econometric time series models, the wavelet transform does not need stationarity (Burrus et al., 1998). Wavelets provide a powerful tool for the analysis and synthesis of data from long-memory processes. At high scales, the wavelet focuses on short-run phenomena. At low scales, the wavelet identifies long-run periodic behavior. By moving from low to high scales, the wavelet zooms in on a process behavior at a point in time, identifying singularities, jumps, and cusps. Alternatively, the wavelet can zoom out to reveal the long, smooth features of a series.

The curse of dimensionality, i.e., the problem where the number of hidden units rises exponentially as the number of input dimensions increases, is the main problem of WNN. To solve the dimensionality problem, Wang et al, (2000) proposed a WNN incorporating a local linear model (LLWNN), where local linear model replaces the connection weights between the hidden layer and output layer of conventional WNN.

In this paper, we propose an innovative hybrid forecasting model that integrates the advantages of the wavelet decomposition (i.e., the MODWT with Daubechies least asymmetric wavelet filter) and the LLWNN model. The model develops through a two-phase approach. In phase one, use of wavelet decomposition improves the forecasting accuracy of the LLWNN, resulting in the Wavelet Local Linear Wavelet Neural Network (WLLWNN) model. We apply the BP and PSO learning algorithms to optimize the WLLWNN structure and avoid overfitting. In phase two, the residuals from an ARFIMA model become inputs to the WLLWNN model. We evaluate the hybrid ARFIMA-WLLWNN model using daily closing prices for the Dow Jones Industrial Average (DJIA) index over the period from January 1, 2010 to February 11, 2020.

The rest of the paper is laid out as follows. Section 2 provides the theoretical background and discusses the three main problems in the design of WLLWNN -- the wavelet decomposition, the learning algorithms for neural network optimization, and the architecture of the WLLWNN model. These problems help to determine an optimal WLLWNN architecture, to arrange the windows of wavelets, and to find the proper orthogonal and non-orthogonal wavelet basis. Section 3 outlines the hybrid ARFIMA-WLLWNN model. Section 4 describes the data and the main results. Section 5 evaluates the predictive performance of the hybrid ARFIMA-WLLWNN model against the LLWNN, WLLWNN, ARFIMA-LLWNN and the ARFIMA-HYAPARCH models. The experimental results indicate that the PSO-optimized hybrid ARFIMA-WLLWNN provides more accurate out-of-sample forecasts over validation horizons of one, five and twenty-two days using three evaluation criteria, namely MAE, MSE, and RMSE. Finally, concluding remarks appear in the last section.

2. Econometric Framework and Methodology

2.1 Wavelet theory

The Fourier theory models a signal as a sum of sines and cosines. The Fourier transform, however, only provides frequency resolution and no time resolution. That is, one cannot identify the timing of the signal. Wavelet theory overcomes this problem.

The wavelet technique allows the decomposition of a signal into several components in multiple scales. In wavelets, low and high pass filters are applied, extracting the low (approximations) and high (details) frequencies of the signal for the level of decomposition chosen, whose sum equals the original series. These constitutive components possess improved statistical properties compared to the original series and, consequently, possess improved forecasting accuracy. Applied to artificial neural network (ANN) models for time-series analysis and forecasting, wavelets decompose the original time-series signal into smoother components and then apply the most appropriate ANN prediction model for each component individually. In this context, the low frequency components contain the general trends of the series and can explain the long-term behavior of the series, while the high frequency components can explain the short-term behavior of the series. That is, wavelet analysis decomposes a given time series on a scale-by-scale basis. Using dilation and translation operations, this technique allows a flexible time-frequency resolution, and can define local features of a given function in a parsimonious way.

Wavelets are orthonormal bases attained through dyadically dilating and translating a pair of specially constructed functions denotes by φ and ψ, which are named father and mother wavelets, respectively, given by:

$$\int \varphi(t)dt = 1, \text{ and} \tag{1}$$

$$\int \psi(t)dt = 0. \tag{2}$$

The smooth low-frequency part of the time series are defined by the father wavelet while the detail high-frequency components are defined by the mother wavelet. The obtained wavelet basis is:

$$\varphi_{j,k}(t) = 2^{j/2}\varphi\left(2^j t - k\right) \text{ and} \tag{3}$$

$$\psi_{j,k}(t) = 2^{j/2}\psi\left(2^j t - k\right), \tag{4}$$

where $j = 1, \ldots, J$ indexes the scale and $k = 1, \ldots, 2^j$ indexes the translation. The parameter j is adopted as the dilation parameter of the wavelet's function. This parameter j adjusts the support of $\psi_{j,k}(t)$ to locally detect the features of high or low frequencies. The parameter k relocates the wavelets in the temporal scale.

The wavelet expansion includes the special localization property, where the coefficient of $\psi_{j,k}(t)$ reveals information content of the function at approximate location $k2^{-j}$and frequency 2^{-j}. We can extend any series over the wavelet basis as a linear combination at arbitrary level $J_0 \in N$ through different scales of the type:

$$X(t) = \sum_{k} s_{J_0,k}\phi_{J_0,k}(t) + \sum_{j \geq J}\sum_{k} d_{j,k}\psi_{j,k}(t), \tag{5}$$

where $\varphi_{J_0,k}$ is a scaling function with the corresponding coarse scale coefficients $s_{J_0,k}$ and $d_{j,k}$ are the detail coefficients given respectively by $s_{J_0,k} = \int X(t)\varphi_{J_0,k}(t)dt$ and $d_{j,k} = \int X(t)\psi_{j,k}(t)dt$. These coefficients measure the contribution of the corresponding wavelet to the function. Equation (5) denotes the decomposition of $X(t)$ into orthogonal components at different resolutions and constitutes the wavelet multiresolution analysis (MRA).

The recursive MRA scheme,[1] which is implemented by a two-channel filter bank (i.e., a high-pass wavelet filter $\{\boldsymbol{h_l}, \quad \boldsymbol{l = 0, \dots, L-1}\}$ and its associated low pass scaling filter $\{\boldsymbol{g_l}, \quad \boldsymbol{l = 0, \dots, L-1}\}$ satisfying the quadrature mirror relationship given by $\boldsymbol{g_l} = (-\boldsymbol{1})^{\boldsymbol{l+1}}\boldsymbol{h_{L-1-l}}$ for $\boldsymbol{l = 0, \dots, L-1}$, where $\boldsymbol{L} \in \boldsymbol{N}$ is the length of the filter. Daubechies (1992) constructed a class of wavelet functions that form an orthonormal basis of piecewise constant functions of length one. The Daubechies wavelet includes many desirable properties. For our purposes, it possesses the smallest support for a given number of vanishing moments (the Daubechies compactly supported wavelet filters) and distinguishes between two choices; the extremal phase filters $\boldsymbol{D}(\boldsymbol{L})$ and the least asymmetric filters $\boldsymbol{La}(\boldsymbol{L})$.

2.2 Learning Algorithms for neural network optimization

Recently, researchers widely apply the artificial neural networks (ANNs) methodology, among computational intelligence systems, for estimating functions and forecasting. The main advantage of ANNs over other nonlinear models is that they are universal approximations that can approximate a large class of functions with a high degree of accuracy (Chen et al. 2003 and Zhang and Qi, 2005). Their performance relates to the parallel processing of the information from the data. In addition, no prior assumption is required in the building process. Instead, the network model largely depends on the features of the data.

[1] In practical applications, we invariably deal with sequences of values indexed by integers rather than functions defined over the entire real axis using short sequences of values referred to as wavelet filters. Hence, the wavelet analysis measured through a filtering perspective is then well suited to time series analysis. Mallat's Multiresolution Analysis (MRA) is considered as a robust theoretical framework for critically sampled wavelet transformation (for more details, see Mallat 1989).

The training process for ANNs is generally complicated due to the high dimensionality. Until today, several researchers choose to adopt Back Propagation (BP) algorithms in the training of the ANNs. BP measures the output error, computing the gradient of this error and adjusting the weights of the network in the direction of descending gradient. Thus, a major difficulty of this algorithm is that it searches for optimal weights, which strongly depends on initial weights. More precisely, if these optimal weights are located close a local minimum; the algorithm becomes stuck at a sub-optimal solution. Therefore, the conventional gradient search method can easily converge at local optima.

Neural network researchers recommend numerous solutions to overcome the trapping by a local minimum and the slow convergence rate. To address this issue, we propose the Particle Swarm Optimization (PSO) as an evolutionary algorithm that performs well in various optimization problems. We use the BP and PSO algorithms to optimize the WLLWNN model.

2.3 The Back Propagation (BP) algorithm

The BP algorithm begins randomly initialized parameters. Then the algorithm measures the error between the output and real values and finally adjusts the weights in the direction of the descendent gradient. The learning rate controls the speed of the training process. For high rates, the ANN model will learn more quickly, but the learning process will never converge if this rate is too high. In contrast, for low rates, the ANN model may converge to a local minimum instead of the global minimum.

The equations of the BP algorithm appear in Burton and Harley (1994) and we briefly describe them below. The objective function to minimize is given as follows:

$$E = \frac{1}{2}\big[y_t - \omega_{1,0}\phi_1(x) - \omega_{1,1}p_1\phi_1(x) - \cdots \omega_{2,0}\phi_2(x)\omega_{2,1}p_2\phi_2(x) -$$

$$\omega_{l,0}\phi_l(x)\omega_{l,1}p_1\phi_l(x) - \cdots \omega_{l,p}p_p\phi_l(x)\big], \qquad (6)$$

where y_t is the desired value, $\phi(x)$ is the active wavelet functions, $\omega_{1,0}$ represents the connection weight, p is the number of inputs $(i = 1,2,\ldots.p)$, and l is the number of the hidden units $(j = 1,2,\ldots.l)$. The model updates the weight from the i^{th} to the $(i+1)^{th}$iteration (i.e., from ω_t to ω_{t+1}) as follows:

$$\omega_{t+1} = \omega_t + \Delta\omega_t = \omega_t + \left(r\frac{\partial E_t}{\partial \omega_t}\right), \qquad (7)$$

where r is the learning rate adopted in the WLLWNN model. The following equations describe $\frac{\partial E}{\partial \omega}$ for all weights:

$$\frac{\partial E}{\partial \omega_{i,0}} = \omega_{i,0} + r.e.\left(\frac{1}{2}\right).\left(x_1^2 + x_2^2 + \cdots + x_p^2\right).exp(-((x_1 - c_i)^2 +$$

$$(x_2 - c_i)^2 + \cdots + (x_p - c_i)^2)), \text{ and} \qquad (8)$$

for $\forall j \neq 0$;

$$\frac{\partial E}{\partial \omega_{i,j}} = \omega_{i,j} + r.e.\left(\frac{1}{2}\right).(x_1^2 + x_2^2 + \cdots + x_n^2).exp(-((x_1 - c_i)^2 +$$

$$(x_2 - c_i)^2 + \cdots + (x_n - c_i)^2)).x_j. \qquad (9)$$

That is,

$$\frac{\partial E}{\partial \omega_{1,0}} = \omega_{1,0} + r.e.\left(\frac{1}{2}\right).\left(x_1^2 + x_2^2 + \cdots + x_p^2\right).exp(-((x_1 - c_1)^2 +$$

$$(x_2 - c_1)^2 + \cdots + (x_p - c_1)^2)), \qquad (10)$$

$$\frac{\partial E}{\partial \omega_{1,2}} = \omega_{1,2} + r.e.\left(\frac{1}{2}\right).\left(x_1^2 + x_2^2 + \cdots + x_p^2\right).exp(-((x_1 - c_1)^2 +$$

$$(x_2 - c_1)^2 + \cdots + (x_p - c_1)^2)).x_2, \qquad (11)$$

$$\frac{\partial E}{\partial \omega_{2,0}} = \omega_{2,0} + r.e.\left(\frac{1}{2}\right).\left(x_1^2 + x_2^2 + \cdots + x_p^2\right).exp(-((x_1 - c_2)^2 +$$

$$(x_2 - c_2)^2 + \cdots + (x_p - c_2)^2)), \qquad (12)$$

$$\frac{\partial E}{\partial \omega_{2,1}} = \omega_{2.1} + r.e.\left(\frac{1}{2}\right).\left(x_1^2 + x_2^2 + \cdots + x_p^2\right).exp(-((x_1 - c_2)^2 +$$

$$(x_2 - c_2)^2 + \cdots + (x_p - c_2)^2)).x_1, \quad (13)$$

where e is the error between output values $\hat{y}$ and real values y $(e = \hat{y} - y)$ and the other weights are also updated in the same way.

2.4 The Particle Swarm Optimization (PSO) algorithm

Kennedy and Eberhart (1995) developed the PSO as an optimization technique. Compared to other learning algorithms, the PSO clearly exhibited its efficiency. The PSO algorithm is established through simulation of bird flocking in two-dimensional space. Each agent's position is denoted by a point in the XY plain and the velocity is represented by vx and vy. The position and velocity information determines the agent position's adjustment. The Bird flocking optimizes the objective function. Each agent knows its best value so far $(pbest)$ and its XY position. In addition, each agent knows the group's best value so far $(gbest)$ among $(pbest)$. In sum, each agent tries to adjust its position using the following information.

(a) The distance between current position and $pbest$.

(b) The distance between the current position and $gbest$.

We can update each agent's velocity by the following equation:

$$v_i^{p+1} = wv_i^p + c_1 * rand_1 * \left(pbes\ _1 - s_i^p\right) + c_2 * rand_2 * (gbest - s_i^p), \quad (14)$$

where v_i^p is the velocity of agent i at iteration p, w is the weight function, c_1 and c_2 are weighting factors i.e., acceleration coefficients controlling the influence of a particle's historical best location and the swarm's historical best location on its next velocity, respectively. s_i^p is the current position of agent i at iteration p, $pbes\ _i$ is the $pbest$ of agent i and $gbest$ is the $gbest$ of the group, and $rand_1$ and $rand_2$ are two separately generated random numbers in the uniform range [0,1].

The first part of equation (14), i.e., $c_1 * rand_1 * (pbest_1 - s_i^p)$, is called the cognitive component, i.e., represents the cognitive perception of the particle, while the second part of equation (14), i.e., $c_2 * rand_2 * (gbest - s_i^p)$, is called the social component, i.e., emulates the social interaction among particles. In other words,, the cognitive component is for self-cognition and the social component is for social learning.

The velocity, which progressively approaches $pbest$ and $gbest$, can be computed using the above equation. The actual position, which characterizes the searching point in the solution space, can be updated using the following equation:

$$s_i^{p+1} = s^p + v_i^{p+1} \tag{15}$$

The first term of equation (14) denote the previous velocity of the agent. The velocity of the agent is updated through the second and third terms.

The general steps, which describe the optimization of the LLWNN using the PSO algorithm, can be summarized as follows.

Step 1. The initial condition is generated for each agent:

The initial searching points of location (s_i^0) and velocity (v_i^0) of each agent are habitually generated randomly within the allowable range. Note that the dimension of search space contains all the parameters of the LLWNN. The current searching point is set to$pbest$ for each agent. The best-evaluated value of $pbest$ is set to $gbest$ and the agent number with the best value is stored.

Step 2. The searching points are evaluated for each agent:

The value of the objective function is calculated for each agent. If this calculated value improves in comparison with the current $pbest$ of the agent, the $pbest$ value is replaced by the current value. If the best value of $pbest$ is better than the current$gbest$, $gbest$ is replaced by the best value and the agent number corresponding to the best value is stored.

Step 3. Modification of each searching point:

Using equations (14) and (15), the actual searching point of each agent is updated.

Step 4. Verification of the exit condition:

If the number of the current iteration reaches the number of the predetermined maximum iteration, then exit. If else; go to step 2.

Contrary to the BP, the PSO algorithm avoids the convergence to a local minimum, since it does not depend on gradient information (Abbass et al. 2001). The PSO produces the best set of weights (particle position), where numerous particles are moving to get the best solution and the total number of weights characterize the dimension of the search space. The optimization finishes when the personal best solution of each particle and the global best of the entire swarm are attended.

2.5 The Wavelet Local Linear Wavelet Neural Network (WLLWNN) model (Phase One)

In this section, we outline the Wavelet Local Linear Wavelet Neural Network (WLLWNN) model, a novel neural network-based wavelet decomposition. The model involves two-steps. First, the historical stock market data are decomposed into wavelet domain constitutive sub-series using Wavelet transform and, second, the decomposed time, are shaped through the Local Linear Wavelet Neural Network (LLWNN) model to produce the set of input variables and form the WLLWNN forecasting model Since stock market data exhibit a richer structure and signal-processing features, the Wavelet Transform is an appropriate tools to bring out the hidden patterns in the series. Equation (5) represents the decomposition of $X(t)$ into orthogonal components at different resolutions and constitutes the so-called wavelet multiresolution analysis (MRA). The flow-chart structure of the WLLWNN model appears in Figure.1. According to wavelet transformation theory, wavelets (used as an activation function) in the following form are a family of functions, generated from one single function $\psi(x)$ by the operation of dilation and translation $\psi(x)$.

$$\psi(x) = \left\{\psi_i = |a_i|^{-1/2}\psi\left(\frac{x-b_i}{a_i}\right); \quad a_i, b_i \in R^n, i \in z\right\}, \tag{16}$$

$$x = (x_1, x_2, \dots x_n),$$

$$a_i = (a_{i1}, a_{i2}, \dots a_{in}), \text{ and}$$

$$b_i = (b_{i1}, b_{i2}, \dots b_{in}).$$

$\psi(x)$, which is localized in both time and scale space, is called a mother wavelet and the parameters a_i and b_i are the scale and translation parameters, respectively.

Instead of the straightforward weight w_i (piecewise constant model), a linear model $v_i = w_{i0} + w_{i1}x_1 + \cdots + w_{in}x_n$ is introduced. The activities of the linear models v_i $(i = 1,2, \dots n)$ are determined by the associated locally active wavelet functions $\psi_i(x)$ $(i = 1,2, \dots n)$. Thus, v_i is only locally significant. Non-linear wavelet basis functions (named wavelets) are localized in both time and scale space. Here $m = n$ and output (Y) of the proposed model is calculated as follows:

$$Y = \sum_{i=1}^{M}(w_{i0} + w_{i1}x_1 + \dots . . w_{in}x_n)\psi_i(x). \tag{17}$$

The mother wavelet is

$$\psi(x) = \frac{-x^2}{2} e^{\frac{-x^2}{\sigma^2}} \text{ and} \tag{18}$$

$$\psi(x) = e^{-\left(\frac{x-c}{\sigma}\right)^2}, \tag{19}$$

where

$$x = \sqrt{d_1^2 + d_2^2 + \dots \dots d_n^2}. \tag{20}$$

2.6 Dual long-memory models

Long-memory models describe strong correlations or dependences across time-series data. This phenomenon is often referred to as "long-memory" or "long-range dependence". It refers to persistent correlation between distant observations in a time series. Dual long-memory models refer to long memory in the first and second moments of a time series. Long-memory models complement the well-known and widely applied stationary and invertible autoregressive and moving average (ARMA) models, whose autocovariances not only sum up but also decay exponentially. Such models are often referred to as "short-memory" models, because negligible correlation exists across distant time intervals. These models, however, often combine with the most basic long-memory models since together they offer the ability to describe both short- and long-memory features in many time series. This holds for the models discussed in this section. In this sub-section, we summarize four dual long-memory models that have received attention in the literature and emphasize some of their salient features: the ARFIMA-FIGARCH model, the ARFIMA-HYGARCH model, the ARFIMA-FIAPRCH model, and the ARFIMA-HYAPARCH model. Long-memory intervenes in the mean equation through the parameter d_m and in the variance equation through the parameter d_v.

2.6.1 The ARFIMA model

The ARFIMA(p, d, q) model was developed by Granger and Joyeux (1980) and Hosking (1981). The model captures the fractionally integrated process I(d) in the conditional mean. The ARFIMA(p, d, q) for the time series r_t is defined as follows:

$$\boldsymbol{\theta}\ (\boldsymbol{L})\ (\mathbf{1}-\boldsymbol{L})^{\boldsymbol{d_m}}\ (\boldsymbol{r_t}-\boldsymbol{\mu}) \quad = \boldsymbol{\phi}\ (\boldsymbol{L})\ \boldsymbol{\varepsilon_t},\ \boldsymbol{\varepsilon_t} \mid \boldsymbol{\Psi_{t-1}} \sim \boldsymbol{D}\ (\mathbf{0},\ \boldsymbol{h_t}), \tag{21}$$

where r_t is the stock market return series, μ is the mean of the series, d_m is a fractional integration parameter of r_t , $\theta(L) = 1 - \theta_1 L - \cdots - \theta_p L^p$ and $\phi(L) = 1 + \phi_1 L + \cdots + \phi_q L^q$ are the AR and MA polynomials in the lag operator, respectively, of orders p and q (with all roots lying outside the unit circle), L denotes the lag operator, ε_t is a white noise disturbance, and $(1 - L)^{d_m}$ stands for the fractional integration lag operator. The AR and MA polynomials constitute the short-memory parameters and affect only the short-run dynamics of the process, whilst the fractional integration parameter d_m detects the long-memory behavior of the process. Following Hosking (1981), various can emerge. If $-0.5 < d_m < 0$, then the process is anti-persistent, i.e., it exhibits negative dependence. memory. If $0 < d_m < 0.5$, then the process is a stationary long-memory process and possesses shocks that disappear hyperbolically. If $0.5 \leq d_m < 1$, then the process is non-stationary, but mean-reverting, with finite impulse response weights. When $d_m = 0$, the process reduces to the standard ARMA and when $d_m = 1$, the process becomes ARIMA and implies infinite persistence of the mean to a shock in the returns. Ψ_{t-1} stands for the information set available at time $t-1$ whereas ε_t follows the conditional distribution D.

2.6.2 The FIGARCH model

The FIGARCH$(\boldsymbol{P}, \boldsymbol{d}, \boldsymbol{Q})$-model, developed by Baillie et al. (1996), models the fractionally integrated process I(d) in the conditional variance of a GARCH (P, Q) process. Formally, the FIGARCH $(\boldsymbol{P}, \boldsymbol{d}, \boldsymbol{Q})$ for the time series $\boldsymbol{h_t}$ is defined as follows:

$$\boldsymbol{h_t = \omega + \left\{1 - \left(1 - \beta(L)\right)^{-1} \varpi(L)(1 - L)^{d_v}\right\} \varepsilon_t^2,} \tag{22}$$

where h_t is the conditional variance of r_t, ω is the mean of the process, d_v is the fractional integration parameter of h_t, and $\beta(L)$ and $\varpi(L)$ are lag polynomials of orders P and Q, respectively, and ε_t is a mean-zero serially uncorrelated process. The FIGARCH model nests both the GARCH model (Bollerslev, 1986) for $d_v = 0$, and the IGARCH model (Engle and Bollerslev, 1986) for $d_v = 1$. In the first case, shocks to the conditional variance decay at an exponential rate while in the second, shocks persist forever and, thus, affect forecasts at all horizons. If $0 < d_v < 1$, then the effect of a shock decreases at a hyperbolic rate The lag polynomials $\beta(L)$ and $\varpi(L)$ account for the short-term dynamics of volatility, while the fractional integration parameter d_v captures the long- term dynamics of volatility. Note that FIGARCH-type processes, although strictly stationary and ergodic for

$0 < d_v < 1$, are not covariance stationary. Furthermore, the interpretation of the long-memory parameter d_v is difficult in the FIGARCH set up. Davidson (2004) shows that long memory increases when d_v approaches zero. This contrasts with the conventional interpretation where long memory increases when d_v increases See Davidson (2004) for additional details.

2.6.3 The HYGARCH model

The hyperbolic GARCH (HYGARCH) was introduced by Davidson (2004) and shows that the HYGARCH model generalizes the FIGARCH model to cope with the deficiencies inherent to the FIGARCH model. The model is covariance stationary, similar to the GARCH model, and exhibits hyperbolic rate of decay similar to the FIGARCH model. The HYGARCH model has the following representation:

$$h_t = \omega + \left\{1 - \left(1 - \beta(L)\right)^{-1} \varpi(L)[1 + \alpha((1 - L)^{d_v} - 1)]\right\} \varepsilon_t^2, \qquad (23)$$

where α is the amplitude parameter. The HYGARCH model nests the FIGARCH process for $\alpha = 1$ and the stable GARCH process for $\alpha = 0$. The process is stationary if $< \alpha < 1$ and nonstionary if $\alpha > 1$. Long memory intervenes in equation (23) through the parameter d_v. Davidson (2004) notes that for $\alpha = 0$, d_v is unidentified, which poses a well-known problem in constructing hypothesis tests. When $d_v = 1$, the parameter α reduces to an autoregressive root reproducing geometric memory and, hence, the model becomes either a stable GARCH model for $\alpha < 1$ or IGARCH for $\alpha = 1$. Thus, testing the restriction $d_v = 1$ allows discrimination between geometric and hyperbolic memory dynamics.

2.6.4 The FIAPARCH model

The. FIGARCH and HYGARCH models successfully deal with volatility clustering and long memory in stock market returns. They fail to consider, however, asymmetry in returns volatility. The "leverage effect" (Black,1976) is an important stylized fact of stock return volatility and corresponds to negative correlations between past returns and future volatility. The FIAPARCH model, developed by Tse (1998) as an extension of the asymmetric power ARCH model (Ding, Granger, and Engle, 1993), models asymmetric responses of volatility to positive and negative shocks along with volatility persistence behavior. The FIAPARCH model expresses the conditional variance as a power transformation of the standard deviation as follows:

$$\boldsymbol{h_t^{\frac{\delta}{2}} = \omega + \left\{1 - \left(1 - \beta(L)\right)^{-1} \varpi(L)(1 - L)^{d_v}\right\}(|\varepsilon_t| - \gamma\varepsilon_t)^{\delta},} \qquad (24)$$

where $-1 < \gamma < 1$ and $\delta > 0$. To accommodate asymmetry in long memory of the conditional variance, the term ε_t^2 of the FIGARCH model in equation (22) is replaced by the term $(|\varepsilon_t| - \gamma\varepsilon_t)^{\delta}$ in equation (24). The parameter δ is the power term that plays the role of a Box-Cox transformation of the conditional standard deviation $h_t^{\frac{1}{2}}$, while γ denotes the asymmetry parameter accounting for the leverage effect. When $\gamma > 0$, negative shocks give rise to higher volatility than positive shocks. The reverse applies if $\gamma < 0$. The magnitude of the shocks is captured by the term $(|\varepsilon_t| - \gamma\varepsilon_t)$.

The use of the power term δ endeavors to avert the Gaussianity assumption. In fact, if the data are assumed to follow a conditional normal density, then the first two moments, (i.e., the mean and the variance) completely typify the distribution of returns. This justifies the common use of a squared term $\delta = 2$ and, hence, a measure of the variance to characterize the volatility structure. Since asymmetry and heavy tails both characterize financial asset returns, however, the hypothesis of normality appears unrealistic and higher-order moments such as skewness and kurtosis are required to specify the true underlying distribution. As such, considering the variance as a measure of the volatility process (i.e., setting $\delta = 2$) can adversely affect the estimation results and the forecasting performance of the model. To deal with this issue, Ding, Granger, and Engle (1993) suggest estimating the volatility measure in the form of a power transformation through allowing an optimal power term δ to be endogenized and freely determined by the data. Note that the FIAPARCH process reduces to the FIGARCH process when $\gamma = 0$ and $\delta = 2$.

2.6.5 The HYAPARCH model

The FIGARCH and FIAPARCH models of Baillie et al (1996) and Tse (1998) are not covariance stationary and, thus, do not permit statements about the autocovariance function due to infinite conditional second moments (Niguez, 2002). The HYGARCH model of Davidson (2004) is covariance stationary, but both the FIGARCH and HYGARCH models fail to allow for asymmetries. The hyperbolic APARCH (HYAPARCH) model proposed by Schoffer (2003) and Dark (2006) addresses some of the limitations of the previous long-memory ARCH models, as it is covariance stationary, accounts for long memory and volatility asymmetries, and releases the unit-amplitude restriction to account for both volatility persistence and covariance stationarity

The HYAPARCH model reproduces the main characteristics of returns of financial time series such as volatility clustering, leptokurtosis, asymmetry, and long memory and estimates the power of the heteroskedastic equation from the data. For this reason, the HYAPARCH model is generally preferred over the previously discussed models.

The HYAPARCH model has the following representation:

$$\boldsymbol{h_t^{\frac{\delta}{2}} = \omega + \left\{1 - \left(1 - \beta(L)\right)^{-1} \varpi(L)\left[1 + \alpha\left((1 - L)^{d_v} - 1\right)\right](1 - L)^{d_v}\right\} (|\varepsilon_t| - \gamma\varepsilon_t)^{\delta}} \qquad (25)$$

where $\delta > 0$ is the power term in the volatility process, $-1 < \boldsymbol{\gamma} < 1$ is the asymmetry parameter, and $\varpi(L)$ and $\beta(L)$ are the ARCH and GARCH polynomials, respectively. Under the condition α < 1 and further restrictions on the remaining parameters of the model, the resulting stochastic process is weakly stationary (Schoffer, 2003).

The HYAPARCH model reduces to the HYGARCH model for $\boldsymbol{\gamma = 0}$ and $\boldsymbol{\delta = 2}$ and to the FIAPARCH model for $\boldsymbol{\alpha = 1}$. In sum, the HYAPARCH process couples the flexibility of a varying exponent with the asymmetry coefficient, thus, capturing asymmetric volatility structure and letting the data determine the power of the heteroscedastic equation. Moreover, it is covariance stationary and enhances the long-memory aspect of the conditional volatility via the fractional differencing parameter $\boldsymbol{d_v}$.

3. The Hybrid ARFIMA-WLLWNN Model (Phase Two)

Our hybrid methodology combines the ARFIMA model and the proposed WLLWNN model. The ARFIMA model offers greater flexibility in modeling simultaneous short- and long-term dependence of a time series. In addition, the choice of WLLWNN in our hybrid model is motivated by the wavelet decomposition and its local linear modeling ability. Consider time series to include two components. The first component is a parametric form with unknown parameters, where a parametric method seems appropriate for such processes. The second component relates to the residuals; which usually presents no specific process. Hence, it is difficult to determine the appropriate model to deal with this part of the time series. For this reason, a non-parametric model seems appropriate for modelling the residuals. This choice is motivated by the fact that non-parametric models can reduce modelling bias by imposing no specific model structure, other than certain smoothness assumptions and, thus, non-parametric models are particularly useful when we little information exists or when we want flexibility in the underling model. The flow-chart structure of the Hybrid ARFIMA-WLLWNN model appears in Figure.2.

Our methodology consists of two steps. The first step models the conditional mean of the DJIA returns using an ARFIMA model. Residuals are important, however, in forecasting time series, since they may contain some information that improves forecasting performance. Thus, the second step treats the residuals from the first step as a novel wavelet local linear wavelet neural network (WLLWNN) model.

Hence, the return series can be written as:

$$r_t = \mu_t + \varepsilon_t \qquad (26)$$

where μ_t denotes the conditional mean of the time series, and ε_t is the residual series. The first step uses the ARFIMA model to reproduce the conditional mean (equation 21).

The second step uses the residuals from the parametric model as a proxy for the corresponding volatility and models them using the WLLWNN model.

Let ε_t denote the residuals at time t from the ARFIMA model, then

$$\varepsilon_t = r_t - \hat{\mu}_t, \qquad (27)$$

were $\hat{\mu}_t$ is the forecast value from equation (21). Thus, the first stage generates the forecast values and the residuals of the semi-parametric modelling..

The second stage models the residuals using the WLLWNN with $n$input nodes. The WLLWNN for the residuals is as follows:

$$\varepsilon_t = f(\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-n}), \qquad (28)$$

where each ε_{t-i} is decomposed using the Wavelet Transform (equation 5) and f is a non-linear, non-parametric function determined by the neural network with the reference to the current state of the data during the training of the neural network. The output layer of the network (equation 17) gives the forecasting results.

$$\hat{r}_t = \widehat{\mu}_t + \widehat{\varepsilon}_t. \qquad (29)$$

Hence, this global prediction represents the result of forecasting both the conditional mean and the conditional variance of the time series.

4. Data and Empirical Results

4.1 Preliminary analysis of the data

The data consist of daily observations on the closing prices P_t of the Dow Jones Industrial Average (DJIA) from 01/01/2010 to 02/11/2020. After excluding non-trading days, the series includes 2544 observations. The data come from the Federal Reserve Economic Data (FRED) of the Federal Reserve Bank of St. Louis (http://research.stlouisfed.org/fred2/). We obtain daily returns by logarithmic differences, i.e., $r_t = \Delta Log P_t$ for t = 1,... N, where r_t is the return for day t, P_t is the closing price of the index for the same day, and N is the sample size. Figure 3 plots the returns on the DJIA index. Visual inspection of the returns suggests that while the mean of the returns is almost zero, certain periods exist that show higher volatility and are riskier than other periods. In particular, while the sample period is untouched by financial crises, one can connect the volatility to changes in the Federal Reserve's monetary policy. Volatility clustering of the returns can easily be seen in Figure 3. Large price changes (i.e., returns with large absolute values) tend to be followed by large price changes, and periods of tranquility alternate with periods of high volatility. This indicates the presence of ARCH effects in the series.

Table 1 provides summary statistics for the daily DJIA returns from 01/01/2010 to 02/11/2020. The series exhibits significant deviations from the normal distribution, as indicated by the kurtosis and skewness statistics. The distribution of returns is negatively skewed, which implies that large negative returns tend to occur more often than large positive returns (Franses and van Dijk, 2000). This reflects the fact that the downturns of the stock market are much steeper than the recoveries, indicating that investors tend to react more strongly to negative news than to positive news. The distribution of the returns is leptokurtic (fat-tailed) relative to the normal distribution. In other words, the shape of the return distribution is more peaked than the normal, implying that small changes are less frequent than in a normal distribution, and extreme events (large price movements) are more likely to occur. The 'fat-tail' problem has important financial implications, especially because it leads to a gross underestimation of risk, since the probability of observing extreme values is higher for fat-tail distributions compared to normal distributions.

Test of normality, autocorrelation, and unit root are provided in Table 2. The Jarque-Bera test rejects, as expected, the hypothesis of normality. The Ljung-Box statistics with up to 20 lags provide evidence of a positive and significant autocorrelation, which does not support the weak form of the EMH. Applied to the absolute returns and the squared returns with up to 20 lags, the Ljung-Box statistics are 1637.9 and 1202.3, respectively, which is highly significant. This, in turn, is evidence of volatility clustering in the returns. Although not reported, the autocorrelations in the absolute returns are generally higher than the autocorrelation in the squared returns. This illustrates what has become known as the 'Taylor property' (Taylor, 1986). That is, when calculating the autocorrelations for the series $|y_t|\delta|$ for various values of δ, one almost invariably finds that the autocorrelations are largest for $\delta = 1$.

The augmented Dickey-Fuller (ADF) and Phillips-Perron (PP) tests reject the hypothesis that the returns are I(1) and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test fails to reject the hypothesis of I(0). We report unit-root and stationarity results for the "intercept only" case. The results for the "no intercept and no trend case" and "intercept and trend case" do not qualitatively differ. The tests to discriminate between I(0) and I(1) behavior exhibit low power under certain types of alternatives such as non-linearities_and long memory.

Tables 3 and 4 test the long-memory property of the returns and the absolute returns. In general, most of the volatility estimation models in the financial markets depend on two proxy variables: (1) the squared returns and (2) the absolute returns. We choose to model volatility based on the absolute returns, which proves more robust against nonnormality (Davidian and Carroll, 1987) and produce better volatility forecasts relative to squared returns (Ding, Granger, and Engle, 1993). Theoretically, Forsberg & Ghysels (2007) show that absolute returns are more persistent and better in predicting future volatility than squared returns. Table 3 provides preliminary evidence of long memory in the conditional variance of the returns, as proxied by the absolute returns of the DJIA series We use the GPH (Geweke and Porter-Hudak, 1983) and local Whittle (LW) (Robinson, 1995) semi-parametric techniques and the parametric ARFIMA (1, d, 0) approach. For the GPH and LW tests, we need to choose a bandwidth m, balancing a high variance caused by staying too close to the origin and using too little information and a bias induced by the contamination of the estimation through the short-memory component of the process. As data-driven bandwidth selection methods do not work well in practice (Andrews and Guggenberger , 2003), we apply four different bandwidths $m = N^{0.5}$, $m = N^{0.6}$, $m = N^{07}$, and $m = N^{0.8}$, where N is the sample size. The results of the GPH and LW tests do not prove sensitive to the choice of the bandwidth. For that reason, we only report the findings for $m = N^{07}$. The three different estimates of the long memory of the conditional variance are close to one another and less than 0.5, implying that the absolute returns series is stationary, mean reverting, and long-memory processes.

In comparison, the evidence of long-memory in the conditional mean is weaker, as shown in Table 4. Although the estimates of the long memory of the returns are markedly small, especially the GPH and LW estimates, they significantly differ from zero. Generally speaking, the parameter estimates support the idea that dual long memory processes exist in the DJIA returns. The short-memory estimates of the ARFIMA(1, *d*, 1) model of the returns are AR(1) = 0.6031*** and MA(1)=-0.4682***, while the short-memory component of the ARFIMA (1, *d*, 0) of the absolute returns is AR(1) = 0.2583***, where *** means rejection of the null hypothesis at 0.01 significance level.

4.2 Results from dual long-memory models

Table 5 reports the estimated parameters of long memory dynamics in the returns and volatility for the ARFIMA-FIGARCH (Baillie et al. 1996), ARFIMA-FIAPARCH (Tse ,1998), ARFIMA-HYGARCH (Davidson 2004), and ARFIMA-HYAPARCH (Dark, 2006) models. The models are estimated by the Quasi-Maximum Likelihood (QML) procedure as implemented in Ox. In view of fat-tail characteristics in the data, the Student t, rather than the normal, distribution is assumed for the disturbances, as suggested by Bollerslev (1987). For the specification of the Student t log-likelihood see Davidson (2004). For a detailed description of the estimation procedure, see Baillie, Bollerslev, and Mikkelsen (1996). We used the Schwarz and the Hannan-Quinn Information criteria to identify the truncation orders of the short-memory polynomials of the ARFIMA model and the ARCH and GARCH polynomials of the conditional variance. The residuals from the ARFIMA model are used to estimate the long-memory behavior in the conditional variance. The models possess the distinctive feature that they simultaneously estimate the long memory in both returns and volatilities. They keep the analytical elegance of the ARMA-GARCH type models while enhancing their dynamics. Thus, the dual specification is more than a simple juxtaposition of two long-memory processes. The joint estimation of the two components of the model proves crucial for estimation and forecasting issues.

Several findings emerge from Table 5. The results indicate strong evidence of long memory both in the conditional mean and the conditional variance. The long-memory parameter d_m of the conditional mean is greater than 0 and less than 0.5, in the range $0.1635 \leq \ d_m \leq 0.1916$. The long-memory parameter d_m statistically differs from zero and one at the 1-percent level. This means that the DJIA returns are neither a unit-root process nor a stationary process with only short memory. Rather, they are stationary and mean returning process exhibiting long memory.

The long-memory parameter d_v of the conditional variance is greater than 0 and less than 0.5, in the range $0.4758 \leq \ d_m \leq 0.4873$. This means that the effect of shocks to the conditional volatility display a hyperbolic rate of decay as opposed to the conventional exponential decay inherent to the stable GARCH process or the infinite persistence pattern distinguishing the IGARCH model. Moreover, the long-memory parameter d_v significantly differs both from zero and one at the 1-percent level, rejecting the validity of both the stable GARCH and the integrated GARCH (IGARCH) specifications. The long memory in the conditional mean implies that stock prices follow a predictable behavior that is inconsistent with the weak form of the EMH. In finance, the weak form EMH asserts that information quickly and efficiently incorporates into asset prices at any point in time, so that past price information cannot be used to predict future price movements. The evidence of long memory in volatility, however, shows that uncertainty or risk importantly helps to determine the behavior of daily stock market data.

As commonly found in the literature, the presence of long memory is stronger in the returns volatility than in the returns, indicating that shocks to volatility in the DJIA index persist longer than shocks to returns. In Table 5, the AR(1) and MA(1) effects are represented by the coefficients θ_1 and φ_1, while the ARCH(1) and GARCH(1) effects are represented by the coefficients β_1 and ϖ_1.

Post-estimation diagnostic statistics provide no evidence of significant residual problems. The Ljung-Box $Q(20)$ and $Q^2(20)$ tests find no significant correlations in the conditional mean and volatility equations. Table 5 also reports the Brock-Dechert-Scheinkma (BDS) statistic (Brock et al., 1996), which tests the null hypothesis that the remaining residuals are independent and identically distributed (i.i.d.). Rejection of the i.i.d. hypothesis implies that some remaining structure exists in the time series, which could include a hidden nonlinearity, hidden nonstationarity, or other type of structure missed by the fit of the model. The test shows no evidence of low dimensional chaotic or nonlinear stochastic processes in the residuals. The parameter υ, representing the number of degrees of freedom, measures the degree of fat-tails of the density of the residuals. In all four models, the estimate is approximately 5, which is low, indicating fatter tails of the density.

The estimate of the asymmetry parameter γ in the ARIMA-FIAPARCH as well as in the ARFIMA-HYAPARCH is negative and significantly differs from zero at the 1-percent level. This means volatility shocks are not symmetric, but that positive shocks cause higher volatility than negative shocks. In other words, the negative sign on $\hat{\gamma}$ suggests that "good news," i.e., an unanticipated increase in the stock market, is more destabilizing than "bad news," i.e., an unanticipated stock market decline. The estimate of the power parameter δ in the ARIMA-FIAPARCH and ARFIMA-HYAPARCH models is positive and not significantly different from 2.

The estimate of the parameter α in the ARFIMA-HYGARCH and ARFIMA-HYAPARCH models is significantly different from zero, leading to the rejection of the stable GARCH model, and from one, leading to the rejection of the ARFIMA-FIGARCH model. Importantly, however, the value of the α parameter estimate exceeds one. This suggests that the driving process of the DJIA returns is not covariance stationary.

The empirical results highlight an important difference, i.e., the volatility shocks are not symmetric. Of the four models, this suggests that we discard the ARFIMA-FIGARCH and ARFIMA-HYGARCH models. Comparing the two remaining models, the ARFIMA-FIAPARCH and ARFIMA-HYAPARCH models, we observe that the latter has a slight edge in terms of loglikelihood and AIC. For this reason, in the forecasting analysis that follows, we use the ARFIMA-HYAPARCH model.

4.3 Results for the WLLWNN model

In this subsection, residuals from the ARFIMA modeling are the input of the novel WLLWNN model to estimate the conditional variance. To avoid the possibility of coupling among different inputs and to accelerate convergence, we normalize all inputs within a range of [0, 1] using the following formula before applying it to the network. This method is the most commonly used data smoothing method. That is, $\varepsilon_{norm} = \frac{\varepsilon_{orig} - \varepsilon_{min}}{\varepsilon_{max} - \varepsilon_{min}}$, where ε_{norm} is the normalized value of the residuals, ε_{orig} is the original value, ε_{min} and ε_{max} are the minimum and maximum values of the corresponding residuals.

These normalized data are then decomposed using the MODWT with Daubechies least asymmetric (La) wavelet filter of length $L = 8$ $(La(8))$. This wavelet filter is frequently adopted in the financial literature and provides the best performance for the wavelet time-series decomposition. Our MODWT decomposition goes up to level $J = 14$ that is specified by $J \leq log_2 \left[\frac{N}{L-1} + 1\right]$ i.e., where N represent the length of the given time series and L denote the length of the filter (Percival and Walden, 2000; Gencay et al., 2002). The time series is decomposed into 14 details $\left(D_t(1), \dots D_t(14)\right)$. The optimization of LLWNN is conducted as follows.

First, the data are divided into three successive parts as follows: (a) a sample of 300 observations to initialize the network training, (b) a training set of 2200 observations, and (c) a test set of 44 observations. The forecasting experiment is performed over the test set using an iterative forecasting scheme and the models are tested for three-time horizons; 1 day, 5 days, and 22 days.

Second, to find the best neural network architecture, at the beginning the parameters are randomly initialized. Then, using two different algorithms, the Back-Propagation algorithm (BP) and the Particle Swarm Optimization algorithm (PSO), these parameters are optimized to minimize the error between the output values and the real values during the training of the network. Table 5 and Table 6 provide the summary of information related to the network architecture. Table 5 defines the BP algorithm architecture and Table 6 states the parameters adopted for running the PSO.

5. Predictive Performance of the WLLWNN Model

This section evaluates the estimated models in a multi-step-ahead forecasting task. Since forecasting is basically an out-of-sample problem, we prefer to apply out-of-sample criteria. Accordingly, three different periods (1 day, 5 days, 22 days) were selected to ensure the quality and robustness of modeling and forecasting results. To evaluate the forecasting accuracy, we apply three evaluation criteria, namely the Mean Absolute Error (MAE), the Mean Squared Error (MSE), and the Root Mean Squared Error (RMSE), given respectively by:

$$MAE = \frac{1}{N-t_1}\sum_{t=t_1}^{N}\left|\left(r_{t+\square} - \hat{r}_{t,t+\square}\right)\right|,$$

(30)

$$MSE = \frac{1}{N-t_1}\sum_{t=t_1}^{N}\left(r_{t+\square} - \hat{r}_{t,t+\square}\right)^2,\ and \qquad (31)$$

$$RMSE = \left(\frac{1}{N-t_1}\sum_{t=t_1}^{N}\left(r_{t+\square} - \hat{r}_{t,t+\square}\right)^2\right)^{1/2}, \qquad (32)$$

where N is the number of observations, $N - t_1$ is the number of observations for predictive performance, r_{t+h} is the return series through period $t + h$, and $\hat{y}_{t,t+h}$ is the predictive log-return series of the predictive horizon h at time t.

We evaluate the predictive performance of the hybrid ARFIMA-WLLWNN against the individual LLWNN model, the hybrid ARFIMA-LLWNN model, the WLLWNN model, and the parametric ARFIMA-HYAPARCH model. In the ANN models, we apply two different learning algorithms (BP and PSO) for the training of the networks. Moreover, we adopted three horizons: 1-day, 5-days, and 22- days ahead forecasting, using the MAE, MSE, and RMSE out of sample criteria. Table 8 reports the forecast evaluation results.

The individual LLWNN based PSO algorithm outperforms the individual LLWNN based BP algorithm. In addition, the individual WLLWNN based PSO algorithm outperforms the individual WLLWNN based BP algorithm. These results prove the superiority of the PSO algorithm for training the neural network model. This result occurs because in the case of the BP algorithms weights are updated in the direction of the negative gradient. Hence, the network training with BP algorithms present some drawbacks such as slow convergence to a local minimum. In the case of training with PSO algorithm, however, weights are characterized by particles position. The hybrid ARFIMA-LLWNN model outperforms the individual LLWNN model, hence, the individual LLWNN model is unable to detect, to model, and to predict the features existing in the DJIA returns.

That is, when it is compared with the hybrid model, this last one provides prediction that is more accurate. Consequently, this network needs an external filter to better estimate the data, since the adoption of the ARFIMA model in the first step enhances the results of forecasting. The ARFIMA-HYAPARCH model outperforms the hybrid ARFIMA-LLWNN model in terms of prediction accuracy. This occurs because the ARFIMA-HYAPARCH model considers the long-memory in both the conditional mean and the conditional variance, making this model a robust tool that can deal with the features of the DJIA index. This is explained by the ability of the HYAPARCH in modelling the long-memory behavior in the conditional variance. This also proves the importance of considering the long-memory behavior to enhance forecasting accuracy.

Besides, the novel WLLWNN overcomes the limitation of the LLWNN that is related to the inability of the network to detect and model the periodic long-memory behavior in the data, since it shows its effectiveness when we compare it with the ARFIMA-HYAPARCH and ARFIMA-LLWNN models. Hence, the proposed hybrid ARFIMA-WLLWNN is a robust tool that can deal with the features of the DJIA index and provide the best forecasting results.

In summary, the ARFIMA-WLLWNN model outperforms all other computing techniques. In fact, this model uses the strength of three techniques at the same time. First, the ARFIMA model that allows detecting and estimating the long memory in the conditional mean. Second, the wavelet decomposition, which can produce a good local representation of the series and, hence, is a good tool to bring out the hidden patterns in the DJIA index. Finally, with the capacity of the LLWNN model as a nonlinear, nonparametric mode, and its particularity in using a wavelet activation function and local linearity, this network can capture more subtle and hidden features of the data.

Figures 5, 6, 7, 8, 9, 10, 11, and 12 confirm that the predictions of the ARFIMA-WLLWNN model based PSO algorithm for the 5-day and 22-day horizons are very close to the real values. That is, these figures confirm the forecasting results in Table 8, which indicate that the ARFIMA-WLLWNN model prediction errors are the smallest for all evaluation criteria. Note that we do not report the forecast graphs for one day, since only one observation is involved.

6. Conclusions

In this paper, we develop a relatively novel neural network model, called Wavelet Local Linear Wavelet Neural Network (WLLWNN). This novel network exhibits a higher generalization performance than the LLWNN. On the other hand, when we deal with neural networks, it is important to choose an appropriate algorithm for training. We present a comparison of the BP and PSO learning algorithms. The BP algorithm updates weights in the direction of the negative gradient. ANNs training with the BP algorithm presents certain drawbacks such as slow convergence that can be trapped in the local minimum. Weights in the PSO algorithm, however, are represented by particles position. The particles velocity and position are updated to search for the personal best and global best values. This avoids convergence to a local minimum. Our experimental results show the superiority of the PSO algorithm. We find that the effectiveness of the novel WLLWNN model is further enhanced by coupling it with the ARFIMA model. The experimental results indicate that the PSO-optimized version of the hybrid ARFIMA-WLLWNN outperforms the LLWNN, WLLWNN, ARFIMA-LLWNN, and the ARFIMA-HYAPARCH models and provides more accurate out-of-sample forecasts over validation horizons of one, five and twenty-two days.

References

Abbass, H. A., Sarker, R., and Newton, C., (2001) PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems. Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546), Seoul, South Korea, pp. 971-978 vol. 2, doi: 10.1109/CEC.2001.934295.

Andrews, D. W. K. and Guggenberger, P., (2003) A Bias-Reduced Log-periodogram Regression Estimator for the Long-Memory Parameter. *Econometrica* 71, 675-712.

Aye, G. C., Balcilar, M., Gupta, R., Kilimani, N., Nakumuryango, A., and Redford, S., (2014) Predicting BRICS stock returns using ARFIMA models. *Applied Financial Economics* 24, 1159-1166.

Baillie, R. T., Bollerslev, T. and Mikkelsen, H. O., (1996) Fractionally integrated generalized autoregressive conditional hetroskedasticity. *Journal of Econometrics* 74, 3-30.

Barkoulas, J. T, and Baum, C F., (1996) Long term dependence in stock returns. *Economics Letters* 53, 253–259

Barkoulas, J. T., Baum, C. F., and Travlos, N., (2000) Long memory in the Greek stock market. *Applied Financial Economics* 10, 177-184.

Baumol, W. J., (1965) *The Stock Market and Economic Efficiency*. New York: Fordham University Press.

Bhardwaj, G., and Swanson, N. R., (2006) An empirical investigation of the usefulness of ARFIMA models for predicting macroeconomic and financial time series*. Journal of Econometrics* 131, 539-578.

Black, F., (1976) Studies of stock price volatility changes. Proceedings of the business and economics section of the American Statistical Association, 177-181.

Bollerslev, T., (1986) Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics* 31, 307-327.

Bollerslev, T., (1987) A conditionally hetroskedastic time series model for speculative prices and rates of return. *The Review of Economics and Statistics* 69, 542-547.

Bollerslev, T., and Mikkelsen, H. O., (1996) Modeling and pricing long memory in stock market volatility. *Journal of Econometrics* 73, 151–184.

Bollerslev, T., and Wooldridge, J. M., (1992) Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. *Econometric Reviews* 11, 143-172.

Bourbonnais, R., and Maftei, M., (2012) ARFIMA process: Tests and applications at a white noise process, a random walk process and the stock exchange index CAC 40. *Economic Computation and Economic Cybernetics Studies and Research* 46, 5-17.

Brock, W. A., Dechert, W. D., and Scheinkman, J., (1996) A test for independence based on the correlation dimension. *Econometric Reviews* 15, 197-235.

Burton, B., and Harley, R. G., (1994) Reducing the computational demands of continually online trained artificial neural networks for system identification and control of fast processes. *Proceedings of the IEEE IAS Annual Meeting*, Denver, CO, 1836–1843.

Cavalcante, J., and Assaf, A., (2005) Long-range dependence in the returns and volatility of the Brazilian stock market. *European Review of Economics and Finance* 5, 5–20.

Chang, P.-C., Liu, C.-H. Lin, J.-L, Fan, C.-Y., and Ng., C. S. P., (2009) A neural network with a case based dynamic window for stock trading prediction. *Expert Systems with Applications* 36, 6889-6898.

Dark, J. G., (2006) Modelling the conditional density using a hyperbolic asymmetric power ARCH model. Mimeo, Monash University.

Dark, J. G., (2010) Estimation of time varying skewness and kurtosis with an application to value at risk. *Studies in Nonlinear Dynamics and Econometrics* 14, 1-50.

Davidian, M., and Carroll, R. J., (1987) Variance function estimation. *Journal of the American Statistical Association* 82, 1079–1091.

Davidson, J., (2004) Moment and memory properties of linear conditional heteroscedasticity models, and a new model. *Journal of Business and Economics Statistics* 22, 16-29.

Dickey, D. A., and Fuller, W. A., (1979) Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association* 74, 427–431

Ding, Z., Granger, C. W. J., and Engle, R. F., (1993) A long memory property of stock market returns and a new model. *Journal of Empirical Finance* 1, 83-106.

Disario, R., Saraoglu, H., McCarthy, J., and Li, H. (2008) Long memory in the volatility of an emerging equity market: the case of Turkey. *Journal of International Financial Markets, Institutions and Money* 18, 305-312

Eberhart, R. C., and Kennedy, J., (1995) A new optimizer using particle swarm theory. *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39–43, Nagoya, Japan. Piscataway: IEEE.

Fama, E. F., (1965) The behavior of stock market prices. *Journal of Business* 38, 34-105.

Floros, C., Jaffry, S., and Lima, G. V., (2007) Long memory in Portuguese stock market, *Studies in Economics and Finance* 24, 220-232.

Forsberg, L., and Ghysels, E., (2006) Why do absolute returns predict volatility so well? *Journal of Financial Econometrics* 6, 31-67

Gencay, R., Seluk, F., and Whitcher, B., (2002) *An Introduction to Wavelets and Other Filtering Methods in Finance and Economics*. Academic Press, New York.

Geweke, J., and Porter-Hudak, S., (1983) The estimation and application of long-memory time series models. *Journal of Time Series Analysis* 4, 221–238.

Gil-Alana, L. A., (2006) Fractional integration in daily stock market returns. *Review of Financial Economics* 15, 28–48.

Granger, C .W. J., and Ding, Z., (1995) Some properties of absolute returns. An alternative measure of risk. *Annals of Economics and Statistics* 40, 67–91

Granger, C. and R. Joyeux 1980. An Introduction to Long Memory Time Series Models and Fractional Differencing, Journal of Time Series Analysis, 1, 15-39.

Gulerce, M., and Unal, G., (2016) Using wavelet analysis to uncover co-movement behavior of multiple energy commodity prices. *International Journal of Wavelets Multiresolution Information Processing* 14, 1650047.

Guresen, E., Kayakutlu, G., and Daim, T. U., (2011) Using artificial neural network models in stock market index prediction. *Expert Systems with Applications* 38, 10389–10397.

Gurgul, H., and Wojtowicz, T., (2006) Long memory on the German stock exchange, Czeck. *Journal of Economics and Finance* 56, 447-468.

Henry, O. T., (2002) Long memory in stock returns: some international evidence. *Applied Financial Economics* 12, 725–729.

Hosking, J. R. M., (1981) Fractional differencing, *Biometrika* 68, 165-176.

Jefferis, K., and Thupayagale, P., (2008) Long memory in Southern African stock markets, *South African Journal of Economics* 75, 384-398.

Kang, H. S., and Yoon, S.-M., (2007) Long memory properties in return and volatility: Evidence from the Korean stock market. *Physica A: Statistical Mechanics and its Applications* 385, 591-600.

Kasman, A., and Torun, E., (2007) Long memory in the Turkish stock market return and volatility. *Central Bank Review* 2, 13-27.

Kasman, A., Kasman, S., and Torun, E., (2009) Dual long memory property in returns and volatility: Evidence from the CEE countries' stock markets, *Emerging Markets Review* 10, 122-139.

Khan, Z. H., Alin, T. S., and Hussain, A., (2011) Price prediction of share market using Artificial Neural Network (ANN). *International Journal of Computer Applications* (0975–8887) 22.

Killic, R., (2004) On the long memory properties of emerging capital markets: Evidence from Istanbul stock exchange. *Applied Financial Economics* 14, 915-922.

Kwiatkowski, D., Phillips, P. C. B., Schmidt, P., and Shin, Y., (1992) Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics* 54, 159–178.

Lee, T. S., and Chiu, C. C., (2002) Neural network forecasting of an opening cash price index. *International Journal of Systems Science* 33, 229-237.

Lin, X.-Q., and Fei, F.-Y., (2013) Long memory revisit in Chinese stock markets: Based on GARCH-class models and multiscale analysis. *Economic Modelling* 31, 265-275.

Lopez-Herrera, F., Ortiz, E., and de Jesus, R. (2012). Long memory behavior in the returns of the Mexican stock market: ARFIMA models and value at risk estimation. *International Journal of Academic Research in Business & Social Sciences* 2, 113–133.

Malkiel, B., (2003) The efficient markets hypothesis and its critics. *Journal of Economic Perspectives* 17, 59-82.

Mallat, S., (1989) Multiresolution approximation and wavelet. *Transactions of the American Mathematical Society* 315, 69-88.

Mandelbrot, B. B., (1971) When can price be arbitraged efficiently? A limit to the validity of the random walk and martingale models. *Review of Economics and Statistics* 53, 225-236

McMillan, D. G., and Thupayagale, P., (2009) Efficiency of the South African equity market, *Applied Financial Economics Letters* 4, \327-330.

McMillan, D. G., and Thupayagale, P., (2008) Efficiency of African equity markets. *Studies in Economics and Finance* 26, 275-292.

Percival, D. B., and Walden, A. T., (2000) *Wavelet Methods for Time Series Analysis*. Cambridge University Press, Cambridge.

Phillips, P. C. B., and Perron, P., (1988) Testing for a unit root in a time series regression. *Biometrika* 75, 335–346.

Reboredo, J. C., and Rivera-Castro, M. A., (2014) Wavelet-based evidence of the impact of oil prices on stock returns. *International Review of Economics and Finance* 29, 145–176.

Robinson, P. M., (1995) Gaussian semiparametric estimation of long range dependence. *Annals of Statistics* 23, 1630–1661

Sadique, S., and Silvapulle, P., (2001) Long-term memory in stock market returns: International evidence. *International Journal of Finance & Economics* 6, 59–67.

Schoffer, O., (2003) HY-A-PARCH: A stationary A-PARCH model with long memory. Mimeo, University of Dortmund.

Tay, F. E. H., and Cao, L., (2001) Application of support vector machines in financial time series forecasting. *Omega* 29, 309–317.

Taylor, S. J., (1986) *Modelling Financial Time Series*. Wiley, New York.

Tse, Y. K., (1998) The conditional heteroscedasticity of the Yen-Dollar exchange rate. *Journal of Applied Econometrics* 13, 49–55.

Table 1. Descriptive statistics of the DJIA returns

Mean	St. Dev.	Minimum	Maximum	Skewness	Kurtosis
0.0004	0.0088	-0.0571	0.0486	-0.4783	7.0063

Notes: The number of observations is 2544. The sample is 01/01/2010 to 02/11/2020. The data are from the Federal Reserve Bank of St. Louis. Federal Reserve Economic Data

Table 2. Tests of normality, autocorrelation and unit root

Jarque-Bera	Q (20)	ADF	PP	KPSS
1798.416***	41.130***	-52.5713***	-53.0383***	0.0241***

Notes: Rejection of the null hypothesis is displayed by *, **, and *** for 10 percent, 5 percent, and 1 percent significance level. ADF is the augmented Dickey Fuller statistic, PP is the Phillips Perron statistic, KPSS is the Kwiatkowski Phillips Schmidt Shin statistic. The Jarque-Bera statistic is chi-squared distributed with two degrees of freedom. $Q(20)$ is the Ljung-Box statistic for serial correlation in the returns for order 20.

Table 3. Long-memory tests in the conditional variance

GPH	LW	ARFIMA (1, *d*, 0)
0.3297***	0.3625***	0.395***

Notes: GPH is the Geweke-Porter-Hudak estimator; LW is the local Whittle estimator; the bandwidth is $m = N^{07}$. Rejection of the null hypothesis is displayed by *, **, and *** for 10 percent, 5 percent, and 1 percent significance level.

Table 4. Long-memory tests in the conditional mean

GPH	LW	ARFIMA (1, *d*, 1)
0.0264***	0.0125***	0.1757***

Notes: GPH is the Geweke-Porter-Hudak estimator; LW is the local Whittle estimator; the bandwidth is N^{07}. Rejection of the null hypothesis is displayed by *, **, and *** for 10 percent, 5 percent, and 1 percent significance level.

Table 5. Estimation of dual memory models

	ARFIMA-FIGARCH	**ARFIMA-FIAPARCH**	**ARFIMA-HYGARH**	**ARFIMA-HYAPARCH**
(p, d_m, q)	$(1, d_m, 1)$	$(1, d_m, 1)$	$(0, d_m, 1)$	$(0, d_m, 1)$
(P, d_v, Q)	$(1, d_v, 1)$	$(1, d_v, 1)$	$(1, d_v, 1)$	$(1, d_v, 1)$
μ	0.0001	0.0000	0.0000	0.0000
	(6.0891)***	(7.3562)***	(5.9851)***	(6.2351)***
d_m	0.1635	0.1786	0.1842	0.1916
	(3.9982)***	(4.2381)***	(4.2843)***	(4.3647) ***
θ_1	-0.4885	-0.4234	-	-
	(-4.6742)***	(-2.1733)**		
φ_1	0.5258	0.5026		0.4736
	(1.6854)*	(2.2611)**		(2.2743)**
ω	0.0000	0.0000	0.0000	0.0000
	(0.1763)	(0.1837)	(0.1793)	(0.1814)
d_v	0.4873	0.4987	0.4875	0.4758
	(4.8983)***	(4.4361)***	(3.9987)***	(4.1857)***
Log(α)	-	-	0.2354	0.238
			(3.8869)***	(4.615)***
γ	-	-0.4172	-	-0.4428
		(-5.6738)***-		(-5.8396)***
δ	-	1.9743	-	2.0581
		(16.6758)***		(16.7866)***
β_1	0.5134	0.4854	0.4786	0.4765
	(4.3268)***	(4.4710)***	(4.7874)***	(4.983)***
ϖ_1	0.3421	0.3192	0.2988	0.3129
	(3.7855)***	(3.9562)***	(3.9586)***	(4.0135)***
$\hat{\upsilon}$	5.2114	5.1556	5.5885	5.6349
	(9.4771)***	(9.3484)***	(9.6744)***	(9.8753)***
Skw	0.4316	0.4662	0.4541	0.4395
	(3.4748)***	(2.0435)*	(3.4786)***	(1.9862)*
Ex. Kurt	2.7342	2.6672	2.6527	2.6281
	(22.345)***	(24.674)***	(24.673)***	(23.729)***
Q(20)	23.7440	23.1452	22.5548	22.5342
$\mathrm{Q}^2(20)$	14.5242	14.5677	14.3437	14.2366
BDS(5)	3.6586	3.5519	3.5332	5.2530
Log(L)	8886.6754	8889.7783	8892.8324	8896.7665
AIC	-6.9869	-6.9925	-6.9783	-6.9781

Notes: The values in parenthesis are the t-Student. $\hat{\upsilon}$ is the degree of freedom of the Student's t distribution. Skw is Skewness. Ex. Kurt is Excess of Kurtosis. Q(20) is the Ljung-Box statistic for serial correlation in the standardized residuals for order 20. $\mathrm{Q}^2(20)$ is the Ljung-Box statistic for serial correlation in the squared standardized residuals for order 20. Log(L) is the value of the maximized Student t log -likelihood, AIC is the Akaike information criteria $*$, $**$ and $***$ denote significance at the 10%, 5% and 1% levels respectively.

Table 6. LLWNN based BP algorithm architecture

Number of hidden layers	10
Learning rate	0.5
Layer conversion function	Wavelet Function
Algorithm	Back Propagation (BP) Algorithm

Table 7. LLWNN based PSO algorithm architecture

Number of populations	20
Number of generations	200
C_1, C_2	1.05
Maximum velocity	1
Minimum velocity	0.3
Number of hidden layers	10
Learning rate	0.5
Layer Activation function	Wavelet Function

Table 8. Out of Sample Forecasts Results

Model	**Criterion**	$\boldsymbol{h = 1}$	$\boldsymbol{h = 5}$	$\boldsymbol{h = 22}$
LLWNN (BP Algorithm)	MAE	0.0088	0.0159	0.0101
	MSE	7.723×10^{-5}	5.0622×10^{-4}	1.6654×10^{-4}
	RMSE	0.0096	0.0228	0.0127
LLWNN (PSO Algorithm)	MAE	0.0084	0.0145	0.0132
	MSE	1.0343×10^{-4}	2.6355×10^{-4}	2.8935×10^{-4}
	RMSE	0.0104	0.0166	0.0171
ARFIMA-LLWNN (BP algorithm)	MAE	0.0041	0.0084	0.0095
	MSE	3.1976×10^{-5}	1.2487×10^{-4}	1.2539×10^{-4}
	RMSE	0.0058	0.0122	0.0118
ARFIMA-LLWNN (PSO algorithm)	MAE	0.0019	0.0054	0.0073
	MSE	3.5129×10^{-6}	2.2690×10^{-5}	3.9264×10^{-5}
	RMSE	0.0019	0.0045	0.0063
WLLWNN (BP Algorithm)	MAE	4.6978×10^{-6}	4.0551×10^{-6}	4.5983×10^{-6}
	MSE	2.2316×10^{-11}	1.7424×10^{-11}	2.6346×10^{-11}
	RMSE	4.7239×10^{-6}	4.1749×10^{-6}	5.1349×10^{-6}
WLLWNN (PSO Algorithm)	MAE	3.7001×10^{-8}	1.6286×10^{-7}	2.7986×10^{-7}
	MSE	1.7781×10^{-15}	3.3395×10^{-14}	9.8627×10^{-14}
	RMSE	4.2170×10^{-8}	1.8274×10^{-7}	3.1405×10^{-7}
ARFIMA-WLLWNN (BP Algorithm)	MAE	2.0409×10^{-6}	2.7731×10^{-6}	6.4121×10^{-7}
	MSE	4.3281×10^{-12}	8.0457×10^{-12}	4.2383×10^{-13}
	RMSE	2.0804×10^{-6}	2.8365×10^{-6}	6.5102×10^{-7}
ARFIMA-WLLWNN (PSO Algorithm)	MAE	4.7726×10^{-9}	3.1504×10^{-8}	2.9496×10^{-8}
	MSE	3.6761×10^{-17}	1.8611×10^{-15}	1.4299×10^{-15}
	RMSE	6.0631×10^{-9}	4.3140×10^{-8}	3.7813×10^{-8}
ARFIMA-HYAPARCH	MAE	5.2316×10^{-7}	4.8637×10^{-6}	3.2478×10^{-8}
	MSE	3.2458×10^{-16}	2.9452×10^{-14}	2.9773×10^{-15}
	RMSE	1.8000×10^{-8}	1.7160×10^{-7}	5.4600×10^{-8}

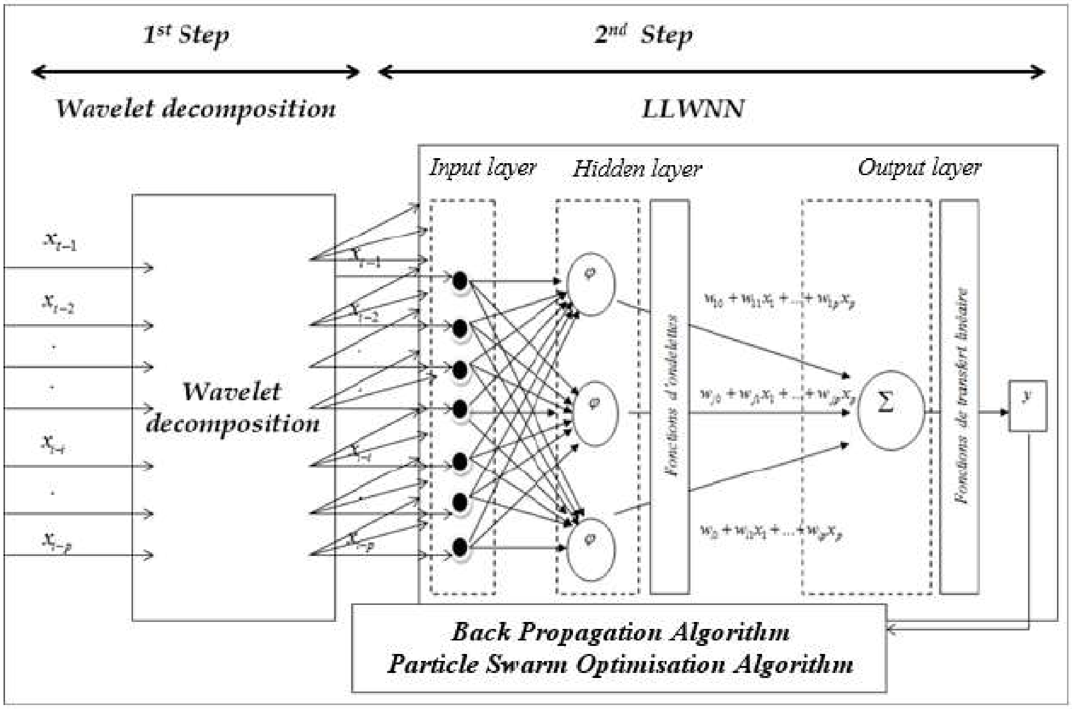

1st Step
2nd Step
Wavelet decomposition
LLWNN
Input layer
Hidden layer
Output layer
x_{t-1}
x_{t-2}
x_{t-i}
x_{t-p}
Wavelet decomposition
φ
φ
φ
Fonctions d'ondelettes
$w_{10} + w_{11}x_1 + \ldots + w_{1p}x_p$
$w_{j0} + w_{j1}x_1 + \ldots + w_{jp}x_p$
$w_{l0} + w_{l1}x_1 + \ldots + w_{lp}x_p$
Σ
Fonctions de transfert linéaire
y
Back Propagation Algorithm
Particle Swarm Optimisation Algorithm

Figure 1. Schematic representation of the WLLWNN model

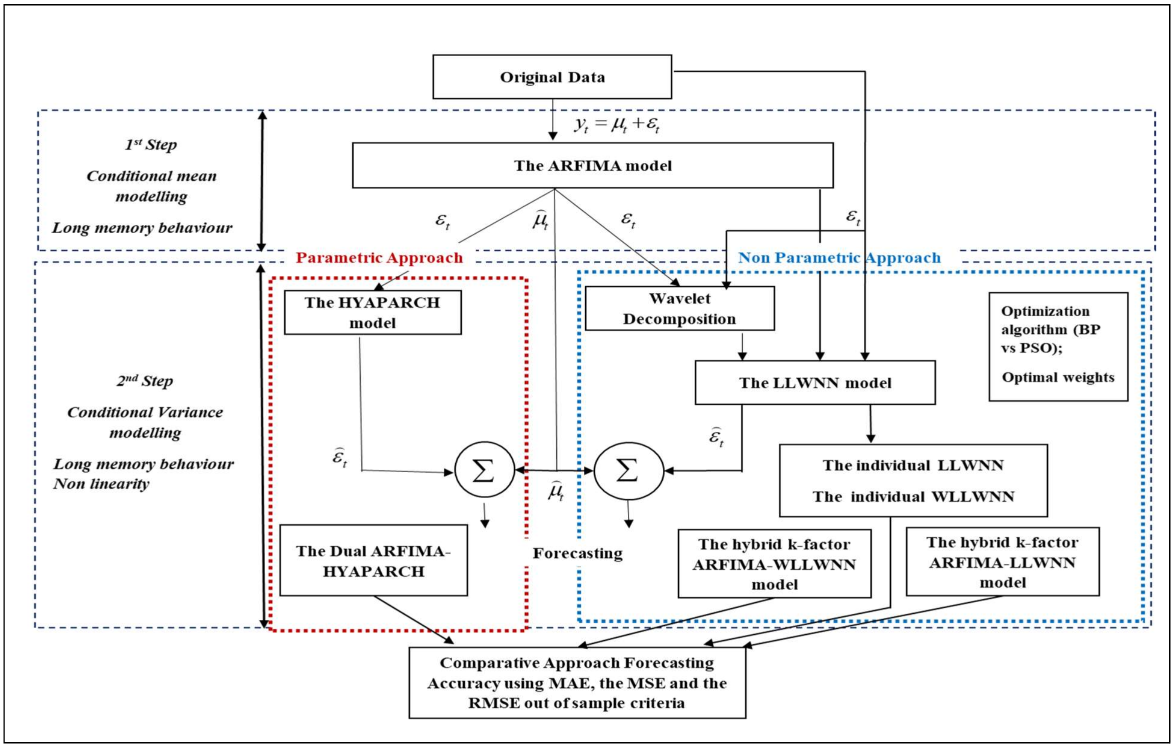


Figure 2. Schematic representation of ARFIMA-WLLWNN vs ARFIMA-HYAPARCH

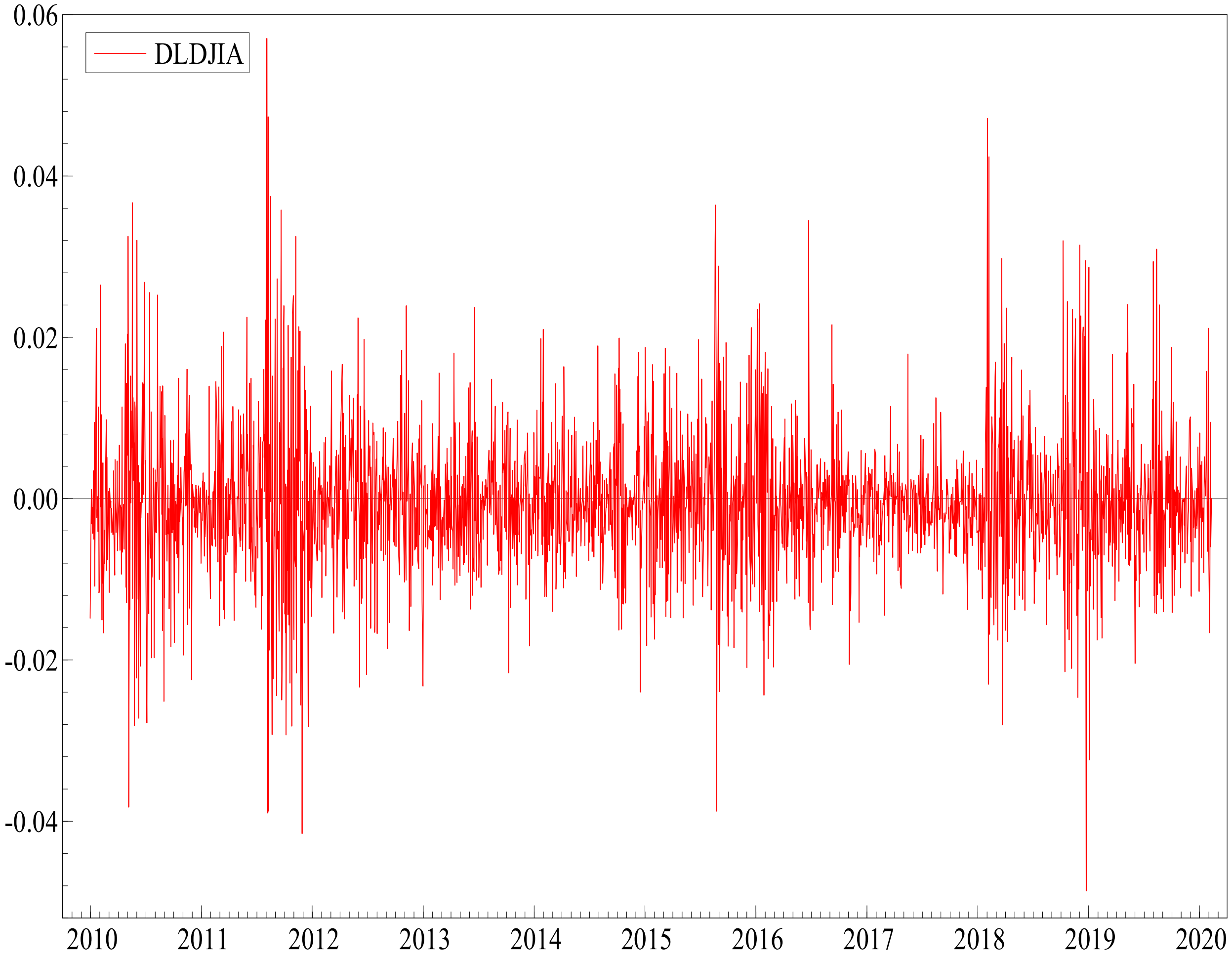
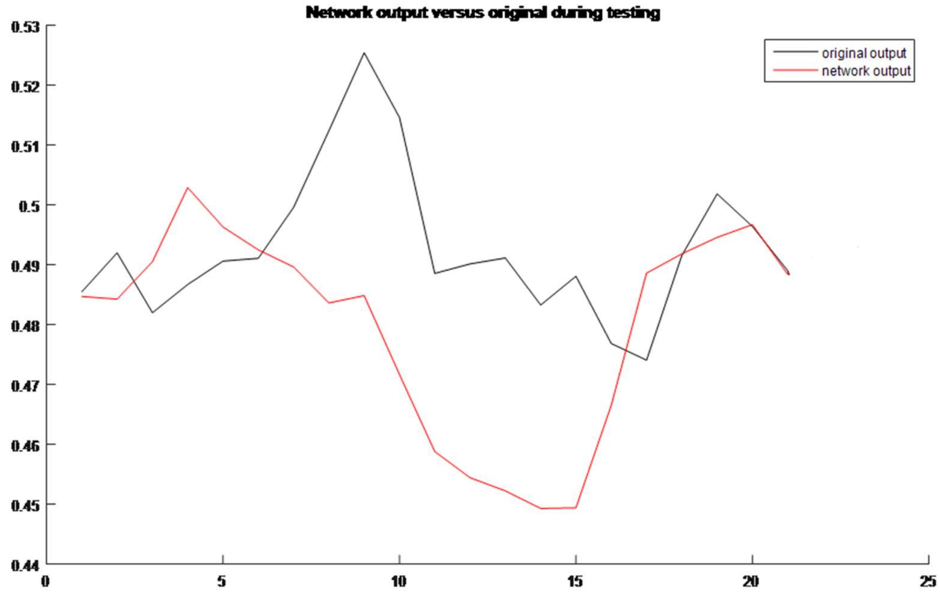
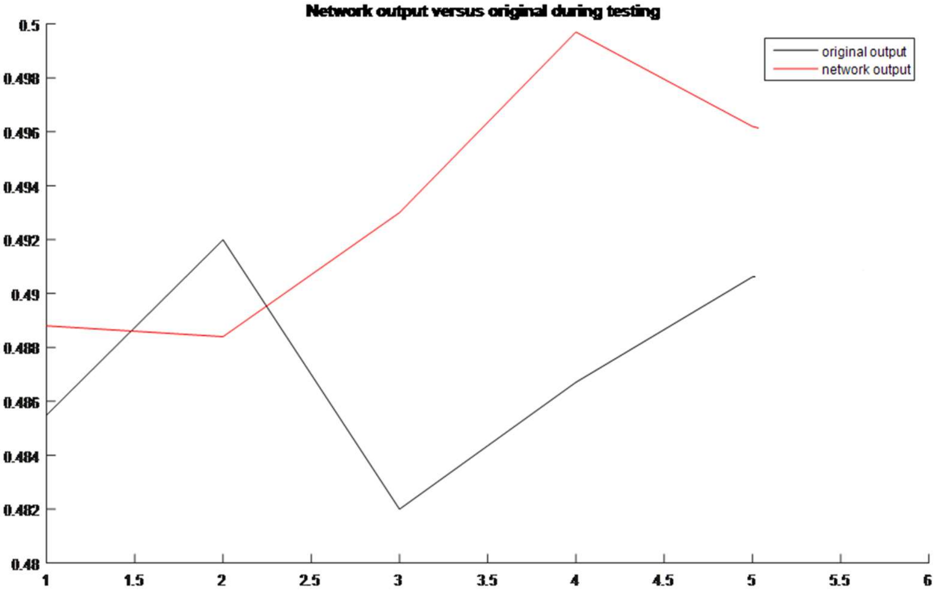


Figure 3: Returns of the DJIA from 01/01/2010 to 02/11/2020



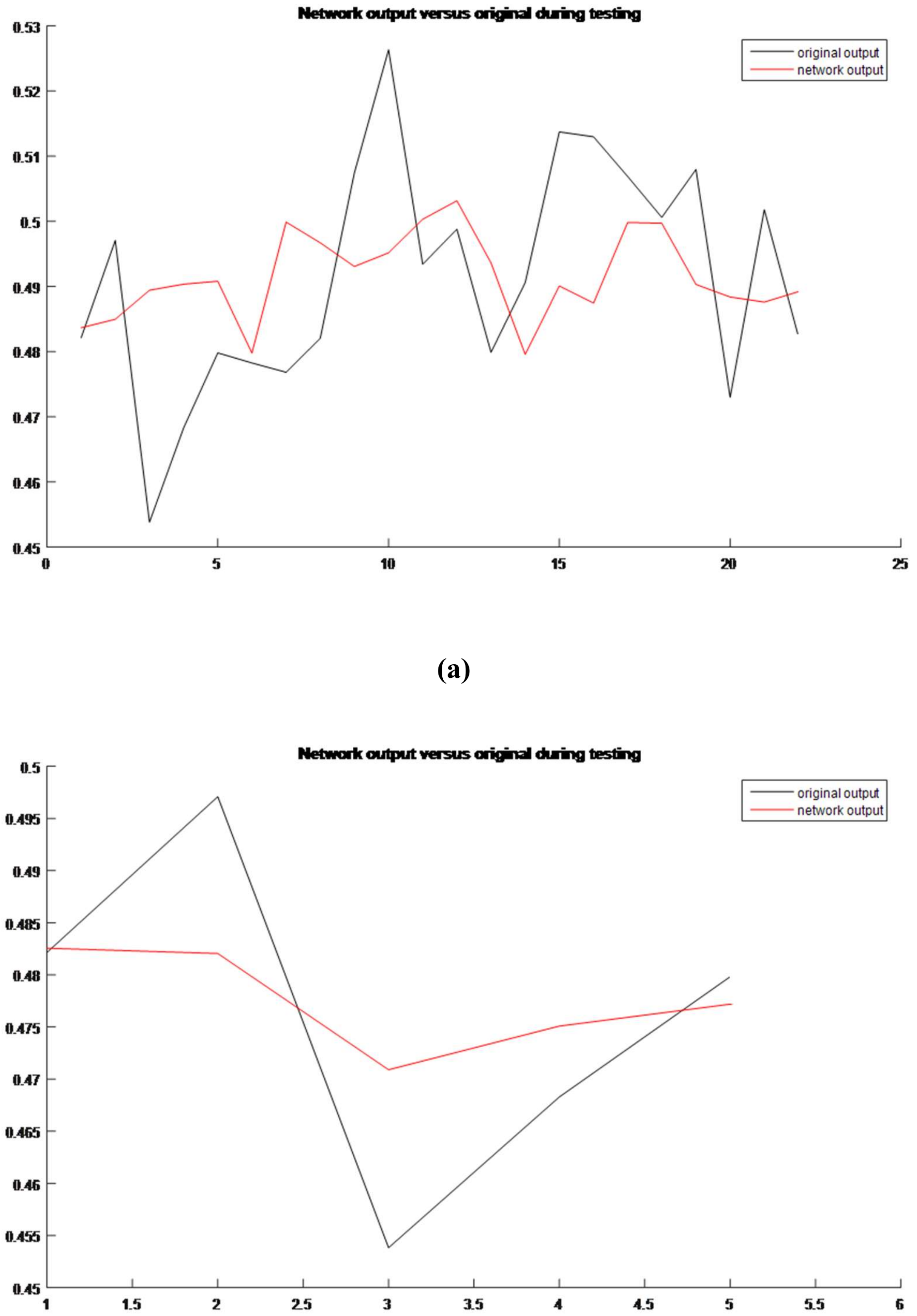
(a)



(b)

Figure 4: Forecasting the DJIA returns using LLWNN based BP Algorithm.

Forecasting horizon (a) h=22-day and (b) h=5-day.

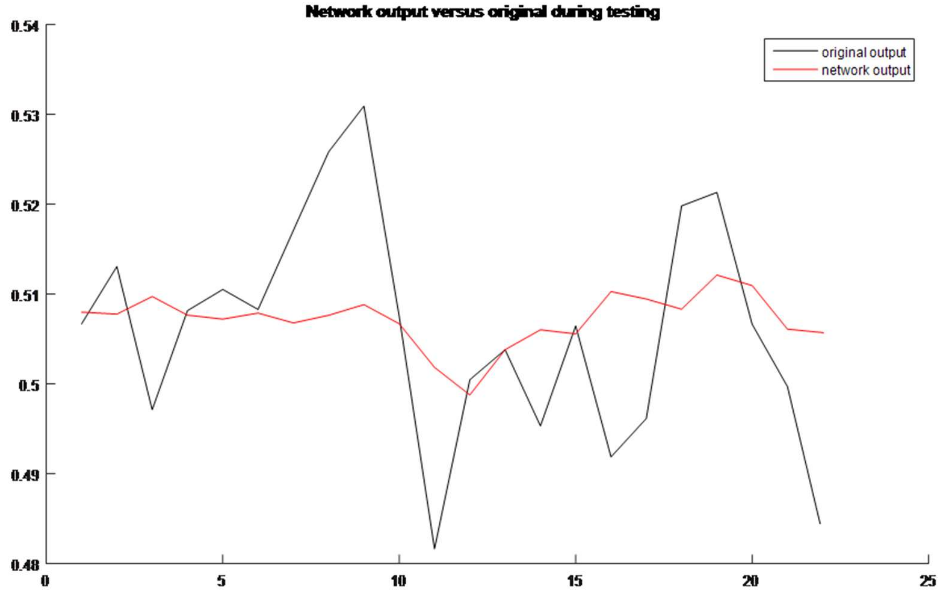


(a)

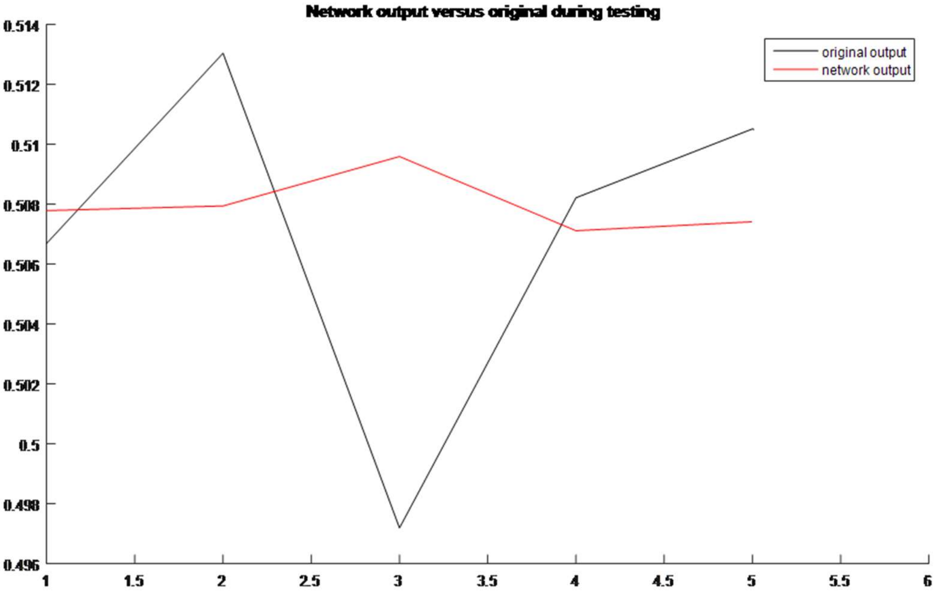
(b)

Figure 5: Forecasting the RDJIA using LLWNN based PSO Algorithm.

Forecasting horizon (a) h=22-day and (b) h=5-day.



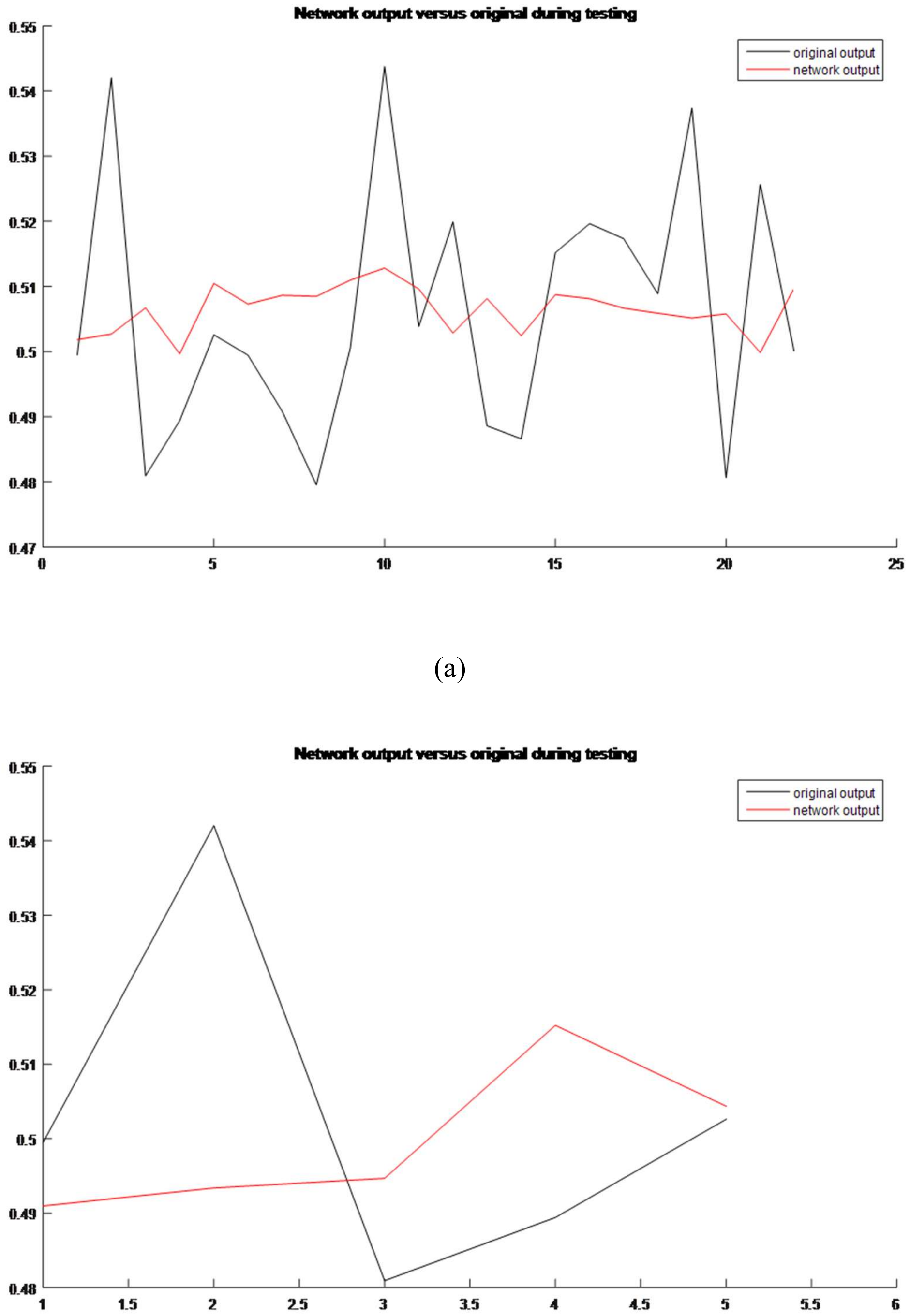
(a)



(b)

Figure 6: Forecasting using the ARFIMA-LLWNN based BP Algorithm.

Forecasting horizon (a) h=22-day and (b) h=5-day.

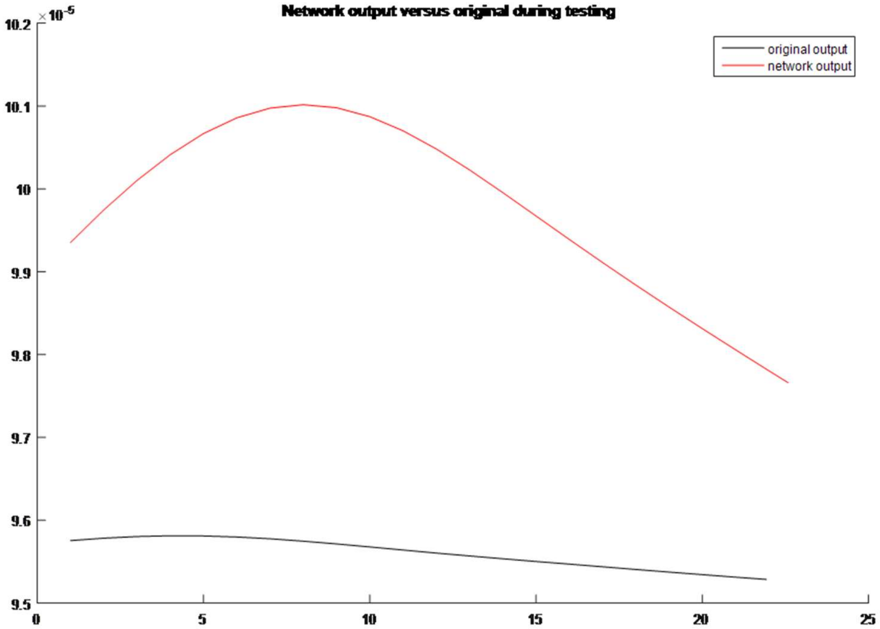


(a)

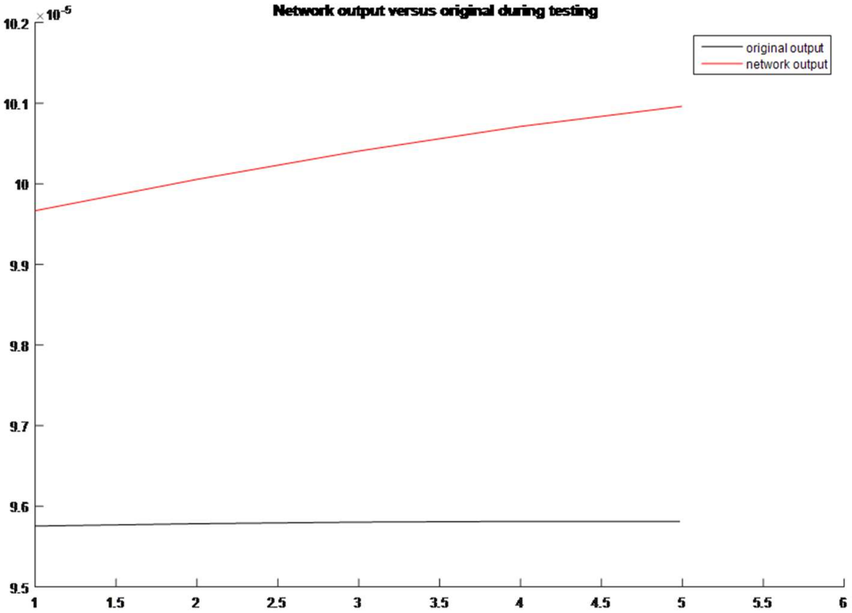
(b)

Figure 7: Forecasting using ARFIMA-LLWNN based PSO Algorithm.

Forecasting horizon (a) h=22-day and (b) h=5-day.



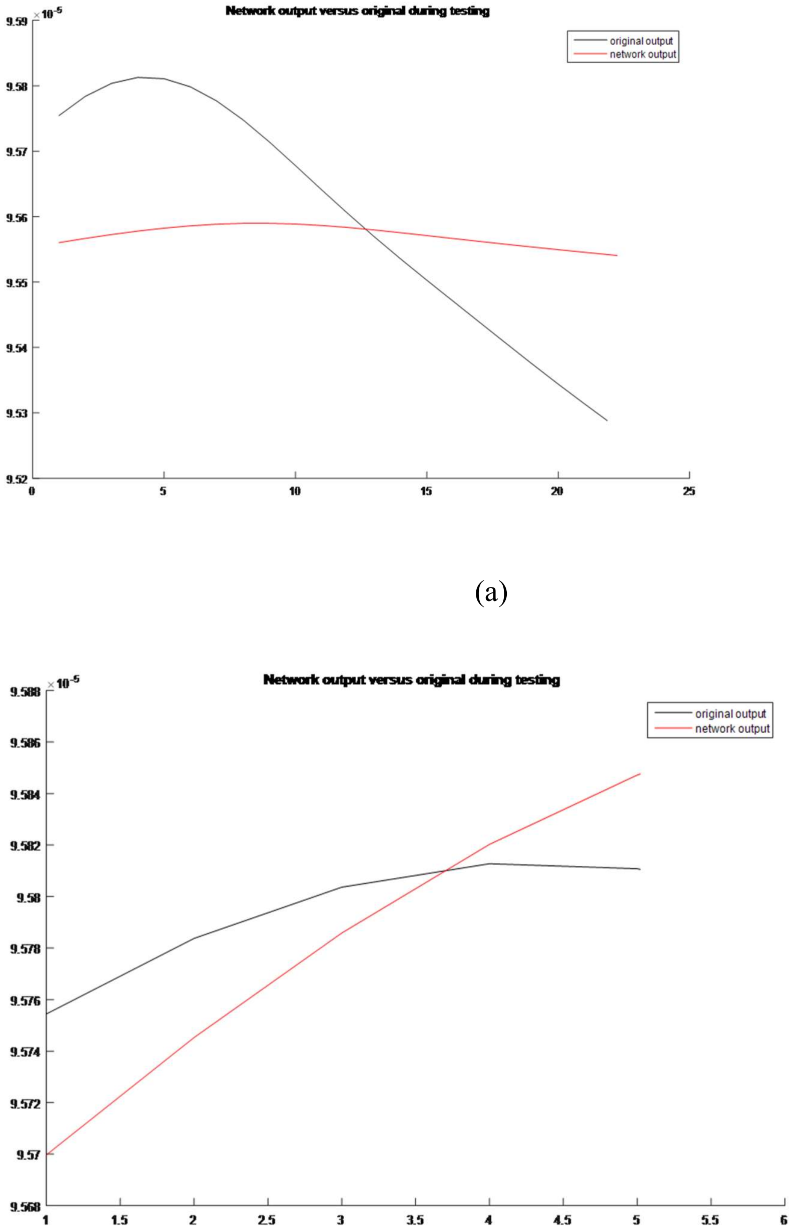
(a)



(b)

Figure 8. Forecasting RDJIA using the WLLWNN based BP Algorithm.

Forecasting horizon (a) h=22-day and (b) h=5-day.

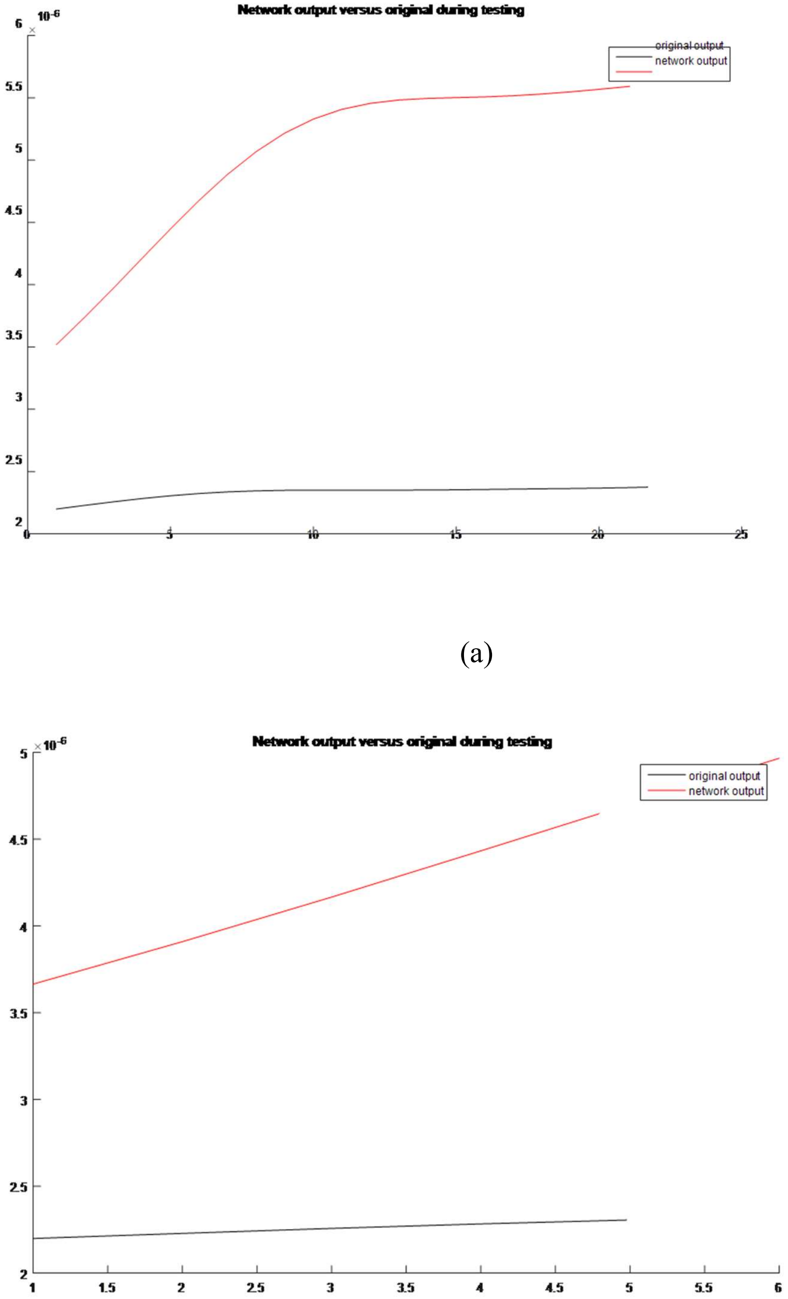


(a)

(b)

Figure 9 Forecasting the RDJIA using WLLWNN based PSO Algorithm.

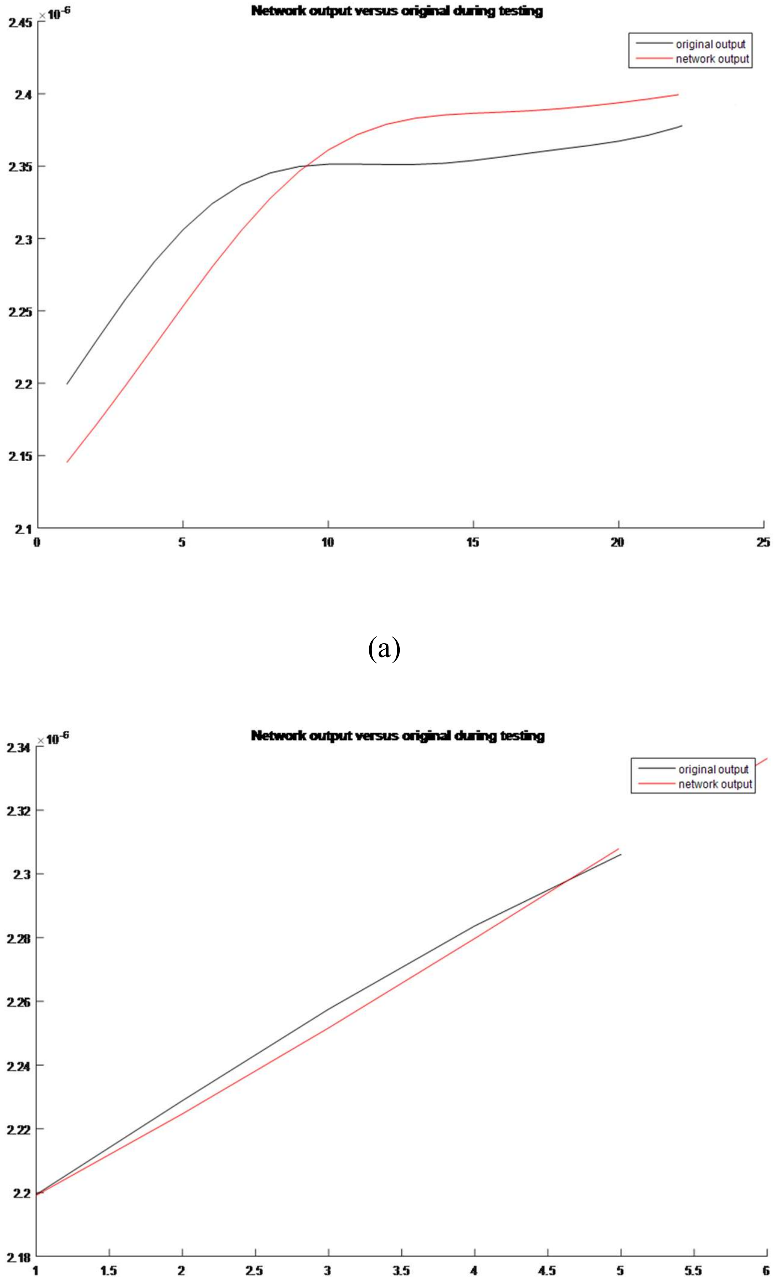
Forecasting horizon (a) h=22-day and (b) h=5-day.



(a)

(b)

Figure 10. Forecasting RDJIA using the ARFIMA-WLLWNN based BP Algorithm; Forecasting horizon (a) h=22-day and (b) h=5-day.



(a)

(b)

Figure 11. Forecasting RDJIA using the ARFIMA-WLLWNN based PSO Algorithm;

Forecasting horizon (a) h=22-day and (b) h=5-day.