# Wavelet Neural Networks for Weather Derivatives Pricing

**Achilleas Zapranis[1], Antonis Alexandridis[2]**

*University of Macedonia*
*of Economic and Social Sciences*
*Department of Accounting and Finance*
*156 Egnatia St.*
*54006 Thessaloniki*
*Greece*

***Abstract***

In this paper we use wavelet neural networks to model and remove the seasonal cycle as well as any possible trends, singularities or jumps of the temperature process. Moreover, we give a complete framework for structuring and training feed forward wavelet neural networks via back-propagation. As we demonstrate here, wavelet networks simplify significantly the mathematics of weather derivatives pricing, since no particular functional form is assumed. Our findings suggest that wavelet networks can model the temperature process very well and consequently they constitute a very accurate and efficient tool for weather derivatives pricing.

## 1. Introduction

Since their inception in 1996, weather derivatives have known a substantial growth. The first parties to arrange for, and issue weather derivatives in 1996, were energy companies, which after the deregulation of energy markets were exposed to weather risk. In September 1999, the Chicago Mercantile Exchange (CME) launched the first exchange traded weather derivatives. In 2004, the notional value of CME weather derivatives was $2.2 billion and grew nine-fold to $22 billion through September 2005, with open interest exceeding 300,000 and volume surpassing 630,000 contracts traded. However, the Over-The-Counter (OTC) market is still more active than the exchange, so the bid-ask spreads are quite large. Today, weather derivatives are being used for hedging purposes by companies and industries, whose profits can be adversely affected by unseasonal weather or, for speculative purposes by hedge funds and others interested in capitalizing on those volatile markets.

A weather derivative is a financial instrument that has a payoff derived from variables such as temperature, snowfall, humidity and rainfall. However, it is estimated that 98-99% of the weather derivatives now traded are based on temperature. This is not surprising since, it is estimated that 30% of the US economy is affected by temperature (CME, 2005). The elec-

[1] zapranis@uom.gr
[2] aalex@uom.gr

tricity sector is especially sensitive to the temperature. According to Li and Sailor (1995) and Sailor and Munoz (1997), temperature is the most significant weather factor explaining electricity and gas demand in the United States. The impact of temperature in both electricity demand and price has been considered in many papers, including Henley and Peirson (1998), Peirson and Henley (1994) and Engle et al (1992). Unlike insurance and catastrophe-linked instruments, which cover high-risk and low probability events, weather derivatives shield revenues against low-risk and high probability events (e.g., mild or cold winters).

Weather risk is unique in that it is highly localized, and despite great advances in meteorological science, still cannot be predicted precisely and consistently. Weather derivatives are also different than other financial derivatives in that the underlying weather index (HDD, CDD, CAT, etc.) cannot be traded. Furthermore, the corresponding market is relatively illiquid. Consequently, since weather derivatives cannot be cost-efficiently replicated with other weather derivatives, arbitrage pricing cannot directly apply to them. The weather derivatives market is a classic incomplete market, because the underlying weather variables are not tradable. When the market is incomplete, prices cannot be derived from the no-arbitrage condition, since it is not possible to replicate the payoff of a given contingent claim by a controlled portfolio of the basic securities. Consequently, the classical Black-Scholes-Merton pricing approach, which is based on no-arbitrage arguments, cannot be directly applied. And market incompleteness is not the only reason for that; weather indices do not follow random walks (as the Black & Scholes approach assumes) and the payoffs of weather derivatives are determined by indices, which are average quantities, whilst the Black-Scholes payoff is determined by the value of the underlying exactly at the maturity date of the contract (European options).

There are several approaches for dealing with incomplete markets. One of them is to introduce the 'market price of risk' for the particular type of the incomplete market, namely a 'factor model', market where there are some non-traded underlying objects. Since, weather derivatives are path depended they are very similar to the average Asian option and similar analytical pricing approaches can be used in this case too. A characteristic example is the approach of Geman and Yor (1993), which used Bessel processes to obtain an exact analytical expression of the Laplace transformation in time of the option price.

A pricing methodology for weather derivatives that is widely used in insurance is the actuarial (or insurance) method. It is based on statistical analysis and it is less applicable in contracts with underlying variables that follow recurrent, predictable patterns. Since, this is the case for most of the weather derivatives contracts, actuarial analysis is not considered the most appropriate pricing approach unless the contract is written on rare weather events such as extreme cold or heat.

Another approach for weather derivatives pricing, is performing simulations based on historical data, known as historical Burn analysis. That is, computing the average payoff of the weather derivatives in the past $n$ years. The central assumption of this method is that the historical record of weather contracts payoffs gives a precise illustration of the distribution of the potential payoffs (Dischel, 1999). If weather risk is calculated as the payoffs standard deviation, then the price of the contract will be $P(t) = D(t, T) \times (\mu \pm \alpha \times \sigma)$, where $D(t, T)$ is the discount factor from contract maturity $T$ to the pricing time $t$, $\mu$ is the historical average payoff, $\sigma$ is the historical standard deviation of payoffs and $a$ is a positive number denoting risk tolerance. However, since the weather processes are not stationary and this approach does not incorporate forecasts, it is bound to be biased and inaccurate. In fact, the historical Burn analysis is considered as the simplest pricing method in terms of implementation, and the most probable to cause large pricing errors.

In contrast to the previous methods, a dynamic model can be used which directly simulates the future behavior of temperature. Using models for daily temperatures can, in principle, lead to more accurate pricing than modeling temperature indices. In the process of calculating the temperature index, such as HDD, as a normal or lognormal process, a lot of information is lost (e.g., HDD is bounded by zero). On the other hand, deriving an accurate model for the daily temperature is not a straightforward process. Observed temperatures show seasonality in all of the mean, variance, distribution and autocorrelations and long memory in the autocorrelations. The risk with daily modeling is that small misspecifications in the models can lead to large mispricing in the contracts.

The continuous processes used for modeling daily temperatures usually take a mean-reverting form, which has to descretized in order to estimate its various parameters. Once the process is estimated, one can then value any contingent claim by taking expectation of the discounted future payoff. Given the complex form of the process and the path-dependent nature of most payoffs, the pricing expression usually does not have closed-form solutions. In that case Monte-Carlo simulations are being used. This approach typically involves generating a large number of simulated scenarios of weather indices to determine the possible payoffs of the weather derivative. The fair price of the derivative is then the average of all simulated payoffs, appropriately discounted for the time-value of money; the precision of the Monte-Carlo approach is depended on the correct choice of the temperature process and the look back period of available weather data.

In this paper, we address the problem of pricing the European CAT options. For this purpose we extent the mean-reverting process with seasonality in the level and volatility proposed by Benth and Saltyte-Benth (2005) - a generalization of (Dornier and Querel, 2000) which is descretized in the form of an AR(1) model. We estimate non-parametrically the temperature process with a wavelet network. We use 10 years of monthly average temperatures in Paris, in order to identify the wavelet network and 1 year to validate it.

Given the temperature model, the first step is to identify and remove from the temperature series the (possible) trend and the non-stationary seasonal cycle, hoping that what is left will be stationary. This is usually done by modeling the seasonal variations as deterministic and the same every year (seasonally stationary). The stochastic variability of the temperature is then moved entirely from the seasonal cycle into the residuals.

In modeling the seasonal cycle deterministically, there are several approaches. The discrete Fourier transform (DTF) is considered to be the most accurate, since, in principle at least, removes the seasonal cycle both in the mean and in the variance. For a detailed discussion on this subject see Jewson and Brix (2005). However, recently Zapranis and Alexandridis (2006, 2007a,b) proposed a novel approach in modeling the seasonal cycle which is an extension of the DFT approach. As it was shown, wavelet analysis is very useful in offering guidance as to which terms of the Fourier series to select. The wavelet decomposition brings out the structure of the underlying temperature series as well as trends, periodicities, singularities or jumps that could not be observed originally (Alaton *et al*., 2000 and Davis, 2001). However wavelet analysis may lead to significant errors since the selection of the terms of Fourier series is done by optical examination. To avoid this a new class of neural network, called wavelet networks, is used in this paper.

Wavelet networks proposed by Zhang & Benveniste (1992) as an alternative to feedforward neural networks. Wavelet Networks are one hidden layer networks that use a wavelet as an activation function instead of the classic sigmoid function. The activation function can be a wavenet (orthogonal wavelets) or a wave frame (continuous wavelets). Wavelet networks are performing excellent in predicting nonlinear behaviors (Gao & Tsoukalas, 2001). Wavelets show local characteristics hence the hidden units of the wavelet network affect the prediction of the network only in a local range. (Postalcioglou & Becerikli, 2007).

Wavelet networks have been used in a variety of applications so far. They first have been used in static and dynamic input-output modeling (Zhang & Benveniste, 1992; Postalcioglu & Becerikli, 2007) and proved that wavelet networks need less training iterations. Szu *et al.* (1992) used for classification of phonemes and speaker recognition. Gao & Tsoukalas (2001) consider wavelet networks one of the most promising tools to solve electricity load prediction problems. Subasi *et al.* (2005) used wavelet networks for classification of electroencephalography (EEG) signals while Khayamian *et al.* (2005) used wavelet networks as a multivariate calibration method for simultaneous determination of test samples of copper, iron and aluminum.

In our knowledge, we are the first to implement wavelet networks in temperature derivates for CAT future pricing. Our aim is to use wavelet networks to automatically model and remove the seasonality and the possible trends of the temperature. Wavelet analysis can decompose a signal into a series of approximations and details. By examining the decomposed signal the terms of the seasonal cycle are selected. Wavelet networks on the other hand approximate the original signal and can automatically remove the seasonal cycle from the data. While we avoid any errors we lose the information of the structure of the underlying temperature process. The same amount of information, as in wavelet analysis, can be obtained using wavelet networks but it needs heavy computation and time.

In this paper we use a wavelet network in order to model and remove from our data the seasonal cycle of the temperature as well as any possible trends, singularities or jumps of the temperature process. More precisely, the rest of the paper is organized as follows. In section 2, we describe the process used to model the average daily temperature in Paris. In section 3 we describe the wavelet network. In section 3.1 we give a brief background in wavelet theory. In section 3.2 we describe the structure of wavelet networks. In section 3.3 we present the training algorithm for the network. In section 3.4 we present the initialization conditions of the network parameters and in section 3.5 the stopping conditions for the training of the network. In section 4 we use wavelet networks to different simulated examples. In section 4.1 we examine a static function. In section 4.2 we give an example of analyzing a simulated upward trend with an AR(3) noise component, in section 4.3 we analyze a Geometric Brownian. In section 5 we apply our model to real data. In section 6 we discuss CAT derivatives pricing and finally, in section 7 we conclude.

## 2. Modeling Temperature Processes

Many different models have been proposed in order to describe the dynamics of a temperature process. The common assumptions in all these models concerning the temperature are the following: it follows a predicted cycle, it moves around a seasonal mean, it is affected by global warming, it appears to have autoregressive changes, its volatility is higher in winter than in summer.

Early models were using AR(1) processes or continuous equivalents (Alaton *et al*., 2002; Davis, 2001; Cao and Wei, 2000). Others like Dornier and Querel (2000) and Moreno (2000) have suggested versions of a more general ARMA(p,q) model. Cabalero *et al.* (2002) have shown, however, that all these models fail to capture the slow time decay of the autocorrelations of temperature and hence lead to significant underpricing of weather options. Thus more complex models were proposed, like an Ornstein-Uhlenbeck process (Brody *et al*, 2002). Also in the noise part of the process, the Brownian noise was at first replaced by a fractional Brownian noise and then by a Levy process (Benth and Saltyte-Benth, 2005). A temperature Ornstein-Uhlenbeck process is:

$$dT(t) = dS(t) - \kappa(T(t) - S(t))dt + \sigma(t)dB(t) \tag{2.1}$$

where, $T(t)$ is the daily average temperature, $B(t)$ is a standard Brownian motion, $S(t)$ is a deterministic function modelling the trend and seasonality of the average temperature, while $\sigma(t)$ is the daily volatility of temperature variations. In Benth's and Saltyte-Benth's (2005) model both $S(t)$ and $\sigma^2(t)$ as a truncated Fourier series:

$$S(t) = a + bt + a_0 + \sum_{i=1}^{I_1} a_i \sin(2i\pi(t - f_i)/365) + \sum_{j=1}^{J_1} b_j \cos(2j\pi(t - g_j)/365) \tag{2.2}$$

$$\sigma^2(t) = c + \sum_{i=1}^{I_2} c_i \sin(2i\pi t/365) + \sum_{j=1}^{J_2} d_j \cos(2j\pi t/365) \tag{2.3}$$

From the Ito formula an explicit solution for (2.1) can be derived:

$$T(t) = s(t) + (T(t-1) - s(t-1))e^{-\kappa t} + \int_{t-1}^{t} \sigma(u)e^{-\kappa(t-u)}dB(u) \tag{2.4}$$

According to this representation $T(t)$ is normally distributed at $t$ and it is reverting to a mean defined by $S(t)$.

A discrete approximation to the Ito formula (2.4), which is the solution to the mean reverting Ornstein-Uhlenbeck process (2.1), is:

$$T(t+1) - T(t) = S(t+1) - S(t) - (1 - e^{-k})\{T(t) - S(t)\} + \sigma(t)\{B(t+1) - B(t)\} \tag{2.5}$$

which can be written as:

$$\tilde{T}(t+1) = a\tilde{T}(t) + \tilde{\sigma}(t)\varepsilon(t) \tag{2.6}$$

where

$$\tilde{T}(t) = T(t) - S(t) \tag{2.7}$$

$$a = e^{-k} \tag{2.8}$$

In order to estimate model (2.6) we need first to remove the trend and seasonality components from the average temperature series.

Zapranis & Alexandridis (2007a) used wavelet analysis in order to identify and remove the trend and the seasonal component of the daily average temperatures in Paris. Then a neural network modelled the seasonal variance of the residuals of eq. (2.6). In a more recent work, Zapranis & Alexandridis (2007b) used both wavelet analysis and neural network in order to identify the seasonal component and proved that this approach gives better results.

In this paper a different approached is adopted. The trend and the seasonality of monthly average temperatures is modelled and removed by a new class of neural networks, called wavelet networks. This method combines both the power of wavelet analysis and neural networks. Instead of identifying the seasonal component by an optical examination, which may lead to wrong conclusions, this is done automatically. Hence, equation (2.6) reduces to:

$$T(t) = \varphi\left(T(t-1)\right) + e_t \tag{2.9}$$

where $\varphi(\bullet)$ is estimated non-parametrically by a wavelet network and $T(t)$ now refers to monthly average temperatures.

## 3. Wavelet Neural Networks for Multivariate Process Modeling

### *3.1 Wavelets for application modeling*

In (Zapranis & Alexandridis; 2007a,b) we give a concise treatment of wavelet theory. Here the emphasis is in presenting the theory and mathematics of wavelet neural networks and thus we give only the very basic notions of wavelets. Very briefly, a *family* of wavelets is constructed by translations and dilations performed on a single fixed function called the *mother wavelet*. A wavelet $\psi_j$ is derived from its mother wavelet $\psi$ by the relation:

$$\psi_j(x) = \psi\left(\frac{x - m_j}{d_j}\right) = \psi(z_j) \tag{3.1}$$

where its translation factor $m_j$ and its dilation factor $d_j$ are real numbers ($d_j > 0$).

In the context of process modeling applications, the wavelets are determined either, by orthogonal wavelet decomposition or, according to space-frequency analysis of the data. Another approach is to construct a feedforward wavelet neural network, which serves as a representation of a family of parameterized non-linear wavelet functions. The translation and dilation factors are real numbers and they are considered as network weights.

Following Oussar *et al* (1998) we use as a mother wavelet the Gaussian function:

$$\psi(a) = -ae^{-\frac{1}{2}a^2} \tag{3.2}$$

which is a differentiable version of the Haar mother wavelet. Other mother wavelets can also be used.

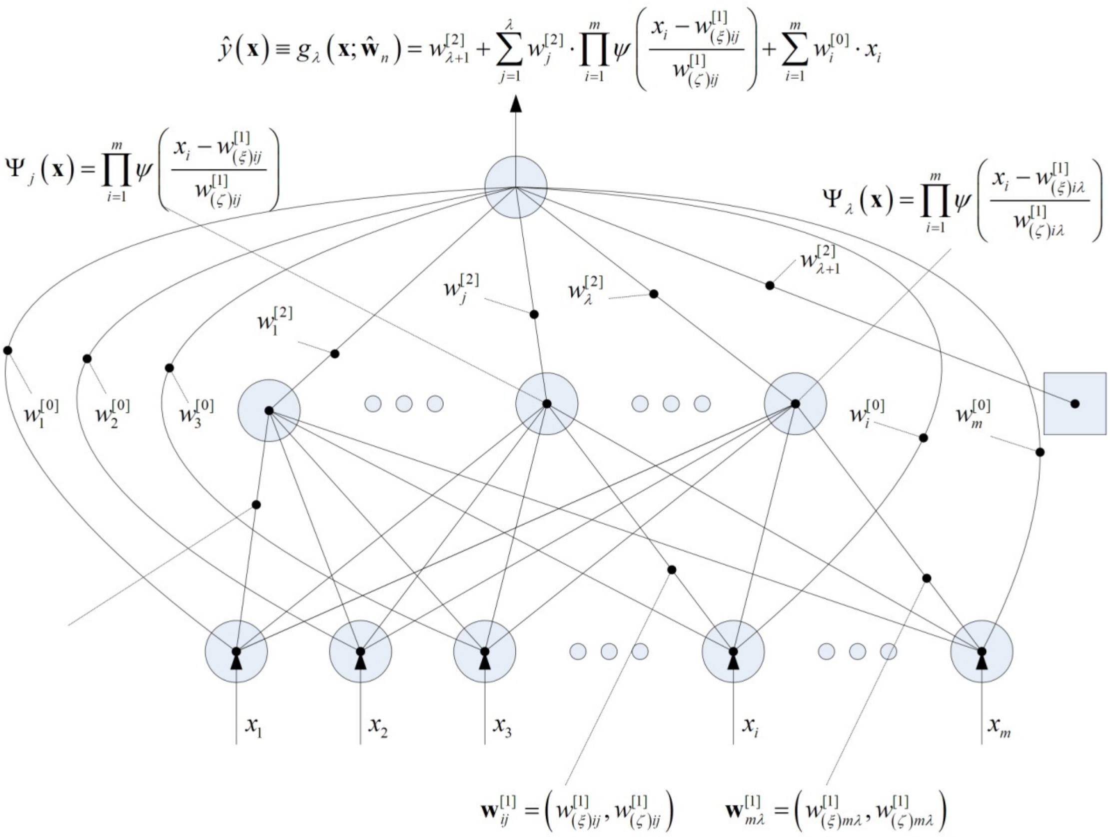


**FIG. 1.** A wavelet feedforward neural network.

*3.2 The structure of wavelet neural networks*

The structure of a single-hidden-layer feedforward wavelet network is given in figure 1. The network output is given by the following expression:

$$\hat{y}(\mathbf{x}) = w_{\lambda+1}^{[2]} + \sum_{j=1}^{\lambda} w_j^{[2]} \cdot \Psi_j(\mathbf{x}) + \sum_{i=1}^{m} w_i^{[0]} \cdot x_i \tag{3.3}$$

In that expression, $\Psi_j(\mathbf{x})$ is a multidimensional wavelet which is constructed by the product of $m$ scalar wavelets, x is the input vector, $m$ is the number of network inputs, $\lambda$ is the number of hidden units and $w$ stands for a network weight. The multidimensional wavelets are computed as follows:

$$\Psi_j(\mathbf{x}) = \prod_{i=1}^{m} \psi(z_{ij}) \tag{3.4}$$

where

$$z_{ij} = \frac{x_i - w_{(\xi)ij}^{[1]}}{w_{(\zeta)ij}^{[1]}} \tag{3.5}$$

In the above expression, $i = 1, \ldots, m$, $j = 1, \ldots, \lambda+1$ and the weights $w$ correspond to the translation ( $w_{(\xi)ij}^{[1]}$ ) and the dilation ( $w_{(\zeta)ij}^{[1]}$ ) factors. The complete vector of the network parameters comprises:

$$\mathbf{w} = \left( w_i^{[0]}, w_j^{[2]}, w_{\lambda+1}^{[2]}, w_{(\xi)ij}^{[1]}, w_{(\zeta)ij}^{[1]} \right) \tag{3.6}$$

Here we have to note that the families of multidimensional wavelets preserve the universal approximation property that characterize neural networks.

*3.3 Training a wavelet network with back-propagation*

The error $e_p$ for pattern $p$ is simply the difference between the target ($y_p$) and the network output ($y_p$ hat in the following expression). By squaring and multiplying by ½ we take the pairwise error $E_p$ which is used in network training:

$$E_p = \frac{1}{2}\left(y_p - \hat{y}_p\right)^2 = \frac{1}{2} e_p^2 \tag{3.7}$$

The training minimizes the quadratic cost functional (or loss function):

$$L_n = \sum_{p=1}^{n} E_p = \frac{1}{2}\sum_{p=1}^{n} e_p^2 \tag{3.8}$$

The minimization is performed by iterative gradient-based methods. In our implementation we have used ordinary back-propagation (less fast but less prone to sensitivity to initial condition than higher order alternatives). The updating of the parameters is performed by the following (delta) rule:

$$w_t = w_{t-1} + \eta \left( -\frac{\partial L_n}{\partial w_{t-1}} \right) + \kappa \left( w_t - w_{t-1} \right) \tag{3.9}$$

where $\eta$ is the learning rate. Also a momentum term, defined by $\kappa$, is induced which increases the training speed. The learning rate and momentum speed take values between 0 and 1.

The partial derivative of the cost function with respect to a weight $w$ is given by:

$$\frac{\partial E_p}{\partial w} = \frac{\partial E_p}{\partial \hat{y}_p}\frac{\partial \hat{y}_p}{\partial w} = -\left(y_p - \hat{y}_p\right)\frac{\partial \hat{y}_p}{\partial w} = -e_p \frac{\partial \hat{y}_p}{\partial w} \tag{3.10}$$

The equivalent expression for the cost functional is:

$$\frac{\partial L_n}{\partial w} = \sum_{p=1}^{n} \frac{\partial E_p}{\partial w} = -e_p \sum_{p=1}^{n} \frac{\partial \hat{y}_p}{\partial w} \tag{3.11}$$

The partial derivatives of the output of the network with respect to the bias term, the direct connections, the translation parameters and the dilation parameters are given by the following equations:

*partial derivatives w.r.t. the bias term*

$$\frac{\partial \hat{y}_p}{\partial w_{\lambda+1}^{[2]}} = 1 \tag{3.12}$$

*partial derivatives w.r.t. the direct connections*

$$\frac{\partial \hat{y}_p}{\partial w_j^{[2]}} = \Psi_j(\mathbf{x}) \quad j = 1,...,\lambda \tag{3.13}$$

*partial derivatives w.r.t. the translation parameters*

$$\begin{aligned} \frac{\partial \hat{y}_p}{\partial w_{(\xi)ij}^{[1]}} &= \frac{\hat{y}_p}{\partial \Psi_j(\mathbf{x})} \cdot \frac{\partial \Psi_j(\mathbf{x})}{\partial \psi(z_{ij})} \cdot \frac{\partial \psi(z_{ij})}{\partial z_{ij}} \cdot \frac{\partial z_{ij}}{\partial w_{(\xi)ij}^{[1]}} \\ &= w_j^{[2]} \cdot \psi(z_{1j}) \cdots \psi'(z_{ij}) \cdots \psi(z_{mj}) \cdot \frac{-1}{w_{(\zeta)ij}^{[1]}} \\ &= -\frac{w_j^{[2]}}{w_{(\zeta)ij}^{[1]}} \psi(z_{1j}) \cdots \psi'(z_{ij}) \cdots \psi(z_{mj}) \end{aligned} \tag{3.14}$$

*partial derivatives w.r.t. the dilation parameters*

$$\begin{aligned} \frac{\partial \hat{y}_p}{\partial w_{(\xi)ij}^{[1]}} &= \frac{\hat{y}_p}{\partial \Psi_j(\mathbf{x})} \cdot \frac{\partial \Psi_j(\mathbf{x})}{\partial \psi(z_{ij})} \cdot \frac{\partial \psi(z_{ij})}{\partial z_{ij}} \cdot \frac{\partial z_{ij}}{\partial w_{(\xi)ij}^{[1]}} \\ &= w_j^{[2]} \cdot \psi(z_{1j}) \cdots \psi'(z_{ij}) \cdots \psi(z_{mj}) \cdot \frac{-1}{w_{(\zeta)ij}^{[1]}} \\ &= -\frac{w_j^{[2]}}{w_{(\zeta)ij}^{[1]}} \psi(z_{1j}) \cdots \psi'(z_{ij}) \cdots \psi(z_{mj}) \end{aligned} \tag{3.15}$$

where the derivative of the mother wavelet is

$$\psi'(z_{ij}) = \frac{\partial \psi(z_{ij})}{\partial z_{ij}} = e^{-\frac{1}{2}z_{ij}^2}\left(z_{ij}^2 - 1\right) \quad (3.16)$$

*3.4 Initialization of the network parameters*

In contrast to neural networks using sigmoid functions selecting initial values of the dilation and translation parameters randomly may not be suitable (Oussar *et al.*; 1998). A wavelet is a waveform of effectively limited duration that has an average value of zero and localized properties. Moreover, the selection of initial values are important because it affects the speed of training and approximation to the global or local minimum (Postalcioglu & Becerikli; 2007). Zhang & Benveniste (1992) propose the following initialization for the translation and dilation parameters.

$$w_{(\xi)ij}^{[1]} = 0.5\left(N_i + M_i\right) \quad (3.17)$$

$$w_{(\zeta)ij}^{[1]} = 0.2\left(M_i - N_i\right) \quad (3.18)$$

where $M_i$ and $N_i$ are defined as the maximum and minimum of input $x_i$

$$M_i = \max_{p=1,\ldots,n}\left(x_{ip}\right) \quad (3.19)$$

$$N_i = \min_{p=1,\ldots,n}\left(x_{ip}\right) \quad (3.20)$$

Although, more complex initialization methods have been proposed, (Oussar & Dreyfus, 2000) the previous heuristic method is simple and efficient. The initialization of the direct connections $w_i^{[0]}$ and the weights $w_j^{[2]}$ is less important and they are initialized in small random values between 0 and 1.

*3.5 Stopping conditions for training*

The weights $w_i^{[0]}$, and $w_j^{[2]}$ and parameters $w_{(\xi)ij}^{[1]}$ and $w_{(\zeta)ij}^{[1]}$ are trained for approximating the target function. When one of the following criteria is met - the cost function reaches a fixed lower bound or the variations of the gradient or the variations of the parameters reaches a lower bound or the number of iterations reaches a fixed maximum, whichever is satisfied first – the training is stopped.

## 4. Examples of Wavelet Networks Applications to Simulated Time-Series

*4.1 Analyzing a static function.*

So far, wavelets have being used in a variety of applications such as signal de-noising, density estimation, variance-covariance estimation and signal compression. Wavelets are able, as it has already been mentioned, to capture changes and events in time-series that are not directly observable.

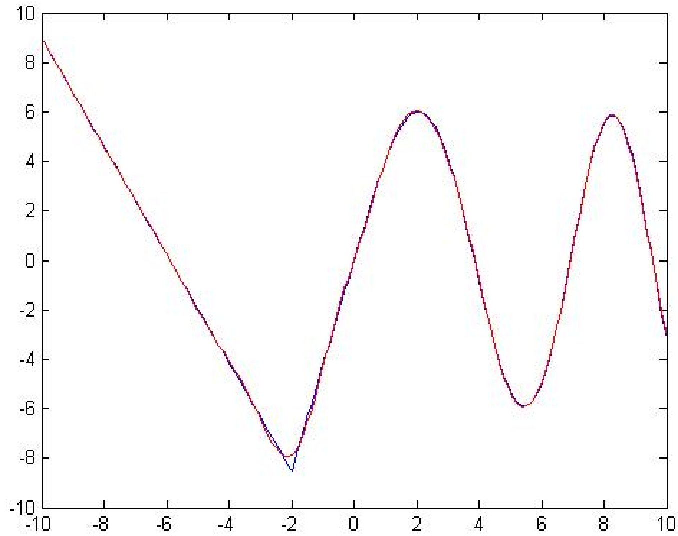


**FIG. 2.** Nonlinear static model and approximation obtained by a W.N. (red line).

On the other hand, neural networks have the ability to model non-linear processes without any assumption of the underlying function that produces the signal. In this section it is shown that the combination of these two methods, wavelet networks, have better results than neural networks or wavelet transform alone.

As a first example a static model given by (4.1) is being used. In $f(x)$ there are 2 breakpoints as well as changes in the function that are hard to model. This example proposed first by Zhang & Benveniste (1992). As shown in that paper wavelet networks can produce a better approximation of the function $f(x)$ than a neural network with sigmoid activation function or a wavelet decomposition. Moreover it was shown that wavelet networks need less parameters to estimate, less training time and less hidden units than the neural networks and the wavelet transform.

$$f(x)\begin{cases} -2.186x-12.864 & x\in[-10,-2] \\ 4.246x & x\in[-2,0] \\ 10e^{-0.05x-0.5}\sin\left(x\left(0.03x+0.7\right)\right) & x\in[0,10] \end{cases} \tag{4.1}$$

The function $f(x)$ and the approximation produced by the wavelet network is shown in figure 2. A wavelet network with 10 hidden units and 1 hidden layer was used which corresponds to a network with 32 parameters. The mother wavelet is given, as in the rest of the paper, by the first derivative of a Gaussian function as shown in (3.2)

First, the data $x$ transformed into the [-1,1] domain. For the training procedure a set of 200 uniformly sampled points used. The learning rate chosen to be 0.1 and the momentum term 0.3.

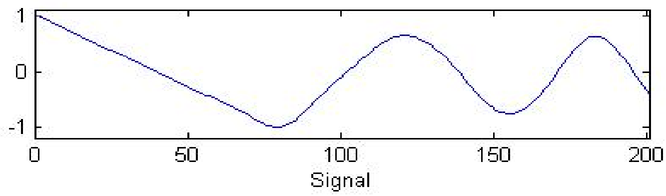


**FIG. 3.** Synthesized Signal.

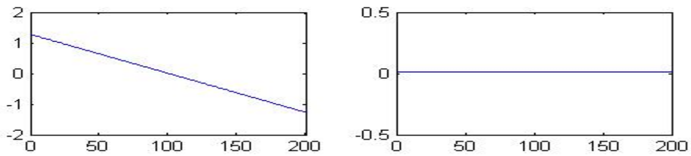

**FIG. 4.** The input $x$ and the bias term.

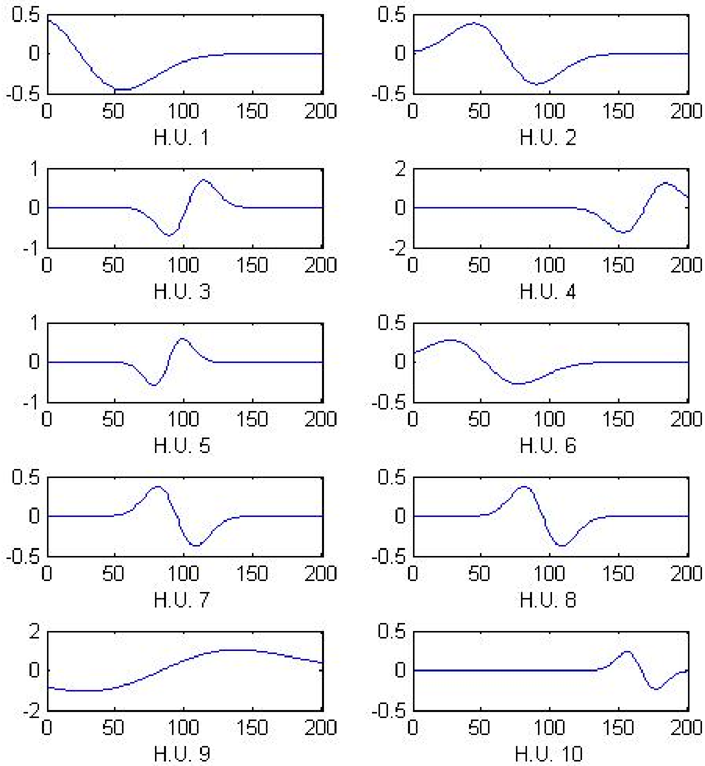

**FIG. 5.** Outputs of each hidden unit.

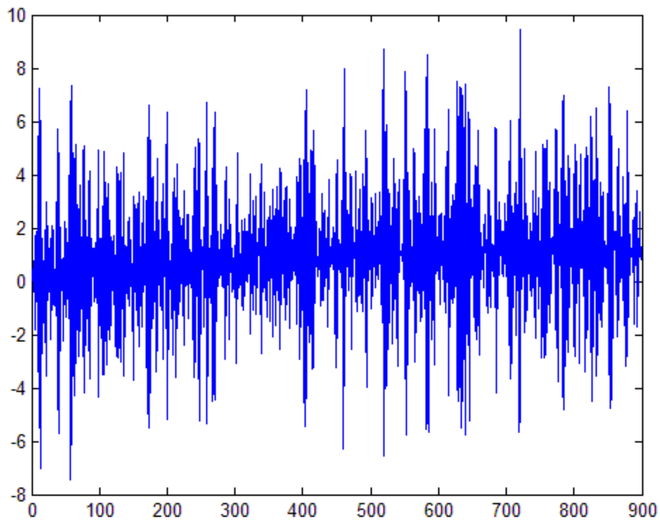


**FIG. 6.** Simulated time-series which consists of an upward trend plus an AR(3) noise component.

The training of the network stopped when the mean square error was less than the maximum error threshold, 0.0001. It is clear that wavelet networks succeeded in approximating (4.1).

Figures 4 show the influence of the bias term and input $x$ to the network output while figure 5 show the output of each hidden unit. In figure 3 the synthesized signal of figures 4 and 5 can be found.

*4.2 Analyzing a Simulated Signal: Upward Trend plus an AR(3) Noise Component*

Next, suppose that the signal $T(t)$ consists of a deterministic underlying component $\varphi(t)$ and a noise part $\varepsilon(t)$:

$$T(t) = \varphi(t) + \varepsilon(t) \tag{4.2}$$

In this example the signal consists of an upward trend plus a colored noise. The colored noise is produced by an AR(3) process:

$$z(t) = -1.5z(t-1) - 0.75z(t-2) - 0.125z(t-3) + 0.5 + \varepsilon(t) \tag{4.3}$$

where $\varepsilon(t)$ are i.i.d. $N(0,1)$.

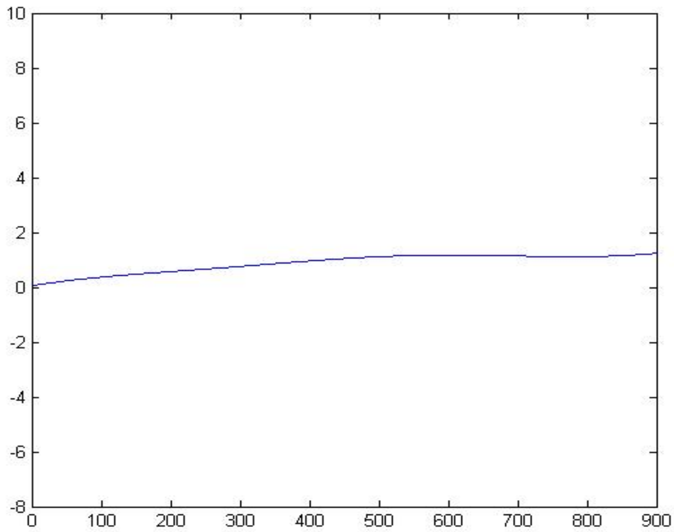


**FIG. 7.** The denoised signal of the simulated time-series which consists of an upward trend plus an AR(3) noise component.

The upward trend is produced by an upward slope, as follows:

$$\varphi(t) = \begin{cases} t/500 + z(t) & \text{t<500} \\ 1 + z(t) & \text{t} \geq 500 \end{cases} \tag{4.4}$$

As it can be observed in the figure 6, the signal seems like noise. None inference can be drawn for its characteristics.

As shown by Zapranis & Alexandridis (2006) wavelets are capable of extracting the underlying signal $\varphi(t)$. This was done using a discrete wavelet transform. In contrast, in this paper a non-orthogonal wavelet frame given by (3.2) is used.

Wavelets networks succeeded to remove the noise and extract the denoised signal, which is depicted as a solid smooth line in figure 7. A network with 2 hidden units was used, which implies that the number of estimated parameters is 8. The learning rate was selected to be 0.1 and the momentum term was selected to be 0.3. The wavelet network not only were able to extract the de-noised signal but also succeeded in distinguishing the breakpoint with good precision.

*4.3 Analyzing a Geometric Brownian Motion*

Many financial pricing models are based on the Geometric Brownian Motion. A GBM is produced by the following model:

$$X(t) = X(0)e^{(\mu - \frac{1}{2}\sigma^2)T + \sigma\varepsilon\sqrt{T}} \tag{4.5}$$

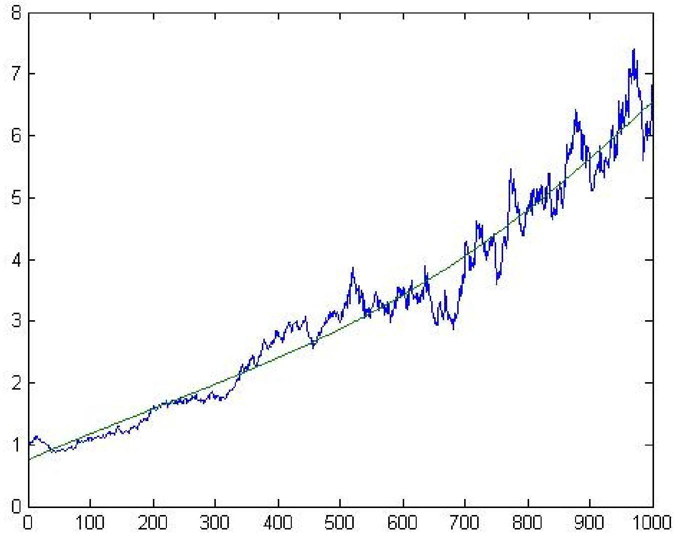


**FIG. 8.** A geometric Brownian motion (1000 steps) with a jump in volatility at $t = 500$, where the initial volatility $\sigma_1 = 0.5$ doubles to $\sigma_2 = 1$.The smooth line represents the exponential upward trend captured by the wavelet network.

A simple GBM depends on the mean $\mu$ and the volatility $\sigma$, so it is essential to know whether and when, one or both parameters change. Here we examine, if wavelet networks can approximate the exponential underlying function and if they can capture changes in the parameters of the GBM model (a simulated path of a GBM process is represented in figure 8).

The volatility of this path changes at $t = 500$ from $\sigma_1 = 0.5$ to $\sigma_2 = 1$. After applying the wavelet network the approximation in figure 4 shows clearly the de-noised signal (smooth line), which is the $e^x$ function. Zapranis & Alexandridis (2006) showed that the wavelet decomposition also captured the volatility change with a very good precision. Observing figure 9 we conclude on the same result. It is clear that the increase of the random part of the GBM is clearly reflected near $t = 510$, while it actually doubles at $t = 500$, i.e.,

$$\sigma_2\varepsilon\sqrt{T} = 2\sigma_1\varepsilon\sqrt{T} \tag{4.6}$$

In this paper we have shown that wavelet networks work better than the classical neural networks or the wavelet transform. We applied wavelet networks in monthly average temperatures in order to remove the seasonal component and any possible trends from the data. The main advantage of this method is that wavelet networks remove the seasonal component (2.2) and the seasonal variance (2.3) automatically in one step. This method avoids any wrong conclusions by the optical examination of the wavelet analysis proposed by Zapranis & Alexandridis (2007b). The drawback is that we lose information of the dynamics of the temperature.

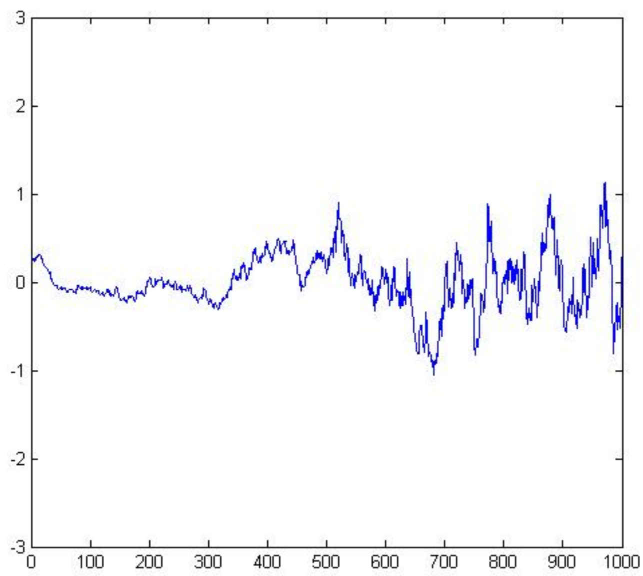


**FIG. 9.** The change in volatility after the original signal removed by a wavelet network.

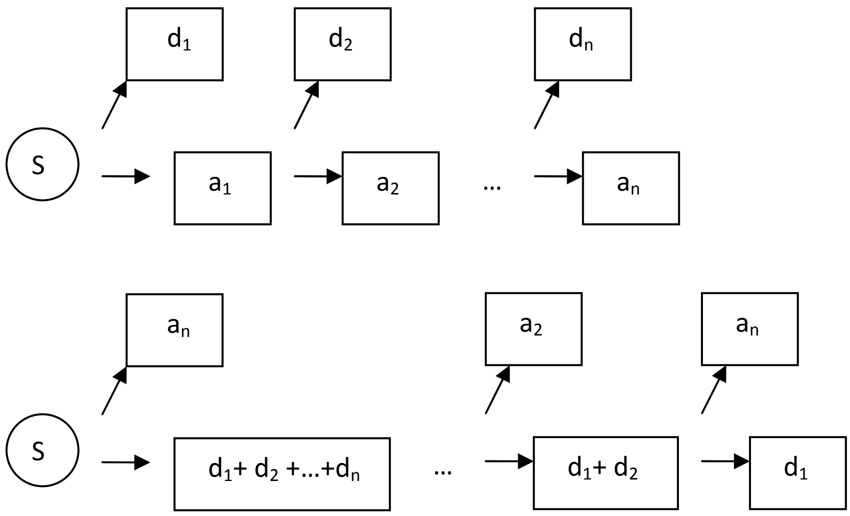


**FIG. 10.** Decomposition of wavelet analysis (top) and wavelet network (bottom).

Figure 10 shows the decomposition of wavelet analysis and wavelet network. Wavelet analysis decompose the signal in a series of approximations and details. On the other hand, wavelet networks produce only one output $a_n$. If a second wavelet network used in the remaining signal ($d_1$+ $d_2$+…+$d_n$) approximation $n$-1 can be obtained. The rest approximations and details can produced by repeating the same methodology. It is clear that this method is much more time consuming and computational heavy.

## 5. Weather Data

In this section real time weather data will be used in order to select and validate our model. The data consists 3650 values, corresponding to the average daily temperatures of 10 years (1991-2000) in Paris, France. The 29th of February was removed from the data. Then the cumulative monthly average temperatures were calculated, resulting to 120 values. Finally, the model was validated in data consisting of 1 year of daily average temperatures (2000-2001).

The most important step in model selection is to select the correct number of hidden units. Zapranis & Alexandridis (2007b) used wavelet transform in order to identify the dynamics of the seasonal component of the temperature. Then a neural network with 10 inputs (the different parts of the seasonal component), one hidden layer, one bias term and 10 hidden units estimated. The number of parameters to estimate is 121. In this paper, the wavelet network consists of one input, $x_1$, representing the average temperatures, one hidden layer, one bias term and one output. In wavelet network framework in order to have the same number of estimated parameters, a network with 40 hidden units must be estimated. Instead, we train the wavelet network for different numbers of hidden units. Then, we validate the different models and we select the one that gives the smallest out-of-sample mean square error. More precisely we train our wavelet network with hidden units varying from 25 to 80. Our experience suggests that larger models will overfit the data while smaller models are not strong enough.

Previous works suggests that weather forecasts beyond 10 days are not accurate. Similar, for averaging periods, monthly or seasonal forecasts must be used. Bigger periods induce significant errors. Figure 12 shows the out-of-sample mean square error for 1,3,6,9 and 12 months.

In this section a wavelet network with learning rate 0.1 and momentum 0.3 was used. The mother wavelet function is given by (3.2) which is the first derivative of a Gaussian function.

Using a wavelet network with 30 hidden units for 1 month forecast, the mean square error is 0.0000672 while the network with 55 hidden units is very large, 2.3498. For 3 months forecast a wavelet network with 50 hidden units gives the smallest error, 0.0677. Smaller models give acceptable results. For 30 hidden units the mean square error is 0.1653. The network with 35 and hidden units give a large error, 1.6397 and 0.8150 respectively.

For 6 months ahead, the wavelet networks with 25 and 30 hidden units give the best forecasts. The mean square errors are 0.6202 and 1.1293 respectively, and they are significant smaller than any other model (e.g. 50 hidden units, error 2.0621).

For larger periods of forecasts the mean square error increases significantly. For 9 months forecast the best model is, again, the one with 30 hidden units and the mean square error is 7.1502. The smallest out-of-sample mean square error for 12 errors is 9.6706 and corresponds to 30 hidden units while the second smallest mean square error is 12.5488, 50 hidden units.

In almost all cases a large spike is clear when a model with 35 hidden units is used. The error is very large in all cases. This is happening probably because the wavelet network finds a local minimum instead of a global one. This is happening because of the initialization of the weights. Different initial conditions for the weights will give a results similar to the models with 60 and 70 hidden units. A better approach is to estimate the wavelet network for a large number of different initial conditions for each number of hidden units. This method guarantees that wavelet network will find the global minimum solution.

Examining figure 12 we conclude that the best model is a wavelet network with 30 hidden units since it has the best accuracy combining the smallest complexity.

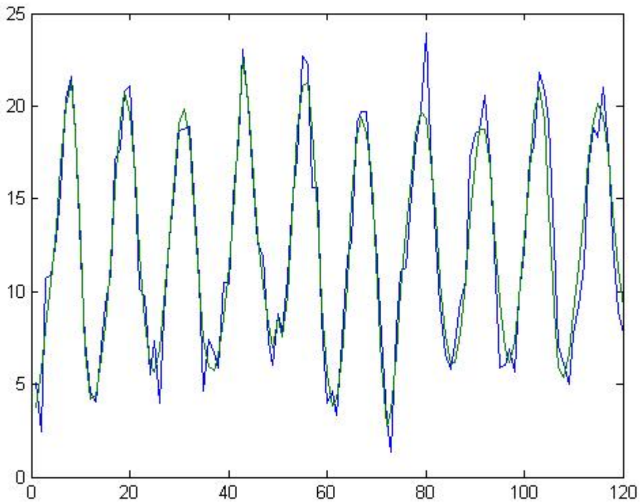


**FIG. 11.** Real time data and model output (green line)

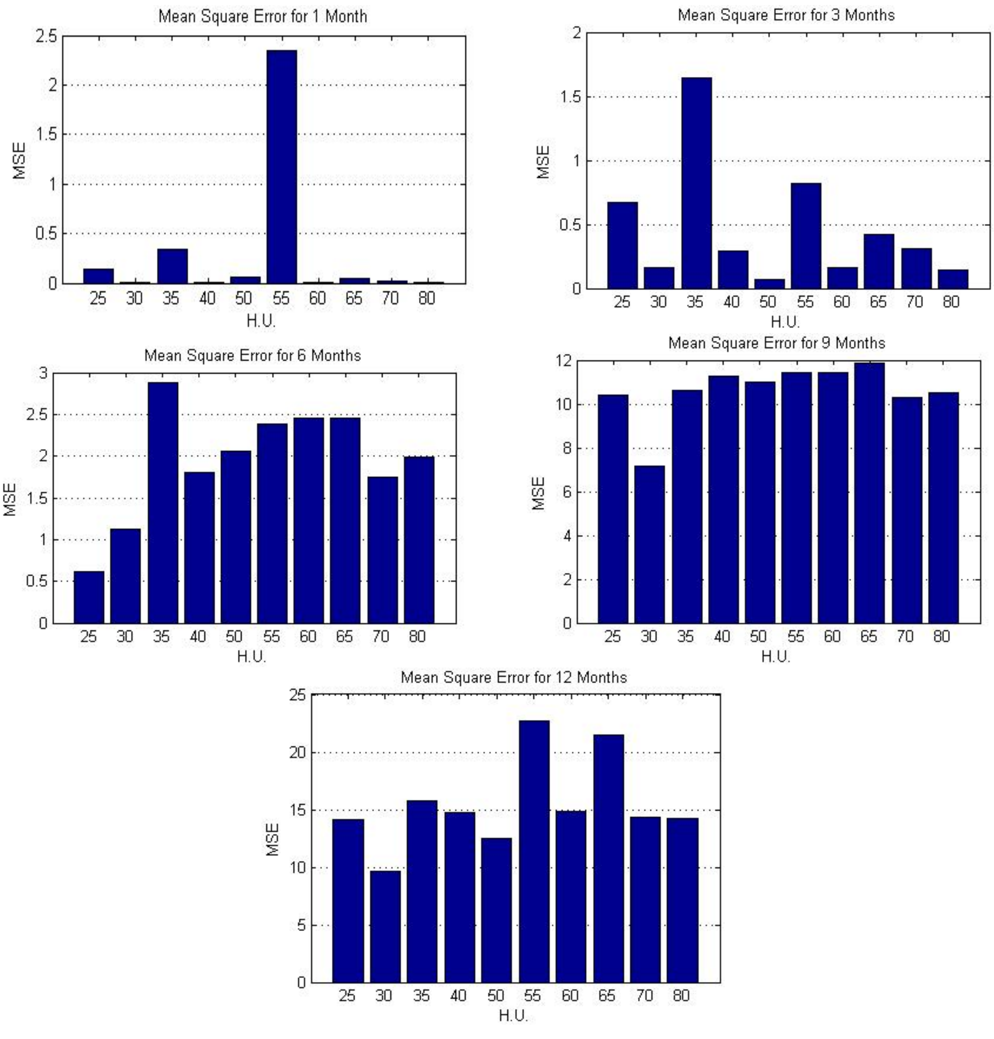


**FIG. 12.** Out-of-Sample Mean Square Error For 1,3,6,9 and 12 months forecast.

Such a network corresponds to 92 parameters and it is significant smaller that the neural network used by Zapranis & Alexandridis (2007b). Figure 11 represents the monthly average temperature and the fitted model output for a wavelet network using 30 hidden units.

## 6. Temperature Derivatives Pricing

The list of traded contracts in weather derivatives market is extensive and constantly evolving. In Europe, CME weather contracts for the summer months are based on an index of Cumulative Average Temperature (CAT). The CAT index is the sum of the daily average temperatures over the contract period. The average temperature is measured as the simple average of the minimum and maximum temperature over one day. Recall, that in this paper *T(t)* represents monthly average temperatures. Hence, the value of a CAT index for the time interval $[\tau_1,\tau_2]$ is given by the following expression:

$$\sum_{t=\tau_1}^{\tau_2} T(t)h(t) \qquad (6.1)$$

where $h$ is the number of days of month $t$ and the temperature is measured in degrees of Celsius. The time period of the CAT index is monthly or seasonal. In USA, CME weather derivatives are based on Heating Degree Days (HDD) or Cooling Degree Days (CDD) index. A HDD is the number of degrees by which daily temperature is below a base temperature, while a CDD is the number of degrees by which the daily temperature is above the base temperature,

i.e., Daily HDD = max (0, base temperature – daily average temperature),
Daily CDD = max (0, daily average temperature – base temperature).

The base temperature is usually 65 degrees Fahrenheit in the US and 18 degrees Celsius in Europe and Japan. HDDs and CDDs are usually accumulated over a month or over a season. To calculate then we simply add up their daily values for that period. For the two Japanese cities, weather derivatives are based on the Pacific Rim index. The Pacific Rim index is simply the average of the CAT index over the specific time period. At the end of 2006, at CME were traded weather derivatives for 18 US cities[3], 9 European cities[4], 2 Japanese cities[5], as well as seasonal strip and frost contracts.

Our aim is to give a mathematical expression for the CAT future price. It is clear that the weather derivative market is an incomplete market. Cumulative average temperature contracts are written on a temperature index which is not a tradable or storable asset. In order to derive the pricing formula, first we must find a risk-neutral probability measure $Q \sim P$ ,where all assets are martingales after discounting. In the case of weather derivatives any equivalent meas-

[3] Atlanta, Baltimore, Boston, Chicago, Cincinnati, Dallas, Des Moines, Detroit, Houston, Kansas City, Las Vegas, Minneapolis-St. Paul, New York, Philadelphia, Portland, Sacramento, Salt Lake City, Tucson.
[4] Amsterdam, Barcelona, Berlin, Essen, London, Madrid, Paris, Rome, Stockholm.
[5] Tokyo, Osaka.

ure $Q$ is a risk neutral probability. If $Q$ is the risk neutral probability and $r$ is the constant compounding interest rate then the arbitrage free future price of a CAT contract at time $t \le \tau_1 \le \tau_2$ is given by:

$$e^{-r(\tau_2 - t)} \mathrm{E}_Q \left[ \sum_{t=\tau_1}^{\tau_2} T(t)h(t) - F_{CAT}(t, \tau_1, \tau_2) \mid \mathfrak{F}_t \right] = 0 \quad (6.2)$$

and since $F_{CAT}$ is $\mathfrak{F}_t$ adapted we derive the price of a CAT futures to be

$$F_{CAT}(t, \tau_1, \tau_2) = \mathrm{E}_Q \left[ \sum_{t=\tau_1}^{\tau_2} T(t)h(t) \mid \mathfrak{F}_t \right] \quad (6.3)$$

*T(t)* is the output of the wavelet network and it is known :

$$\mathrm{T}(\mathbf{x}) = \hat{y}(\mathbf{x}) = w_{\lambda+1}^{[2]} + \sum_{j=1}^{\lambda} w_j^{[2]} \cdot \Psi_j(\mathbf{x}) + \sum_{i=1}^{m} w_i^{[0]} \cdot x_i \quad (6.4)$$

$$F_{CAT}(t, \tau_1, \tau_2) = \sum_{t=\tau_1}^{\tau_2} T(t)h(t) = T(\mathbf{x}) \cdot h(\mathbf{x}) \quad (6.5)$$

## 7. Conclusions and Further Work

In this paper we have used a wavelet network in order to model and remove the seasonal cycle as well as any possible trends, singularities or jumps of the temperature process. Moreover, we have given a complete framework for structuring and training feed forward wavelet neural networks via back-propagation. Our findings suggest that wavelet networks can model the temperature process very well and consequently they can be used for predicting the CAT index. As we have shown, applying wavelet networks simplifies significantly the mathematics of weather derivatives pricing, since no particular functional form is assumed.

Our initial results are very promising. However, an explicit formulation that connects the structure of the wavelet network to the ordinary wavelet decomposition would increase significantly the information that we can extract from the trained network. Furthermore, a complete statistical framework for model selection and identification, specifically designed for this type of networks, would add impetus to their successful everyday use in the context of application modelling and specifically weather derivatives pricing. These two topics constitute the main focus of our future work.

## 8. References


P. Alaton, B. Djehince & D. Stillberg "On Modeling and Pricing Weather Derivatives", Applied Mathematical Finance, 9, pp 1-20, 2000.

F.E. Benth & J. Saltyte-Benth "The Volatility Of Temperature And Pricing Of Weather Derivatives" *Pure Mathematics,* E-Print, 12, Department of Mathematics. University of Oslo, 2005.

F.E. Benth & J. Saltyte-Benth "Stochastic Modeling of Temperature Variations With a View Towards Weather Derivatives", *Applied Mathematical Finance*, 12 (1), pp 53-85, 2005.

A. Brix, S. Jewson & C. Ziehmann. *Weather Derivative Modeling and Valuation*, Risk Books 2002.

C.D. Brody, J. Syroka & M. Zervos "Dynamical Pricing of Weather Derivatives" *Quantitative Finance*, 2, pp. 189-198, 2002.

R. Caballero, S. Jewson & A. Brix "Long Memory in Surface Air Temperature: Detection, Modeling and Application to Weather Derivative Valuation" *Climate Research*, 21, pp127-140, 2002.

M. Cao & J. Wei "Pricing The Weather" *Risk,* 13 (5), 2000.

M. Davis "Pricing Weather Derivatives by Marginal Value" *Quantitative Finance*, 1, pp.305-308, 2001.

I. Daubechies. *Ten Lectures on Wavelets (CBMS-NSF Regional Conference Series in Applied Mathematics)*, 1992.

F. Dornier & M. Querel "Caution to the Wind" *Energy Power Risk Management*, pp. 30-32, 2000.

R. Gao & L.H. Tsoukalas. "Neural-wavelet Methodology for Load Forecasting".*Journal of Intelligent & Robotic Systems.*31,pp 149-157.2001.

S. Jewson & A. Brix *Weather Derivative Valuation*, Cambridge University Press, 2005.

T. Khayamian, A.A. Ensafi, R. Tabaraki, M. Esteki. "Principal Component-Wavelet Networks as a New Multivariate Calibration Model". *Analytical Letters*. 38 (9)., pp 1477-1489. 2005

S.G. Mallat *A Wavelet Tour of Signal Processing.* San Diego: Academic Press, 1999.

M. Moreno "Riding the Temp" *Futures and Options World*, 11, 2000

Y. Oussar & G. Dreyfus. "Initialization by Selection for Wavelet Network Training". *Neurocomputing.* 34, pp 131-143. 2000.

Y Oussar, I. Rivals, L. Presonnaz, G. Dreyfus. "Training Wavelet Networks for Nonlinear Dynamic Input Output Modeling". *Neurocomputing.* 20, pp 173-188. 1998.

S. Postalcioglu & Y. Becerikli. "Wavelet Networks for Nonlinear System Modeling". *Neural Computing & Applications*. 16, pp 443-441, 2007

A. Subasi, A. Alkan, E. Koklukaya, M.K. Kiymik. "Wavelet Neural Network Classification of EEG Signals by Using AR Model With MLE Pre-processing". *Neural Networks.* 18. pp 985-997. 2005.

H. Szu, B. Telfer, S. Kadambe. "Neural Network adaptive wavelets for signal representation and classification". *Opt. Engineering.* 31. pp 1907-1916. 1992.

M. Thuillard. "A Review of Wavelet Networks, Wavenets, Fuzzy Wavenets and Their Applications". In proc. of ESIT, Aachen, Germany, 14-15 September 2000.

P. Wojtaszczyk. *A Mathematical Introduction to Wavelets.* Cambridge: Cambridge University Press, 1997.

J. Xu & D.W.C. Ho. "A Basis Algorithm for Wavelet Neural Networks". *Neurocomputing.* 48. pp 681-689. 2002.

A. Zapranis & A. Alexandridis "Weather Analysis & Weather Derivative Pricing" in proc. HFAA, Thessaloniki, December 2006.

A. Zapranis & A. Alexandridis "Weather Derivatives Pricing: Modeling The Seasonal Residual Variance of an Ornstein-Uhlenbeck Temperature Process With Neural Networks" *Neurocomputing* (accepted, to appear), 2007a.

A. Zapranis & A. Alexandridis "Modelling Temperature Time Dependent Speed of Mean Reversion in the Context of Weather Derivative Pricing". *Applied Mathematical Finance.* (submitted October 2007), 2007b.

A. Zapranis & A.-P. Refenes, *Principles of Neural Model Identification, Selection and Adequacy: With Applications to Financial Econometrics*, Springer-Verlag, 1999.

Q. Zhang & A. Benveniste. "Wavelet Networks". *IEEE Trans. Neural Networks.* 3 (6), pp 889-898, 1992.