International Journal of Forecasting 33 (2017) 21–47

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

A comparison of wavelet networks and genetic programming in the context of temperature derivatives

Antonis K. Alexandridis [a,*], Michael Kampouridis [b], Sam Cramer [b]

[a] School of Mathematics, Statistics and Actuarial Science, University of Kent, United Kingdom
[b] School of Computing, University of Kent, United Kingdom

ARTICLE INFO

Keywords:
Weather derivatives
Wavelet networks
Temperature derivatives
Genetic programming
Modelling
Forecasting

ABSTRACT

The purpose of this study is to develop a model that describes the dynamics of the daily average temperature accurately in the context of weather derivatives pricing. More precisely, we compare two state-of-the-art machine learning algorithms, namely wavelet networks and genetic programming, with the classic linear approaches that are used widely in the pricing of temperature derivatives in the financial weather market, as well as with various machine learning benchmark models such as neural networks, radial basis functions and support vector regression. The accuracy of the valuation process depends on the accuracy of the temperature forecasts. Our proposed models are evaluated and compared, both in-sample and out-of-sample, in various locations where weather derivatives are traded. Furthermore, we expand our analysis by examining the stability of the forecasting models relative to the forecasting horizon. Our findings suggest that the proposed nonlinear methods outperform the alternative linear models significantly, with wavelet networks ranking first, and that they can be used for accurate weather derivative pricing in the weather market.

1. Introduction

This paper uses wavelet networks (WNs) and genetic programming (GP) to describe the dynamics of the daily average temperature (DAT), in the context of weather derivatives pricing. The proposed methods are evaluated both in-sample and out-of-sample against various linear and non-linear models that have been proposed in the literature.

Recently, a new class of financial instruments, known as "weather derivatives" has been introduced. Weather derivatives are financial instruments that can be used by organizations or individuals to reduce the risk associated with adverse or unexpected weather conditions, as part of a risk management strategy (Alexandridis & Zapranis, 2013a). Just like traditional contingent claims, the payoffs of which depend upon the price of some fundamental, a weather derivative has an underlying measure such as rainfall, temperature, humidity, or snowfall. However, they differ from other derivatives in that the underlying asset has no value and cannot be stored or traded, but at the same time must be quantified in order to be introduced in the weather derivative. To do this, temperature, rainfall, precipitation, or snowfall indices are introduced as underlying assets. However, the majority of the weather derivatives have a temperature index as the underlying asset. Hence, this study focuses only on temperature derivatives.

Studies have shown that about $1 trillion of the US economy is exposed directly to weather risk (Challis, 1999;

* Corresponding author.
E-mail address: A.Alexandridis@kent.ac.uk (A.K. Alexandridis).

http://dx.doi.org/10.1016/j.ijforecast.2016.07.002

Hanley, 1999). Today, weather derivatives are used for hedging purposes by companies and industries whose profits can be affected adversely by unseasonal weather, and for speculative purposes by hedge funds and others who are interested in capitalising on these volatile markets. Weather derivatives are used to hedge volume risk, rather than price risk.

It is essential to have a model that (i) describes the temperature dynamics accurately, (ii) describes the evolution of the temperature accurately, and (iii) can be used to derive closed form solutions for the pricing of temperature derivatives. In complete markets, the cash flows of any strategy can be replicated by a synthetic one. In contrast, the weather market is an incomplete market, in the sense that the underlying asset has no value and cannot be stored, and hence, no replicating portfolio can be constructed. Thus, modelling and pricing the weather market are challenging issues. In this paper, we focus on the problem of temperature modelling. It is of paramount importance to address this problem before doing any investigation into the actual pricing of the derivatives.

There has been quite a significant amount of work done to date in the area of modelling the temperature over a certain time period. Early studies tried to model different temperature indices directly, such as heating degree days (HDD) or the cumulative average temperature (CAT).[1] Following this path, a model is formulated so as to describe the statistical properties of the corresponding index (Davis, 2001; Dorfleitner & Wimmer, 2010; Geman & Leonardi, 2005; Jewson, Brix, & Ziehmann, 2005). One obvious drawback of this approach is that a different model must be used for each index when formulating the temperature index, such as HDD, as a normal or lognormal process, meaning that a lot of information both in common and extreme events is lost; e.g., HDD is bounded by zero (Alexandridis & Zapranis, 2013a).

More recent studies have utilized dynamic models, which simulate the future behavior of DAT directly. The estimated dynamic models can be used to derive the corresponding indices and price various temperature derivatives (Alexandridis & Zapranis, 2013a). In principle, using models for daily temperatures can lead to more accurate pricing than modelling temperature indices. The continuous processes used for modeling DAT usually take a mean-reverting form, which has to be discretized in order to estimate its various parameters.

Most models can be written as nested forms of a mean-reverting Ornstein–Uhlenbeck (O–U) process. Alaton, Djehince, and Stillberg (2002) propose the use of an O–U model with seasonalities in the mean, using a sinusoidal function and a linear trend in order to capture urbanization and climate changes. Similarly, Benth and Saltyte-Benth (2007) use truncated Fourier series in order to capture the seasonality in the mean and volatility. In a more recent paper, Benth, Saltyte-Benth, and Koekebakker (2007) propose the use of a continuous autoregressive model. Using 40 years of data in Stockholm, their results indicate that their proposed framework is sufficient to explain the autoregressive temperature dynamics. Overall, the fit is very good; however, the normality hypothesis is rejected even though the distribution of the residuals is close to normal.

A common denominator in all of the works mentioned above is that they use linear models, such as autoregressive moving average models (ARMA) or their continuous equivalents (Benth & Saltyte-Benth, 2007). However, a fundamental problem of such models is the assumption of linearity, which cannot capture some features that occur commonly in real-world data, such as asymmetric cycles and outliers (Agapitos, OŃeill, & Brabazon, 2012b). On the other hand, nonlinear models can encapsulate the time dependency of the dynamics of the temperature evolution, and can provide a much better fit to the temperature data than the classic linear alternatives.

One example of a nonlinear work is that by Zapranis and Alexandridis (2008), who used nonlinear non-parametric neural networks (NNs) to capture the daily variations of the speed at which the temperature reverts to its seasonal mean. Their results indicated that they had managed to isolate the Gaussian factor in the residuals, which is crucial for accurate pricing. Zapranis and Alexandridis (2009) used NNs to model the seasonal component of the residual variance of a mean-reverting O–U temperature process, with seasonality in the level and volatility. They validated their proposed method on more than 100 years of data collected from Paris, and their results showed a significant improvement over more traditional alternatives, regarding the statistical properties of the temperature process. This is important, since small misspecifications in the temperature process can lead to large pricing errors. However, although the distributional statistics were improved significantly, the normality assumption of the residuals was rejected.

NNs have the ability to approximate any deterministic nonlinear process, with little knowledge and no assumptions regarding the nature of the process. However, the classical sigmoid NNs have a series of drawbacks. Typically, the initial values of the NN's weights are chosen randomly, which is generally accompanied by extended training times. In addition, when the transfer function is of sigmoidal type, there is always a significant chance that the training algorithm will converge to a local minimum. Finally, there is no theoretical link between the specific parametrization of a sigmoidal activation function and the optimal network architecture, i.e., model complexity.

In this paper, we continue to look into nonlinear models, but we move away from neural networks. Instead, we look into two other algorithms from the field of machine learning (Mitchell, 1997): wavelet networks (WNs) and genetic programming (GP). The two proposed nonlinear methods will then be used to model the DAT. There are various reasons why we focus on these two nonlinear models. First, we want to avoid the black-boxes produced by alternative nonlinear models, such as NNs and support vector machines (SVM). Second, both models have many desirable properties, as it is explained below.

One of the main advantages of GP is its ability to produce white-box (interpretable) models, which allows traders to visualise the candidate solutions, and thus the

[1] The CAT and HDD indices are explained in Section 2.

temperature models. Another advantage of GP is that, unlike other models, it does not make any assumptions about the weather data. Furthermore, it does not require any assumptions about the shape of the solution (equation); we just feed in the algorithm with the appropriate components, and it creates solutions via its evolutionary approach. To the best of our knowledge, the only works that have applied GP to temperature weather derivatives are those of Agapitos, OŃeill, and Brabazon (2012a); Agapitos et al. (2012b). However, the GP proposed by Agapitos et al. (2012a,b) was used for the seasonal forecasting of temperature indices. Nevertheless, in principle, using models for daily temperatures can lead to more accurate pricing than modelling temperature indices (Jewson et al., 2005). Therefore, this study uses the GP to forecast DAT.

WNs, on the other hand, while not producing white-box models, can be characterised as grey-box models, since they can provide information on the participation of each wavelon to the function approximation and estimated dynamics of the generating process. In addition, WNs use wavelets as activation functions. We expect the waveforms of the wavelet activation function to capture the seasonalities and periodicities that govern the temperature process accurately in both the mean and variance. WNs were proposed by Pati and Krishnaprasad (1993) as an alternative to NNs that would alleviate the weaknesses associated with NNs and wavelet analysis, while preserving the advantages of both methods. In contrast to other transfer functions, wavelet activation functions have various desirable properties (Alexandridis & Zapranis, 2014). In particular, first, wavelets have high compression abilities, and secondly, computing the value at a single point or updating the function estimate from a new local measure involves only a small subset of coefficients. In contrast, other nonlinear regression algorithms, such as SVMs, have little theory about choosing the kernel functions and their parameters. In addition, these other algorithms encounter problems with discrete data, require very large training times, and need extensive memory for solving the quadratic programming (Burges, 1998). This study uses 11 years of detrended and deseasonalized DAT, resulting to 4,015 training patterns. WNs have been used in a variety of applications to date, such as short term load forecasting, time-series prediction, signal classification and compression, signal de-noising, static, dynamic and nonlinear modelling, and nonlinear static function approximation (Alexandridis & Zapranis, 2014); in addition, they can also constitute an accurate forecasting method in the context of weather derivatives pricing, as was shown by Alexandridis and Zapranis (2013a,b).

Earlier work using WNs and GP was presented by Alexandridis and Kampouridis (2013). The current study expands the work of Alexandridis and Kampouridis (2013) by comparing the results produced by the GP and the WN with those from the two state-of-the-art linear temperature modelling methods proposed by Alaton et al. (2002) and Benth and Saltyte-Benth (2007). Furthermore, the two proposed methods are also compared with three state-of-the-art machine learning algorithms that are used commonly in regression problems: neural networks (NN), radial basis functions (RBF), and support vector regression (SVR). The different models are compared in one-day-ahead and period-ahead out-of-sample forecasting on *180 different data sets*. Moreover, we perform an in-depth analysis of predictive power and a statistical ranking of each method. Finally, we study the evolution of the prediction errors of the methods across different time horizons.

Lastly, it should be mentioned that the problem of temperature prediction in the context of weather derivatives is completely different to the problem of weather forecasting. In the latter, meteorologists aim to predict the temperature accurately over a short time period (e.g., 3–5 days) and in the near future (e.g., next week). With weather derivatives, a trader is faced with the problem of pricing a derivative where the measurement period is (possibly) a year later. Thus, s/he has to have an accurate expectation of the temperature properties, such as the cumulative average over a certain long-term period (e.g., a year). Thus, predicting the temperature accurately on a daily basis is not the issue here, and therefore, once the temperature predictions have been obtained, they are then used as parameters to decide on the price at which the derivatives are going to be traded.

The rest of the paper is organized as follows. Section 2 briefly presents the weather derivatives market. Section 3 presents our methodology. More precisely, the linear and nonlinear models are presented in Sections 3.1 and 3.2, respectively. The WN and the GP are discussed in Sections 3.3 and 3.4 respectively, and the three machine learning benchmark models (NN, RBF, SVR) are presented in Section 3.5. The data sets are described in Section 4, while our results are presented in Section 5. The in-sample comparison of all models is discussed in Section 5.1, while Section 5.2 presents the out-of-sample forecasting comparison. Finally, Section 6 concludes and discusses future work.

2. The weather market

Chicago Mercantile Exchange (CME) offers various weather futures and options contracts. These are index-based products that are geared to the average seasonal and monthly weather in 47 cities[2] around the world: 24 in the U.S., 11 in Europe, 6 in Canada, 3 in Australia and 3 in Japan. Temperature derivatives are usually settled based on four main temperature indices: CAT, HDDs, cooling degree days (CDD) and the Pacific Rim (PAC).

In Europe, CME weather contracts for the summer months are based on an index of CAT. The CAT index is the sum of the DATs over the contract period. The value of a CAT index for the time interval $[\tau_1, \tau_2]$ is given by:

$$CAT = \int_{\tau_1}^{\tau_2} T(s)ds, \tag{1}$$

where the temperature is measured in degrees Celsius and the DAT is the average of the daily maximum and minimum temperatures. One CAT index futures contract

[2] This is the number of cities for which the CME trades weather contracts at the end of 2014.

costs £20 per index point in London, and €20 per index unit in all other European locations. CAT contracts have either monthly or seasonal durations. CAT futures and options are traded on the following moths: May, June, July, August, September, April and October.

In the USA, Canada and Australia, CME weather derivatives are based on either the HDD or CDD indices. HDD is the number of degrees by which the daily temperature is below a base temperature, and CDD is the number of degrees by which the daily temperature is above the base temperature. The base temperature is usually 65 degrees Fahrenheit in the USA and 18 degrees Celsius in Europe and Japan. Mathematically, this can be expressed as

$$HDD(t) = \big(18 - T(t)\big)^{+} = \max\big(18 - T(t), 0\big)$$

$$CDD(t) = \big(T(t) - 18\big)^{+} = \max\big(T(t) - 18, 0\big).$$

HDDs and CDDs are accumulated over a period, usually a month or a season. Hence, the accumulated HHDs and CDDs over the period $[\tau_1, \tau_2]$ are given by:

$$AccHDD(t) = \int_{\tau_1}^{\tau_2} \max\big(18 - T(t), 0\big)ds$$

$$AccCDD(t) = \int_{\tau_1}^{\tau_2} \max\big(T(t) - 18, 0\big)ds.$$

CME also trades HDD contracts for the European cities. Contracts on the following months can be found: November, December, January, February, March, October and April.

It can be shown easily that the HDD, CDD and CAT indices are linked by the following formula:

$$\max\big(18 - T(t), 0\big) = 18 - T(t) + \max\big(T(t) - 18, 0\big). \quad (2)$$

For the three 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:

$$PAC = \frac{1}{\tau_2 - \tau_1} \int_{\tau_1}^{\tau_2} T(s)ds. \quad (3)$$

In this study, we focus only on the CAT and HDD indices. The PAC and CDD indices can be retrieved using the relationships in Eqs. (2) and (3).

A trader is interested in finding the price of a temperature contract written on a specific temperature index. The price of a futures contract written in a temperature index under the risk neutral probability Q at time $t \leq \tau_1 < \tau_2$ is

$$e^{-r(T-t)}\mathbb{E}_Q\Big[Index - F_{Index}(t, \tau_1, \tau_2) \mid \mathcal{F}_t\Big] = 0,$$

where *Index* is the CAT, PAC, AccHDD or AccCDD and F_{Index} is the price of a futures contract written on the specific index, r is the risk-free interest rate, and $\mathcal{F}_t$ is the history of the process until time t. Since F_{Index} is $\mathcal{F}_t$-adapted, we derive the price of the futures contract to be

$$F_{Index}(t, \tau_1, \tau_2) = \mathbb{E}_Q\Big[Index \mid \mathcal{F}_t\Big],$$

which is the expected value of the temperature index under the risk-neutral probability Q and the filtration $\mathcal{F}_t$.

3. Methodology

According to Alexandridis and Zapranis (2013a) and Cao and Wei (2004), the temperature has the following characteristics: it follows a predicted cycle, it moves around a seasonal mean, it is affected by global warming and urban effects, it appears to have autoregressive changes, and its volatility is higher in winter than in summer.

Various different models have been proposed in an attempt to describe the dynamics of a temperature process. Early models used AR(1) processes or continuous equivalents (Alaton et al., 2002; Cao & Wei, 2000). A more general version of an ARMA(p, q) model was suggested by Dornier and Queruel (2000) and Moreno (2000). However, Caballero and Jewson (2002) showed that all of these models fail to capture the slow time decay of the autocorrelations of temperature, hence leading to a significant underpricing of weather options. More complex models utilize an O–U process where the noise part of the process can be a Brownian, fractional Brownian or Lévy process (Benth & Saltyte-Benth, 2005; Brody, Syroka, & Zervos, 2002).

When the noise process follows a Brownian motion, the temperature dynamics are given by the following model, where the DAT is described by a mean-reverting O–U process:

$$dT(t) = dS(t) + \kappa \times \big(T(t) - S(t)\big)dt + \sigma(t)dB(t), \quad (4)$$

where $T(t)$ is the average daily temperature, κ is the speed of mean reversion (i.e., how fast the temperature returns to its seasonal mean), $S(t)$ is a deterministic function that models the trend and seasonality, $\sigma(t)$ is the daily volatility of temperature variations, and $B(t)$ is the driving noise process. As was shown by Dornier and Queruel (2000), the term $dS(t)$ should be added in order to ensure a proper mean-reversion to the historical mean, $S(t)$. For more details on temperature modelling, we refer the reader to Alexandridis and Zapranis (2013a).

The following sections present the models that this paper uses to predict the daily temperature. First, Section 3.1 presents two state-of-the-art linear models that are typically used for daily temperature prediction in the context of weather derivatives: those of Alaton et al. (2002), and Benth and Saltyte-Benth (2007). Then, Section 3.2 presents the nonlinear equations that act as the motivation behind the research into machine learning algorithms that we discuss in the following sections. Next, Section 3.3 presents the WNs and their setup, along with parameter tuning. Section 3.4 then presents the GP algorithm and its experimental setup, along with parameter tuning. Finally, Section 3.5 discusses the three different state-of-the-art machine learning algorithms that are used commonly for regression problems, and are used as benchmarks in our paper.

3.1. Linear models

This section presents the two linear models that will be used for the comparison of temperature modelling in the context of weather derivatives pricing. The first one was proposed by Alaton et al. (2002) and will be referred to as the Alaton model, while the second one was proposed by Benth and Saltyte-Benth (2007) and will be referred to as the Benth model. Both models have been proposed

previously, and are presented well and extensively in the literature. Here, we present the basic aspects of both models briefly, for the sake of completeness. For analytical presentations of the two models, the reader is referred to Alaton et al. (2002) and Benth and Saltyte-Benth (2007).

3.1.1. The Alaton model

Alaton et al. (2002) use the model given by Eq. (4), where the seasonality in the mean is incorporated using a sinusoid function:

$$S(t) = A + Bt + C\sin(\omega t + \phi), \tag{5}$$

where ϕ is the phase parameter that defines the days of the yearly minimum and maximum temperatures. Since it is known that the DAT has a strong seasonality with a one year period, the parameter ω is set to $\omega = 2\pi/365$. The linear trend due to urbanization or climate change is represented by $A + Bt$. The time, measured in days, is denoted by t. The parameter C defines the amplitude of the difference between the yearly minimum and maximum DATs. Using the Itô formula, a solution to Eq. (4) is given by:

$$T(t) = S(t) + \big(T(s) - S(s)\big)e^{-\kappa(T-s)} + \int_s^t e^{-\kappa(t-s)}\sigma(\tau)dB(\tau). \tag{6}$$

Another innovative characteristic of the framework presented by Alaton et al. (2002) is the introduction of seasonality to the standard deviation, modelled by a piecewise function. They assume that $\sigma(t)$ is a piecewise constant function, with a constant value each month.

3.1.2. The Benth model

Benth and Saltyte-Benth (2007) suggested the use of a mean reverting O–U process, where the noise process is modelled by simple Brownian motion, as in Eq. (4). The discrete form of the model in Eq. (4) can be written as an AR(1) model with a zero constant:

$$\tilde{T}(t+1) = a\tilde{T}(t) + \tilde{\sigma}(t)\epsilon(t) \tag{7}$$

where $\tilde{T}(t)$ is the detrended and deseasonalised DAT given by $\tilde{T}(t) = T(t) - S(t)$, $a = e^{-\kappa}$ and $\tilde{\sigma}(t) = a\sigma(t)$.

Strong seasonality is evident in the autocorrelation function of the squared residuals of the AR(1) model. Both the seasonal mean and the (square of the) daily volatility of temperature variations are modelled using truncated Fourier series:

$$S(t) = a + bt + \sum_{i=1}^{I_1} a_i \sin\big(2\pi i(t - f_i)/365\big) + \sum_{j=1}^{J_1} b_j \cos\big(2\pi j(t - g_j)/365\big) \tag{8}$$

$$\sigma^2(t) = c + \sum_{i=1}^{I_2} c_i \sin\big(2\pi it/365\big) + \sum_{j=1}^{J_2} d_j \cos\big(2\pi jt/365\big). \tag{9}$$

Using truncated Fourier series allows us to obtain a good fit for both the seasonality and variance components, while keeping the number of parameters relatively low (Benth & Saltyte-Benth, 2007). The representation above simplifies the calculations needed for the estimation of the parameters and for the derivation of the pricing formulas. Eqs. (8) and (9) allow both larger and smaller periodicities in the mean and variance than the classical one-year temperature cycle.

3.2. Nonlinear models

The speed of mean reversion, κ, indicates how quickly the temperature process reverts to the seasonal mean. Intuitively, it is expected that the speed of mean reversion will not be constant. If the temperature today is away from the seasonal average (a cold day in summer), then the speed of mean reversion will be expected to be high; i.e., the difference between today's and tomorrow's temperatures is expected to be high. In contrast, if the temperature today is close to the seasonal variance, we expect the temperature to revert to its seasonal average slowly. We capture this feature by using a time-varying function $\kappa(t)$ to model the speed of mean reversion. Hence, the structure for modelling the dynamics of the temperature evolution becomes:

$$dT(t) = dS(t) + \kappa(t) \times \big(T(t) - S(t)\big)dt + \sigma(t)dB(t). \tag{10}$$

Eq. (7) is a lineal AR(1) model with a zero constant. Since our analysis considers the speed of mean reversion to be a time-varying function, not a constant, Eq. (7) can be written as:

$$\tilde{T}(t) = a(t-1)\tilde{T}(t-1) + \sigma(t)\epsilon(t), \tag{11}$$

where

$$a(t) = 1 + \kappa(t). \tag{12}$$

The impact of a false specification of a on the accuracy of the pricing of temperature derivatives is significant (Alaton et al., 2002). Using nonlinear models, the generalized version of Eq. (11) is estimated nonlinearly and non-parametrically, that is:

$$\tilde{T}(t+1) = \phi\big(\tilde{T}(t), \tilde{T}(t-1), \ldots\big) + e(t). \tag{13}$$

It is clear that Eq. (13) is a generalisation of Eq. (7). In other words, the difference between the linear and nonlinear models is the definition of ϕ. The previous section estimated ϕ using two different linear models. The next section estimates the function ϕ using a range of nonlinear models, such as WNs, GP, SVRs, RBFs and NNs.

Eq. (13) uses past temperatures (detrended and deseasonalized) over one period. We expect the use of more lags to overcome the strong correlation found in the residuals in models such as those of Alaton et al. (2002), Benth and Saltyte-Benth (2007) and Zapranis and Alexandridis (2008). However, the length of the lag series must be selected. This is described for each nonlinear model in the sections that follow.

3.3. Wavelet networks

WNs are a theoretical formulation of a feed-forward NN in terms of wavelet decompositions. WNs are networks

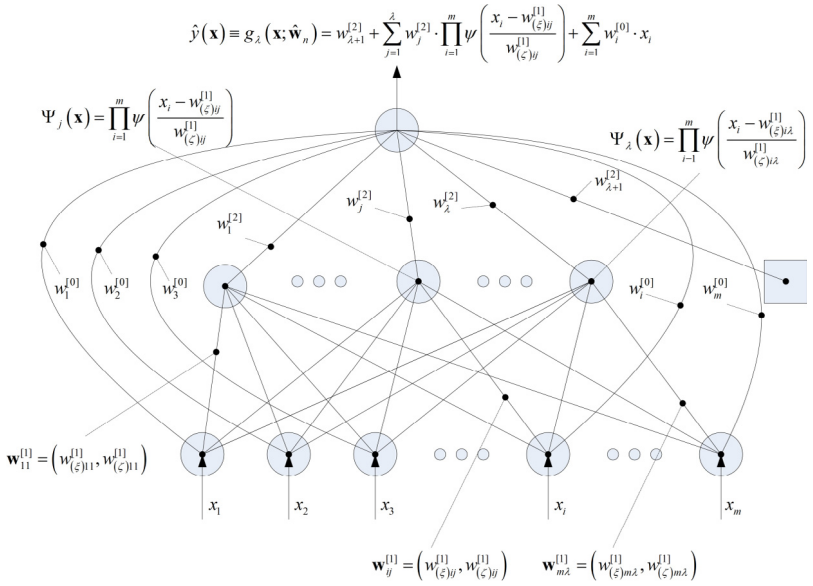

Fig. 1. A feedforward wavelet network.

with one hidden layer that use a wavelet as an activation function, instead of the classic sigmoidal family. They are a generalization of radial basis function networks. WNs overcome the drawback associated with neural networks and wavelet analysis, while at the same the time preserving the "universal approximation" property that characterizes neural networks. In contrast to the classic transfer functions, wavelets have high compression abilities; and in addition, computing the value at a single point or updating the function estimate from a new local measure involves only a small subset of coefficients (Bernard, Mallat, & Slotine, 1998). In contrast to classical "sigmoid NNs", WNs allow for constructive procedures that initialize the parameters of the network efficiently. The use of wavelet decomposition allows a "wavelet library" to be constructed. In turn, each wavelon can be constructed using the best wavelet in the wavelet library. The main characteristics of these procedures are: (i) convergence to the global minimum of the cost function, and (ii) initial weight vector into close proximity of the global minimum, leading to drastically reduced training times (Zhang, 1997; Zhang & Benveniste, 1992). In addition, WNs provide information on the relative participation of each wavelon in the function approximation, and the estimated dynamics of the generating process. Finally, efficient initialization methods will approximate the same vector of weights that minimize the loss function each time.

3.3.1. Model setup

Our proposed WN has the structure of a three-layer network. We propose a multidimensional WN with a linear connection between the wavelons and the output, and also include direct connections from the input layer to the output layer in order to be able to approximate accurately linear problems. Hence, a network with zero HUs is reduced to the linear model.

The structure of a single hidden-layer feedforward WN is given in Fig. 1. The network output is given by:

$$g_\lambda(\mathbf{x};\mathbf{w}) = \hat{y}(\mathbf{x}) = w_{\lambda+1}^{[2]} + \sum_{j=1}^{\lambda} w_j^{[2]} \cdot \Psi(\mathbf{x}) + \sum_{i=1}^{m} w_i^{[0]} \cdot x_i, \quad (14)$$

where $\Psi(\mathbf{x})$ is a multidimensional wavelet which is constructed as the product of m scalar wavelets, $\mathbf{x}$ is the input vector, m is the number of network inputs, λ is the number of HUs, and w stands for a network weight. The multidimensional wavelets are computed as

$$\Psi(\mathbf{x}) = \prod_{i=1}^{m} \psi(z_{ij}), \quad (15)$$

where ψ is the mother wavelet and

$$z_{ij} = \frac{x_i - w_{(\xi)ij}^{[1]}}{w_{(\zeta)ij}^{[1]}}. \quad (16)$$

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

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

These parameters are adjusted during the training phase.

Following Becerikli, Oysal, and Konar (2003), Billings and Wei (2005), and Zhang (1994), we take as our mother wavelet the Mexican Hat function, which has been shown to be useful and to work satisfactorily in various applications, and is given by:

$$\psi(z_{ij}) = (1 - z_{ij}^2)e^{-\frac{1}{2}z_{ij}^2}. \quad (17)$$

Table 1
Variable selection with backward elimination in Berlin.

Step	Variable to remove (lag)	Variable to enter (lag)	Variables in model	Hidden units (parameters)	n/p ratio	Empirical loss	Prediction risk
	–	–	7	5 (83)	43.9	1.5928	3.2004
1	X_6	–	6	2 (33)	110.4	1.5922	3.1812
2	X_7	–	5	1 (17)	214.3	1.5927	3.1902
3	X_5	–	4	1 (14)	260.2	1.6004	3.2056
4	X_4	–	3	1 (11)	331.2	1.5969	3.1914

The algorithm concluded in four steps. In each step, we present the following: which variable is removed, the number of hidden units for the particular set of input variables and parameters used in the wavelet network, the empirical loss and the prediction risk.

3.3.2. Parameter tuning

The WN is constructed and trained by applying the model selection and variable selection algorithms developed and presented by Alexandridis and Zapranis (2014, 2013b). The algorithms are presented analytically by Alexandridis and Zapranis (2014), while the flowchart of the model identification algorithm is presented in Fig. 2. Eq. (13) implies that the number of lags of the detrended and deseasonalized temperatures must be decided. The lagged series will be used as inputs for the training of the WN, where the output/target time series is today's detrended and deseasonalized temperature.

Initially, the training set contains the dependent variable and seven lags. Hence, the training set consists of seven inputs, one output and 3643 training pairs. Table 1 summarizes the results of the model identification algorithm for Berlin. The results for the remaining cities are similar. Both the model selection and variable selection algorithms are included in Table 1. The algorithm concluded in four steps, and the final model contains only three variables. In the final model the prediction risk is 3.1914, while that for the original model was 3.2004. A closer inspection of Table 1 reveals that the empirical loss increased slightly, from 1.5928 for the initial model to 1.5969 for the reduced model, indicating that the explained variability (unadjusted) decreased slightly, but that the explained variability (adjusted for degrees of freedom) was increased from 63.98% initially to 64.61% for the reduced model. Finally, the number of parameters in the final model is reduced significantly. The initial model needed five HUs and seven inputs, resulting to 83 parameters. Hence, the ratio of the number of training pairs n to the number of parameters p was 43.9. In the final model, only one HU and three inputs were used. Hence, only 11 parameters were adjusted during the training phase, and the ratio of the number of training pairs n to the number of parameters p was 331.2. In all cities, a WN with only one HU is sufficient to model the detrended and deseasonalized DATs.

The backward elimination method was used for the efficient initialisation of the WN, as was described by Alexandridis and Zapranis (2014, 2013b). Efficient initialization will result in fewer iterations in the training phase of the network and training algorithms that will avoid local minima of the loss function in the training phase. After the initialization phase, the network is trained further in order to obtain the vector of the parameters $w = \hat{\mathbf{w}}_n$ that minimizes the loss function. The ordinary back-propagation algorithm is used.

Panel (a) of Fig. 3 presents the initialization of the final model using only one HU. The initialization is very good and the WN converged after only 19 iterations. The training stopped when the minimum velocity, 10^{-5}, of the training algorithm was reached. The minimum velocity can be expressed mathematically as

$$\left| \frac{L_{n,t} - L_{n,t-1}}{L_{n,t-1}} \right|,$$

where $L_{n,t}$ is the training error of the WN at iteration t. The fit of the trained WN is shown in panel (b) of Fig. 3.

3.4. Genetic programming

Genetic programming (GP; see Banzhaf, Nordin, Keller, & Francone, 1998; Koza, 1992; Poli, Langdon, & McPhee, 2008) is an evolutionary technique that is inspired by natural evolution, where computer programs act as the individuals in a population. We apply the GP algorithm by following the procedure described below. First, a random population of individuals is initialized, by using terminals and functions that are appropriate to the problem domain. The former are the variables and constants of the programs, and the latter are responsible for processing the values of the system, either terminals or other functions' outputs. After the population has been initialized, each individual is measured in terms of a pre-specified fitness function. The fitness function measures the performance of each individual on the specified problem. The fitness value determines which individuals from the current generation will have their genetic material passed into the next generation (the new population) via genetic operators. We ensure that the best material is chosen by enforcing a selection strategy. Typically, this is done by using a tournament selection, where t candidate parents are selected from the population at random, and the best of these t individuals becomes the first parent. If necessary, the process is repeated in order to select the second parent (e.g., for the crossover operator). These parent individuals are then manipulated by genetic operators, such as crossover and mutation, in order to produce offspring, which constitute the new population. In addition, elitism can be used to copy the best individuals into the new population, in order to ensure that the best solutions are not lost between generations. Finally, a new fitness function is assigned to each individual in the new population, and the whole process is repeated until a given termination criterion is met. Usually, the process ends after a specified number of generations. In the last generation, the program with the best fitness is considered to be the result of that run. For a relatively up-to-date perspective on the field of GP, including open issues, see Miller and Poli (2010).

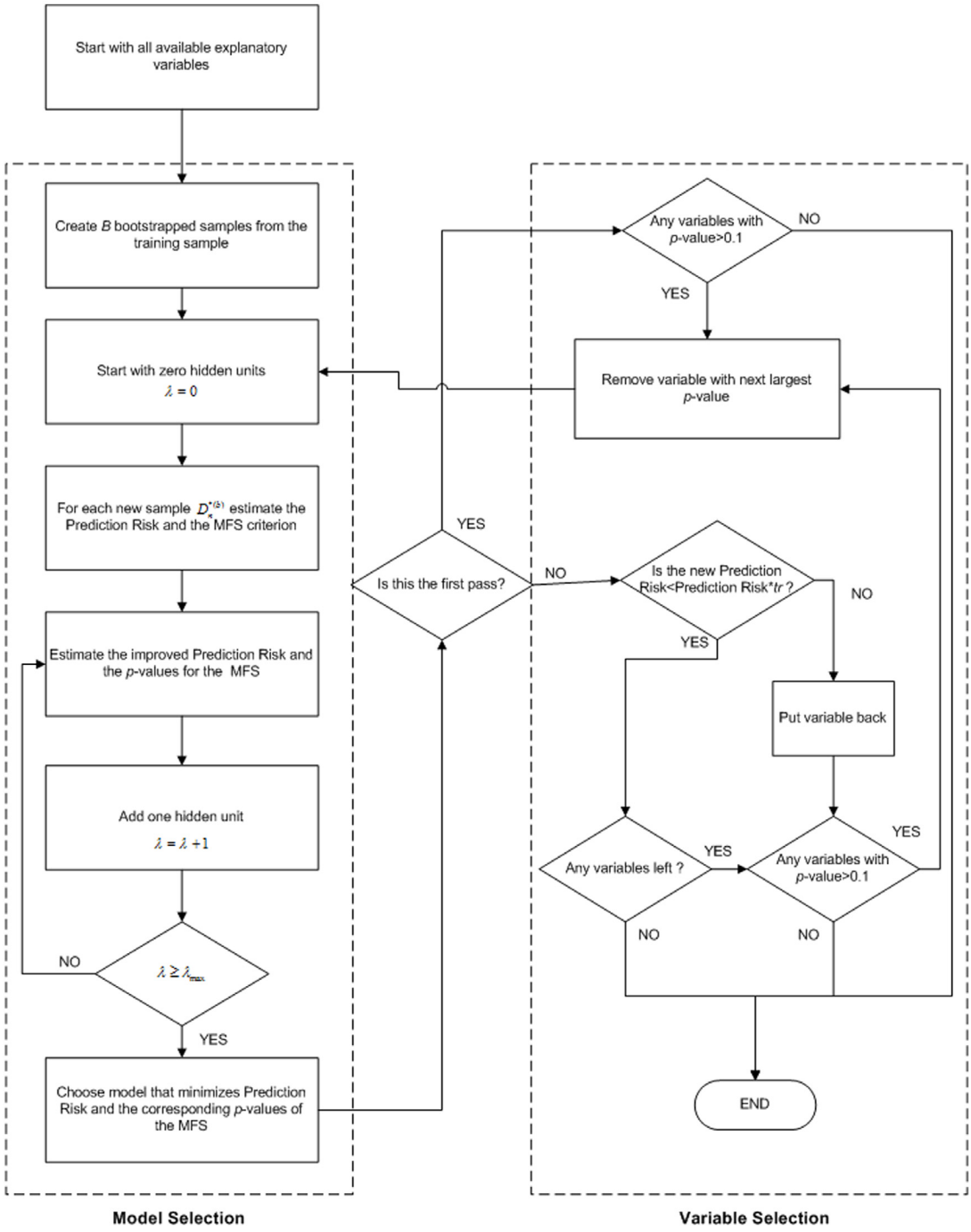

Fig. 2. Model identification: model selection and variable selection algorithms using wavelet networks.

As was explained at the beginning of this paper, we chose to apply the GP to the problem of modelling the temperature in the context of weather derivatives for several reasons: they are white-box (interpretable) models, and require no assumptions about the weather data or the shape of the solution (equation). This provides the advantage of flexibility, since a different temperature model can be derived for each city that we are interested in, in contrast to the linear models of Alaton and Benth, which assume fixed functional forms.

3.4.1. Model setup

This study uses our GP to evolve trees that predict the temperatures of a given city over a future period. The

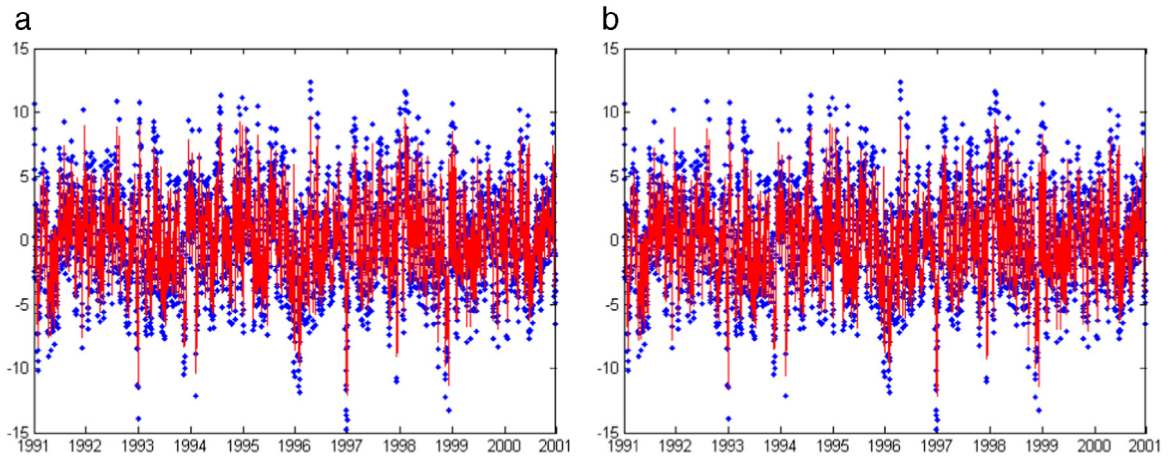

Fig. 3. Initialization of the final model for the temperature data in Berlin using the BE method (a) and the fit of the trained network with one HU (b). The WN converged after 19 iterations.

function set of the GP contains standard arithmetic operators (ADD, SUB, MUL, DIV (protected division)), along with MOD (modulo), LOG(x), SQRT(x) and the trigonometric functions of sine and cosine. The terminal set consists of the index t representing the current day, $1 \leq t \leq$ (size of training and testing set); the temperatures of the last N days,[3] $\tilde{T}(t-1), \tilde{T}(t-2), \ldots, \tilde{T}(t-N)$; the constant π; and 10 random numbers in the range $(-10, 10)$. A sample tree, which was the best tree produced by the GP for the Stockholm dataset, is presented in Fig. 4. According to this tree, today's temperature $\tilde{T}_t$ is equivalent to

$$(\alpha \times \beta \times \tilde{T}_{t-2} + \tilde{T}_{t-1}) \times \cos\left(\frac{\sin\gamma}{\delta + \tilde{T}_{t-5}}\right),$$

where $\tilde{T}_{t-1}$, $\tilde{T}_{t-2}$ and $\tilde{T}_{t-5}$ are the temperatures at times $t-1$, $t-2$ and $t-5$, respectively, and α, β, γ, and δ are constants. As can be seen from the equation above, the temperature take into account not only very short-term historical values ($\tilde{T}_{t-1}$, $\tilde{T}_{t-2}$), but also longer-term values ($\tilde{T}_{t-5}$).

The genetic operators that we use are subtree crossover, subtree mutation and point mutation (Banzhaf et al., 1998; Koza, 1992; Poli et al., 2008). In our algorithmic setup, the probability of point mutation, P_{PM}, is equal to $(1 - P_{SC} - P_{SM})$, where P_{SC} and P_{SM} are the probabilities of subtree crossover and subtree mutation, respectively. The fitness function is the mean square error (MSE). Next, Section 3.4.2 discusses the tuning of some important GP parameters.

3.4.2. Parameter tuning

The tuning of the parameters took place in four different phases. Thus, we were creating different model setups, where a different set of values would be used in each setup. Then, we tested each setup under three different datasets, namely the DATs for Madrid, Oslo, and Stockholm. It is important to note here that these datasets are different from those that are used for our comparative experiments in Section 5. This was done deliberately in order to avoid having a biased algorithmic setup due to parameter tuning.

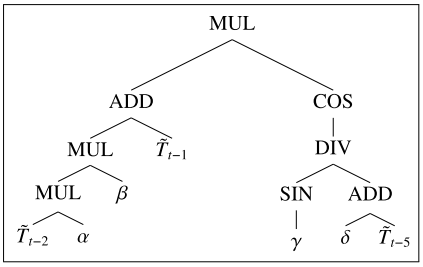

Fig. 4. Best tree returned for the Stockholm database. The equivalent equation is $(\alpha \times \beta \times \tilde{T}_{t-2} + \tilde{T}_{t-1}) \times \cos(\frac{\sin\gamma}{\delta + \tilde{T}_{t-5}})$.

In the first phase, we were interested in optimising the population size and the number of generations. We experimented with four different population sizes, namely 100, 300, 500 and 1000, and four numbers of generations, 30, 50, 75 and 100. Combining these population and generation values created 16 different model setups. After 50 runs of each setup, we used the non-parametric Friedman test to rank them in terms of average testing fitness. The setup that ranked the highest was the one using a population of 500 individuals and 50 generations.

In the second parameter-tuning phase, we were interested in tuning the genetic operators' probabilities. We experimented with probabilities of 0.1, 0.3 and 0.5 for both subtree crossover and subtree mutation.[4] This set of values created nine different model setups. Each setup was ranked in terms of its average testing fitness after 50 individual runs. Our results indicate that the highest ranking setup was $P_{SC} = 0.3$, $P_{SM} = 0.5$ and $P_{PM} = 0.2$.

Next, in the third parameter-tuning phase, we were interested in increasing the generalisation chances of our

[3] The value of N, which is the number of different lags, as presented in Eq. (13), was determined by parameter tuning, and is presented in Section 3.4.2.

[4] We found during the early experimentation phase that high crossover values (e.g., a crossover probability of 0.9) did not lead to good results, and therefore we did not include such high values during the parameter tuning phase.

training temperature models. We achieved this by using the machine learning ensemble algorithm of bootstrap aggregating (a.k.a. bagging), which generates m new training sets from a training set D of size n, with each new set being of size n', by sampling from D uniformly and with replacement. We set size $n' = n$, and then experimented with m different training sets. More specifically, we experimented with ensembles of sizes ranging from two to 10. Our experiments showed that the best-performing ensemble size was seven.

Finally, in the last phase we were interested in determining the number of lags of the past temperatures of Eq. (13). As in the case of the WN, we experiment with seven lags, with 50 individual runs for each number of lags. However, we should note that in this case our methodology was applied to the datasets used in the results section (Section 5), namely Amsterdam, Berlin, and Paris. We experimented with these datasets here because the tuning of lags would only be meaningful if it took place on the actual datasets that we are interested in, not the ones used for tuning purposes. The Friedman non-parametric test showed that the best testing results were achieved when using five variables: detrended and deseasonalised temperatures at times $t-1$, $t-2$, $t-3$, $t-4$, and $t-5$. Thus, we decided to use five lags for our comparative experiments.

Table 2 summarises the experimental parameters used by our GP, as a result of parameter tuning.[5] Finally, given that the GP is a stochastic algorithm, we perform 50 independent runs of the algorithm, with the GP results reported in Section 5 being the averages of these 50 runs. In addition, we also present the performance of the best GP tree over the 50 runs, as in the real world one would be using a single tree, which would be the best tree returned during the training phase.

Table 2
GP experimental parameters.

Parameter	Value
Max initial depth	2
Max depth	4
Generations	50
Population size	500
Tournament size	4
Subtree crossover	30%
Subtree mutation	50%
Point mutation	20%
Fitness function	Mean square error (MSE)
Function set	ADD, SUB, MUL, DIV, MOD, LOG, SQRT, SIN, COS
Terminal set	Index t corresponding to current day $\tilde{T}_{t-1}$, $\tilde{T}_{t-2}$, $\tilde{T}_{t-3}$, $\tilde{T}_{t-4}$, $\tilde{T}_{t-5}$, Constant π 10 random constants in $(-10, 10)$

3.5. Benchmark nonlinear methods

Here, we outline the three nonlinear benchmarks (Chang & Lin, 2011; Hall et al., 2009) that are to be compared against the performances of WN and GP. For each algorithm, we first provide a brief introduction, then present the model setup. Lastly, we discuss the parameter tuning process.

3.5.1. Neural networks

A multilayer perceptron (MLP) is a feed-forward NN that utilizes a back-propagation learning algorithm in order to enhance the training of the network (Rumelhart, McClelland, & PDP Research Group, 1986). NNs consist of multiple layers of nodes that are able to construct nonlinear functions. A minimum of three layers are constructed, namely an input layer and an output layer, with l hidden layers in between. Each node in one layer connects to each node in the next layer with a weight w_{ij}, where ij is the connection between two nodes in adjacent layers within the network. Each node in the hidden layer will be a sigmoid (a nonlinear function; see Cybenko, 1989), but for the purposes of a regression problem, the output layer is a linear activation function.

On each pass through, the NN calculates the loss between the predicted output $\hat{y}_n$ at the output layer and the expected output y_n for the nth iteration (epoch). The loss function used in this paper is usually the sum of squared errors, given by:

$$L_n = \frac{1}{2}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \tag{18}$$

where N represents the total number of training points. Once the loss has been calculated, the back-propagation step begins by tracking the output error back through the network. The errors from the loss function are then used to update the weights for each node in the network, such that the network converges. Therefore, minimising the loss function requires w_{ij} to be updated repeatedly using gradient descent, so we update the weights at step $t+1$, $w_{ij,t+1}$, using:

$$w_{ij,t+1} = w_{ij,t} - \eta\frac{\delta L}{\delta w_{ij,t}} + \mu\Delta w_{ij,t}, \tag{19}$$

where $w_{ij,t+1}$ is the updated weight, η is the learning rate, Δ represents the gradient, and μ is the momentum. The derivative $\frac{\delta L}{\delta w_{ij,t}}$ is used to calculate how much and in which direction the weights should be modified. The learning rate, $\eta > 0$, indicates the distance to be travelled along the gradient descent at each update. To ensure convergence, the value of η should remain relatively small. However, too small a value of η will either cause slow convergence or potentially trap the training in a local minimum. A momentum term, μ, is used to speed up the learning process, and μ reduces the possibility of falling into a local minimum by making larger movements down the gradient descent in the same direction. In addition, in order to prevent the network from diverging, the learning rate

[5] We did not do any tuning for the maximum initial or overall depth of the trees, as we were interested in keeping a low value of the depth in order to retain the human comprehensibility of the trees. In addition, previous experiments had shown that the algorithm was not sensitive to different values of the depth.

will decay by:

$$\eta_n = \frac{\eta_0}{1 + n\eta_d}, \tag{20}$$

where $\eta_d = \frac{\eta}{I}$ and I is the total number of epochs.

3.5.2. Radial basis function

RBFs are a variant of feed-forward NNs that rely only on a two-layered network (input and output; see Broomhead & Lowe, 1988). Between the two layers exists a hidden layer, in which each node implements a radial basis function (or radial kernel), which is tuned to a specific region of the feature space. The activation of each radial kernel is based on the distance between the input vector x and a dummy vector μ_j, given by:

$$\phi_j(x) = f(\|x - \mu_j\|), \tag{21}$$

where j is the total number of radial kernels and $\phi_j(x)$ is a nonlinear function for each radial kernel in the network (input-hidden mapping). The most common radial basis, which is the one used in this paper, is the Gaussian kernel given by:

$$\phi_j(x) = \exp\left[-\frac{1}{2}(x - \mu_j)'\sigma_j^{-1}(x - \mu_j)\right], \tag{22}$$

where μ_j and σ_j are the mean and covariance matrix of the jth Gaussian function. Finally, each radial kernel is mapped to an output (hidden-output mapping) via a weighted sum of each radial kernel, given by:

$$y_o(x) = \sum_{j=1}^{K} \lambda_{jo}\phi_j(x), \tag{23}$$

where λ are the output weights, K represents the number of radial kernels in the hidden layer, and o represents the number of output nodes in the output layer. We train the network using the k-means clustering unsupervised technique in order to find the initial centres for the Gaussian kernels. Once the initial centres have been selected, the network adjusts itself to the minimum distance $\|x_i - \hat{\mu}_j\|$ for each radial kernel, given the data x_i. Finally, the hidden-output weights that map each radial kernel to the output nodes can be optimised by minimising the least squares estimate, producing an $f(x)$ that consists of the optimised weighted sum of all of the radial kernels.

3.5.3. Support vector regression

SVR is a very specific class of algorithm without local minima, which facilitates the usage of kernels and promotes sparseness and the ability to generalise (Vapnik, 1995). SVR essentially learns a non-linear function by mapping linear functions into high dimensional kernel induced feature space. This paper uses a type of SVR called ϵ-SV regression, where we attempt to find a function $f(x)$ that has at most ϵ error between the predicted value $\hat{y}_n$ and the actual value y_n for all of the training data. Therefore, the only considerations are that the predicted output must be within the margin ϵ at all times, no error larger than ϵ should be accepted, and at the same time the output should be as *flat* as possible. We aim to fit the following function:

$$f(x) = \langle \omega, x_n \rangle + b \quad \omega \in \chi, b \in \mathbb{R} \tag{24}$$

where $\langle\ ,\ \rangle$ represents the dot product in χ. We strive for a small ω in order to ensure the flattest curve, thus making the predictions less sensitive to random shocks in the training data. This is formulated as a convex optimisation problem by minimising $\frac{1}{2}\|\omega^2\|$, subject to $|y_n - (\langle \omega, x_n \rangle + b)| \leq \epsilon,\ \forall n$. It is probable that there is no function $f(x)$ that satisfies the constraint ϵ at all points. We detract from this violation by employing a "*soft-margin*", which introduces two slack variables η_n and η_n^* for each data point. Hence, we aim to minimise the objective function:

$$\min \frac{1}{2}\|\omega^2\| + C\sum_{n=1}^{N}(\eta_n + \eta_n^*), \tag{25}$$

subject to:

$$y_n - (\langle \omega, x_n \rangle + b) \leq \epsilon + \eta_n \quad \forall n \tag{26}$$

$$(\langle \omega, x_n \rangle + b) - y_n \leq \epsilon + \eta_n^* \quad \forall n \tag{27}$$

$$\eta_n, \eta_n^* \geq 0 \quad \forall n, \tag{28}$$

where the constant $C > 0$ (cost) represents the balance between the flatness of $f(x)$ and the extent to which violations of ϵ are tolerated. The loss function is the distance between the observed value y_n and the margin of allowed error ϵ, given by:

$$\text{Loss}_\epsilon = \begin{cases} 0, & \text{if } |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon & \text{otherwise.} \end{cases} \tag{29}$$

The optimisation of Eq. (25) is only possible if the training data are strictly linear. The production of nonlinear functions requires a nonlinear kernel function $G(x_i, x) = \langle \phi x_i, \phi x \rangle$, where $\phi(x)$ is a transformation that maps x to a high-dimensional space. Then, a linear model is constructed in this new feature space. This requires Eq. (25) to be transformed into a Langrange dual formula by introducing non-negative multipliers α_n and α_n^* for each observation x_n and minimising the objective function:

$$\min \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i, x) + \epsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*), \tag{30}$$

subject to:

$$\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0 \tag{31}$$

$$0 \leq \alpha_n \leq C \quad \forall n \tag{32}$$

$$0 \leq \alpha_n^* \leq C \quad \forall n. \tag{33}$$

Any predictions that lie within the ϵ margin have Lagrange multipliers $a_n = 0$ and $a_n^* = 0$. Those outside the ϵ margin are called support vectors. Therefore, the regression function is given by:

$$f(x) = \sum_{i=1}^{n_{sv}}(\alpha_i - \alpha_i^*)K(x_i, x), \tag{34}$$

where n_{sv} refers to the number of support vectors. Here, we use the radial basis function (RBF) kernel, which takes an additional parameter γ, given by:

$$K(x_i, x) = \exp(-\gamma |x_i - x|^2). \quad (35)$$

3.5.4. Parameter tuning

The three methods above were tuned on the DATs of the cities used in the case of GP (Madrid, Oslo and Stockholm), in order to avoid bias in the results. We tuned the parameters using the iRace optimisation package (López-Ibáñez, Dubois-Lacoste, Stützle, & Birattari, 2011). The parameters for the three methods NN, RBF and SVR can be found in Table 3.

The correct lags of the data were selected by following a procedure similar to that used in the case of WNs. For each city in the results section, we used the optimal parameters found by iRace and performed a backwards elimination of the nonsignificant lags.

4. Data description

For this study, we selected DATs for several cities from around the world. We used cities from four continents: Europe, America, Asia, and Australia. These cities were: Amsterdam, Berlin, Paris, Atlanta, Chicago, New York, Osaka, Tokyo and Melbourne. Temperature derivatives in these cities are traded actively through the CME. The data for the European cities were provided by the ECAD,[6] while data for the remaining cities were obtained from Bloomberg.

We have downloaded 11 years of DATs, resulting in 4,015 values between 1991 and 2001. Our dataset was split into in-sample and out-of-sample subsets. The in-sample subset was used to estimate the various models described in the previous section, while the out-of-sample data were used to evaluate the forecasting power of each method. The in-sample data consists of the first 10 years, i.e., 1991–2000, while the out-of-sample period is 2000–2001. Table 4 presents the descriptive statistics of the in-sample datasets. The mean temperature ranges from 9.94° C (Chicago) to 17.18° C (Atlanta). As we can observe, the variation in the DAT is large in every city. The standard deviation ranges from 4.60 in Melbourne to 10.80 in Chicago. In addition, the difference between the maximum and minimum temperatures is around 30° C in Melbourne, but 60° C in the case of Chicago. The maximum and minimum temperatures vary from city to city, but are explained by their location. These figures indicate that the temperature is very volatile, and is expected to be difficult to model and predict accurately. A closer inspection of Table 5 reveals that the descriptive statistics of the out-of-sample data set are similar.

In order for each year to have an equal number of observations, the 29th of February was removed from the data. Next, the seasonal mean and trend were removed from the data, using Eq. (5) for Alaton's method and Eq. (8) for Benth's and the GP, NNs, RBFs and SVR methods. In the case of WNs, the seasonal mean was captured using wavelet analysis (Alexandridis & Zapranis, 2013a).

In our analysis, all algorithms will be used to model and forecast detrended, deseasonalized DATs. We do this in order to avoid possible problems with over-fitting in the presence of seasonalities and periodicities. Then, the forecasts are transformed back to the original temperature time series in order to compare the performances of the algorithms.

The objective is to forecast two temperature indices accurately, namely accumulated HDDs and CAT. Temperature derivatives are commonly written on these two temperature indices. The PAC and CDD indices can be retrieved using the relationships in Eqs. (2) and (3).

5. Results

5.1. In-sample comparison: distributional statistics

In this section, we conduct an in-sample comparison of the seven models (Alaton, Benth, WN, GP, NN, RBF, SVR). More precisely, our comparison is based on a statistical analysis of the fit and the descriptive statistics of the residuals. The two linear models proposed by Benth and Alaton, as well as our proposed WN, assume that the residuals are independent and identically distributed (iid) and follow a normal distribution with mean zero and variance one, i.e., $e_t \sim N(0, 1)$.[7]

If the above assumption is violated, then the seasonal variance cannot be estimated correctly. In addition, if the residuals are not distributed independently, the proposed model is not complicated enough to explain the dynamics of the temperature evolution; furthermore, there are parts of the dynamics of the time series that are not captured by the model. As a result, such models cannot be used for forecasting, since the predicted values would be biased.

We test the above assumption by first examining the mean and standard deviation of the residuals. Then, the kurtosis and skewness are examined and a Kolmogorov–Smirnov (KS) normality test is performed in order to test the normality. The skewness should be equal to zero, while

[6] European Climate Assessment & Dataset project: http://eca.knmi.nl.

[7] Although a normal distribution is not necessary for either WN or GP, the assumption is very convenient for deriving closed form solutions of the pricing equations, as was presented by Benth and Saltyte-Benth (2007) and Alexandridis and Zapranis (2013a). We want to point out that this assumption, which is essential for the linear models, is violated frequently, leading to an underestimation of the variance, and therefore wrong pricing of the weather derivatives. On the other hand, when using WN or GP we can fit alternative distributions to the residuals and choose the correct one without restrictions. For example, Alexandridis and Zapranis (2013a) presented an extensive study of the selection of the distribution of the residuals of a temperature process using WN. We found the residuals to follow a hyperbolic distribution. Furthermore, we used WN and the hyperbolic distribution to derive the pricing equations of various weather derivatives. In addition, although this was not the aim of this study, WN can provide both confidence and prediction intervals, as was described by Alexandridis and Zapranis (2013b) and Alexandridis and Zapranis (2014). Similar procedures can be followed for the GP. Finally, the GP is used to forecast the temperature process and construct the temperature index, e.g., the CAT or HDD indices. This temperature index value can then be translated into its corresponding dollar value.

Table 3
Optimal parameters for the three benchmark non-linear models: SVR, RBF and NN.

SVR		RBF-k-means parameters		NN	
SVM Type	epsilon-SVR	Minimum standard deviation	1.53	Decay	True
Cost	5.32	NumClusters	46	Hidden layers	Number of lags
Gamma	3.23	Ridge	0.636	Learning rate	0.64
Kernel type	RBF			Momentum	0.5
Epsilon	1.39			Epochs	474

Table 4
Descriptive statistics of the daily temperature for the in-sample period: 1991–2000.

	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	p-value
Atlanta	17.18	8.21	32.50	18.05	−10.85	−0.36	2.18	57.84	0.0000
New York	13.32	9.47	33.89	13.35	−16.10	−0.16	2.09	51.58	0.0000
Chicago	9.94	10.80	33.60	10.28	−26.65	−0.26	2.27	43.42	0.0000
Melbourne	14.15	4.60	33.00	13.39	2.70	0.75	3.47	60.40	0.0000
Tokyo	16.31	7.67	32.50	16.50	1.00	0.09	1.85	60.10	0.0000
Osaka	16.30	8.59	33.50	16.50	−1.00	0.04	1.74	59.25	0.0000
Amsterdam	10.23	6.08	25.80	10.10	−10.90	−0.18	2.67	54.13	0.0000
Berlin	10.01	7.91	30.40	10.00	−14.70	−0.08	2.38	49.04	0.0000
Paris	12.51	6.44	29.90	12.40	−9.10	−0.04	2.48	56.38	0.0000

Table 5
Descriptive statistics of the daily temperature for the out-of-sample period: 2000–2001.

	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	p-value
Atlanta	16.97	7.47	28.06	17.78	−2.22	−0.46	2.23	18.38	0.0000
New York	13.93	9.24	33.89	13.89	−6.11	−0.10	1.88	16.47	0.0000
Chicago	10.43	10.39	29.44	11.11	−15.00	−0.22	2.07	14.16	0.0000
Melbourne	14.43	4.50	29.94	13.42	6.70	1.01	3.75	19.11	0.0000
Tokyo	15.96	7.89	31.00	16.50	1.50	−0.01	1.81	18.78	0.0000
Osaka	16.48	9.02	32.50	17.00	0.50	0.05	1.72	18.67	0.0000
Amsterdam	10.61	6.16	24.70	11.10	−3.60	−0.09	2.12	17.01	0.0000
Berlin	9.78	7.73	27.40	10.60	−7.20	0.00	2.00	14.75	0.0000
Paris	12.65	6.51	27.30	12.60	−1.60	0.06	2.22	18.11	0.0000

the kurtosis should be equal to three. The KS statistic quantifies the distance between the empirical distribution function of the sample and the cumulative distribution function (CDF) of the reference distribution; in our case, the normal distribution. Hence, the two hypotheses are:

H_0 : The data have the hypothesized, continuous CDF

H_1 : The data do not have the hypothesized, continuous CDF.

The critical value of the Kolmogorov–Smirnov test is 1.36 for a 95% confidence interval.

Finally, a Ljung–Box lack-of-fit hypothesis test is performed in order to test whether the residuals are iid. The Ljung–Box test is based on the Q statistic. The two hypothesis are:

H_0 : The data are distributed independently

H_1 : The data are not distributed independently,

and the Q statistic is given by:

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k},$$

where n is the sample size, $\hat{\rho}_k^2$ is the sample autocorrelation at lag k, and h is the number of lags being tested. The critical value of the Ljung–Box test is 31.41, for a confidence interval of 95%.

Table 6 provides descriptive statistics of the residuals of the Alaton model. The mean is almost zero and the standard deviation almost one for all cities. The kurtosis is positive (excessive) for all cities except Paris and New York, while the skewness is negative for all but Berlin, Amsterdam and Melbourne. The KS test results indicate that the normality hypothesis is rejected in Amsterdam, while there is not enough evidence to reject the normality hypothesis at the 10% confidence level for Berlin, New York or Paris. However, a closer inspection of Table 6 reveals very high values of the Ljung–Box lack-of-fit Q-statistic, revealing a strong autocorrelation in the residuals; i.e., the iid assumption is rejected. Hence, the results of the previous test for normality may not lead to substantial values of the KS test.

Table 7 provides descriptive statistics of the residuals of the Benth model. The standard deviation ranges between 0.56 and 0.82, in contrast to the initial hypothesis that the residuals follow a $N(0, 1)$ distribution. This has implications for the estimation of the seasonal variance. As the variance is underestimated, Benth's model will underestimate the prices of the corresponding temperature derivatives. In addition, the normality hypothesis is rejected in all cities. Finally, the Ljung–Box lack-of-fit Q-statistic reveals strong autocorrelation in the residuals. Hence, the forecast temperature values and prices of temperature derivatives will be biased, leading to large pricing errors.

Table 6
Descriptive statistics of the residuals of the Alaton model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
Atlanta	0.00	0.98	3.56	0.13	−5.30	−0.67	3.66	3.84	0.0000	205.86	0.0000
New York	0.00	0.97	3.11	0.03	−3.26	−0.08	2.94	1.05	0.2147	189.95	0.0000
Chicago	0.00	0.98	3.11	0.02	−3.56	−0.16	3.27	1.64	0.0092	141.65	0.0000
Melbourne	0.00	0.96	4.01	−0.09	−3.33	0.53	3.59	2.95	0.0000	292.24	0.0000
Tokyo	0.00	0.97	5.49	0.05	−5.91	−0.22	4.60	1.81	0.0028	297.02	0.0000
Osaka	0.00	0.99	7.02	0.01	−7.39	−0.22	5.55	1.65	0.0083	254.16	0.0000
Amsterdam	0.00	0.99	3.42	−0.04	−4.05	0.16	3.40	1.89	0.0015	193.43	0.0000
Berlin	0.00	0.99	4.40	−0.02	−3.85	0.01	3.43	0.99	0.2799	87.82	0.0000
Paris	0.00	0.99	3.03	0.00	−3.61	−0.13	2.95	0.75	0.6156	100.63	0.0000

St.Dev = standard deviation
K–S = Kolmogorov–Smirnov goodness-of-fit
LBQ = Ljung–Box lack-of-fit Q-statistic.

Table 7
Descriptive statistics of the residuals of the Benth model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
Atlanta	0.00	0.75	2.77	0.10	−3.79	−0.69	3.67	6.21	0.0000	187.31	0.0000
New York	0.00	0.67	2.15	0.02	−2.23	−0.10	3.01	6.20	0.0000	176.70	0.0000
Chicago	0.00	0.73	2.21	0.03	−2.76	−0.18	3.29	5.18	0.0000	136.55	0.0000
Melbourne	0.00	0.56	2.44	−0.05	−1.97	0.43	3.67	9.28	0.0000	159.90	0.0000
Tokyo	0.00	0.59	3.44	0.03	−3.65	−0.23	5.21	8.96	0.0000	88.85	0.0000
Osaka	0.00	0.67	5.03	0.01	−4.95	−0.23	6.08	6.86	0.0000	113.12	0.0000
Amsterdam	0.00	0.82	2.86	−0.03	−3.18	0.15	3.42	3.82	0.0000	197.39	0.0000
Berlin	0.00	0.81	3.60	−0.01	−3.25	0.00	3.49	3.91	0.0000	82.29	0.0000
Paris	0.00	0.80	2.53	0.00	−3.10	−0.15	2.99	3.57	0.0000	98.97	0.0000

St.Dev = standard deviation
K–S = Kolmogorov–Smirnov goodness-of-fit
LBQ = Ljung–Box lack-of-fit Q-statistic.

The results from the previous models indicate that the AR(1) model given by Eq. (7) is not complicated enough for modelling the time dependency of the temperature dynamics. This is evident from the strong autocorrelation found in the residuals. Moreover, the correct value of the seasonal variance is not computed, and there are indications that the choice of the Brownian noise process may not be correct. These conclusions reveal a very important limitation of the two linear models, which can have serious implications. The forecasts may not represent the real evolution of the temperature dynamics, leading to biased forecasts and a significant mispricing of weather derivatives.

On the other hand, a closer inspection of Table 8 reveals that the proposed WN model outperforms the two linear models in terms of distributional statistics of the residuals. First, in contrast to the models of Alaton and Benth, our tests indicate a absence of autocorrelation in the residuals, with an exception of the Asian cities. The normality hypothesis cannot be rejected in the cases of Berlin, New York and Paris, while being rejected at the 5% significance level but not the 1% level in the case of Amsterdam. For the remaining cities, normality is rejected, but the KS values are much smaller than in the alternative methods. Finally, wavelet analysis successfully identifies all of the seasonal cycles that affect the temperature dynamics. Hence, the initial assumption of the WN model holds, and the WN can be used for forecasting.

Next, we present the in-sample descriptive statistics of the residuals of the GP model. Note that no assumptions about the distributional properties of the residuals were made in the case of the GP. Thus, the results are only provided for the sake of completeness, and are presented in Table 9. Since GP is a stochastic algorithm, we present the results for both the best tree and the average performance. Panel A of Table 9 presents the descriptive statistics of the residuals of the best tree, while panel B presents the descriptive statistics of the mean residuals of the 50 trees. A closer inspection of the statistics reveals that the standard deviation is significantly larger than one and the KS test rejects the normality hypothesis. The max and min values of the residuals are significantly larger than for the other three methods. Finally, the Ljung–Box lack-of-fit Q-statistic reveals no autocorrelation in the residuals in Chicago, Amsterdam, Berlin and Paris at the 1% level. The results for the mean residuals of the 50 runs are similar, as is shown in panel B. Lastly, it should be noted that the best GP trees returned for each dataset follow a structure similar to that of the sample tree presented in Fig. 4.

Next, we focus on the results of the three benchmark nonlinear non-parametric models. Like the GP, these models do not make any assumptions about the distributional properties of the residuals. Thus, the results for NN, RBF, and SVR are only provided for completeness in the Appendix, in Tables A.1–A.3 respectively. As we can see, the mean is zero in the case of the RBF, but fluctuates around zero for the NNs and the SVR. The standard deviation ranges from 1.77 to 3.26 for all models, while normality is rejected for all cities. On the other hand, the autocorrelation hypothesis is rejected only for the Asian cities for the three benchmark models.

In conclusion, the normality hypothesis is rejected for the two linear models, while strong autocorrelation is evident in the residuals. The normality hypothesis is also

Table 8
Descriptive statistics of the residuals of the WN model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
Atlanta	−0.01	1.00	4.64	0.13	−4.85	−0.65	3.76	3.77	0.0000	32.70	0.0364
New York	0.00	1.00	3.62	0.03	−3.38	−0.09	3.06	0.96	0.3128	17.42	0.6256
Chicago	0.00	1.00	3.12	0.03	−4.04	−0.20	3.34	1.79	0.0033	19.65	0.4803
Melbourne	0.01	1.00	4.66	−0.07	−3.62	0.43	3.79	2.18	0.0001	74.33	0.0000
Tokyo	0.00	1.00	5.83	0.05	−5.91	−0.18	5.19	1.68	0.0070	60.12	0.0000
Osaka	0.01	1.00	7.13	0.01	−7.51	−0.22	6.09	1.98	0.0007	82.93	0.0000
Amsterdam	0.00	1.00	3.80	−0.04	−4.16	0.13	3.50	1.49	0.0237	23.07	0.2855
Berlin	0.00	1.00	4.45	0.00	−4.02	−0.02	3.53	0.96	0.3086	29.62	0.0763
Paris	0.00	1.00	2.89	0.02	−4.23	−0.17	3.01	0.89	0.3960	21.19	0.3859

St.Dev = standard Ddviation
K–S = Kolmogorov–Smirnov goodness-of-fit
LBQ = Ljung–Box lack-of-fit Q-statistic.

Table 9
Descriptive statistics of the residuals of the GP model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
(a) Panel A: Residuals of the best tree											
Atlanta	0.08	2.57	16.66	0.34	−11.03	−0.53	4.51	13.09	0.0000	68.14	0.0000
New York	−0.07	2.84	11.75	0.02	−12.07	−0.11	3.42	13.76	0.0000	63.93	0.0000
Chicago	−0.02	3.28	10.80	0.13	−12.64	−0.23	3.68	14.15	0.0000	30.10	0.0682
Melbourne	−0.05	2.52	11.87	−0.28	−8.25	0.63	4.57	12.90	0.0000	120.32	0.0000
Tokyo	0.00	2.08	10.37	0.11	−12.84	−0.35	5.06	9.29	0.0000	76.59	0.0000
Osaka	0.05	1.87	8.58	0.09	−12.67	−0.30	5.43	8.05	0.0000	88.85	0.0000
Amsterdam	0.10	1.78	7.06	0.02	−8.23	0.15	3.72	8.32	0.0000	25.90	0.1693
Berlin	−0.08	2.33	11.05	−0.08	−9.86	−0.03	3.68	11.82	0.0000	32.57	0.0376
Paris	−0.02	2.00	5.63	0.01	−8.46	−0.19	3.06	10.13	0.0000	36.50	0.0134
(b) Panel B: Mean residuals of the 50 runs											
Atlanta	0.01	2.58	15.97	0.30	−12.16	−0.61	4.59	12.30	0.0000	41.63	0.0031
New York	−0.04	2.85	11.50	0.04	−12.06	−0.11	3.42	13.54	0.0000	73.79	0.0000
Chicago	0.09	3.28	10.89	0.22	−12.82	−0.22	3.67	15.12	0.0000	40.45	0.0044
Melbourne	0.01	2.51	11.70	−0.19	−8.19	0.56	4.51	11.72	0.0000	97.85	0.0000
Tokyo	0.00	2.08	10.44	0.11	−13.39	−0.40	5.23	9.32	0.0000	99.05	0.0000
Osaka	0.02	1.86	8.00	0.06	−12.70	−0.27	5.14	7.47	0.0000	86.57	0.0000
Amsterdam	0.03	1.79	6.98	−0.04	−8.61	0.13	3.75	7.70	0.0000	55.93	0.0000
Berlin	0.01	2.33	11.13	0.00	−9.75	−0.03	3.71	11.11	0.0000	34.63	0.0222
Paris	0.02	2.00	5.58	0.06	−8.36	−0.20	3.06	10.52	0.0000	35.57	0.0173

Panel A shows the descriptive statistics for the residuals of the best tree, while Panel B shows those for the mean residuals of the 50 runs.
St.Dev = standard deviation
K–S = Kolmogorov–Smirnov goodness-of-fit
LBQ = Ljung–Box lack-of-fit Q-statistic.

rejected in the case of the GP. Finally, some autocorrelation is present, but the Q-statistic is very close to the critical value. The NN, RBF and SVR yield results similar to those from the GP. Finally, the WN outperforms the previous methods because it is the only model for which the resulting residuals are idd $N(0, 1)$, although the normality hypothesis is rejected for some cities. Similarly to the other methods, the autocorrelation was removed for all cities except the Asian ones.

5.2. Out-of-sample forecasting

In this section, we provide an out-of-sample validation of our proposed models. Our proposed nonlinear models are validated and compared against two forecasting methods proposed in prior studies, those of Alaton and Benth. In addition, they are also compared against three other state-of-the-art machine learning algorithms that are used commonly for regression problems: NNs, RBFs and SVR. These seven models will be used to forecast out-of-sample DATs for different periods. Usually, temperature derivatives are written for a period of a month or a season, and sometimes even for a year. Hence, DATs for one, two, three, six and 12 months will be forecast. The out-of-sample period corresponds to the period 1st January–31st December 2001, and each time interval starts on 1st January 2001. Note that the DATs from 2001 were not used in the estimation of the parameters of the seven models. Since we are studying nine cities (Atlanta, New York, Chicago, Melbourne, Tokyo, Osaka, Amsterdam, Berlin, Paris) and two indices (HDD, CAT) for five different time periods (1, 2, 3, 6 and 12 months) using two forecasting schemes (1-day-ahead, out-of-sample), the seven models (Alaton, Benth, GP, WN, NN, RBF, SVR) are *compared across 180 different datasets*.

The predictive power of each method will be assessed by computing the relative absolute percentage error (APE), given by:

$$APE = \left| \frac{y - \hat{y}}{y} \right|,$$

where y is the corresponding index and $\hat{y}$ is the index predicted according to each method.

Table 10
Relative percentage errors of the one-day-ahead CAT index.

1 month	Alaton	Benth	GPA	GPB	WN	SVR	NN	RBF
Atlanta	**0.79**%	1.56%	5.29%	2.41%	1.57%	5.22%	3.12%	3.42%
New York	11.18%	**9.49**%	52.80%	26.28%	13.27%	15.34%	16.09%	18.87%
Chicago	3.90%	7.24%	6.79%	6.89%	**2.77**%	10.00%	10.05%	8.85%
Melbourne	7.62%	**6.39**%	22.63%	8.20%	6.65%	8.16%	7.08%	7.47%
Tokyo	23.19%	25.75%	28.53%	28.21%	**19.47**%	20.97%	20.95%	21.60%
Osaka	13.33%	18.73%	15.98%	18.50%	**11.33**%	15.40%	17.36%	16.57%
Amsterdam	3.30%	5.43%	0.56%	0.26%	**0.19**%	21.32%	20.63%	20.31%
Berlin	9.49%	**1.71**%	7.83%	7.70%	3.92%	20.50%	19.63%	21.30%
Paris	2.30%	**0.66**%	2.59%	1.82%	0.86%	6.59%	6.35%	5.85%
2 months								
Atlanta	3.56%	2.00%	1.00%	1.70%	2.86%	**0.68**%	2.79%	2.73%
New York	**0.01**%	15.89%	42.28%	23.63%	4.20%	16.29%	14.85%	16.00%
Chicago	11.39%	15.12%	50.86%	14.51%	**9.61**%	14.66%	14.08%	11.64%
Melbourne	8.15%	**6.79**%	16.87%	8.63%	7.08%	8.93%	7.60%	7.79%
Tokyo	13.43%	17.02%	39.62%	18.76%	**11.39**%	12.98%	13.06%	14.06%
Osaka	5.36%	10.73%	30.12%	10.35%	**4.51**%	7.66%	9.05%	8.66%
Amsterdam	1.39%	3.34%	1.19%	**0.55**%	0.72%	16.09%	16.13%	15.68%
Berlin	1.95%	7.26%	**1.84**%	3.03%	4.10%	19.27%	19.26%	19.77%
Paris	1.64%	**0.13**%	2.07%	1.57%	0.41%	6.03%	6.07%	5.43%
3 months								
Atlanta	**0.01**%	1.20%	1.98%	1.44%	0.63%	2.31%	0.76%	0.70%
New York	8.07%	16.96%	33.09%	22.72%	**5.28**%	17.12%	17.34%	17.72%
Chicago	17.97%	24.51%	78.13%	23.38%	**16.25**%	26.89%	26.09%	23.05%
Melbourne	6.34%	**5.50**%	15.61%	7.17%	5.51%	6.98%	5.95%	6.21%
Tokyo	7.45%	11.42%	18.52%	12.57%	**6.87**%	8.52%	8.84%	9.12%
Osaka	3.25%	7.80%	14.19%	7.41%	**2.91**%	5.63%	6.68%	6.33%
Amsterdam	3.69%	5.21%	1.75%	2.92%	**0.36**%	17.28%	16.98%	16.85%
Berlin	8.08%	11.04%	7.42%	7.63%	**4.16**%	22.09%	22.12%	22.94%
Paris	2.01%	**0.72**%	2.30%	1.76%	0.82%	4.65%	4.47%	4.04%
6 months								
Atlanta	**0.57**%	1.23%	1.75%	1.36%	0.71%	1.97%	1.36%	1.31%
New York	**0.20**%	2.33%	2.75%	3.77%	0.41%	2.35%	2.35%	2.32%
Chicago	2.84%	5.24%	13.22%	5.12%	**2.60**%	5.51%	5.44%	4.78%
Melbourne	4.26%	4.11%	13.42%	5.75%	**3.73**%	4.90%	4.12%	4.43%
Tokyo	2.06%	4.44%	6.40%	5.00%	**1.82**%	3.22%	3.35%	3.45%
Osaka	0.61%	2.88%	4.59%	2.60%	**0.55**%	1.76%	2.24%	2.09%
Amsterdam	2.15%	2.83%	1.53%	2.34%	**0.14**%	8.46%	8.21%	8.32%
Berlin	3.72%	4.37%	3.25%	3.23%	**1.43**%	8.87%	8.78%	9.13%
Paris	0.37%	1.02%	**0.06**%	0.56%	0.11%	4.64%	4.43%	4.31%
12 months								
Atlanta	0.25%	0.45%	0.75%	0.54%	**0.16**%	1.09%	0.59%	0.45%
New York	1.83%	0.38%	**0.05**%	1.25%	1.68%	0.31%	0.28%	0.17%
Chicago	**0.12**%	1.79%	4.94%	1.84%	0.24%	1.93%	1.84%	1.49%
Melbourne	2.43%	2.82%	8.36%	4.33%	**2.13**%	3.20%	2.57%	2.86%
Tokyo	1.91%	3.74%	8.10%	4.18%	**1.54**%	2.88%	2.86%	3.13%
Osaka	0.35%	2.13%	5.67%	1.93%	**0.32**%	1.29%	1.65%	1.55%
Amsterdam	0.47%	1.13%	0.70%	1.29%	**0.09**%	2.75%	2.43%	2.62%
Berlin	1.50%	2.15%	1.32%	1.35%	**0.65**%	5.54%	5.28%	5.62%
Paris	0.59%	1.16%	0.37%	0.89%	**0.18**%	4.19%	3.93%	3.91%

5.2.1. Predictive performance

The models' predictive power will be evaluated using two out-of-sample forecasting methods. First, we will estimate out-of-sample forecasts over a specific period; second, we will estimate one-day-ahead forecasts over a specific period. In the first case, the out-of-sample forecasts, today's (time-step 0) temperature is known and is used to forecast the temperature tomorrow (time-step 1). However, tomorrow's temperature is unknown and cannot be used to forecast the temperature two days ahead. Hence, we use the temperature forecast at time-step 1 to forecast the temperature at time-step 2, and so on. We call this method the out-of-sample over a period forecast. For the second case, the one-day-ahead forecast, the procedure is as follows. The temperature today (time-step 0) is known, and is used to forecast tomorrow's temperature (time-step 1). Then tomorrow's real temperature is used to forecast the temperature at time-step 2, and so on. We will refer to this method as the one-day-ahead over a period forecast. Naturally, the second method is expected to be more accurate.

In the case of the stochastic GP algorithm, the temperature forecasts are calculated by computing the average of the 50 independent forecasting models (i.e., one forecasting model for each independent GP run), as was described at the end of Section 3.4.2. It should be noted here that, as GP is a stochastic algorithm, the average performance over the 50 runs (denoted by GPA) is used

Table 11
Relative percentage errors of the one-day-ahead HDD index.

1 month	Alaton	Benth	GPA	GPB	WN	SVR	NN	RBF
Atlanta	**0.34**%	0.67%	2.26%	1.03%	0.67%	2.23%	1.33%	1.46%
New York	0.77%	**0.65**%	3.62%	1.80%	0.91%	1.05%	1.10%	1.30%
Chicago	0.74%	1.37%	1.28%	1.30%	**0.52**%	1.89%	1.90%	1.67%
Melbourne	85.23%	54.17%	876.23%	4.26%	24.95%	**0.82**%	20.69%	12.64%
Tokyo	8.22%	9.12%	10.11%	10.00%	**6.90**%	7.43%	7.42%	7.65%
Osaka	3.97%	5.57%	4.76%	5.51%	**3.37**%	4.58%	5.17%	4.93%
Amsterdam	0.67%	1.11%	0.11%	0.05%	**0.04**%	4.34%	4.20%	4.14%
Berlin	0.65%	**0.12**%	0.53%	0.52%	0.27%	1.40%	1.34%	1.45%
Paris	1.09%	**0.31**%	1.23%	0.86%	0.41%	3.12%	3.01%	2.77%
2 months								
Atlanta	2.65%	1.47%	0.71%	1.24%	2.12%	**0.46**%	2.07%	2.03%
New York	**0.00**%	1.60%	4.26%	2.38%	0.42%	1.64%	1.50%	1.61%
Chicago	1.97%	2.62%	8.80%	2.51%	**1.66**%	2.54%	2.44%	2.01%
Melbourne	80.53%	55.04%	542.77%	10.40%	34.54%	**0.88**%	30.03%	8.54%
Tokyo	5.69%	7.21%	16.78%	7.94%	**4.82**%	5.50%	5.53%	5.96%
Osaka	1.97%	3.96%	11.10%	3.82%	**1.66%**	2.82%	3.34%	3.19%
Amsterdam	0.37%	0.88%	0.31%	**0.14**%	0.19%	4.25%	4.26%	4.14%
Berlin	0.19%	0.70%	**0.18**%	0.29%	0.40%	1.87%	1.86%	1.91%
Paris	0.85%	**0.07**%	1.07%	0.81%	0.21%	3.12%	3.14%	2.81%
3 months								
Atlanta	**0.05**%	1.14%	1.86%	1.37%	0.62%	2.17%	0.73%	0.68%
New York	1.33%	2.79%	5.44%	3.73%	**0.87**%	2.81%	2.85%	2.91%
Chicago	1.84%	2.51%	8.01%	2.40%	**1.67**%	2.76%	2.68%	2.36%
Melbourne	36.56%	21.09%	202.13%	**3.99**%	22.21%	8.16%	18.20%	12.28%
Tokyo	4.44%	6.82%	11.05%	7.50%	**4.10**%	5.08%	5.28%	5.44%
Osaka	1.68%	4.04%	7.35%	3.84%	**1.51**%	2.92%	3.46%	3.28%
Amsterdam	1.08%	1.53%	0.51%	0.86%	**0.10**%	5.08%	4.99%	4.95%
Berlin	1.07%	1.47%	0.99%	1.01%	**0.55**%	2.94%	2.94%	3.05%
Paris	1.36%	**0.48**%	1.56%	1.19%	0.56%	3.15%	3.03%	2.74%
6 months								
Atlanta	**1.11**%	2.18%	2.93%	2.42%	1.38%	3.04%	1.63%	1.55%
New York	2.76%	4.50%	8.05%	5.77%	**2.15**%	4.36%	4.48%	4.29%
Chicago	3.02%	3.98%	9.40%	3.82%	**2.79**%	4.05%	4.10%	3.75%
Melbourne	2.82%	0.76%	42.64%	5.83%	1.33%	1.62%	**0.27**%	1.18%
Tokyo	3.04%	6.44%	11.13%	7.23%	**2.91**%	4.48%	4.75%	4.82%
Osaka	1.31%	4.30%	8.63%	3.95%	**1.21**%	2.89%	3.55%	3.33%
Amsterdam	2.08%	2.61%	1.56%	2.25%	**0.03**%	7.23%	7.03%	7.11%
Berlin	2.91%	3.32%	2.55%	2.55%	**1.11**%	6.15%	6.13%	6.35%
Paris	1.86%	2.60%	1.30%	1.95%	**0.32**%	7.24%	7.11%	6.92%
12 months								
Atlanta	0.01%	1.67%	1.79%	1.90%	**0.00**%	2.47%	0.70%	0.35%
New York	**0.45**%	2.36%	4.06%	3.60%	0.67%	1.90%	2.00%	1.80%
Chicago	1.00%	2.51%	6.19%	2.48%	**0.75**%	2.32%	2.35%	1.94%
Melbourne	1.74%	1.17%	21.45%	5.05%	1.06%	1.42%	**0.05**%	1.01%
Tokyo	5.91%	9.07%	21.71%	9.98%	**4.95**%	6.92%	7.00%	7.49%
Osaka	3.14%	6.06%	18.91%	5.48%	**2.66**%	4.23%	5.01%	4.80%
Amsterdam	1.45%	2.19%	1.66%	2.55%	**0.16**%	4.28%	3.92%	4.12%
Berlin	2.45%	2.96%	2.20%	2.22%	**1.01**%	5.94%	5.78%	6.08%
Paris	2.91%	3.65%	2.34%	3.21%	**0.90**%	8.44%	8.17%	8.15%

for comparison purposes; in addition, we also present the performance of the best GP tree out of the 50 runs (denoted by GPB), as was explained in Section 3.4.2. In the cases of WNs, NNs and RBFs, we face the problem of local minima during training. Section 3.3 provided an analytical presentation of our way of dealing with this problem in the case of WNs, and a similar method is followed for NNs and RBFs. More precisely, we find the optimal architecture of the NNs and the RBFs in the first step, then we train 50 different models for each method in the second. Since the optimal architecture is used, we want to keep the network that produces the minimum of the loss function, i.e., the mean square error between the real and the fitted data. It is expected that the global minimum of the loss function can be found by following this method. The model that produces the minimum mean square error is used for forecasting.

The relative percentage errors are presented in Tables 10–13. The best value (i.e., lowest error) for each city and algorithm is shown in boldface. As we can see, WN has the lowest relative percentage errors for the most cities for the one-day-ahead predictions (Tables 10–11),

Table 12
Relative percentage errors of the out-of-sample CAT index.

1 month	Alaton	Benth	GPA	GPB	WN	SVR	NN	RBF
Atlanta	**11.50%**	16.05%	27.46%	29.52%	12.15%	34.56%	33.67%	29.00%
New York	**4.51%**	54.33%	50.49%	145.95%	31.43%	94.01%	87.01%	87.97%
Chicago	14.85%	28.30%	35.26%	31.88%	**14.70%**	46.46%	46.19%	47.16%
Melbourne	13.00%	14.30%	17.05%	14.78%	**12.20%**	13.12%	13.30%	13.58%
Tokyo	43.64%	61.61%	53.88%	63.36%	**36.23%**	70.62%	67.05%	68.65%
Osaka	31.95%	53.26%	44.90%	57.46%	**31.75%**	54.45%	56.90%	56.01%
Amsterdam	12.46%	23.61%	3.50%	10.85%	**3.21%**	33.64%	29.10%	32.21%
Berlin	43.01%	14.45%	34.35%	27.56%	27.71%	**11.33%**	18.89%	19.29%
Paris	15.94%	10.93%	15.59%	15.02%	10.89%	5.03%	3.23%	**2.07%**
2 months								
Atlanta	10.07%	5.78%	1.20%	**0.45%**	11.81%	5.01%	4.66%	0.63%
New York	**3.88%**	40.53%	41.34%	106.57%	24.47%	66.37%	61.56%	60.28%
Chicago	28.01%	45.86%	50.45%	48.49%	**26.62%**	63.92%	64.00%	65.29%
Melbourne	14.03%	15.43%	16.94%	15.75%	**13.41%**	14.17%	14.38%	14.66%
Tokyo	26.08%	42.05%	39.33%	43.72%	**18.94%**	50.16%	46.82%	48.34%
Osaka	13.80%	32.56%	29.80%	35.44%	**13.03%**	32.66%	34.87%	34.12%
Amsterdam	**1.81%**	11.73%	8.06%	10.79%	6.18%	22.38%	19.51%	22.98%
Berlin	12.87%	9.86%	8.60%	**5.41%**	5.61%	29.13%	34.82%	33.29%
Paris	11.34%	5.52%	11.50%	11.07%	6.47%	9.37%	7.42%	**1.75%**
3 months								
Atlanta	**0.44%**	3.96%	6.93%	7.38%	2.36%	12.99%	12.77%	8.86%
New York	18.72%	48.29%	49.88%	91.68%	**6.50%**	64.76%	61.67%	60.35%
Chicago	58.45%	91.69%	98.87%	96.53%	**56.15%**	124.22%	124.63%	127.14%
Melbourne	10.82%	12.35%	13.48%	12.65%	**10.10%**	10.99%	11.23%	11.52%
Tokyo	12.94%	25.81%	24.86%	27.16%	**6.63%**	32.31%	29.58%	30.82%
Osaka	6.99%	21.95%	21.13%	24.05%	**6.36%**	21.79%	23.57%	22.98%
Amsterdam	16.62%	26.23%	8.97%	**7.63%**	9.44%	36.88%	34.52%	38.05%
Berlin	**34.07%**	51.67%	35.31%	37.23%	37.05%	66.59%	70.94%	69.27%
Paris	9.66%	4.66%	10.08%	9.76%	5.65%	8.04%	6.31%	**1.30%**
6 months								
Atlanta	1.74%	4.54%	5.57%	5.70%	**0.20%**	9.38%	9.29%	6.85%
New York	**0.04%**	7.63%	8.25%	18.56%	2.56%	11.64%	10.87%	10.43%
Chicago	9.36%	19.93%	21.50%	21.16%	**8.74%**	29.40%	29.61%	30.41%
Melbourne	7.19%	9.12%	9.97%	9.45%	**5.81%**	7.43%	7.75%	8.10%
Tokyo	3.87%	10.72%	10.82%	11.43%	**0.01%**	14.17%	12.69%	13.37%
Osaka	1.56%	8.74%	8.95%	9.65%	**1.17%**	8.54%	9.41%	9.13%
Amsterdam	11.60%	16.53%	9.00%	8.89%	**8.64%**	22.16%	21.12%	23.03%
Berlin	17.61%	22.79%	**16.70%**	17.05%	16.99%	27.07%	28.31%	27.70%
Paris	2.02%	5.46%	**1.52%**	1.70%	4.54%	13.73%	12.54%	9.14%
12 months								
Atlanta	**1.19%**	1.35%	1.84%	1.91%	3.22%	5.36%	5.31%	3.12%
New York	4.65%	**0.99%**	1.38%	8.75%	6.38%	3.87%	3.31%	2.94%
Chicago	**0.94%**	6.00%	6.72%	6.26%	1.52%	12.10%	12.25%	12.79%
Melbourne	4.15%	6.35%	7.07%	6.70%	**2.52%**	4.39%	4.76%	5.16%
Tokyo	3.59%	9.05%	9.26%	9.52%	**0.46%**	11.80%	10.61%	11.16%
Osaka	0.89%	6.51%	6.80%	7.10%	**0.51%**	6.31%	6.99%	6.77%
Amsterdam	**2.03%**	5.92%	3.60%	3.73%	3.20%	10.42%	9.67%	11.22%
Berlin	6.82%	10.88%	**6.64%**	6.85%	6.75%	14.16%	15.12%	14.59%
Paris	2.36%	5.42%	**2.16%**	2.31%	4.85%	12.73%	11.66%	8.61%

followed by Alaton and Benth. The picture is similar for the out-of-sample predictions (Tables 12–13), but here the GP seems to have the lowest errors for some cities as well.

In summary, for the one-day-ahead forecasts, the WN outperformed the alternative methods in 53 of the 90 cases. The Alaton and Benth methods produced the most accurate forecasts 11 and 13 times each. GPA and GPB produced the most accurate forecasts four and three times, respectively. Finally, SVR, NN and RBF had the smallest forecasting errors in four, two and zero cases, respectively. For the out-of-sample forecasts, the WN outperformed the other methods in 41 of the 90 cases, followed by Alaton, GPA and GBP with 20, nine and seven cases respectively. On the other hand, RBF scored best in six cases, and Benth's model, SVR and NN scored best in three, two and two cases respectively.

In total, the WN had the best predictive performance in 52% of the samples, followed by Alaton with 18%. Benth's model had the best predictive performance in 9% of the cases, while the GPA and GPA were best in 7% and 6% respectively. It is worth mentioning that the performance of the GP increases to 9%, the same as Benth's model, when only the GPA or the GPB is considered. For the three benchmark models, SVR and RBF gave the best results in 3% of the cases each, while NN was last, with only 2%. A summary of these results is presented in Table 14.

Table 13
Relative percentage errors of the out-of-sample HDD index.

1 month	Alaton	Benth	GPA	GPB	WN	SVR	NN	RBF
Atlanta	**4.91**%	6.85%	11.72%	12.60%	5.19%	14.75%	14.37%	12.38%
New York	**0.31**%	3.73%	3.47%	10.02%	2.16%	6.45%	5.97%	6.04%
Chicago	2.81%	5.35%	6.66%	6.02%	**2.78**%	8.78%	8.73%	8.91%
Melbourne	100.00%	100.00%	101.32%	**89.60**%	100.00%	100.00%	100.00%	100.00%
Tokyo	15.46%	21.83%	19.09%	22.45%	**12.84**%	25.02%	23.76%	24.33%
Osaka	9.51%	15.85%	13.37%	17.10%	**9.45**%	16.21%	16.94%	16.67%
Amsterdam	2.54%	4.81%	0.71%	2.21%	**0.65**%	6.85%	5.93%	6.56%
Berlin	2.93%	0.98%	2.34%	1.88%	1.89%	**0.77**%	1.29%	1.31%
Paris	7.55%	5.18%	7.39%	7.12%	5.16%	2.38%	1.53%	**0.98**%
2 months								
Atlanta	7.62%	4.35%	0.86%	**0.29**%	8.94%	3.88%	3.61%	0.54%
New York	**0.39**%	4.08%	4.17%	10.74%	2.47%	6.69%	6.20%	6.07%
Chicago	4.85%	7.94%	8.73%	8.39%	**4.61**%	11.06%	11.07%	11.30%
Melbourne	95.06%	74.81%	**30.66**%	66.27%	99.15%	94.78%	92.30%	89.05%
Tokyo	11.05%	17.81%	16.66%	18.52%	**8.02**%	21.25%	19.83%	20.48%
Osaka	5.08%	12.00%	10.98%	13.06%	**4.80**%	12.04%	12.85%	12.57%
Amsterdam	**0.48**%	3.10%	2.13%	2.85%	1.63%	5.91%	5.15%	6.07%
Berlin	1.25%	0.95%	0.83%	**0.52**%	0.54%	2.82%	3.37%	3.22%
Paris	5.87%	2.86%	5.96%	5.73%	3.35%	4.85%	3.84%	**0.91**%
3 months								
Atlanta	**0.37**%	3.68%	6.41%	6.83%	2.13%	11.98%	11.77%	8.18%
New York	3.08%	7.93%	8.20%	15.06%	**1.07**%	10.64%	10.13%	9.92%
Chicago	5.99%	9.40%	10.14%	9.90%	**5.76**%	12.74%	12.78%	13.04%
Melbourne	43.15%	22.99%	**2.94**%	18.82%	52.18%	41.11%	37.89%	34.30%
Tokyo	7.72%	15.40%	14.84%	16.21%	**3.96**%	19.28%	17.66%	18.40%
Osaka	3.62%	11.37%	10.95%	12.46%	**3.30**%	11.30%	12.22%	11.91%
Amsterdam	4.89%	7.71%	2.64%	**2.24**%	2.78%	10.84%	10.15%	11.19%
Berlin	**4.53**%	6.87%	4.69%	4.95%	4.93%	8.85%	9.43%	9.21%
Paris	6.54%	3.16%	6.83%	6.61%	3.82%	5.44%	4.27%	**0.88**%
6 months								
Atlanta	**1.13**%	5.95%	8.60%	8.99%	1.77%	15.03%	14.82%	10.71%
New York	4.16%	10.45%	10.94%	19.27%	**1.63**%	13.80%	13.18%	12.85%
Chicago	6.37%	11.04%	11.83%	11.64%	**5.93**%	15.25%	15.33%	15.67%
Melbourne	2.63%	4.72%	8.37%	5.94%	8.47%	1.68%	**0.47**%	0.86%
Tokyo	5.57%	15.30%	15.03%	16.26%	**0.17**%	20.03%	18.02%	18.94%
Osaka	2.44%	12.08%	11.93%	13.29%	**1.72**%	11.86%	12.99%	12.62%
Amsterdam	10.19%	14.30%	8.02%	7.94%	**7.72**%	18.84%	18.01%	19.52%
Berlin	11.93%	15.08%	**11.53**%	11.76%	11.73%	17.68%	18.41%	18.06%
Paris	5.96%	10.18%	**5.40**%	5.65%	9.14%	20.07%	18.72%	14.74%
12 months								
Atlanta	6.42%	**0.41**%	1.22%	1.53%	11.12%	9.56%	9.39%	4.40%
New York	4.47%	**3.20**%	3.54%	13.09%	7.45%	7.08%	6.34%	5.89%
Chicago	**0.20**%	5.68%	6.28%	5.74%	0.49%	10.48%	10.59%	10.99%
Melbourne	1.51%	4.86%	7.08%	6.02%	6.44%	0.83%	**0.23**%	1.38%
Tokyo	10.50%	20.83%	20.55%	21.48%	**4.30**%	25.78%	23.66%	24.63%
Osaka	7.06%	17.31%	17.23%	18.27%	**6.14**%	17.03%	18.24%	17.84%
Amsterdam	**5.99**%	10.88%	7.91%	8.04%	7.45%	16.21%	15.32%	17.09%
Berlin	**9.61**%	13.14%	9.62%	9.83%	9.73%	15.98%	16.79%	16.36%
Paris	9.39%	13.86%	**9.02**%	9.24%	12.94%	24.14%	22.71%	18.50%

Specifically, Table 14 shows the absolute numbers and percentages of samples in which each method outperforms the others, i.e., has the best predictive accuracy.

We investigate the above results further by proceeding to rank the algorithms statistically. We do this by applying the non-parametric Friedman test with Hommel's post-hoc test (Demsar, 2006; Garcia & Herrera, 2008). The results of the test are presented in Table 15. More precisely, Table 15 presents the average rank of each algorithm,[8] along with the adjusted p-value according to Hommel's post-hoc test. The p-value represents the comparison between an algorithm's average rank and the algorithm with the best rank (control algorithm). The statistical tests were conducted for all different setups, i.e., the combined results of HDD and CAT, over both the one-day-ahead and out-of-sample experiments.

One can observe from Table 15 that the proposed WN ranks first and statistically outperforms every other method. Alaton's method ranks second, while Benth, GPB and GPA rank third, fourth and fifth, respectively, but

[8] The lower the average rank, the better the algorithm's performance.

Table 14
Predictive performances of all algorithms.

	One-day-ahead	Out-of-sample	Total
Alaton	11 (12%)	20 (22%)	31 (18%)
Benth	13 (14%)	3 (3%)	16 (9%)
GPA	4 (4%)	9 (10%)	13 (7%)
GPB	3 (3%)	7 (8%)	10 (6%)
WN	53 (59%)	41 (46%)	94 (52%)
SVR	4 (4%)	2 (2%)	6 (3%)
NN	2 (2%)	2 (2%)	4 (2%)
RBF	0 (0%)	6 (7%)	6 (3%)

The numbers of datasets on which each method has the best predictive accuracy. Percentages are reported in parentheses.

Table 15
Statistical test results according to the non-parametric Friedman test with Hommel's post-hoc test.

Algorithm	Average rank	Adjusted P_{Hommel}
WN (c)	2.1722	–
Alaton	3.0055	0.00012
Benth	4.3194	1.81E−16
GPB	4.8472	1.12E−24
GPA	4.8944	2.18E−25
RBF	5.4777	7.95E−37
NN	5.5444	3.31E−38
SVR	5.7388	1.47E−42

with their rankings being very close to each other. Then, RBF ranks sixth, and NN and SVR rank seventh and eight, respectively. From this, we can conclude, first, that WN is clearly superior to all of the other methods tested in this paper. Second, pairwise comparisons revealed that both GPA and GPB were able to outperform traditional state-of-the-art machine learning algorithms, such as NN and SVR, while there was no statistical significance relative to RBF. Third, the pairwise comparisons showed that GP's performance was not statistically different to that of Benth's model, a traditional model for temperature forecasting in the context of weather derivatives.

5.2.2. Comparisons over different forecasting horizons

In this section, we expand our analysis by studying the evolution of the error with respect the forecast horizon. Figs. 5–8 present the evolution of the error across forecasting horizons for each city. Fig. 5 presents the one-day-ahead results for the CAT index, while Fig. 6 presents the one-day-ahead results for the HDD index. Similarly, Figs. 7 and 8 present the results for the out-of-sample forecasting method for the CAT and HDD indices, respectively. In each figure, the x-axis represents the algorithms tested (Alaton, Benth, WN, GPA, GPB, SVR, NN, RBF), the y-axis represents the relative percentage errors, and the z-axis represents the different forecasting horizons (one month, two months, three months, six months, and 12 months).

A closer inspection of Fig. 5 reveals that the relative absolute error is more stable for the WN than for the alternative methods. Although the evolution of the error varies across cities, in general an increase in the error is observed at the mid-term horizon (three to six months) for the European and U.S. cities, while for Melbourne, Tokyo and Osaka, the error is very high in the short term and decreases as the forecasting horizon increases. On the other hand, the changes in the relative absolute error for the remaining methods are abrupt, with large spikes. Similar results are observed in Fig. 6. A closer inspection of Fig. 6 reveals that the error for the WN increases at the mid-term horizons for Melbourne, Chicago, New York, Atlanta and Amsterdam, while the opposite is true for Osaka and Tokyo. Finally, the error increases with the time horizon for Paris and Berlin. The error patterns for the remaining algorithms are similar, although the changes in the error between periods are more abrupt and large spikes are observed frequently.

Focusing on out-of-sample forecasting, Figs. 7 and 8 reveal that the relative absolute error for the European cities increases with the forecasting horizon, as expected. However, the opposite is true for the Asian cities. For the U.S. cities, the error increases until it reaches a maximum at the six-month horizon, after which it drops. Finally, it should be noted that the GPB usually outperforms the GPA, with a performance similar to Benth.

6. Conclusions

In this paper, we have proposed two novel nonlinear models, namely WN and GP, and compared them with three popular nonlinear models (NN, RBF, SVR) and two linear models that have been proposed previously in the literature in the context of temperature modelling and weather derivative pricing. The seven models were compared in the forecasting of two temperature indices, in nine cities in which weather derivatives are traded, for five different time periods, using two forecasting schemes. As a result, the seven models were compared over 180 datasets. Our results indicate that WNs outperformed all other models.

Both in-sample and out-of-sample comparisons were performed. The in-sample comparison was based on the distributional statistics of the residuals. An understanding of the dynamics that govern the residuals would provide additional information regarding the validity of the proposed models. We found that, in most cases, the initial assumption of normality was accepted only for the WN. In addition, strong autocorrelation was found in the residuals of the two linear models. As a result, only the residuals of the WN satisfy the initial assumptions. This is a very important limitation of the alternative methods, since they can lead to forecasts that do not represent the real evolution of the temperature dynamics, leading to biased forecasts and a significant mispricing of the weather derivatives.

In the out-of-sample comparison, we tested our models using two forecasting schemes, namely one-day-ahead forecasting and out-of-sample forecasting. In both cases, the WN outperformed all of the other methods, followed by Alaton, Benth and GP. The above results were also con-

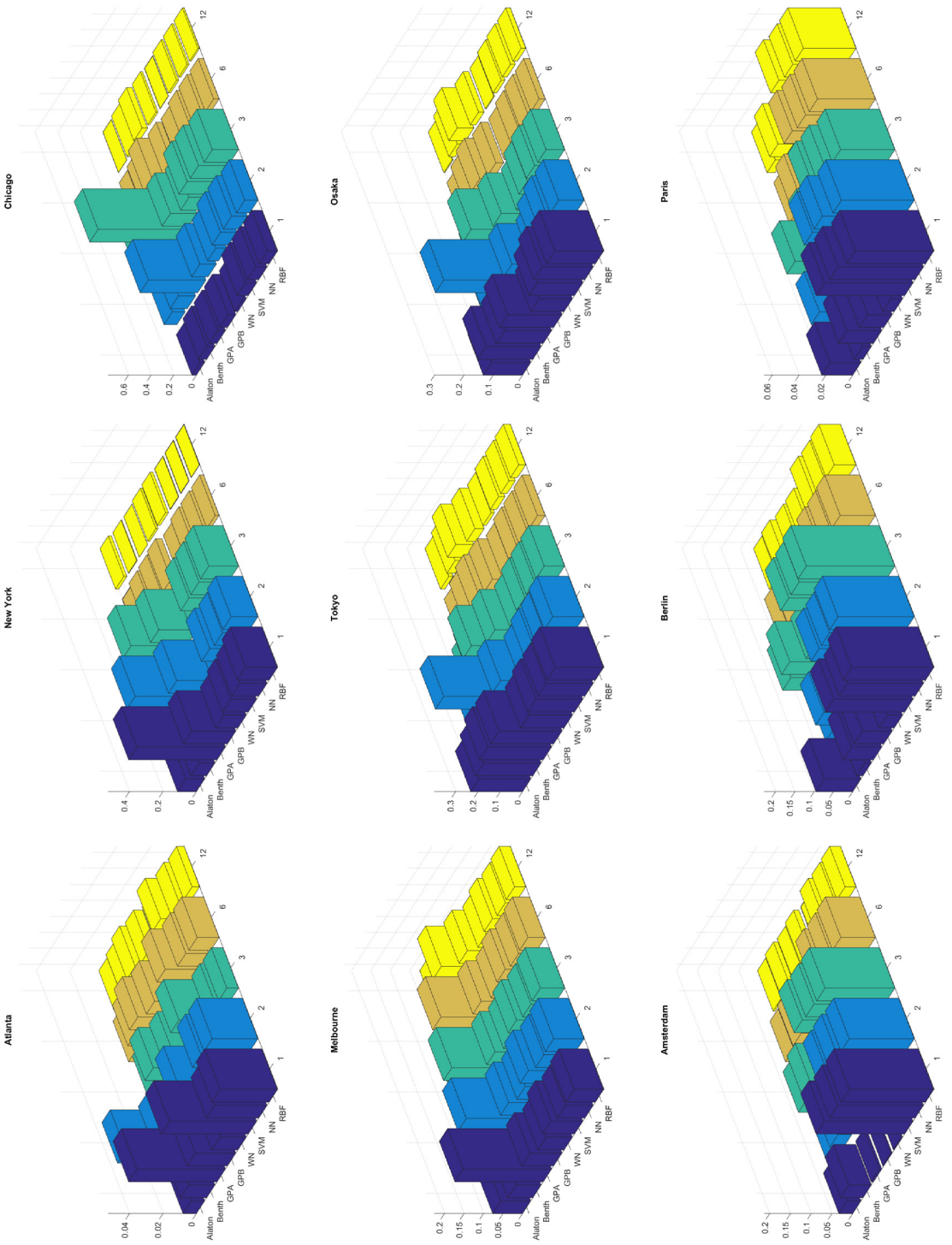

Fig. 5. Relative percentage errors of the one-day-ahead CAT index. The *x*-axis of each figure presents the algorithms tested, the *y*-axis presents the relative percentage errors, and the *z*-axis presents the different horizons (one month, two months, three months, six months, and 12 months).

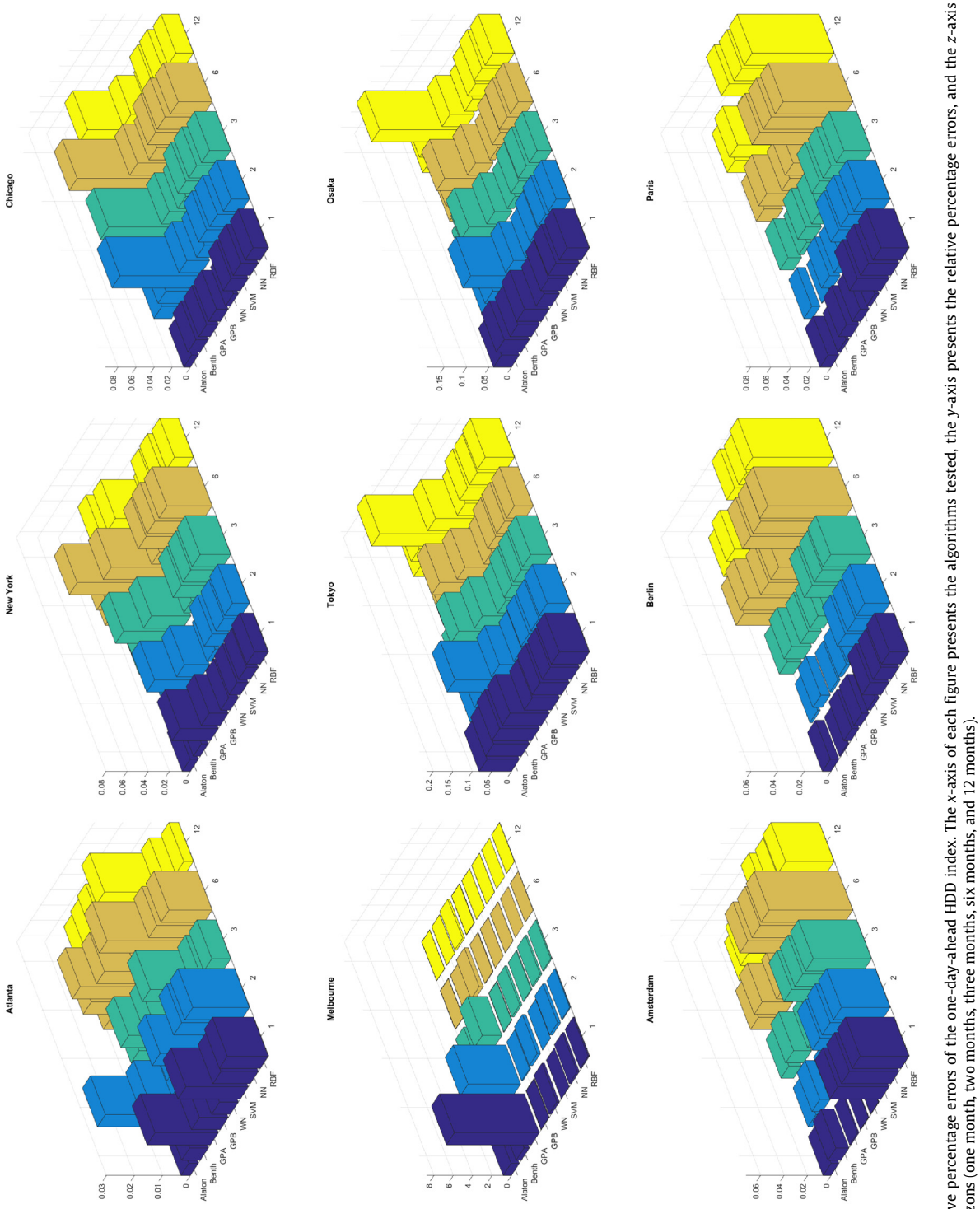

Fig. 6. Relative percentage errors of the one-day-ahead HDD index. The *x*-axis of each figure presents the algorithms tested, the *y*-axis presents the relative percentage errors, and the *z*-axis presents the different horizons (one month, two months, three months, six months, and 12 months).

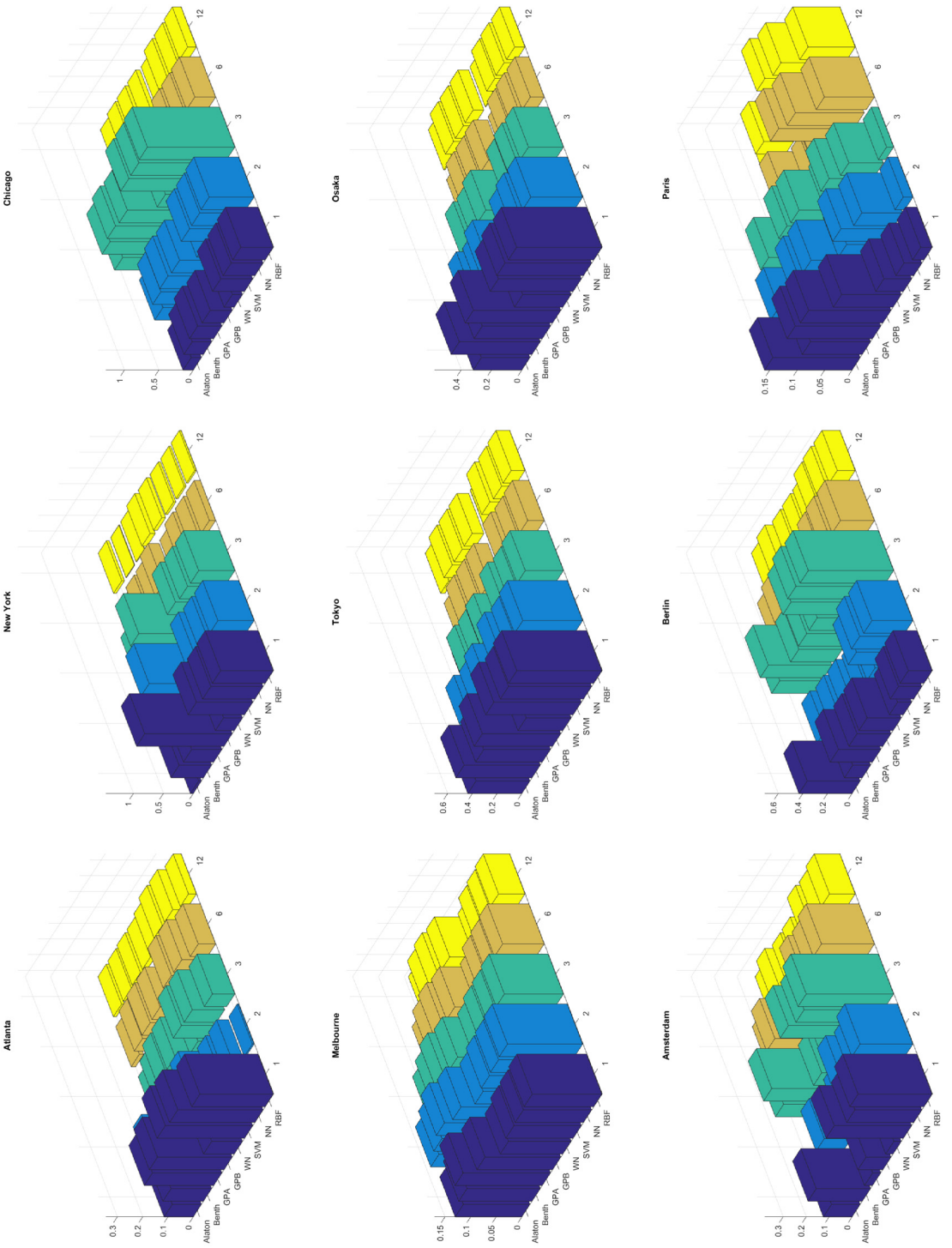

Fig. 7. Relative percentage errors of the out-of-sample CAT index. The *x*-axis of each figure presents the algorithms tested, the *y*-axis presents the relative percentage errors, and the *z*-axis presents the different horizons (one month, two months, three months, six months, and 12 months).

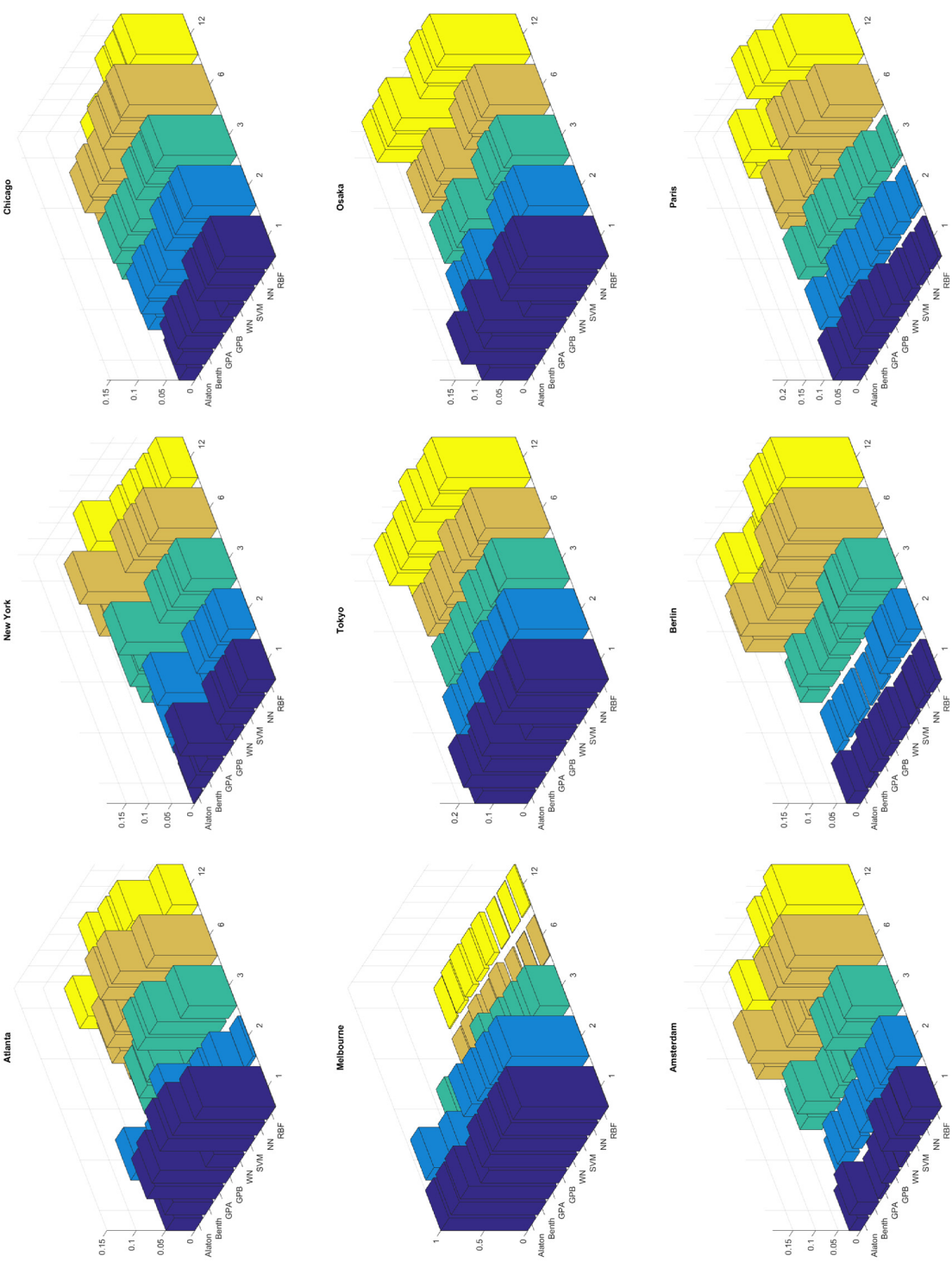

Fig. 8. Relative percentage errors of the out-of-sample HDD index. The *x*-axis of each figure presents the algorithms tested, the *y*-axis presents the relative percentage errors, and the *z*-axis presents the different horizons (one month, two months, three months, six months, and 12 months).

firmed by a non-parametric Friedman test, with Hommel's post-hoc test. The test revealed that the WN was ranked first, Alaton second, Benth third and GP fourth, followed by NN, RBF and SVR. It is worth mentioning that the difference between the rankings of Benth and GP was not statistically significant, while GP statistically outperformed two state-of-the-art machine learning algorithms (NN and SVR).

Finally, we examined the stability of each forecasting method relative to the forecasting horizon. Our results indicate that the WN outperforms the alternative methods, in the sense that the forecasting error is more stable. The error patterns for the remaining algorithms are similar, although the changes in the error between periods are more abrupt, and large spikes are observed frequently.

The previous analysis demonstrates our results to be very promising. Modelling the DAT using the proposed method (WNs) enhanced the predictive accuracy of the temperature process. WNs can model the dynamics of the temperature very well, and can constitute an accurate method for temperature derivatives pricing. The additional accuracy of the proposed model will have an impact on the accurate pricing of temperature derivatives. In addition, the GP outperformed state-of-the-art machine learning regression algorithms, as well as Benth's model for the out-of-sample forecasting, indicating the usefulness of GP for pricing weather contracts before the temperature measuring period.

There is a lot of future work that could be done on the WN and GP algorithms. At the moment, the GP fitness function is a simple MSE function, and is not tailored to the problem of weather derivatives. We believe that it would be beneficial to investigate other fitness functions, which would take into account the HDD and CAT indices. Furthermore, another potential extension of the fitness function would be to build in information about the pricing of weather derivatives, thus offering a generalized framework that can be applied to the pricing of temperature weather derivatives. In addition, instead of using a parametric equation for the seasonal mean, WNs could be used to approximate it nonlinearly and non-parametrically. We expect this method to provide a better fit to the data and to reveal the true dynamics of the evolution of the seasonal mean of the temperature. Furthermore, our promising results suggest that it would be worthwhile to examine the performances of more advance machine learning techniques, such as deep networks and self-organising fuzzy neural networks.

Acknowledgments

We would like to thank the associate editor and the anonymous referees for their constructive comments, which improved the final version of this paper substantially.

Appendix. In-sample descriptive statistics for NN, RBF and SVR

See Tables A.1–A.3.

Table A.1
Descriptive statistics of the residuals of the NN model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
Atlanta	−0.04	2.56	16.53	0.26	−11.27	−0.53	4.53	11.48	0.0000	25.43	0.1855
New York	−0.02	2.82	10.69	0.05	−11.34	−0.06	3.43	13.44	0.0000	18.10	0.5809
Chicago	−0.03	3.26	11.10	0.09	−12.51	−0.17	3.62	14.15	0.0000	18.36	0.5635
Melbourne	−0.04	2.50	11.41	−0.25	−8.31	0.54	4.54	12.09	0.0000	60.83	0.0000
Tokyo	0.01	2.08	10.31	0.10	−13.14	−0.29	5.25	9.15	0.0000	62.91	0.0000
Osaka	−0.01	1.87	8.44	0.03	−12.72	−0.23	5.06	7.72	0.0000	107.47	0.0000
Amsterdam	0.02	1.78	7.13	−0.05	−8.56	0.15	3.73	7.70	0.0000	22.23	0.3282
Berlin	0.04	2.32	11.05	0.03	−9.80	−0.02	3.69	11.32	0.0000	33.16	0.0324
Paris	−0.02	1.99	5.46	0.03	−7.17	−0.19	3.01	10.04	0.0000	21.21	0.3849

Table A.2
Descriptive statistics of the residuals of the RBF model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
Atlanta	0.00	2.54	14.60	0.28	−11.11	−0.54	4.38	11.86	0.0000	23.47	0.2664
New York	0.00	2.79	9.60	0.03	−11.33	−0.07	3.37	13.29	0.0000	17.91	0.5935
Chicago	0.00	3.24	11.11	0.12	−12.11	−0.16	3.57	14.36	0.0000	15.49	0.7479
Melbourne	0.00	2.48	11.45	−0.19	−8.34	0.51	4.51	11.37	0.0000	57.55	0.0000
Tokyo	0.00	2.03	10.23	0.10	−12.71	−0.25	4.81	8.88	0.0000	51.16	0.0002
Osaka	0.00	1.86	7.83	0.04	−12.72	−0.25	5.05	7.64	0.0000	108.61	0.0000
Amsterdam	0.00	1.77	7.11	−0.06	−8.78	0.14	3.72	7.52	0.0000	23.50	0.2648
Berlin	0.00	2.31	11.09	−0.01	−9.86	−0.02	3.70	10.95	0.0000	35.54	0.0174
Paris	0.00	1.98	5.39	0.03	−7.35	−0.19	3.01	10.09	0.0000	20.86	0.4054

Table A.3
Descriptive statistics of the residuals of the SVR model.

City	Mean	St.Dev	Max	Median	Min	Skewness	Kurtosis	K–S	*p*-value	LBQ	*p*-value
Atlanta	−0.14	2.56	15.82	0.17	−11.46	−0.58	4.58	10.61	0.0000	25.65	0.1779
New York	−0.03	2.82	10.82	0.01	−11.62	−0.08	3.52	13.05	0.0000	18.47	0.5566
Chicago	−0.05	3.26	10.93	0.06	−12.47	−0.18	3.66	14.06	0.0000	17.81	0.6001
Melbourne	0.06	2.49	11.62	−0.17	−8.38	0.59	4.73	10.71	0.0000	64.66	0.0000
Tokyo	0.03	2.06	10.21	0.11	−13.90	−0.25	5.37	8.97	0.0000	55.55	0.0000
Osaka	0.03	1.87	7.76	0.07	−12.69	−0.25	5.08	7.57	0.0000	105.71	0.0000
Amsterdam	−0.01	1.78	7.14	−0.08	−8.69	0.15	3.76	7.73	0.0000	22.09	0.3355
Berlin	−0.01	2.32	11.13	0.01	−10.01	−0.03	3.72	10.89	0.0000	35.47	0.0177
Paris	−0.04	1.98	5.44	−0.01	−7.38	−0.19	3.02	9.65	0.0000	21.78	0.3526

St.Dev = standard deviation.
K–S = Kolmogorov–Smirnov goodness-of-fit.
LBQ = Ljung–Box Q-statistic lack-of-fit.

References

Agapitos, A., OŃeill, M., & Brabazon, A. (2012a). Evolving seasonal forecasting models with genetic programming in the context of pricing weather-derivatives. In *Applications of evolutionary computation* (pp. 135–144). Springer.

Agapitos, A., OŃeill, M., & Brabazon, A. (2012b). Genetic programming for the induction of seasonal forecasts: A study on weather derivatives. In *Financial decision making using computational intelligence* (pp. 159–188). Springer.

Alaton, P., Djehince, B., & Stillberg, D. (2002). On modelling and pricing weather derivatives. *Applied Mathematical Finance*, *9*, 1–20.

Alexandridis, A.K., & Kampouridis, M. (2013). Temperature forecasting in the concept of weather derivatives: A comparison between wavelet networks and genetic programing. In *13th EANN*.

Alexandridis, A. K., & Zapranis, A. (2013a). *Weather derivatives: modeling and pricing weather-related risk*. New York: Springer.

Alexandridis, A. K., & Zapranis, A. (2014). *Wavelet networks: methodologies and applications in financial engineering, classification and chaos*. New Jersey, USA: Wiley.

Alexandridis, A. K., & Zapranis, A. D. (2013b). Wavelet neural networks: A practical guide. *Neural Networks*, *42*, 1–27.

Banzhaf, W., Nordin, P., Keller, R. E., & Francone, F. D. (1998). *Genetic programming an introduction: On the automatic evolution of computer programs and its applications, dpunkt*. San Francisco, California: Verlag and Morgan Kaufmann Publishers. Inc..

Becerikli, Y., Oysal, Y., & Konar, A. F. (2003). On a dynamic wavelet network and its modeling application. *Lecture Notes in Computer Science*, *2714*, 710–718.

Benth, F. E., & Saltyte-Benth, J. (2005). Stochastic modelling of temperature variations with a view towards weather derivatives. *Applied Mathematical Finance*, *12*(1), 53–85.

Benth, F. E., & Saltyte-Benth, J. (2007). The volatility of temperature and pricing of weather derivatives. *Quantitative Finance*, *7*(5), 553–561.

Benth, F. E., Saltyte-Benth, J., & Koekebakker, S. (2007). Putting a price on temperature. *Scandinavian Journal of Statistics*, *34*, 746–767.

Bernard, C., Mallat, S., & Slotine, J.-J. (1998). Wavelet interpolation networks. In *The proc. of ESANN '98* (pp. 47–52).

Billings, S. A., & Wei, H.-L. (2005). A new class of wavelet networks for nonlinear system identification. *IEEE Transactions on Neural Networks*, *16*(4), 862–874.

Brody, C. D., Syroka, J., & Zervos, M. (2002). Dynamical pricing of weather derivatives. *Quantitave Finance*, *2*, 189–198.

Broomhead, D. S., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, *2*, 321–355.

Burges, C.J.C. (1998). A tutorial on support vector machines for pattern recognition (pp. 121–167).

Caballero, R., & Jewson, S. (2002). Multivariate long-memory modeling of daily surface air temperatures and the valuation of weather derivative portfolios.

Cao, M., & Wei, J. (2000). *Pricing the weather. Risk weather risk special report*. (pp. 67–70). Energy And Power Risk Management, May.

Cao, M., & Wei, J. (2004). Weather derivatives valuation and market price of weather risk. *Journal of Future Markets*, *24*(11), 1065–1089.

Challis, S. (1999). Bright forecast for profits. Reactions June edition.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*, 27:1–27:27.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Davis, M. (2001). Pricing weather derivatives by marginal value. *Quantitative Finance*, *1*, 1–4.

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30.

Dorfleitner, G., & Wimmer, M. (2010). The pricing of temperature futures at the chicago mercantile exchange. *Journal of Banking & Finance*, *34*(6), 1360–1370.

Dornier, F., & Queruel, M. (2000). *Caution to the wind. Weather risk special report*. (pp. 30–32). Energy Power Risk Management, August.

Garcia, S., & Herrera, F. (2008). An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. *Journal of Machine Learning Research*, *9*(2677–2694), 66.

Geman, H., & Leonardi, M.-P. (2005). Alternative approaches to weather derivatives pricing. *Managerial Finance*, *31*(6), 46–72.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Exploration Newsletter*, *11*(1), 10–18.

Hanley, M. (1999). Hedging the force of nature. *Risk Professional*, *1*, 21–25.

Jewson, S., Brix, A., & Ziehmann, C. (2005). *Weather derivative valuation: the meteorological, statistical, financial and mathematical foundations*. Cambridge, UK: Campbridge University Press.

Koza, J. (1992). *Genetic Programming: On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.

López-Ibáñez, M., Dubois-Lacoste, J., Stützle, T., & Birattari, M. (2011). *The Rpackageirace package, iterated race for automatic algorithm configuration. Tech. rep.*. Belgium: IRIDIA, Université Libre de Bruxelles.

Miller, J., & Poli, R. (Eds.) (2010). *Tenth anniversary issue: progress in genetic progarmming and evolvable machines*. Springer.

Mitchell, T. (1997). *Machine learning*. Boston: McGraw-Hill.

Moreno, M. (2000). Riding the temp. Weather Derivatives, FOW Special Supplement December.

Pati, Y., & Krishnaprasad, P. (1993). Analysis and synthesis of feedforward neural networks using discrete affine wavelet transforms. *IEEE Transactions on Neural Networks*, *4*(1), 73–85.

Poli, R., Langdon, W. W. B., & McPhee, N. F. (2008). *Field guide to genetic programming*. Lulu Enterprises Uk Limited.

Rumelhart, D. E., McClelland, J. L., & PDP Research Group, C. (Eds.) (1986). *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1: foundations*. Cambridge, MA, USA: MIT Press.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc..

Zapranis, A., & Alexandridis, A. K. (2008). Modelling temperature time dependent speed of mean reversion in the context of weather derivetive pricing. *Applied Mathematical Finance*, *15*(4), 355–386.

Zapranis, A., & Alexandridis, A. K. (2009). Weather derivatives pricing: Modelling the seasonal residuals variance of an ornstein-uhlenbeck temperature process with neural networks. *Neurocomputing*, *73*, 37–48.

Zhang, Q. (1994). *Using wavelet network in nonparametric estimation. Tech. Rep. 2321, Techincal report*. INRIA.

Zhang, Q. (1997). Using wavelet network in nonparametric estimation. *IEEE Transactions on Neural Networks*, *8*(2), 227–236.

Zhang, Q., & Benveniste, A. (1992). Wavelet networks. *IEEE Transactions on Neural Networks*, *3*(6), 889–898.

Antonios K. Alexandridis is a Lecturer in Finance at the School of Mathematics, Statistics and Actuarial Science, University of Kent, UK. His research interests are close related to Artificial Intelligence and Financial Engineering. So far, he has published several research papers in leading, international and well recognized journals. He has also authored 2 books in the area of weather derivatives and wavelet networks (*Springer*: Weather Derivatives: Modeling and Pricing Weather-Related Risk, *Wiley*: Wavelet Neural Networks: Methodology and Applications in Financial Engineering, Classification and Chaos).

Michael Kampouridis is a lecturer at the School of Computing at the University of Kent, UK. His main research interests lie on the intersection of Computational Intelligence and Computational Finance. Areas of particular interest include algorithmic trading, financial forecasting, and intelligent decision support systems.

Sam Cramer is a Ph.D. student at the School of Computing at the University of Kent, UK. His main research interests lie on the intersection of Computational Intelligence and Computational Finance. Areas of particular interest include weather derivatives, financial forecasting, and intelligent decision support systems.