Genetic Programming for the Induction of Seasonal Forecasts: A Study on Weather-derivatives

Alexandros Agapitos, Michael O'Neill, Anthony Brabazon

Financial Mathematics and Computation Research Cluster
Natural Computing Research and Applications Group
Complex and Adaptive Systems Laboratory
University College Dublin, Ireland
alexandros.agapitos@ucd.ie, m.oneill@ucd.ie, anthony.brabazon@ucd.ie

Abstract. The last ten years has seen the introduction and rapid growth of a market in weather derivatives, financial instruments whose payoffs are determined by the outcome of an underlying weather metric. These instruments allow organisations to protect themselves against the commercial risks posed by weather fluctuations and also provide investment opportunities for financial traders. The size of the market for weather derivatives is substantial, with a survey suggesting that the market size exceeded $45.2 Billion in 2005/2006 with most contracts being written on temperature-based metrics. A key problem faced by buyers and sellers of weather derivatives is the determination of an appropriate pricing model (and resulting price) for the financial instrument. A critical input into the pricing model is an accurate forecast of the underlying weather metric. In this study we induce seasonal forecasting temperature models by means of a Machine Learning algorithm. Genetic Programming (GP) is applied to learn an accurate, localised, long-term forecast of a temperature profile as part of the broader process of determining appropriate pricing model for weather-derivatives. Two different approaches for GP-based time-series modelling are adopted. The first is based on a simple system identification approach whereby the temporal index of the time-series is used as the sole regressor of the evolved model. The second is based on iterated single-step prediction that resembles autoregressive and moving average models in statistical time-series modelling. The major issue of effective model generalisation is tackled though the use of an ensemble learning technique that allows a family of forecasting models to be evolved using different training sets, so that predictions are formed by averaging the diverse model outputs. Empirical results suggest that GP is able to successfully induce seasonal forecasting models, and that search-based autoregressive models compose a more stable unit of evolution in terms of generalisation performance for the three datasets considered. In addition, the use of ensemble learning of 5-model predictors enhanced the generalisation ability of the system as opposed to single-model prediction systems. On a more general note, there is an increasing recognition of the utility of evolutionary methodologies for the

modelling of meteorological, climatic and ecological phenomena, and this work also contributes to this literature.

1 Introduction

Weather conditions affect the cash flows and profits of businesses in a multitude of ways. For example, energy company sales will be lower if a winter is warmer than usual, leisure industry firms such as ski resorts, theme parks, hotels are affected by weather metrics such as temperature, snowfall or rainfall, construction firms can be affected by rainfall, temperatures and wind levels, and agricultural firms can be impacted by weather conditions during the growing or harvesting seasons [1]. Firms in the retail, manufacturing, insurance, transport, and brewing sectors will also have weather "exposure". Less obvious weather exposures include the correlation of events such as the occurrence of plant disease with certain weather conditions (i.e. blight in potatoes and in wheat) [2]. Another interesting example of weather risk is provided by the use of "Frost Day" cover by some of the UK town/county councils whereby a payout is obtained by them if a certain number of frost days (when roads would require gritting - with an associated cost) are exceeded. Putting the above into context, it is estimated that in excess of $1 trillion of activity in the US economy is weather-sensitive [3].

A key component of the accurate pricing of a weather derivative are forecasts of the expected value of the underlying weather variable and its associated volatility. The goal of this study is to produce seasonal predictive models by the means of Genetic Programming (GP) of the stochastic process that describes temperature. On a more general attempt to induce good-generalising seasonal models, an ensemble learning method (bagging) is employed to minimise high-variance models that are often associated with unstable learning algorithms as is the case of GP.

This contribution is organised as follows. Sections 2, 3 and 4 provide the background information to the problem domain tackled, as well as to the problem-solving methods employed. Background information is divided into three major parts. These are:

1. Section 2 introduces weather derivatives, discusses various methods for pricing these financial instruments, and finally motivates the need for seasonal temperature forecasting as part of a more general model for their pricing.
2. Section 3 introduces basic prior approaches to the task of seasonal temperature forecasting, and distinguishes between a number of possible scenarios in considering the use of weather forecast information for derivatives pricing. This section also motivates our choice of time-series index modelling.
3. Section 4 reviews the machine learning method of GP and its application to time-series forecasting with an emphasis on weather, climate, and ecology forecasting. The major statistical techniques for time-series modelling are also described in this section with the aim of linking these methods with

similar frameworks employed by GP-based time-series modelling systems. The ensemble learning method of *bagging* for improving model generalisation is also introduced in this section.

Following the background sections, Section 5 details our current scope of research. Section 6 describes the data utilised, the experimental setup, and the evolutionary model development framework adopted. Section 7 discusses the empirical findings, and finally Section 8 draws our conclusions.

2 A Brief Introduction to Weather Derivatives

2.1 Managing Weather Risk

In response to the existence of weather risk, a series of financial products have been developed in order to help organisations manage these risks. Usually, the organisation that wishes to reduce its weather risk buys "protection" and pays a premium to the seller who then assumes the risk. If the weather event occurs, the risk taker then pays an amount of money to the buyer. The oldest of these financial products are insurance contracts. However, insurance only provides a partial solution to the problem of weather risk as insurance typically concentrates on the provision of cover against damage to physical assets (buildings, machinery) or cash flows which arise from high-risk, low-probability, events such as floods or storm damage. The 1990s saw a convergence of capital and insurance markets and this led to the creation of additional tools for financial weather risk management. One example of this is provided by "catastrophe bonds" whereby a firm issues debt in the form of long-term bonds. The terms of these bonds include a provision that the payment of principal or interest (or both) to bondholders will be reduced in the event of specified natural disasters - thereby transferring part of the risk of these events to the bondholders. This reduction in capital or interest payments would leave the seller with extra cash to offset the losses caused by the weather disaster. As would be expected, the buyers of catastrophe bonds will demand a risk premium in order to compensate them for bearing this weather risk.

The above financial products do not usually provide cover against lower risk, higher probability, events such as the risk of higher than usual rainfall during the summer season, which could negatively impact on the sales and profits of (for example) a theme park. This "gap" in the risk transfer market for weather eventually led to the creation of a market for weather derivatives which allow counter-parties to trade weather risks between each other. In essence, weather derivatives are financial products that provide a payout which is related to the occurrence of pre-defined weather events [4]. These derivatives allow commercial organisations to reduce the volatility of future cash flows by hedging against one of the factors which contribute to volatility, namely the weather. Weather derivatives offer several advantages over insurance contracts as unlike insurance cover there is no need to file a claim or prove damages. Weather derivatives also permit a user to create a hedge against a "good" weather event elsewhere. For

example, for an agricultural firm, good weather in another location may increase the harvest in that locality, thereby reducing the price that the firm gets for its own produce due to over supply. Weather derivatives also remove the problem of 'moral hazard' that can occur under traditional insurance.

In addition to the trading of weather derivatives in order to manage weather risks, substantial trading in weather derivatives markets is driven by the trading of weather risk as an investment product. As weather is not strongly correlated with the systemic risk in general financial markets, weather derivatives represent an asset class which can provide diversification benefits for investors [5]. Weather derivatives also provide short-term traders with speculative investment possibilities as well as opening up cross trading strategies between weather and commodities markets (as both are impacted by weather) [5].

The scale of weather markets can be gleaned from the fifth annual industry survey by the Weather Risk Management Association (WRMA) (a Washington-based trade group founded in 1999) which suggests that the number of contracts transacted globally in the weather market had risen to more than 1,000,000 in the year ending March 2006, with a notional value of $45.2 billion [6].

2.2 Development of market for weather derivatives

The earliest weather derivative contracts arose in the US in 1997 [7]. A number of factors promoted their introduction at this time. Federal deregulation of the power sector created a competitive market for electricity. Before deregulation, utilities had the opportunity to raise prices to customers in the event of weather-related losses occurring, competition removed this safety net. This created a demand for financial products to allow the newly deregulated utilities to hedge against reductions in sales volume, caused by weather (temperature) fluctuations. Most of the early weather derivatives involved utilities and their imprint on the market remains in that the most-heavily traded weather derivatives are still temperature-based (for this reason, this paper concentrates on temperature-based derivatives). Apart from deregulation of the power sector, the 1997 El Nino brought an unusually mild winter to parts of the US. Many firms, including heating oil retailers, utilities and clothing manufacturers, saw their revenue dip during what should have been their peak selling season. This enhanced the visibility of weather-related risks. At the same time, the insurance industry faced a cyclical downturn in premium income, and seeking alternative income sources, was prepared to make capital available to hedge weather risks providing liquidity to the fledgling market [7].

The earliest weather derivatives were traded over-the-counter (OTC) as individually negotiated contracts. The absence of market-traded derivatives restricted the liquidity of the OTC market. In September 1999, the Chicago Mercantile Exchange (CME) (*www.cme.com*) created the first standardized, market-traded, weather derivatives (futures and options) and this led to a notable increase in their use. The CME also acted as a clearing house for all transactions, reducing substantially the counter-party risk faced by market participants. Currently the CME offer weather derivative contracts on a wide variety of underlying

weather metrics including temperature, rainfall, snowfall, frost and hurricanes. The most popular contracts are those based on temperature in 24 US cities including Colorado Springs, Las Vegas, Los Angeles, Portland,Sacramento, Salt Lake City, Tucson, Atlanta, Dallas, Houston, Jacksonville, Little Rock, Raleigh, Chicago, Cincinnati, Des Moines, Detroit, Kansas City, Minneapolis, Baltimore, Boston, New York, Philadelphia and Washington D.C. Weather derivatives are also available based on weather events outside the US.

2.3 OTC Weather Derivatives

Weather derivative contracts typically have a number of common attributes [8]:

- A contract period with a specified start and end date.
- A defined measurement station (location) at which the weather variable is to be measured.
- An index which aggregates the weather variable over the contract period.
- A payoff function which converts the index value into a monetary amount at the end of the contract period.

Contracts can be sub-divided into three broad categories [9]:

1. OTC weather derivatives.
2. Traded weather futures (equivalent to a swap - in essence this is a combined put and call option - each with the same strike price - with each party taking one side).
3. Traded weather options.

The earliest weather derivatives were traded over-the-counter (OTC) as individually negotiated contracts. In OTC contracts, one party usually wishes to hedge a weather exposure in order to reduce cash flow volatility. The payout of the contract may be linked to the value of a weather index on the Chicago Mercantile Exchange (CME) or may be custom-designed. The contract will specify the weather metric chosen, the period (a month, a season) over which it will be measured, where it will be measured (often a major weather station at a large airport), the scale of payoffs depending on the actual value of the weather metric and the cost of the contract. The contract may be a simple "swap" where one party agrees to pay the other if the metric exceeds a pre-determined level while the other party agrees to pay if the metric falls below that level. Thus if an energy firm was concerned that a mild winter would reduce demand for power, it could enter into a swap which would provide it with an increasing payout if average temperature over (for example) a month exceeded 66^oF. Conversely, to the extent that average temperature fell below this, the energy firm, benefiting from higher power sales, would pay an amount to the counterparty. OTC contracts usually have a fixed maximum payout and therefore are not open ended. As an alternative to swap contracts, contracts may involve call or put options. As an interesting example of an OTC contract, a London restaurant entered into a contract which provided for a payout based on the number of days in a

month when the temperature was less than 'x' degrees [9]. This was designed to compensate the restaurant for lost outdoor table sales when the weather was inclement.

In the US, many OTC (and all exchange-traded) contracts are based on the concept of a *"degree-day"*. A degree-day is the deviation of a day's average temperature from a reference temperature. Degree days are usually defined as either *"Heating Degree Days"* (HDDs) or *"Cooling Degree Days"* (CDDs). The origin of these terms lies in the energy sector which historically (in the US) used 65 degrees Fahrenheit as a baseline, as this was considered to be the temperature below which heating furnaces would be switched on (a heating day) and above which air-conditioners would be switched on (a cooling day). As a result HDDs and CDDs are defined as

$$HDD = \text{Max (0, } 65^{o}\text{F - average daily temperature)} \tag{1}$$

$$CDD = \text{Max (0, average daily temperature - } 65^{o}\text{F)} \tag{2}$$

For example, if the average daily temperature for December 20th is $36^{o}F$, then this corresponds to 29 HDDs (65 - 36 = 29). The payoff of a weather future is usually linked to the aggregate number of these in a chosen time period (one HDD or CDD is typically worth $20 per contract). Hence, the payoff to a December contract for HDDs which (for example) trades at 1025 HDDs on 1st December - assuming that there was a total of 1080 HDDs during December - would be $1,100 ($20 * (1080-1025)). A comprehensive introduction to weather derivatives is provided by [8].

2.4 Pricing a Weather Derivative

A substantial literature exists concerning the pricing of financial derivatives. However, models from this literature cannot be simply applied for pricing of weather derivatives as there are a number of important differences between the two domains. The *underlying* (variable) in a weather derivative (a weather metric) is non-traded and has no intrinsic value in itself (unlike the underlying in a traditional derivative which is typically a traded financial asset such as a share or a bond). It is also notable that changes in weather metrics do not follow a pure random walk as values will typically be quite bounded at specific locations. Standard (arbitrage-free) approaches to derivatives pricing (such as the Black-Scholes option pricing model [10]) are inappropriate as there is no easy way to construct a portfolio of financial assets which replicates the payoff to a weather derivative [11].

In general there are four methods used to price weather risk which vary in their sophistication:

1. *Business Pricing*. This approach considers the potential financial impact of particular weather events on the financial performance of a business. This information combined with the degree of risk adverseness of the business

(a utility function [12]), can help determine how much a specific business should pay for "weather insurance".
2. *Burn Analysis*. This approach uses historical payout information on the derivative in order to estimate the expected payoff to the derivative in the future. This approach makes no explicit use of forecasts of the underlying weather metric.
3. *Index modelling*. These approaches attempt to build a model of the distribution of the underlying weather metric (for example, the number of seasonal cumulative heating degree days), typically using historical data. A wide variety of forecasting approaches such as time-series models, of differing granularity and accuracy, can be employed. The fair price of the derivative is the expected value based on this, discounted for the time value of money.
4. *Physical models of the weather*. These employ numerical weather prediction models of varying time horizon and granularity. This approach can incorporate the use of monte-carlo simulation, by generating a large number of probabilistic scenarios (and associated payoffs for the weather derivative) with the fair price of the derivative being based on these, discounted for the time value of money [13].

As with financial asset returns, weather has volatility and hence, a key component of the accurate pricing of a weather derivative such as an option are forecasts of the underlying weather variable (an estimate of its expected value) and its associated volatility. As can be seen, the latter two methods above explicitly rely on the production of forecasts of the underlying variable using historic and/or current weather forecast information. This paper focuses on *index modelling*, whereby temperature composes the weather metric of interest. The section that follows contains a brief introduction to the complex task of weather forecasting for the purposes of pricing a weather derivative. Our discussion concentrates on seasonal temperature forecasting.

3 Weather Forecasting for Pricing a Weather Derivative

Weather forecasting is a complex process which embeds a host of approaches and associated time horizons. At one end of the continuum we have short-run weather forecasts which typically are based on structural physical models of atmospheric conditions (known as Atmospheric General Circulation Models - AGCMs). These models divide the atmosphere into a series of "boxes" of defined distance in north-south, east-west, and vertical directions. Starting from a set of initial conditions in each box, the evolution of atmospheric conditions is simulated forward in time using these values and the set of equations assumed to explain atmospheric conditions.

As the outputs from these models are sensitive to initial conditions the most common approach is to develop an ensemble forecast (which consists of multiple future weather scenarios, each scenario beginning from slightly different initial conditions). These models usually have good predictive ability up to about 10

days with rapidly reducing predictive ability after that. Forecasts produced by these models are relatively large-scale in nature and hence, to obtain a regional or localised weather forecast the output from the AGCM must be "downscaled" (this refers to the process of developing a statistical model which attempts to relate large-scale AGCM forecasts to the weather at a specific location). It should be noted that as weather derivatives are usually written for a specific location, course-grained forecasts from AGCMs are not especially useful for weather derivative pricing (at a specific location).

Longer term forecasts having a time horizon beyond one month are typically termed *seasonal forecasts* [14]. There are a variety of methods for producing these forecasts ranging from the use of statistical time series models based on historic data to the use of complex, course-grained, simulation models which incorporate ocean and atmospheric data. Given the range of relevant phenomena it has proven to be a very difficult task to build structural models for accurate long-term seasonal forecasting and non-structural time series approaches (which bypass atmospheric data and science) can produce long-run forecasts which are at least as good as those produced by structural models once the forecast horizon exceeds a few weeks [13]. Very long-term climate forecasts are also produced by various groups but these are not relevant for the purposes of weather derivative pricing.

In considering the use of weather forecast information for derivatives pricing, we can distinguish between a number of possible scenarios. As weather derivatives can often be traded long before the start of the relevant "weather period" which will determine the payoff to the derivative. In this case we can only use seasonal forecasting methods as current short run weather forecasts have no useful information content in predicting the weather than will arise during the weather period. The second case is that the derivative is due to expire within the next 10 or so days, so the current short-run weather forecast (along with the weather record during the recent past) has substantial information content in pricing the derivative. Obviously the closer the derivative gets to its expiry date, the less important the weather forecast will become, as the payoff to the derivative will have been substantially determined by weather that has already occurred. The final (and most complex) case is where the derivative has several weeks or months left to run in its weather period, hence its value will need to be ascertained using a synthesis of short-run weather forecasts and information from a longer-run seasonal forecast. The process of integrating these sources of information has been the subject of several studies [15].

3.1 Prior Approaches to Seasonal Temperature Forecasting

A number of prior studies have examined the prediction of seasonal temperature in the context of pricing weather derivatives. The historical time-series of temperatures for a given location exhibit the following characteristics [9]:

1. Seasonality.
2. Mean reversion.

3. Noise.

A simple linear model for capturing the seasonality component is proposed by [9] :

$$T_t^m = A + Bt + C\sin(\omega t + \varphi) \tag{3}$$

where T_t^m is the mean temperature at (day) time t, ω represents a phase angle as the maximum and minimum do not necessarily occur on 1st January and 1st July each year, φ represents the period of the seasonal temperature cycle ($2\pi/365$). Bt permits mean temperature to change each year, allowing for a general warming or cooling trend, and A provides an intercept term. Daily temperatures display marked mean-reversion, and this supports the idea that the process can be modelled using autoregressive methods. These models can capture the key properties of temperature behavior such as seasonality and other variations throughout the year [3]. The variance of temperatures is not constant during the annual cycle, varying between months but remaining fairly constant within each month [9]. In particular, variability of temperature is higher in winter (in the Northern Hemisphere) than in summer. Thus, the noise component is likely to be complex. In a study of this issue [16] noted that while assuming that the noise component was i.i.d. did result in reasonable predictions, they could be improved by allowing the distribution of the noise component to vary dynamically. In modeling temperature, attention can be restricted to discrete estimation processes [16]. Although temperature is continually measured, the values used to calculate the temperature metrics of interest (HDDs or CDDs) are discrete, as they both rely on the mean daily temperature.

Seasonal temperature forecasting can be reduced to the task of index modelling as discussed in Section 2.4. Two major families of *heuristic* and *statistical* time-series modelling methods are described in the next section, with the aim of introducing the general problem-solving framework employed.

4 Machine Learning of Time-series Forecasting Models

Modern machine learning *heuristic* methods for time-series modelling are based on two main natural computing paradigms, those of *Artificial Neural Networks* and *Evolutionary Automatic Programming* (EAP). Both methods rely on a training phase, whereby a set of adaptive parameters or data-structures are being adjusted to provide a model that is able to uncover sufficient structure in training data in order to allow useful predictions. This work makes use of the main thread of EAP that comes under the incarnation of Genetic Programming.

There are a number of reasons to suppose that the use of GP can prove fruitful in the seasonal modelling of the temperature at a specific location. As noted, the problem of seasonal forecasting is characterised by a lack of a strong theoretical framework, with many plausible, collinear explanatory variables. Rather than attempt to uncover a theoretical cause and effect model of local temperature for each location, this study undertakes a time-series analysis of historical temperature data for the locations of interest. A large number of functional forms,

lag periods and recombinations of historic data could be utilized in this process. This gives rise to a high-dimensional combinatorial problem, a domain in which GP has particular potential. The major issue of effective model generalisation is tackled though the use of an ensemble learning technique that allows a family of forecasting models to be evolved using different training sets, so that predictions are formed by averaging the diverse model outputs. This section introduces the GP paradigm and its application to time-series modelling. Special attention is given to the modelling of ecologic and atmospheric data. The dominant statistical time-series modelling methods are also reviewed in an attempt to motivate the forecasting model representations that will be employed as part of the evolutionary learning algorithm in later sections. Finally, ensemble learning and its impact on model generalisation is discussed in the final sub-section.

4.1 Genetic Programming

Genetic Programming [17–20] (GP) is an automatic programming technique that employs an Evolutionary Algorithm (EA) to search the space of candidate solutions, traditionally represented using expression-tree structures, for the one that optimises some sort of program-performance criterion. The highly expressive representation capabilities of programming languages allows GP to evolve arithmetic expressions that can take the form of regression models. This class of GP application has been termed "Symbolic Regression", and is potentially concerned with the discovery of both the functional form and the optimal coefficients of a regression model. In contrast to other statistical methods for data-driven modelling, GP-based symbolic regression does not presuppose a functional form, i.e. polynomial, exponential, logarithmic, etc., thus the resulting model can be an arbitrary arithmetic expression of regressors [21]. GP-based regression has been successfully applied to a wide range of financial modelling tasks [**?**].

GP adopts an Evolutionary Algorithm (EA), which is a class of stochastic search algorithms inspired by principles of *natural genetics* and *survival of the fittest*. The general recipe for solving a problem with an EA is as follows:

1. Define a *representation space* in which candidate solutions, computer programs, can be specified.
2. Design the *fitness criteria* for evaluating the quality of a solution.
3. Specify a *parent selection and replacement* policy.
4. Design a *variation mechanism* for generating offspring programs from a parent or a set of parents.

In GP, programs are usually expressed using hierarchical representations taking the form of syntax-trees, as shown in Figure 1. It is common to evolve programs into a constrained, and often problem-specific user-defined language. The variables and constants in the program are leaves in the tree (collectively named as terminal set), whilst arithmetic operators are internal nodes (collectively named as function set). In the simplest case of symbolic regression, the function set consists of basic arithmetic operators, while the terminal set consists of random numerical constants and a set of regressor variables. Figure

1 illustrates an example expression-tree representing the arithmetic expression $x + (2 - y)$.

GP finds out how well a program works by executing it, and then testing its behaviour against a number of test cases; a process reminiscent of the process of black-box testing in conventional software engineering practice. In the case of symbolic regression, the test cases consist of a set of input-output pairs, where a number of input variables represent the regressors and the output variable represents the regressand. GP relies on an error-driven model optimisation procedure, assigning program fitness that is based on some sort of error between the program output value and the actual value of the regressand variable. Those programs that do well (i.e. high fitness individuals) are chosen to be take part to a *program variation* procedure, and produce offspring programs. The primary program variation procedures that compose the main search operators of the space of computer programs are *crossover* and *mutation*.

The most commonly used form of crossover is *subtree crossover*, depicted in Figure 1. Given two parents, subtree crossover randomly (and independently) selects a cross-over point (a node) in each parent tree. Then, it creates two offspring programs by replacing the subtree rooted at the crossover point in a copy of the first parent with a copy of the subtree rooted at the crossover point in the second parent, and vice-versa. Crossover points are not typically selected with uniform probability. This is mainly due to the fact that the majority of the nodes in an expression-tree are leaf-nodes, thus a uniform selection of crossover points leads to crossover operations frequently exchanging only very small amounts of genetic material (i.e., small subtrees). To counteract this tendency, inner-nodes are randomly selected 90% of the time, while leaf-nodes are selected 10% of the time.

The dominant form of mutation in GP is *subtree mutation*, which randomly selects a mutation point in a tree and substitutes the subtree rooted there with a new randomly generated subtree. An example application of the mutation operator is depicted in Figure 1. Another common form of mutation is *point mutation*, which is roughly equivalent to the bit-flip mutation used in genetic algorithms. In point mutation, a random node is selected and the primitive stored there is replaced with a different random primitive of the same rarity taken from the primitive set. When subtree mutation is applied, this involves the modification of exactly one subtree. Point mutation, on the other hand, is typically applied on a per-node basis. That is, each node is considered in turn and, with a certain probability, it is altered as explained above. This allows multiple nodes to be mutated independently in one application of point mutation.

Like in any EA, the initial population of GP individuals is randomly generated. Two dominant methods are the *full* and *grow* methods, usually combined to form the *ramped half-and-half* expression-tree initialisation method [21]. In both the *full* and *grow* methods, the initial individuals are generated so that they do not exceed a user-specified maximum depth. The depth of a node is the number of edges that need to be traversed to reach the node starting from the tree's root node (the depth of the tree is the depth of its deepest leaf). The *full*

method generates full tree-structures where all the leaves are at the same depth, whereas the *grow* method allows for the creation of trees of more varied sizes and shapes.

4.2 Genetic Programming in Time-series Modelling

This section describes the approach adopted by GP in time-series forecasting with an emphasis to weather, climate, and ecology forecasting. In GP-based time-series prediction [22–24] the task is to induce a model that consists of the best possible approximation of the stochastic process that could have generated an observed time-series. Given *delayed vectors* v, the aim is to induce a model f that maps the vector v to the value x_{t+1}. That is,

$$x_{t+1} = f(v) = f(x_{t-(m-1)\tau}, x_{t-(m-2)\tau}, \ldots, x_t) \tag{4}$$

where m is embedding dimension and τ is delay time. The embedding specifies on which historical data in the series the current time value depends. These models are known as *single-step predictors*, and are used to predict to predict one value x_{t+1} of the time series when all inputs $x_{t-m}, \ldots, x_{t-2}, x_{t-1}, x_t$ are given. For long-term forecasts, *iterated single-step prediction models* are employed to forecast further than one step in the future. Each predicted output is fed back as input for the next prediction while all other inputs are shifted back one place. As a result, the input consists partially of predicted values as opposed to observables from the original time-series. That is,

$$\begin{aligned} x'_{t+1} &= f(x_{t-m}, \ldots, x_{t-1}, x_t); m < t \\ x'_{t+2} &= f(x_{t-m+1}, \ldots, x_t, x'_{t+1}); m < t \\ &\vdots \\ x'_{t+k} &= f(x_{t-m+k-1}, \ldots, x'_{t+k-2}, x'_{t+k-1}); m < t, k \geq 1 \end{aligned} \tag{5}$$

where k is the prediction step.

Long-term predictions involve a substantially more challenging task than short-term ones. The fact that each newly predicted value is partially dependent on previously generated predictions creates a reflexive relationship among program outputs, often resulting in inaccuracy propagation and an associated rapid fitness decrease with each additional fitness-case evaluation. Long-term forecasting models are generally sensitive to their initial output values, and inaccuracies of initial predictions are quickly magnified with each subsequent fitness evaluation iteration.

Examining prior literature reveals that evolutionary model induction methodologies have been applied to a number of problems in weather, climate and ecology forecasting. Examples include [25] which used GP to downscale forecasts based on course-grained Atmospheric General Circulation model outputs to estimate local daily extreme (maximum and minimum) temperatures. The results obtained from application of GP to data from the Chute-du-Diable weather

station in North Eastern Canada outperformed benchmark results from commonly used statistical downscaling models. GP has also been used for climate prediction problems including rainfall-runoff modelling [26], groundwater level fluctuations [27], short-term temperature prediction [28] and CO_2 emission modelling [29], the combination of ensemble forecasts [30], the forecasting of El Nino [31], evapotranspiration modelling (the process by which water is lost to the atmosphere from the ground surface via evaporation and plant transpiration) [32], modelling the relationship between solar activity and earth temperature [33], stream flow forecasting (forecasting of stream flow rate in a river) [34], modelling of monthly mean maximum temperature [35], modelling of water temperature [36], and wind prediction [37]. Hence we can see that there has been fairly widespread use of GP in this domain, although no previous application to the problem of seasonal forecasting was noted.

4.3 Statistical Time-series Forecasting Methods

Statistical time-series forecasting methods fall into the following five categories; the first three categories can be considered as linear, whereas the last two are non-linear methods:

1. Exponential smoothing methods.
2. Regression methods.
3. Autoregressive Integrated Moving Average methods (ARIMA).
4. Threshold methods.
5. Generalised Autoregressive Conditionally Heteroskedastic methods (GARCH).

In *exponential smoothing*, a forecast is given as a weighted moving average of recent time-series observations. The weights assigned decrease exponentially as the observations get older. In *regression*, a forecast is given as a linear combination of one or more explanatory variables. ARIMA models give a forecast as a linear function of past observations and error values between the time-series itself and past observations of explanatory variables. These models are essentially based on a composition of *autoregressive models* (linear prediction formulas that attempt to predict an output of a system based on the previous outputs), and moving average models (linear prediction model based on a *white noise* stationary time-series). For a discussion on smoothing, regression and ARIMA methods see [38]. Linear models cannot capture some featured that commonly occur in real-world data such as asymmetric cycles and outliers.

Threshold methods [38] assume that extant asymmetric cycles are cause by distinct underlying phases of the time-series, and that there is a transition period between these phases. Commonly, the individual phases are given a linear functional form, and the transition period is modelled as an exponential or logistic function. GARCH methods [39] are used to deal with time-series that display non-constant variance of residuals (error values). In these methods, the variance of error values is modelled as a quadratic function of past variance values and past error values.

Both linear and non-linear methods above, although capable of characterising features such as asymmetric cycles and non-constant variance of residuals, assume that the underlying data-generation process is stationary. For many real-world problems, this assumption is often invalid as shifting environmental conditions may cause the underlying data-generating process to change. In applying the statistical forecasting methods listed above, expert judgement is required to initially select the most appropriate method, and hence select an appropriate model-parameter optimisation technique. In the likely event that the underlying data-generating process is itself evolving, a modelling method must be reevaluated. This is one of the main reasons that forecasting models that can handle dynamic environments are desired.

4.4 Ensemble Learning for Model Generalisation

The idea of *supervised ensemble learning* is to induce multiple *base models*, and combine their predictions in order to increase *generalisation* performance, that is the performance on previously unseen instances. This was originally conceived in the context of *learning algorithm instability*, in which small changes in the training instances can lead to substantially different models with significant fluctuations in accuracy [40]. Ensembles of models approach the phenomenon of overfitting using the statistical concept of *bias-variance tradeoff*, under which the generalisation error of a model is decomposed into the sum of *bias* plus the *variance* [40]. Bias measures the extent to which the learned model is different from the target model, whereas variance measures the extent to which the learned model is sensitive on a particular sample training dataset [41].

There is a trade-off between bias and variance, with very flexible models having low bias and high variance, whereas relatively rigid models having high bias and low variance. To better illustrate the concept of bias and variance, consider that we are constructing a fixed model completely independent of a dataset. In this case, the bias will be high since we are not learning anything from the data, however the variance will vanish. In the opposite case, where we induce a function that fits the training data perfectly, the bias term disappears whereas the variance becomes pronounced. Best generalisation is achieved when we have the best balance between the conflicting requirements of small bias and small variance. Ensemble methods are typically based on inducing families of accurate models that are trained on various distributions over the original training dataset. They form an approach to minimise both bias and variance.

A *parallel ensemble* combines independently constructed accurate (low-bias) and diverse (low-variance) base models. In this case, an individual base model is trained on a specific subsample of the training instances, and the ultimate requirement is that different base models should make errors of different magnitude when confronted with new instances. Parallel ensembles obtain better generalisation performance than any single one of their components using a variance-reduction technique, and in the majority of cases, they are applied to unstable, high-variance learning algorithms (i.e. decision-tree induction, GP model induction [42]) *Bagging* [43] (bootstrap aggregation) is the earliest parallel ensemble

learning method that has been proven very effective for training *unstable* classifiers. The method creates multiple instances of the training dataset by using a bootstrapping technique [44]. Each of these different datasets are used to train a different model. The outputs of the multiple models are hence combined by averaging (in the case of regression), or voting (in the case of classification) to create a single output.

5 Scope of Research

The goal of this study is to produce predictive models of the stochastic process that describes temperature. More specifically, we are interested in modelling aggregate monthly HDDs using data from three US airport weather stations. Our main objective is to determine whether GP is capable of uncovering sufficient structure in historical data for a series of US locations, to allow useful prediction of the future monthly HDDs profile for those locations. The incorporation of the induced models into a complete pricing model for weather derivatives is left for future work. We also restrict attention to the case where the contract period for the derivative has not yet commenced. Hence, we ignore short-run weather forecasts, and concentrate on seasonal forecasting.

We investigate two families of program representations for time-series modelling. The first is the standard GP technique, genetic symbolic regression (GSR), applied to the forecasting problem in the same way that it is applied to symbolic regression problems. The task is to approximate a periodic function, where *temperature* (HDDs) is the dependent variable (regressand), and *time* is the sole regressor variable. The second representation allows the induction of iterated single-step predictors that can resemble autoregressive (GP-AR) and autoregressive moving average (GP-ARMA) time-series models that were described in Section 4.3. In an attempt to provide good generalising forecasting models, ensembles of predictors are evolved within the general bagging framework for training set resampling and model-output combination. The sections that follow describe the experiment design, discuss the empirical results, and draw our conclusions.

6 Experiment Design

6.1 Model Data

Three US weather stations were selected: (a) Atlanta (ATL); (b) Dallas, Fort Worth (DEN); (c) La Guardia, New York (DFW). All the weather stations were based at major domestic airports and the information collected included date, maximum daily temperature, minimum daily temperature, and the associated HDDs and CDDs for the day. This data was preprocessed to create new time-series of *monthly* aggregate HDDs and CDDs for each weather station respectively.

There is generally no agreement on the appropriate length of the time-series which should be used in attempts to predict future temperatures. Prior studies have used lengths of twenty to fifty years, and as a compromise this study uses data for each location for the period 01/01/1979 - 31/12/2002. The monthly HDDs data for each location is divided into a *training set* (15 years) that measures the performance during the learning phase, and a *test set* (9 years) that quantifies model generalisation.

6.2 Forecasting Model Representations and Run parameters

Table 1. Learning algorithm parameters

EA	panmictic, generational, elitist GP with an expression-tree representation
No. of generations	51
Population size	1,000
Tournament size	4
Tree creation	ramped half-and-half (depths of 2 to 6)
Max. tree depth	17
Subtree crossover	30%
Subtree mutation	40%
Point mutation	30%
Fitness function	Root Mean Squared Error (RMSE)

This study investigates the use of two families of seasonal forecast model representations, where the forecasting horizon is set to 6 months. The first is based on standard GP-based symbolic regression (GSR), where *time* serves as the regressor variable (corresponding to a month of a year), and *monthly HDDs* is the regressand variable. Assuming that time t is the start of the forecast, we can obtain a 6-month forecast by executing the program with inputs $\{t+1, \ldots, t+6\}$.

The second representation for evolving seasonal forecasting models is based on the iterated single-step prediction that can emulate autoregressive models, as described in Section 4.3. This method requires that delayed vectors from the monthly HDDs time-series are given as input to the model, with each consecutive model output being added at the end of the delayed input vector, while all other inputs are shifted back one place.

Table 2 shows the primitive single-type language elements that are being used for forecasting model representation in different experiment configurations. For GSR, the function set contains standard arithmetic operators (protected division) along with e^x, $log(x)$, $\sqrt{x}$, and finally the trigonometric functions of sine and cosine. The terminal set is composed of the index t representing a month, and random constants within specified ranges. GP-AR(12), GP-AR(24), GP-AR(36), all correspond to standard autoregressive models that are implemented as iterated single-step prediction models. The argument in the parentheses specifies the number of past time-series values that are available as input to the model. The function set in this case is similar to that of GSR excluding the

Table 2. Forecasting model representation languages

Forecasting model	Function set	Terminal set
GSR	`add`, `sub`, `mul`, `div`, `exp`, `log`, `sqrt`, `sin`, `cos`	index t corresponding to a month 10 rand. constants in -1.0, ..., 1.0 10 rand. constants in -10.0, ..., 10.0
GP-AR(12)	`add`, `sub`, `mul`, `div`, `exp`, `log`, `sqrt`	10 rand. constants in -1.0, ..., 1.0 10 rand. constants in -10.0, ..., 10.0 HDD_{t-1}, ..., HDD_{t-12}
GP-AR(24)	`add`, `sub`, `mul`, `div`, `exp`, `log`, `sqrt`	10 rand. constants in -1.0, ..., 1.0 10 rand. constants in -10.0, ..., 10.0 HDD_{t-1}, ..., HDD_{t-24}
GP-AR(36)	`add`, `sub`, `mul`, `div`, `exp`, `log`, `sqrt`	10 rand. constants in -1.0, ..., 1.0 10 rand. constants in -10.0, ..., 10.0 HDD_{t-1}, ..., HDD_{t-36}
GP-ARMA(36)	`add`, `sub`, `mul`, `div`, `exp`, `log`, `sqrt`	10 rand. constants in -1.0, ..., 1.0 10 rand. constants in -10.0, ..., 10.0 HDD_{t-1}, ..., HDD_{t-36} M(HDD_{t-1},..., HDD_{t-6}), SD(HDD_{t-1},..., HDD_{t-6}) M(HDD_{t-1},..., HDD_{t-12}), SD(HDD_{t-1},..., HDD_{t-12}) M(HDD_{t-1},..., HDD_{t-18}), SD(HDD_{t-1},..., HDD_{t-18}) M(HDD_{t-1},..., HDD_{t-24}), SD(HDD_{t-1},..., HDD_{t-24}) M(HDD_{t-1},..., HDD_{t-30}), SD(HDD_{t-1},..., HDD_{t-30}) M(HDD_{t-1},..., HDD_{t-36}), SD(HDD_{t-1},..., HDD_{t-36})

trigonometric functions, whereas the terminal set is augmented with historical monthly HDD values. For the final model configuration, GP-ARMA(36), the function set is identical to the one used in the other autoregressive models configurations, however the terminal set contains moving averages, denoted by $M(HDD_{t-1}, \ldots, HDD_{t-\lambda})$, where λ is the time-lag and HDD_{t-1} and $HDD_{t-\lambda}$ represent the bounds of the moving average period. For every moving average, the associated standard deviation for that period is also given as model input, and is denoted by $SD(HDD_{t-1}, \ldots, HDD_{t-\lambda})$. Finally, Table 1 presents the parameters of our learning algorithm.

6.3 Bagging of GP Time-series Models

Bagging produces redundant training sets by sampling with replacement from the original training instances. This effectively produces training sets that focus on various distributions over the original learning points. For a number of trials equal to the ensemble size, a training set of equal size to the original training set is sampled with replacement from the original instances. This means that some instances may not appear in it while others appear more than once. An independent GP time-series model is being evolved for every bootstrapped training set, and the outputs of the multiple models are hence combined using a simple averaging procedure in order to predict unseen instances. In this study we are considering ensembles of sizes 5, 10 and 20 independent predictors.

Table 3. Comparison of training and testing RMSE obtained by different forecasting configurations, each experiment was ran for 50 times. Standard error for mean in parentheses. Bold face indicates best performance on test data for single base models. Bold face combined with underline indicates best test performance among all experiment series.

Dataset	Forecasting configuration	Mean Training RMSE	Best Training RMSE	Mean Testing RMSE	Best Testing RMSE
ATL	GSR	140.52 (9.55)	68.82	149.53 (8.53)	**72.73**
	GP-AR(12)	92.44 (0.54)	81.78	111.87 (0.41)	103.60
	GP-AR(24)	91.33 (0.68)	83.33	96.15 (0.51)	91.26
	GP-AR(36)	88.96 (0.81)	77.30	90.38 (0.81)	79.44
	GP-ARMA	85.20 (0.86)	75.84	85.71 (0.82)	74.31
DEN	GSR	165.76 (11.46)	103.09	180.46 (11.74)	**95.23**
	GP-AR(12)	133.18 (0.43)	121.38	126.78 (0.25)	117.19
	GP-AR(24)	130.41 (0.73)	111.48	124.36 (0.66)	110.31
	GP-AR(36)	131.13 (1.08)	114.86	111.41 (0.57)	103.73
	GP-ARMA	126.46 (1.29)	106.18	108.90 (0.64)	101.57
DFW	GSR	118.96 (8.02)	66.49	118.69 (7.20)	66.12
	GP-AR(12)	88.75 (0.66)	80.64	90.37 (0.26)	86.57
	GP-AR(24)	96.14 (0.95)	83.55	85.36 (0.42)	78.24
	GP-AR(36)	89.52 (0.69)	81.12	62.11 (0.43)	55.84
	GP-ARMA	87.09 (0.82)	75.41	60.92 (0.52)	**55.10**

Dataset	Forecasting configuration	Ensemble size	Mean Training RMSE	Best Training RMSE	Mean Testing RMSE	Best Testing RMSE
ATL	GSR	5	144.90 (4.62)	82.82	150.26 (4.27)	93.29
	GP-AR(12)	5	90.70 (0.38)	84.62	111.40 (0.28)	106.94
	GP-AR(24)	5	85.22 (0.49)	77.32	92.06 (0.29)	88.13
	GP-AR(36)	5	80.01 (0.40)	75.08	80.94 (0.57)	75.65
	GP-ARMA	5	81.60 (0.83)	75.60	80.57 (0.34)	**70.96**
DEN	GSR	5	247.27 (22.70)	121.47	215.87 (7.70)	108.38
	GP-AR(12)	5	131.47 (0.36)	123.37	136.36 (11.13)	120.14
	GP-AR(24)	5	127.64 (0.60)	116.79	122.04 (0.50)	114.35
	GP-AR(36)	5	123.73 (0.86)	110.45	106.42 (0.44)	**92.93**
	GP-ARMA	5	116.86 (0.51)	109.19	109.38 (0.48)	103.49
DFW	GSR	5	165.29 (3.75)	87.93	145.76 (4.05)	75.76
	GP-AR(12)	5	87.11 (0.42)	80.91	89.20 (0.22)	82.71
	GP-AR(24)	5	87.65 (0.49)	80.99	79.21 (0.33)	74.66
	GP-AR(36)	5	86.41 (0.44)	79.74	59.56 (0.33)	**53.07**
	GP-ARMA	5	87.16 (0.60)	77.40	67.20 (0.17)	63.71
ATL	GSR	10	261.62 (18.76)	153.55	190.13 (2.99)	133.39
	GP-AR(12)	10	91.07 (0.30)	85.90	111.71 (0.23)	108.17
	GP-AR(24)	10	85.65 (0.49)	81.25	91.53 (0.21)	88.32
	GP-AR(36)	10	78.82 (0.28)	74.62	79.44 (0.25)	76.43
	GP-ARMA	10	79.95 (0.43)	75.14	80.00 (0.26)	77.67
DEN	GSR	10	295.79 (4.46)	223.11	287.47 (4.73)	203.87
	GP-AR(12)	10	131.20 (0.27)	125.50	125.15 (0.18)	120.60
	GP-AR(24)	10	128.37 (0.41)	122.67	122.59 (0.36)	118.53
	GP-AR(36)	10	122.99 (0.70)	115.29	105.68 (0.31)	101.55
	GP-ARMA	10	116.52 (0.34)	112.35	109.26 (0.37)	104.42
DFW	GSR	10	152.20 (5.85)	117.91	144.53 (2.35)	109.15
	GP-AR(12)	10	92.88 (5.21)	83.11	94.55 (0.65)	87.53
	GP-AR(24)	10	87.02 (0.25)	82.93	78.80 (0.18)	76.47
	GP-AR(36)	10	84.98 (0.35)	80.19	58.91 (0.27)	54.32
	GP-ARMA	10	86.97 (0.50)	79.66	66.82 (0.14)	63.70
ATL	GSR	20	245.24 (3.97)	189.02	206.78 (1.79)	178.42
	GP-AR(12)	20	90.76 (0,78)	86.16	110.44 (0.20)	107.24
	GP-AR(24)	20	85.05 (0.24)	82.21	91.21 (0.14)	89.50
	GP-AR(36)	20	78.76 (0.24)	75.95	78.82 (0.13)	77.51
	GP-ARMA	20	79.26 (0.13)	76.18	79.19 (0.16)	76.95
DEN	GSR	20	336.83 (4,43)	286.20	323.56 (3.15)	270.80
	GP-AR(12)	20	131.16 (0.22)	127.63	125.22 (0.14)	123.32
	GP-AR(24)	20	127.53 (0.27)	123.45	121.87 (0.24)	118.17
	GP-AR(36)	20	123.33 (0.52)	115.27	105.91 (0.30)	102.10
	GP-ARMA	20	116.26 (0.40)	111.86	108.52 (0.23)	105.34
DFW	GSR	20	215.47 (2.97)	179.29	189.28 (1.57)	166.87
	GP-AR(12)	20	87.32 (2.09)	82.32	88.90 (0.11)	86.24
	GP-AR(24)	20	85.88 (0.20)	79.72	78.41 (0.12)	76.62
	GP-AR(36)	20	85.40 (0.23)	82.31	59.11 (0.20)	56.43
	GP-ARMA	20	86.37 (0.20)	80.95	67.19 (0.16)	65.19

7 Results

We performed 50 independent evolutionary runs for each forecasting model configuration presented in Table 2. A summary of average and best training and test results obtained using each model configuration is presented in Table 3. The first part of the table refers to single-model forecasting, while the second part presents the results obtained by multi-model predictions using different ensemble sizes. The distributions of test-data RMSE obtained by best-of-run models are illustrated in Figures 2, 3, 4 for ATL, DEN, and DFW datasets respectively.

For the case of single-model forecasting, the results suggest that the family of autoregressive moving average models perform better on average than those obtained with standard symbolic regression. A statistically significance difference (unpaired t-test, $p < 0.0001$, degrees of freedom $df = 98$) was found between the average test RMSE for GSR and GP-ARMA in all three datasets. Despite the fact that the ARMA representation space offers a more stable unit for evolution than the essentially free-of-domain-knowledge GSR space, best testing RMSE results indicated that GSR models are better performers in ATL and DEN datasets, as opposed to the DFW dataset, where the best-of-50-runs GP-ARMA model appeared superior. Given that in time-series modelling it is often practical to assume a *deterministic* and a *stochastic* part in a series' dynamics, this result can well corroborate on the ability of standard symbolic regression models to effectively capture the deterministic aspect of a time-series, and successfully forecast future values in the case of time-series with a weak stochastic or volatile part. Another interesting observation is that there is a difference in the generalisation performance between GP-AR models of different order, suggesting that the higher the order of the AR process the better its performance on seasonal forecasting. Statistically significant differences (unpaired t-test, $p < 0.0001$, $df = 98$) were found in mean test RMSE between GP-AR models of order 12 and those of order 36, in all three datasets. During the learning process, we monitored the test-data performance of the best-of-generation individual, and we adopted a model selection strategy whereby the best-generalising individual from all generations is designated as the outcome of the run. Figures 5(a), (b), (c) illustrate the distributions of the generation number where model selection was performed, for the three datasets. It can be seen that GSR models are less prone to overfitting, then follows GP-ARMA, and finally it can be noted that GP-AR models of high order are the most sensitive to overfitting the training data. Interestingly is this fact is observed across all three datasets. In addition to this observation, Figure 9 illustrates the RMSE curves during training. It can be seen that under the GSR model configuration, there is a slower rate of training-error minimisation, with initial models being poorer performers compared to the respective ones under the GP-AR and GP-ARMA model configurations. Eventually, however, we observe that all model configurations reach the same training error rates. This observation makes the GP-AR and GP-ARMA model configurations much more efficient in terms of search effort required to find the best-of-run generalising models, however, rendering any additional training prone to overfitting.

Looking at the results of Table 3 obtained with multi-model predictors, we observe that ensembles of size 5 generalised the best in all datasets, improving the results upon single-model predictors. Interestingly, the best-generalising ensemble GP-AR and GP-ARMA models outperformed their GSR counterparts in all datasets. Statistically significant differences (unpaired t-test, $p < 0.0001$, $df = 98$) were found between the mean test RMSE of ensembles of size 5 of autoregressive models and standard symbolic regression models. This is mainly attributed to the unstable performance of GSR models indicated by the high variance in test RMSE in different evolutionary runs (Figures 2(a), 3(a), 4(a)), and the fact the bagging generates models from resampling the training data and learning models using each sub-sample separately. An additional interesting observation is that the use of greater ensemble size have an effect in reducing the RMSE variance in the case of GSR; however, increasing the ensemble size shows no pronounced effect in the variance of autoregressive models. Overall, it is noted that increasing the ensemble size beyond 5 models results in worsening the generalisation performance. This observation is consistent across all datasets.

Finally, Figures 6, 7, 8 show the target and predicted values from the best-performing 5-model autoregressive models of Table 3, for ATL, DEN, and DFW datasets respectively. It can be seen that the evolved ensemble models achieved a good fit for most of the in-sample and out-of-sample data range. Table 4 presents a gallery of good-generalising GP-AR(36) evolved models.

Table 4. Sample evolved GP-AR(36) models

$f(t) = \sqrt{HDD_{t-12} * \left(HDD_{t-36} + \sqrt{HDD_{t-12} * \left(\frac{HDD_{t-26}}{-0.92 + (HDD_{t-7} * log(HDD_{t-21}))} \right)} \right)}$
$f(t) = \sqrt{(HDD_{t-24} * HDD_{t-36}) - HDD_{t-24} + 11.51}$
$f(t) = HDD_{t-36} * 0.94$
$f(t) = HDD_{t-36} - \sqrt{HDD_{t-12}} + 0.41 * (HDD_{t-36} - HDD_{t-12})$
$f(t) = \frac{HDD_{t-36}}{\sqrt{\frac{HDD_{t-36} + 5.11}{HDD_{t-12}}}}$
$f(t) = \sqrt{(HDD_{t-36} + 0.17) * (HDD_{t-12} - 0.84)}$

8 Conclusion

This study adopted a time-series modelling approach to the production of a seasonal weather-metric forecast, as a constituent part of a general method for pricing weather derivatives. Two GP-based methods for time series modelling were used; the first one is based on standard symbolic regression; the second one is based on autoregressive time-series modelling that is realised via an iterated single-step prediction process and a specially crafted terminal set of historical time-series values.

Results are very encouraging, suggesting that GP is able to successfully evolve accurate seasonal temperature forecasting models. The use of ensemble learning of 5-model predictors enhanced the generalisation ability of the system, as opposed to single-model predictions. Standard symbolic regression was seen to be able to capture the deterministic aspect of the modelled data and attained the best test performance, however its overall performance was marked as unstable, producing some very poor-generalising models. On the other hand, the performance of search-based autoregressive and moving average models was deemed on average the most stable and best-performing in out-of-sample data.

References

1. A. Garcia and F. Sturzenegger, "Hedging corporate revenues with weather derivatives: A case study", Master's thesis, Universite de Lausanne, 2001.
2. Van Sprundel, *Using weather derivatives for the financial risk management of plant diseases: A study on Phytophthora infestans and Fusarium head blight*, PhD thesis, Wageningen University, 2011.
3. Cao M. and Wei J., "Equilibrium valuation of weather derivatives", Working paper, School of Business, York University, Toronto, 2002.
4. Vining R, "Weather derivatives: Implications for australia", in *Proceedings of Hawaii Conference on Business*, 2001.
5. Weather Risk Management Association, "Introduction to the weather market", April 2011.
6. Weather Risk Management Association, "Results of 2006 annual industry-wide survey", April 2006.
7. Considine G., "Introduction to weather derivatives", Tech. Rep., Weather Derivatives Group, 1999.
8. Jewson S., Brix A., and Ziehmann C., *Weather Derivative Valuation: The Meteorological, Statistical, Financial and Mathematical Foundations*, Cambridge University Press, 2005.
9. Alaton P., Djehiche B., and Stillberger D., "On modelling and pricing weather derivatives", *Applied Mathematical Finance*, vol. 9, no. 1, pp. 1–20, 2002.
10. Black F. and Scholes M., "The pricing of options and corporate liabilities", *Journal of Political Economy*, vol. 81, pp. 637–654, 1973.
11. Campbell S. and Diebold F., "Weather forecasting for weather derivatives", *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 6–16, 2005.
12. Davis M., "Pricing weather derivatives by marginal value", *Quantitative Finance*, vol. 1, pp. 305–308, 2001.

13. Taylor J. and Buizza R., "Density forecasting for weather derivative pricing", *International Journal of Forecasting*, vol. 22, pp. 29–42, 2006.
14. Weigel A., Baggenstos D., and Liniger M., "Probabilistic verification of monthly temperature forecasts", *Monthly Weather Review*, vol. 136, pp. 5162–5182, 2008.
15. Jewson S. and Caballero R., "The use of weather forecasts in the pricing of weather derivatives", *Metereological Applications*, vol. 10, pp. 367–376, 2003.
16. Moreno M., "Riding the temperature", 2000.
17. R. Poli, W. B. Langdon, and N. F. McPhee, *A field guide to genetic programming*, Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`, 2008, (With contributions by J. R. Koza).
18. M. O'Neill, L. Vanneschi, S. Gustafson, and W. Banzhaf, "Open issues in genetic programming", *Genetic Programming and Evolvable Machines*, vol. 11, no. 3/4, pp. 339–363, September 2010, Tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines.
19. R. Poli, L. Vanneschi, W. B. Langdon, and N. F. McPhee, "Theoretical results in genetic programming: The next ten years?", *Genetic Programming and Evolvable Machines*, vol. 11, no. 3/4, pp. 285–320, September 2010, Tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines.
20. J. R. Koza, "Human-competitive results produced by genetic programming", *Genetic Programming and Evolvable Machines*, vol. 11, no. 3/4, pp. 251–284, September 2010, Tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines.
21. J.R. Koza, *Genetic Programming: on the programming of computers by means of natural selection*, MIT Press, Cambridge, MA, (1992).
22. Alexandros Agapitos, Matthew Dyson, Jenya Kovalchuk, and Simon Mark Lucas, "On the genetic programming of time-series predictors for supply chain management", in *GECCO '08: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, 2008.
23. Neal Wagner, Zbigniew Michalewicz, Moutaz Khouja, and Rob Roy McGregor, "Time series forecasting for dynamic environments: The DyFor genetic program model", *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 433–452, August 2007.
24. Emiliano Carreno Jara, "Long memory time series forecasting by using genetic programming", *Genetic Programming and Evolvable Machines*, vol. 12, no. 4, pp. 429–456, December 2012.
25. Coulibaly P., "Downscaling daily extreme temperatures with genetic programming", *Geophysical Research Letters*, 2004.
26. Peter A. Whigham and Peter F. Crapper, "Time series modelling using genetic programming: An application to rainfall-runoff models", in *Advances in Genetic Programming 3*, Lee Spector, William B. Langdon, Una-May O'Reilly, and Peter J. Angeline, Eds., chapter 5, pp. 89–104. MIT Press, Cambridge, MA, USA, June 1999.
27. Yoon-Seok Hong and Michael R. Rosen, "Identification of an urban fractured-rock aquifer dynamics using an evolutionary self-organizing modelling", *Journal of Hydrology*, vol. 259, no. 1-4, pp. 89–104, 2002.
28. Katya Rodriguez-Vazquez, "Genetic programming in time series modelling: An application to meteorological data", in *Proceedings of the 2001 Congress on Evolutionary Computation CEC2001*, COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea, 27-30 May 2001, pp. 261–266, IEEE Press.

29. Marcos Alvarez-Diaz, Gonzalo Caballero Miguez, and Mario Solino, "The institutional determinants of CO2 emissions: A computational modelling approach using artificial neural networks and genetic programming", FUNCAS Working Paper 401, Fundacion de las Cajas de Ahorros, Madrid, July 2008.
30. Bakhshaii A. and Stull R., "Deterministic ensemble forecasts using gene-expression programming", *Weather and Forecasting*, vol. 24, no. 5, pp. 1431–1451, 2009.
31. De Falco I., Della Cioppa A., and Tarantino E., "A genetic programming system for time series prediction and its application to el nino forecast", in *Advances in Intelligent and Soft Computing - Soft Computing: Methodologies and Applications*, vol. 32 of *151-162*. Springer, 2005.
32. Ozgur Kisi and Aytac Guven, "Evapotranspiration modeling using linear genetic programming technique", *Journal of Irrigation and Drainage Engineering*, vol. 136, no. 10, pp. 715–723, October 2010.
33. Julio J. Valdes and Antonio Pou, "Central England temperatures and solar activity: A computational intelligence approach", in *International Joint Conference on Neural Networks (IJCNN 2010)*, Barcelona, Spain, 18-23 July 2010, IEEE Press.
34. A. Makkeasoyrn, Ni-Bin Chang, and Xiaobing Zhou, "Short-term streamflow forecasting with global climate change implications - A comparative study between genetic programming and neural network models", *Journal of Hydrology*, vol. 352, no. 3-4, pp. 336–354, 2008.
35. S. Shahid, M. Hasan, and R. U. Mondal, "Modeling monthly mean maximum temperature using genetic programming", *International Journal of Soft Computing*, vol. 2, no. 5, pp. 612–616, 2007.
36. Maritza Arganis, Rafael Val, Jordi Prats, Katya Rodriguez, Ramon Dominguez, and Josep Dolz, "Genetic programming and standardization in water temperature modelling", *Advances in Civil Engineering*, vol. 2009, 2009.
37. Juan J. Flores, Mario Graff, and Erasmo Cadenas, "Wind prediction using genetic algorithms and gene expression programming", in *Proceedings of the International Conference on Modelling and Simulation in the Enterprises. AMSE 2005*, Morelia, Mexico, April 2005.
38. Makridakis S., Wheelright S., and Hyndman R., *Forcasting: Methods and Applications*, New York: Willey, 1998.
39. Bollerslev T., "Generalised autoregressive conditional heteroskedasticity", *Journal of Econometrics*, vol. 31, pp. 307–327, 1986.
40. Richard Duda, Peter Hart, and David Stork, *Pattern Classification*, John Wiley and Sons, 2nd edition, 2001.
41. Christopher M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1996.
42. Hitoshi Iba, "Bagging, boosting, and bloating in genetic programming", in *Proceedings of the Genetic and Evolutionary Computation Conference*, Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, Eds., Orlando, Florida, USA, 13-17 July 1999, vol. 2, pp. 1053–1060, Morgan Kaufmann.
43. Leo Breiman and Leo Breiman, "Bagging predictors", *Machine Learning*, pp. 123–140, 1996.
44. B Efron and R Tibshirani, *An introduction to the bootstrap*, Chapman and Hall, 1993.

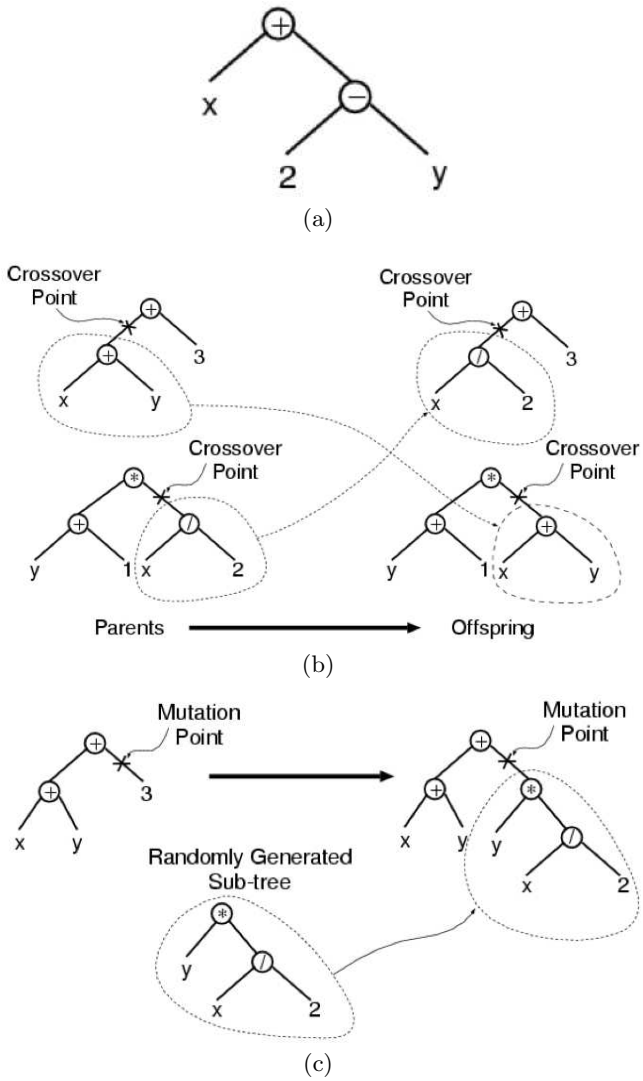
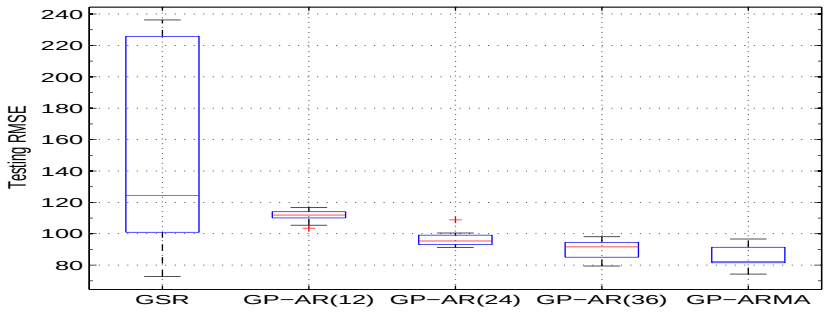
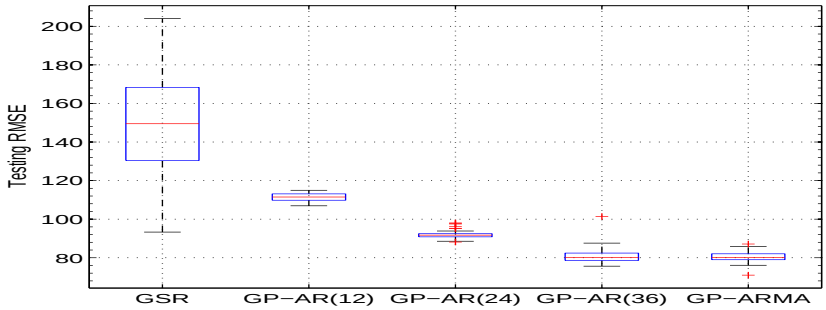


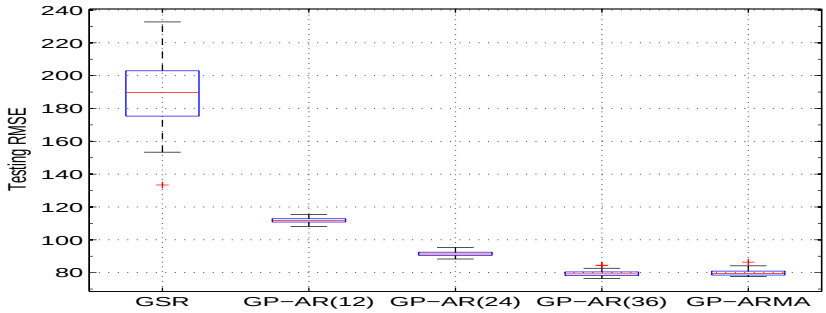
Fig. 1. Genetic programming representation and variation operators.



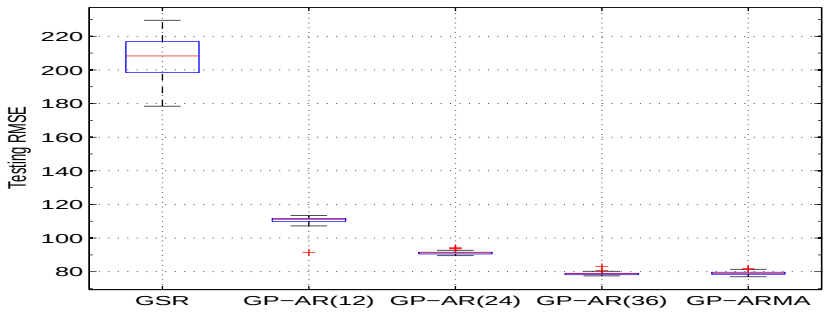
(a)



(b)

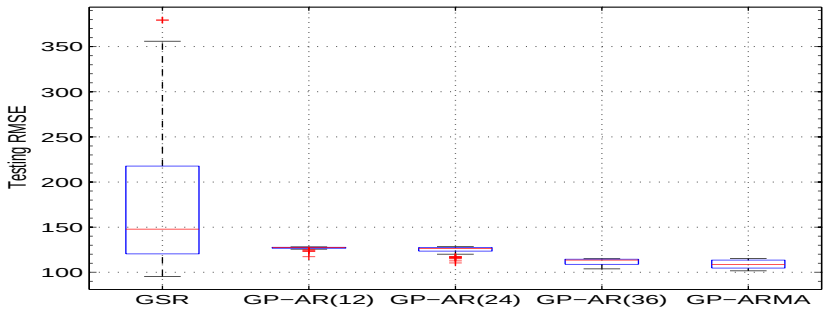


(c)

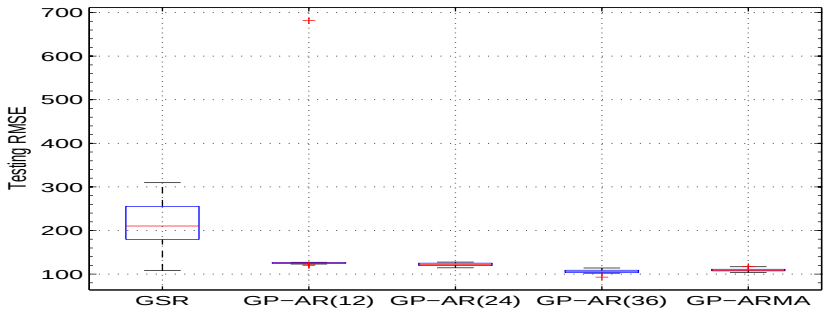


(d)

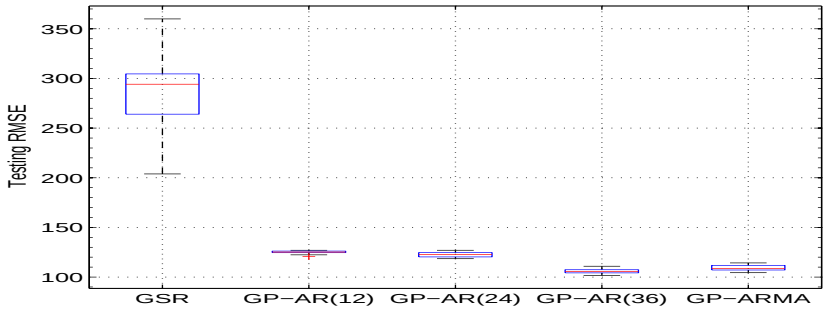
Fig. 2. Distribution of best-of-run test RMSE accrued from 50 independent runs for the ATL dataset. (a) Single model; (b) Ensemble size 5; (c) Ensemble size 10; (d) Ensemble size 20.



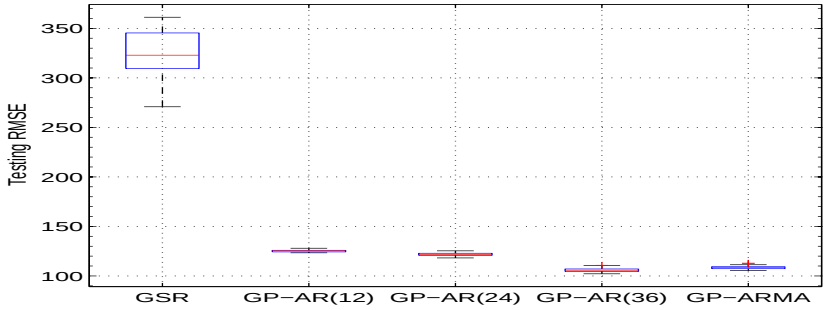
(a)



(b)

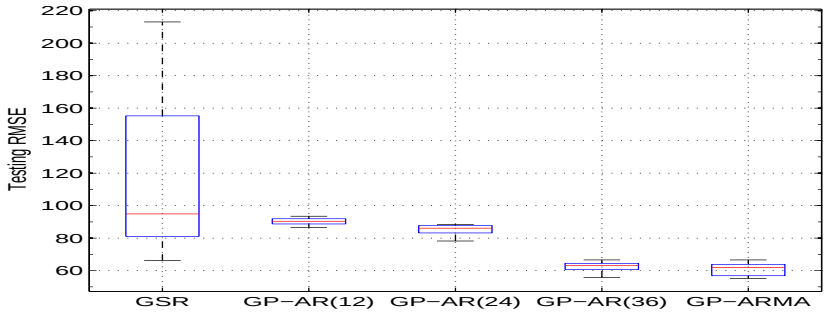


(c)

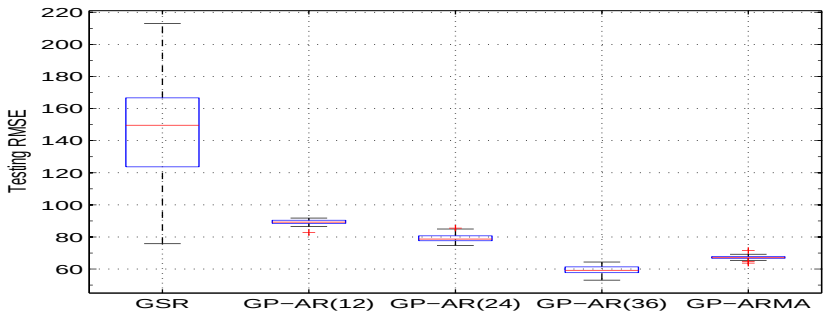


(d)

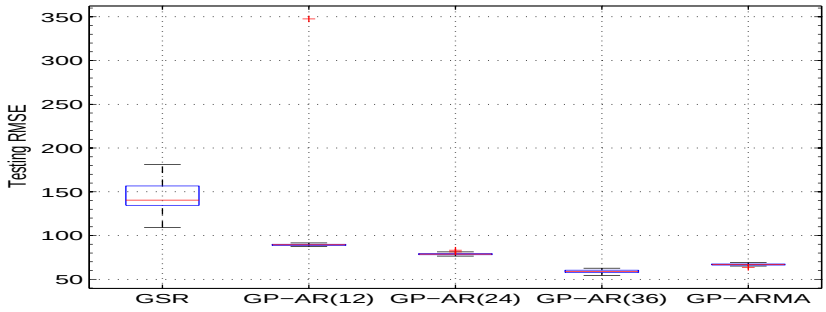
Fig. 3. Distribution of best-of-run test RMSE accrued from 50 independent runs for the DEN dataset. (a) Single model; (b) Ensemble size 5; (c) Ensemble size 10; (d) Ensemble size 20.



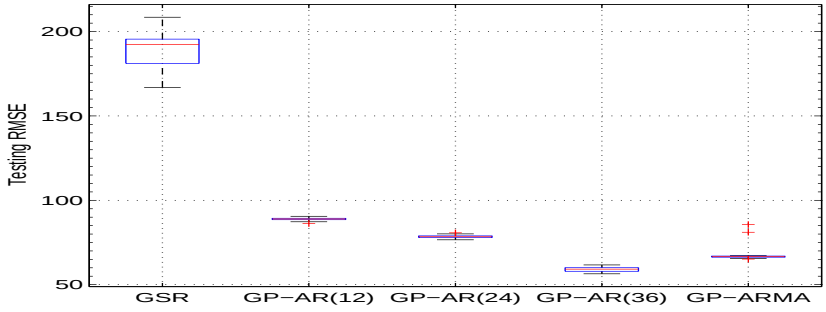
(a)



(b)

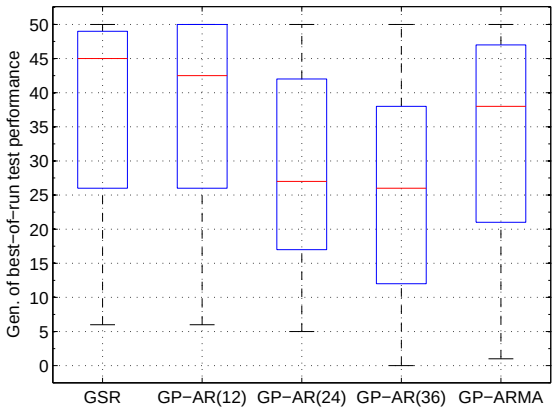


(c)

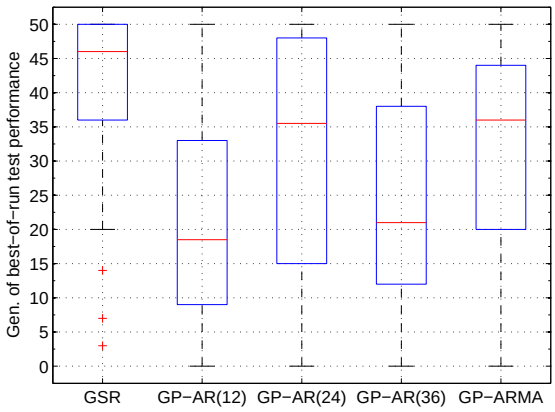


(d)

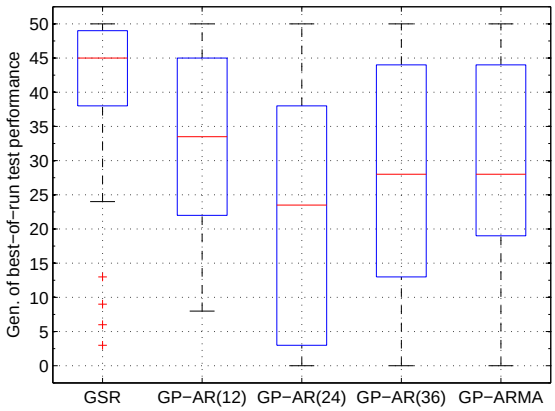
Fig. 4. Distribution of best-of-run test RMSE accrued from 50 independent runs for the DFW dataset. (a) Single model; (b) Ensemble size 5; (c) Ensemble size 10; (d) Ensemble size 20.



(a)

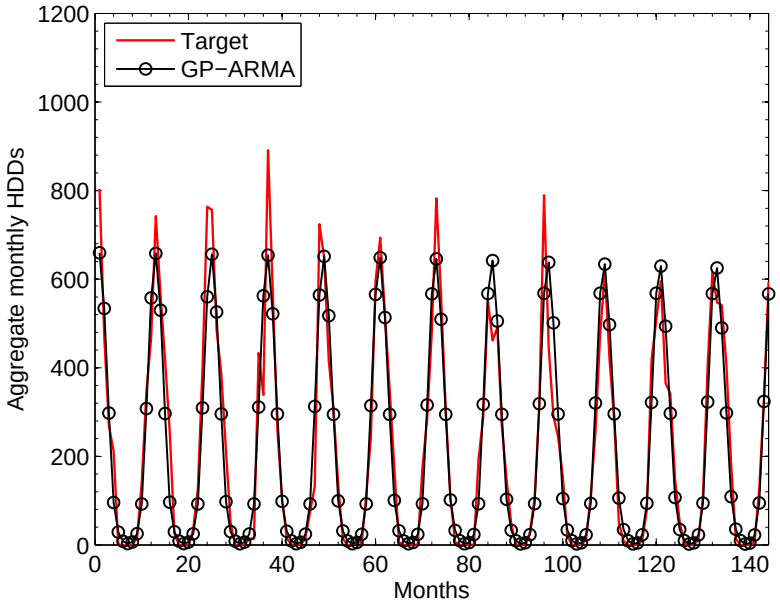


(b)

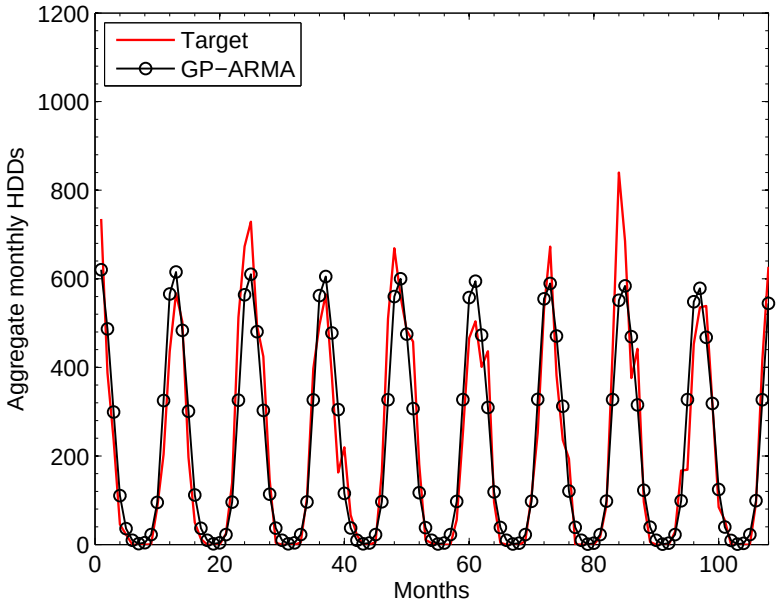


(c)

Fig. 5. Figures (a), (b), (c) show the distribution of generation number where each best-of-run individual on test data was discovered for the cases of ATL, DEN, and DFW respectively. Cases for single model predictions.

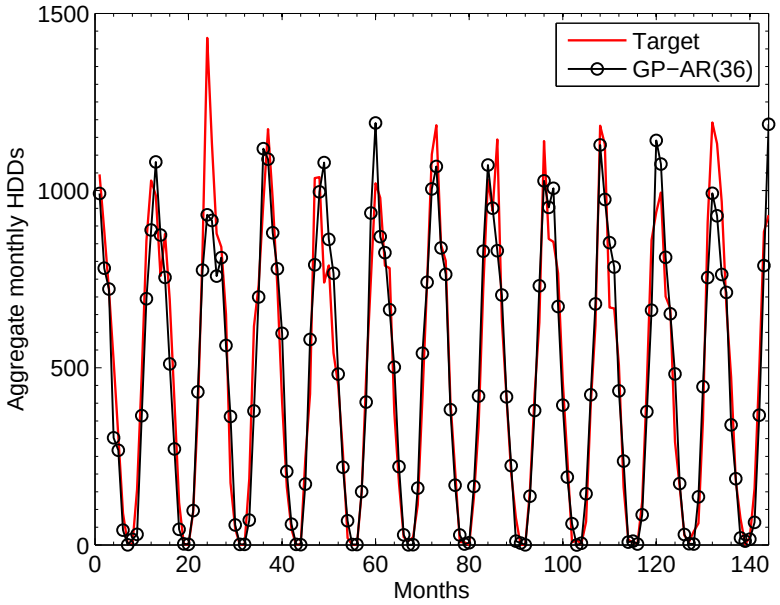


(a)

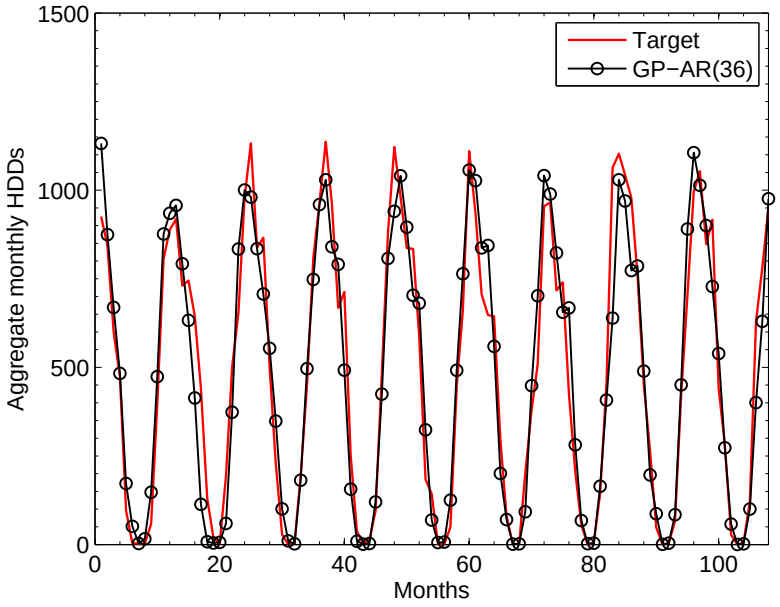


(b)

Fig. 6. Target vs. Prediction for best-performing models of GP-ARMA (ensemble size 5) for the ATL dataset. (a) training data; (b) test data.

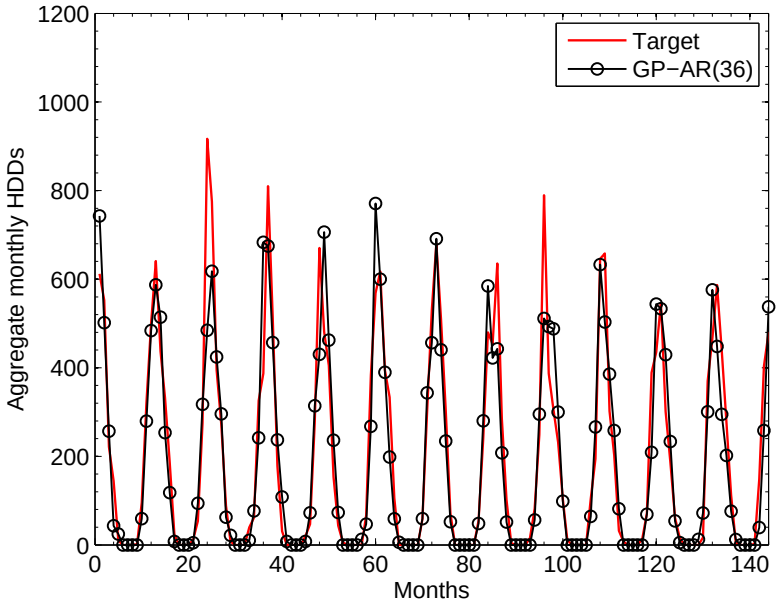


(a)

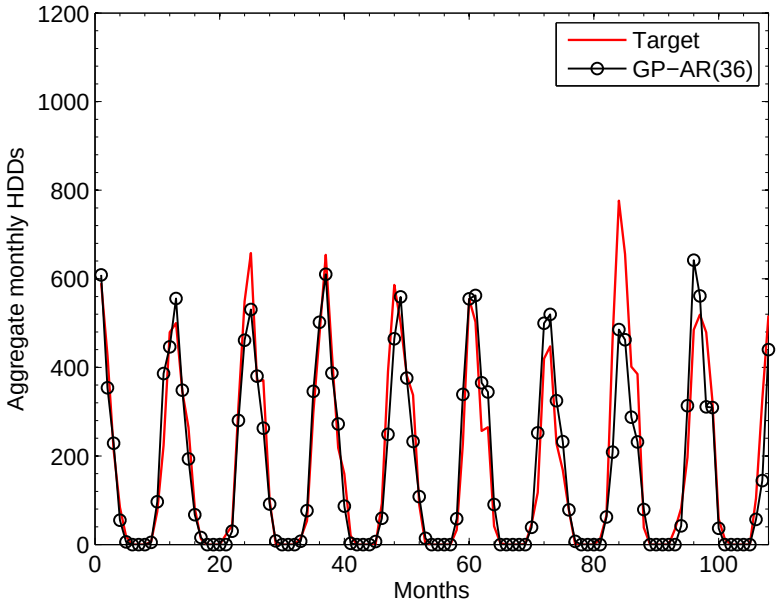


(b)

Fig. 7. Target vs. Prediction for best-performing models of GP-AR(36) (ensemble size 5) for the DEN dataset. (a) training data; (b) test data.



(a)



(b)

Fig. 8. Target vs. Prediction for best-performing models of GP-AR(36) (ensemble size 5) for the DFW dataset. (a) training data; (b) test data.

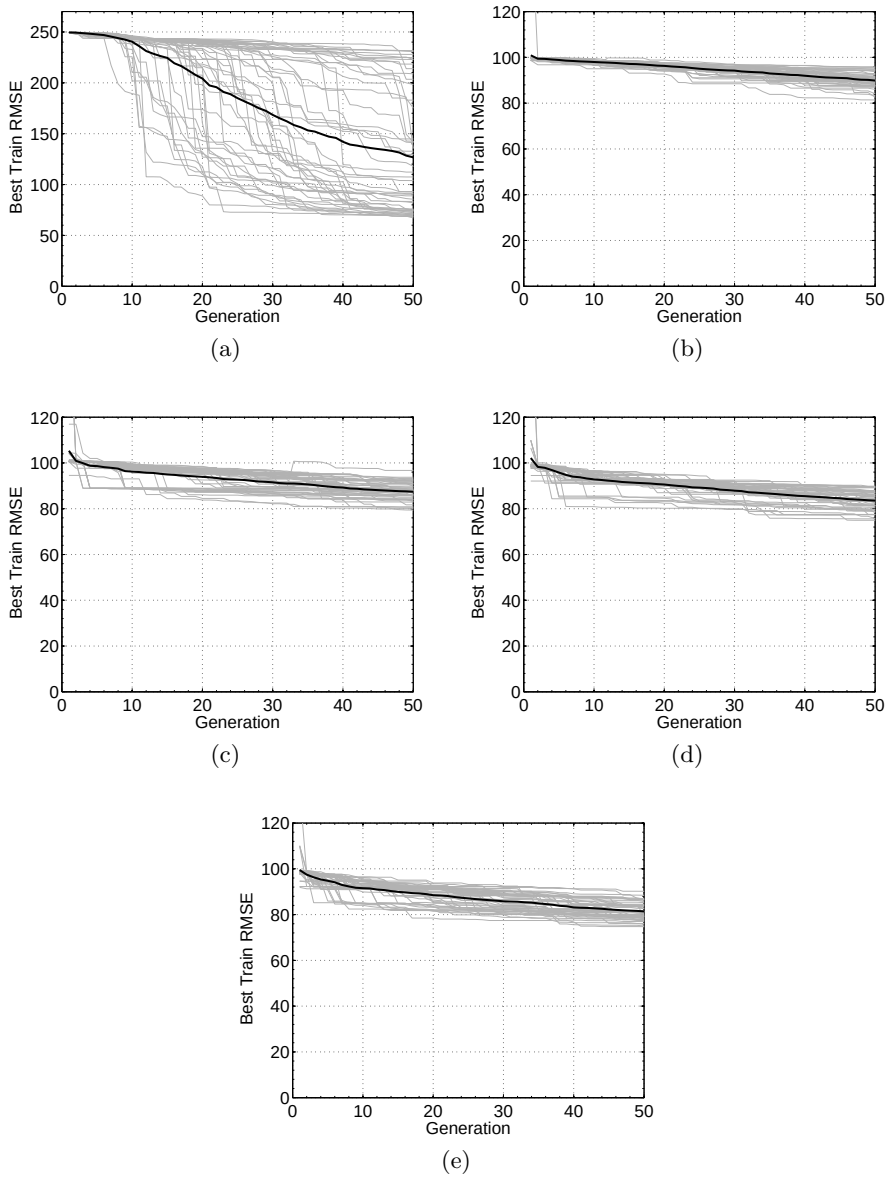


Fig. 9. RMSE histograms for the ATL dataset. Each figure illustrates 50 independent evolutionary runs. Average is indicated with bold. (a) GSR; (b) GP-AR(12); (c) GP-AR(24); (d) GP-AR(36); (e) GP-ARMA.