Kent Academic Repository

Cramer, Sam, Kampouridis, Michael, Freitas, Alex A. and Alexandridis, Antonis (2019) *Stochastic Model Genetic Programming: Deriving Pricing Equations for Rainfall Weather Derivatives.* Swarm and Evolutionary Computation, 46 . pp. 184-200. ISSN 2210-6502.

Downloaded from

https://kar.kent.ac.uk/72082/ The University of Kent's Academic Repository KAR

The version of record is available from

https://doi.org/10.1016/j.swevo.2019.01.008

This document version

Author's Accepted Manuscript

DOI for this version

https://doi.org/10.22024/UniKent/01.02.72082.3147127

Licence for this version

CC BY-NC-ND (Attribution-NonCommercial-NoDerivatives)

Additional information

Versions of research works

Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal* , Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

Enquiries

If you have questions about this document contact ResearchSupport@kent.ac.uk. Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies).

Stochastic Model Genetic Programming: Deriving Pricing Equations for Rainfall Weather Derivatives

Sam Cramer, Michael Kampouridis, Alex A. Freitas[a], Antonis Alexandridis[b,1]

[a] *School of Computing, University of Kent, UK*
[b] *Kent Business School, University of Kent, UK*

Abstract

Rainfall derivatives are in their infancy since starting trading on the Chicago Mercantile Exchange (CME) in 2011. Being a relatively new class of financial instruments there is no generally recognised pricing framework used within the literature. In this paper, we propose a novel Genetic Programming (GP) algorithm for pricing contracts. Our novel algorithm, which is called Stochastic Model GP (SMGP), is able to generate and evolve stochastic equations of rainfall, which allows us to probabilistically transform rainfall predictions from the risky world to the risk-neutral world. In order to achieve this, SMGP's representation allows its individuals to comprise of two weighted parts, namely a seasonal component and an autoregressive component. To create the stochastic nature of an equation for each SMGP individual, we estimate the weights by using a probabilistic approach. We evaluate the models produced by SMGP in terms of rainfall predictive accuracy and in terms of pricing performance on 42 cities from Europe and the USA. We compare SMGP to 8 methods: its predecessor DGP, 5 well-known machine learning methods (M5 Rules, M5 Model trees, k-Nearest Neighbors, Support Vector Regression, Radial Basis Function), and two statistical methods, namely AutoRegressive Integrated Moving Average (ARIMA) and Monte Carlo Rainfall Prediction (MCRP). Results show that the proposed algorithm is able to statistically outperform all other algorithms.

Keywords: Weather derivatives, rainfall, pricing, stochastic model genetic programming

1. Introduction

In this paper we are concerned with the pricing of rainfall derivatives. However, in order to achieve this, a large focus will be on the prediction of rainfall that directly underpins a derivatives contracts value. The weather provides many obstacles and has always been a considerable risk factor for various individuals with businesses' profit being greatly affected by the state of the weather. Until 1996 there was no suitable financial protection available to businesses; previously, one would have to rely on insurance, but it is considerably hard to prove that a business has been adversely affected by the weather. The insurance market can cover extreme weather such as hurricanes, but not unfavourable fluctuations, because these are harder to prove the effect on business. Nowadays, an individual can get financial protection against the weather elements, by the use of weather derivatives.

Weather derivatives are a type of contract held between two or more parties, whose value depends upon the underlying weather variable. Various different types exist and are commonly traded on temperature, rainfall and wind. What makes weather derivatives different to other derivatives is that the market is incomplete. This is due to not being able to physically hold, store or trade the weather variables. Without a tangible asset, it is not possible to recreate a riskless hedge, hence opportunities of arbitrage may exist. As a result, this violates the assumptions underpinning commonly used pricing models. This is a major problem and currently there exists no generally accepted pricing framework to value these contracts. The contracts of interest are based on rainfall, which are a relatively new financial instrument that started trading in 2011 on the Chicago Mercantile Exchange (CME).

[1] Corresponding author: Michael Kampouridis, School of Computing, Medway, ME4 4AG, UK. Tel: +44 1634 88 8837. Email: M.Kampouridis@kent.ac.uk

A rainfall derivative contract will protect against too much or too little rainfall, as the contracts payout is dependent on the level of rainfall that falls at a specified location. Therefore, a farmer could engage in a contract and would pay a fair price to protect themselves from too much rainfall. If that event happens then the farmer would be rewarded with a payout the equivalent of the amount of rainfall that fell multiplied by a amount per index point. Contracts that have previously traded tend to be \$50 per 0.1 inch of rainfall (which is 1 index point), but this can vary per contract. In this paper we are concerned with looking to predict the fair price of that contract. In other words, what is the expected level of rainfall to fall. This level of rainfall is used as the index itself in order to calculate the payout from the realisation of the event. As the level of rainfall is used as the index itself, estimating rainfall as accurately as possible becomes a necessity for improved pricing. The payout can be considered in the perspective of a farmer, the lost income from too much rainfall from damage to their crops. They would receive a payout per index point above his contract. There is of course a risk he would lose money if rainfall was below the specified rainfall level.

Rainfall derivatives are less commonly traded compared to other types of weather derivatives, but are just as important, especially for those in agriculture. Rainfall derivatives are a much more recent addition, because rainfall is considered the hardest weather variable to model and price [1]. If the modelling is insufficient, this can lead to large pricing errors, since future rainfall forecasts will not reflect future events. Hence, this leads to a derivatives price being far away from the true value, increasing the volatility and uncertainty within the market. In turn this reduces the prospect of attracting new investors to the market. Unlike other domains, the time series of rainfall is highly discontinuous with little to no connection between each pair of consecutive days. The binary event of wet or dry day that underpins the rainfall amount is largely random. Moreover, the daily data does not follow a trend, with little to no seasonality being present. Therefore, by addressing these issues through machine learning and statistical methods, the pricing accuracy should increase.

To price derivatives there are two techniques that are formulated for rainfall derivatives: indifference pricing [2] and the arbitrage-free pricing approach [3]. Since contracts began trading on the CME, the latter became the standard pricing technique. The technique works by probabilistically transforming the predictions from the risky world to the risk-neutral. Thus, a large number of future rainfall pathways are required to calculate the probability of a rainfall amount occurring. Synonymous to general derivative pricing, Monte Carlo simulations is required to generate the possible rainfall values.

The majority of published works has so far focused on creating rainfall derivatives models. Nevertheless, as the concept of rainfall derivatives is relatively new, there exists little literature on this subject. Moreover, the difficulty in predicting rainfall has deterred the attention of researchers, unlike other weather derivatives such as temperature[2]. To estimate future levels of rainfall, the Markov-chain extended with rainfall prediction (MCRP) [10] method has been commonly applied in a wide range of the literature, including rainfall derivatives [3, 11, 12, 13]. The general MCRP approach is often referred to as a 'chain-dependent process' [14], which splits the model into capturing first the rain occurrence pattern, and then predicting the rainfall intensities. The occurrence pattern is produced by a Markov-chain, where state 0 is a dry day and state 1 is a wet day. If a wet day is produced then the rainfall intensity is calculated by generating a random number from a given distribution (typically the Gamma or Mixed-Exponential distribution), otherwise a value of 0 is assigned (zero rainfall). We refer the reader to [10] for a complete description of MCRP. Despite being a popular approach, MCRP is very simplistic and does not truly capture the irregularities of rainfall. The final result tends to fluctuate around the observable mean of the training data. Moreover, there exists a large number of rainfall pathways that do not reflect future behaviour.

Machine learning methods can be seen as an alternative and have become more popular over recent years. Typical applications within machine learning revolve around short term predictions (e.g. rainfall-runoff models up to a few hours [15], monthly amounts [16] [17]) or mid range forecasts of up to a month [18, 19] . For daily predictions, [20] used a feed-forward back-propagation neural network for daily rainfall prediction in Sri Lanka, which was inspired by the chain-dependent approach from statistics. The work in [21] also applied GP to daily rainfall data, but the GP performed poorly by itself, although when assisted by wavelets the predictive accuracy improved. In the context of rainfall derivatives a selection of machine learning algorithms was explored in detail in [22], which showed that Radial Basis Function (RBF), Support Vector Regression (SVR) and GP outperformed the commonly applied method of MCRP following a transformation of the data. In addition, [23] presented a tailored GP for the problem of rainfall

[2] In fact, temperature weather derivatives have attracted a lot of research, both from the statistical and mathematical community [4, 5], as well as the machine learning community [6, 7, 8, 1, 9].

prediction, and [24] extended the above work by exploring the use of feature extraction. Both works showed promising results, where the GP could outperform MCRP, the current-state-of-the art. Furthermore, [25, 26] extended the above GP works, by proposing a new algorithm called Decomposition GP (DGP). This was a novel hybrid algorithm (comprising of a Genetic Algorithm (GA) part and a Genetic Programming part) that decomposes the problem of rainfall into subproblems. The motivation for doing this was to allow the GP to focus on each subproblem, before combining them back into the full problem. The GP did this by having a separate regression equation for each subproblem, determined based on the level of rainfall; in addition, the GA determined which regression equation should be used (solving a classification problem). By turning our attention to subproblems, this allowed the DGP to reduce the difficulty when dealing with data sets with high volatility and extreme rainfall values, since these values can be focused on independently.

Nevertheless, while DGP was a very effective algorithm in terms of rainfall prediction, it is not appropriate for deriving rainfall pricing equations. In order to price a derivative a series of future predictions are required to determine the probability of an event occurring. A probability is required, because there is a level of uncertainty regarding the future. As DGP (like a standard machine learning technique) was utilised, a deterministic model would be produced, and would only produce a single prediction regarding the future. One limitation for pricing is the model generated cannot assign a probability of a future event from a single observation. A model that is capable of providing many future predictions can be used to estimate the probability of an event happening. Consequently, this allows pricing models including arbitrage-free approach and indifference pricing methods to be utilised in order to find the fair value of a derivative contract. Therefore, within this paper we aim to overcome the downside of DGP and propose a framework for generating many predictions using DGP as the backbone to drive the predictive accuracy.

With respect to the pricing literature, [27] proposed the arbitrage-free approach for the problem of rainfall derivatives, but did not apply it to generate any prices. This is closely linked with [3], which used the framework of [27] to apply a range of distributions to the output from MCRP. Based on maximising the result of the Kolmogorov-Smirnov test, they found that the Normal-inverse Gaussian (NIG) distribution is the most suitable distribution. They applied the arbitrage-free pricing method to price rainfall futures at the CME for three cities in the USA, namely Detroit, Jacksonville and New York. This was the first work of pricing real futures prices. [28] followed up the work using the Poisson-cluster model [29, 30, 31] to apply to the rainfall futures prices at Detroit. The findings suggested that both models were suitable for pricing at Detroit, but the results indicated that the Poisson-cluster fitted the data better. In terms of pricing performance, both models were very similar. Finally, [32] proposed a risk-neutral density of rainfall predictions generated by DGP and supported by Markov Chain Monte Carlo (MCMC). Moreover, instead of having a single rainfall model for all contracts, [32] also proposed having a separate rainfall model for each contract. Their results showed to produce prices closer to the CME, when compared to prices derived by MCRP and Burn Analysis. However, these two methods had the disadvantage of being computationally inefficient, as a large computational overhead was required to extrapolate a density. Also, having to produce a separate model for each contract was cumbersome.

To overcome the above issues, we propose a novel GP algorithm, which is able to generate and evolve a stochastic equation of rainfall. We call this algorithm Stochastic Model Genetic Programming (SMGP). SMGP individuals comprise of two parts, a seasonal component and an autoregressive component (DGP). In addition, we introduce the use of weights for these two components. To create the stochastic nature of an equation for each SMGP individual, we estimate the weights by using a probabilistic approach. This allows us to perform a random walk on our rainfall values, and to estimate a density that reflects each day in the testing set. Hence, by calculating the probability that a rainfall event occurs, we can translate this into the risk-neutral measure. More information about the SMGP algorithm will be given in Section 3.

Therefore the goal of this paper is twofold: (i) to predict rainfall as accurately as possible, and (ii) to derive rainfall pricing equations. In order to test the effectiveness of the proposed algorithm, we run two sets of experiments. First, we are interested in investigating how effectively SMGP can predict rainfall amounts. This is very important, because the ability to price rainfall derivatives relies heavily on predicting the level of rainfall as accurately as possible, to minimise problems of mispricing [1, 33]. We report the Root Mean Square Error (RMSE) of the rainfall predictions, and compare it to DGP, and five popular machine learning algorithms, including M5 Rules, M5 Model trees, k-Nearest Neighbors, Support Vector Regression, and Radial Basis Function. In addition, we will compare the performance of the SMGP to MCRP, which as we explained earlier is the current state-of-the-art algorithm for rainfall prediction in the context of weather derivatives. In the second set of experiments, we focus on deriving pricing equations. We compare the prices derived by the SMGP, not only to the previously mentioned machine learning algorithms, but also to Burn Analysis (BA), which is a common benchmark for derivatives pricing.

The rest of this paper is organised as follows. We begin with Section 2, which presents background information on the problem of pricing in the context of weather derivatives. In Section 3, we present in detail our new algorithm based on producing a stochastic white-box potentially interpretable model, which will be used for deriving pricing equations. In Section 4, we outline the experimental set-up and tuning for our experiments. In Section 5, we show the experimental results for the proposed SMGP and the other benchmarks on the rainfall prediction problem and also how they relate to the pricing of rainfall derivatives. Finally, in Section 6 we conclude and discuss future work. In addition, a Glossary of financial and mathematical terms is included after the References section, to help readers unfamiliar with financial terminology.

2. Pricing Within Rainfall Derivatives

2.1. Overview

The general method for pricing a derivative contract for the rainfall amount is given by:

$$F(t;\tau_1,\tau_2) = E^Q[I(\tau_1,\tau_2)|f_t] = E^Q\left[\sum_{\tau=\tau_1}^{\tau_2} R_T|f_t\right] \tag{1}$$

where $F(t;,\tau_1,\tau_2)$ represents a futures contract priced at time point t for a contract period from time point τ_1 until time point τ_2. For this, t does not have to equal to τ_1, because contracts are priced for a future date. $E^Q[I(\tau_1,\tau_2)|f_t]$ represents the index I of the rainfall amount over the contract period τ_1 till τ_2, given the available data at time point f_t. This index level is calculated at the risk-neutral expectation denoted by E^Q. This gives us the final part of the equation that is the sum of the total rainfall (R_T) over the contract period given the available historical data that we have under risk-neutral conditions. As the rainfall index is explicitly used in the formulation of a derivatives price, the prediction of the underlying variable of rainfall is required. Note, Q (risk-neutral measure) does not have anything to do with the objective probability of occurrence of scenarios, i.e. the probability of a certain rainfall prediction pathway happening. Q in our case is a probability measure on the set of scenarios, which is a *bet* on the occurrence of this event. In other words, we are trying to measure the probability of us betting on the occurrence of this outcome, rather than the probability of the outcome.

Rainfall derivatives is an incomplete market, as rainfall amounts do not have a price, nor can they be held or traded. Therefore, one cannot assume *arbitrage-free* pricing (there exists the opportunity for risk-free profit), as a result pricing directly on the accumulated amount of rainfall is considered risky. Because of this, additional methods are required to transform rainfall amounts from the real world to the risk-neutral world. Therefore, the rainfall amount is directed towards the more likely scenario in order to achieve neutrality. Another perspective is finding the expectation of the index that has been calculated and then what is the probability for the index to take that value. Arbitrage pricing and risk-neutrality are key concepts, which need to be addressed within derivative pricing. The absence of arbitrage imposes constraints on the way derivatives are priced within a market. Risk-neutrality allows the price of any derivative within an arbitrage-free market to discount the expected payoff under an approximated probability measure called a risk-neutral measure.

Our rainfall estimates $I(\tau_1,\tau_2)$ are considered the expected price under the canonical measure P (i.e., the probability space (Ω, f, P)), but are within the 'risky' world. Ω is the sample space, a set of all possible outcomes, f is a set of events, where each event is a set containing zero or more outcomes and P the assignment of probabilities to the functions. Therefore, we require $Q \sim P$, such that all tradable assets in the market are martingales after discounting taking into account investors' exposure to risk. A martingales is a sequence of values of a random variable, such as a stochastic process, where at a particular time in the realised sequence, the expectation of the future value is equal to the present observed value. The expectation is also conditioning on the given knowledge of all prior observed values. To establish the risk preferences of investors we require the Market Price of Risk (MPR), which is the additional return or risk premium expected by investors for being exposed to undertaking the futures contract. Within complete markets, where the modelled quantity is tradable, the MPR does not explicitly feature in the formulation of the price. This is because investors are able to hedge away the risk in any position by dynamically buying and selling the underlying asset, allowing the equivalent martingale measure of Q to be calculated. It is crucial to derive the equivalent martingale measure, which verifies that there is arbitrage-free pricing. Therefore, we must specify the risk-neutral probability of Q. The weather derivatives are traded in an incomplete market, so there will exist many different martingales (Q). Hence, it is not possible to find a unique risk-neutral measure Q [34, 27], such that Q is equivalent to the physical measure P. As mentioned previously, Q is the betting on the outcome of P. Therefore, the derivative price is arbitrage-free, if and only if there exists a probability measure $Q \sim P$, such that the derivative payoffs are martingales with respect to Q. For this reason Q is an equivalent martingale measure. The Black-Scholes model, the first and most well known pricing model achieves its equivalent martingale measure by modifying the drift in the Brownian motion.

Since it is not possible to construct a portfolio that can be perfectly hedged (has a replication strategy), it is not possible to find a unique risk-measure, or unique equivalent martingale $Q \sim P$. Instead, many different martingales exist and prices can be derived directly only based on the basis of no-arbitrage. Due to this reason, we are looking to estimate Q_θ, where theta is the MPR, a parameter for finding the unique equivalent martingale.

There are two main approaches for approximating the unique (a generalisation of many) equivalent martingale and to find the MPR, which is the indifference pricing and arbitrage-free pricing. We cannot use Brownian motion like in the Black-Scholes pricing model for three reasons. Firstly, rainfall is a binary event with extremely volatile peaks making the data non smooth. Secondly, there is no mean-reverting value, i.e. there is no seasonal mean. Thirdly, rainfall is strictly non-negative and does allow for an unbounded random walk. As arbitrage-free pricing is the pricing method currently used within rainfall derivatives [27, 3, 28], we will only focus on this method in this work.

2.2. Arbitrage-Free Pricing

The arbitrage-free pricing approach uses the Esscher transform [35] (synonymous to exponential tilting), which is a generalisation of the Girsanov transform for Brownian processes. The Esscher transform can be seen as a method to change the index value, whilst in most cases retaining the original probability density function (PDF). Numerous distribution functions can be used to achieve this shift as part of the Esscher transform, as long as they are within the exponential distribution family [35]. Therefore, there is a greater choice available and we can fit a distribution that is more suitable to the problem. The use of the Esscher transform changes the jump intensity and jump size under P to the new probability Q_θ. Thus, achieving risk-neutral and arbitrage-free pricing from the predicted rainfall amounts. [36] generalise the transformation to a stochastic process driven by a Lévy process and is applied across a variety of different pricing applications [36, 37, 38].

The Esscher transform changes the probability density $f(x)$ of a random variable X (in our case a probability density of all rainfall pathways based on the accumulated rainfall amount for a given period) to a new probability density $f(x;\theta)$ with parameter θ denoting the MPR, given by:

$$f(x;\theta) = \frac{\exp(\theta x) f(x)}{\int_{-\infty}^{\infty} \exp(\theta x) f(x) dx}. \tag{2}$$

Here we see the Radon-Nikodym derivative with θ being the level of risk exposed to investors from the jumps of the driving process of rainfall. The Esscher transform reflects the corresponding risk by exponentially tilting the jump measure shown by Equation 2 through θ. Many distributions from the exponential family can be used. Those applied within the literature are: Bernoulli, Binomial, Normal, Poisson and Normal Inverse Gamma (NIG) distributions. All of these distributions can take the θ into consideration. The next step is to fit one of the chosen distributions to $f(x)$, the most common one is the NIG, which has 4 parameters to tune: μ for the location, β the skewness, σ the scaling and α for the steepness. Other than the good fit, using the NIG($\alpha, \beta, \mu, \sigma$) benefits from the distribution maintaining its shape [39] under the Esscher transform with parameter θ becoming NIG($\alpha, \beta + \theta, \mu, \sigma$).

The NIG distribution has four parameters and belongs to the generalised hyperbolic distributions. It is used for several applications of risk-neutral modelling across a variety of financial problems, with a PDF in the closed form of:

$$f(x|\alpha,\beta,\mu,\delta) = \frac{\alpha\delta \exp(\delta\sqrt{\alpha^2 - \beta^2} + \beta(x-\mu))}{\pi\sqrt{\delta^2 + (x-\mu)^2}} K_1\left(\alpha\sqrt{\delta^2 + (x-\mu)^2}\right), \tag{3}$$

where K_1 denotes the modified Bessel function of the second kind. The NIG distribution is infinitely divisible and creates a Lévy process $L_t, t \geq 0$, making it ideal for the Esscher transform.

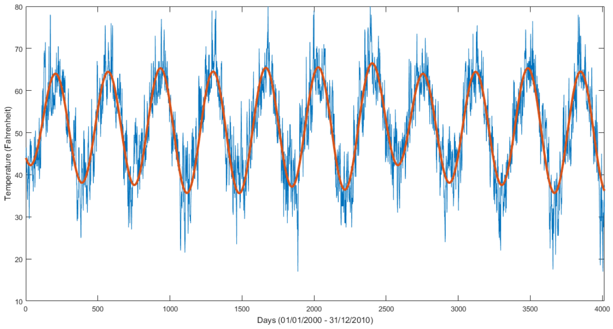


Figure 1: The annual seasonality that exists within temperature as modelled via a truncated Fourier series.

Theoretical prices under Q_θ can be estimated by taking the mean value of the sampled index (MPR = 0) or of the transformed outcome (MPR $\neq$ 0) with a given MPR, which can be negative or positive. This can be assumed and picked arbitrarily at first (constant over time), but it would be wise to consider the value changing over time to deal with different time periods. Having the MPR calibrated with the real market data would go towards finding the most appropriate MPR and hence calculate over time the risk present to investors. Once the MPR has been chosen accordingly, then the unique equivalent martingale is found and prices can then be derived using the formula from Equation 1.

3. Generating Stochastic Equations Through GP

Our proposed method, which we refer to as Stochastic Model GP (SMGP), is a new algorithm for pricing rainfall derivatives. This novel algorithm will overcome the problems from [32], which provided a cumbersome and inefficient methodology for calculating prices. This algorithm aims to provide a better solution (in both predictive and pricing accuracy), whilst not requiring a contract-specific set-up or Monte Carlo Markov Chain (MCMC) to facilitate deterministic solutions. Therefore, our aim in this section is to outline the SMGP's evolutionary process, which can evolve a single stochastic equation for pricing in the entire contract period. The considerations to take into account within the algorithm are the dynamic nature of the time series and to avoid having eight distinct models[3] to run for each city. From our methodology, we deter from building city-specific models and contract-specific models to allow for a generalised framework to facilitate flexible pricing on an ad-hoc basis. For example, a new contract or another use case where rainfall can benefit the final goal. A general framework allows for a plug and play, instead of redefining the problem and model space each time.

3.1. General Model

A general framework for each GP individual (candidate equation) is given by Equation 4:

$$y_t = \phi_t + \kappa_t + \epsilon_t, \tag{4}$$

where t denotes each day, ϕ a seasonal component, κ an autoregressive component, and ϵ a noise component. The motivation for having the ϕ component is to extend each individual into the construction of a stochastic equation (which will be described later). On analysis of the data through rainfall's autocorrelation function, we did not detect any reoccurring seasonality in rainfall on an annual basis like temperature (Figure 1). However, from visually inspecting the time series, there appears to be some element of seasonality on an irregular time scale when examining the time series (Figure 3).

3.2. GP Individual Representation

Each GP individual can be represented, at a high level of abstraction, by the general model given in Figure 2.

[3] One for each monthly derivatives contract traded in a year; rainfall monthly contracts are only traded between March and October.

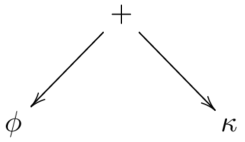


Figure 2: The high-level representation of each individual in the population, consisting of a seasonal and a autoregressive component.

Here we have a GP whose root node takes a “plus” symbol, which combines parameters κ and ϕ. We have a single population of individuals, each one consists of two branches (Equation 4). One for the parameter of ϕ and the other for the parameter of κ; they jointly evolve to minimise the RMSE. The fittest individual for breeding is decided on minimising the RMSE of Equation 4. We choose this procedure to reduce the randomness and to encourage more emphasis on solving the combined problem. Usually, solving the subproblem for the seasonal and the autoregressive component separately is more beneficial, but the GP needs to learn how much emphasis to put on the seasonal and autoregressive part. For example, both models may duplicate the patterns observed and cannot be combined easily. This would be very difficult to generalise considering different seasonal patterns.

Within this framework, parameter κ is a decomposition GP (DGP), which solves the autoregressive part. DGP was first presented in [25], and its main components are summarised in Section 3.4. The only modification of DGP required for it to be used within SMGP is the wrapper that protects trees producing negative values. The modification replaces checking if the prediction is less than zero by checking if the statement $\phi_t + \kappa_t$ is less than zero. If so, the output of the DGP is then modified to satisfy the equation $\phi_t + \kappa_t + d = 0$, where d is the value to offset the output of a GP individual at time t producing a nonnegative output.

3.3. Seasonal component ϕ

Within SMGP, the seasonal component of ϕ is required to create a stochastic equation. It allows SMGP to decide whether the predicted value lies above or below the seasonal effect. Within this section, we outline the methods used to estimate a seasonal pattern. The most common procedure for representing any seasonal effect is through fitting a truncated Fourier series, given by:

$$\phi(t) = \frac{a_0}{2} + \sum_{n=0}^{N} a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right), \tag{5}$$

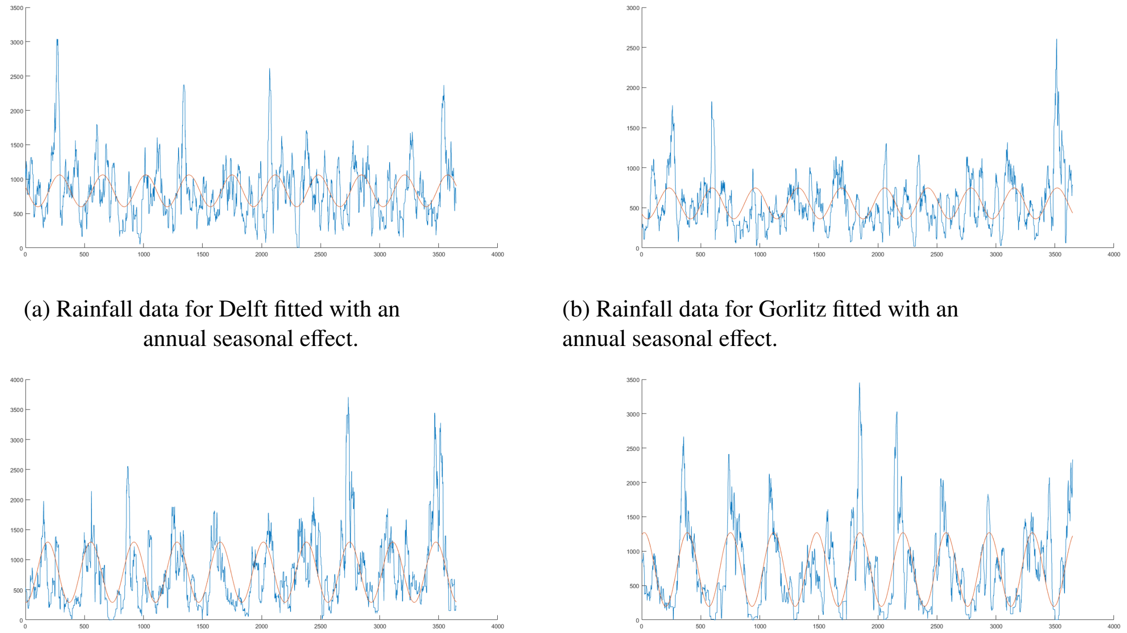
where a and b are constants fitted for the data, n is the order of the fourier series and T is the time period of the seasonal effect. Ideally, we expect a seasonal pattern for $T = 365$, which represents seasonality on an annual basis. For our problem, the effects of seasonality after the data transformation is not consistently the same over a year, which can be observed in Figures 3a, 3b, 3c and 3d. We observe no clear seasonal pattern, which is similar to Figure 1. The truncated Fourier series overestimates and underestimates significant periods over the years.

This shows the problem with detecting and removing seasonality from our time series. We witness some level of seasonality, but not on a consistent scale depending on the data set. For example, we see the same spikes for all time series, but the lags between the spikes varies between 9 months to 15 months across the years. Therefore, there exists some level of seasonality following an irregular pattern, which is difficult to capture correctly. Fourier series unfortunately does not allow for this behaviour as the frequency of sine and cosine waves must be consistent. In order to have the desired behaviour to look for irregular patterns, we design a GP to perform the fitting of seasonality. This uses the fundamental behaviour of Fourier series as a guidance for our model.

3.3.1. Structure of GP for Seasonality

For the GP to include a seasonality feature, we enforce a syntactic structure similar to that of a truncated Fourier series. However, the components within the sine and cosine terms allow for seasonal patterns of variable length. The main components of the proposed GP are as follows:

- All individuals consist of a root node (addition) with the first argument being an intercept and the second being any function.



(a) Rainfall data for Delft fitted with an annual seasonal effect.

(b) Rainfall data for Gorlitz fitted with an annual seasonal effect.

(c) Rainfall data for Des Moines fitted with an annual seasonal effect.

(d) Rainfall data for Portland fitted with an annual seasonal effect.

Figure 3: An attempt at fitting a truncated fourier series for an annual frequency.

- If a sine or cosine node is chosen, the first argument is the amplitude of that wave.
- The amplitude and intercept terms are strictly constants.
- Only within sine or cosine environments there can be terminals that affect the frequency of the curve.

By enforcing these syntactic structures, we are able to control the seasonality, which allows the GP plenty of flexibility to evolve solutions for varying seasonality. In order to enforce the structure of valid solutions and to maintain it throughout evolution, we use a Strongly-Typed GP, the same as the DGP.

3.3.2. Terminals

The terminals we use for ϕ are specifically designed for the seasonal part. The first terminal is the intercept, which is the equivalent to a_0 from the Fourier series. The second terminal is an amplitude, that is similar to the terms a and b from the Fourier series prior to the sine and cosine. It multiplies the output from the sine or cosine function. The third terminal is a dynamic terminal that reflects the time index t of the function, which is incremented with each day till it reaches its seasonal length before repeating from 0. Finally, we have the frequency of the wave. This final term can only exist within a sine or cosine environment.

Similar to the DGP, we have a set of constants specifically for the exponent of the power function, which are in the same range -4 to 4, with 0.25 increments excluding 0.[4]

3.3.3. Functions

The function set includes the same functions as the DGP, also includes sine, cosine and a root node, which must be addition. The list of terminals and functions is summarised in Table 1.

[4] The range was decided during previous tuning experimentations conducted in [25, 26].

Table 1: Genetic Programming functions and terminal sets.

Set	Value
Functions	ADD, SUB, MUL, DIV, POW, SQRT, LOG SIN, COS, ROOT
Terminals	Amplitude, Frequency, Intercept, Dynamic time index, ERC, Constants in the range [-4,4]

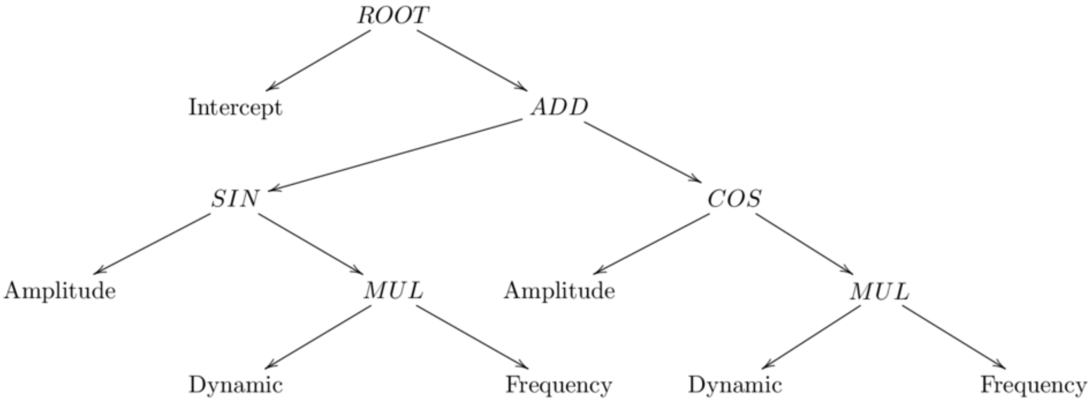


Figure 4: An example tree showing the syntactic structure of a GP individual for the following equation expressed as a truncated Fourier series: f(x) = $a_0 + a_1 sin(h_1 x) + b_1 cos(h_2 x)$, where h refers to $\frac{2\pi}{T}$. Note, more elaborate trees can be developed that use the full range of functions in Table 1, as long as the syntactic constraints are satisfied.

3.3.4. Management of Trees

Additionally, to ensure that only the terminals *frequency* and *dynamic time index* could be chosen inside the sine and cosine environment, we modify the type system to include two types of add, subtract, multiply and divide. One set of types accepts only functions as their arguments, whereas the other set can only be chosen directly within a sine or cosine environment. This allows for other functions and the terminals *dynamic time index* and *frequency*. An example tree showing its syntactic structure is given in Figure 4.

3.4. Autoregressive Component (κ)

The autoregressive component (κ) is based on the DGP, initially proposed in [25]. We refer the reader to this paper for a full explanation. DGP consists of a number of individuals split into two separate populations, a GP part and a GA part. The GP part consists of b expression trees, where nodes represent functions or terminals as usual in a GP. For our implementation we define b to equal 3, such that we have 3 GP equations to predict low, medium and high rainfall amounts. The GA part consists of a linear chromosome with a string of n rules, each with g genes.

The idea behind DGP is that by partitioning the rainfall data into three different partitions (low, medium and high rainfall amounts), we are simplifying the prediction process. Therefore, the GP part of the DGP algorithm will be creating three different rainfall equations, one for each partition. The GA component of the DGP algorithm acts as a classifier, indicating in which partition of rainfall amount each rainfall data point belongs.

3.4.1. Terminals

To be consistent with previous works, we use the same terminals as outlined in [25]. The terminals for κ are defined by the r_t's and r_y's calculated based on the data from Section 4.4, where r_t is the accumulated rainfall amount in the last non-overlapping sliding window t periods ago. For example, t_1 for a data point on the 1^{st} April with a sliding window of 31 days would be the accumulated values from 1st March until 31st March. The same data point for t_2 would consist of the accumulated values from 29th January until 28th February. Similarly, r_y is the accumulated rainfall amount in the current sliding window y years ago.

The second element is an ephemeral random constant (ERC), whose value is a uniformly distributed random number. The third element is a set of constants from -4 to 4, at 0.25 intervals specific to the power function.

3.4.2. Function set

The function set includes: Add (ADD), Subtract (SUB), Multiply (MUL), Divide (DIV), Power (POW), Square root (SQRT), and Log (LOG). The functions LOG, SQRT and DIV are protected. Since we allow for fractional powers, we force a whole number for the second argument of the POW function, if the first argument is negative. The function and terminal sets are summarised in Table 2.

Table 2: GP function an terminal sets

Set	Value
Functions	ADD, SUB, MUL, DIV, POW, SQRT, LOG
Terminals	11 r_t periods $\{r_{t-1}, r_{t-2} \dots r_{t-11}\}$, 10 r_y periods $\{r_{y-1}, r_{y-2} \dots r_{y-10}\}$, ERC, Constants in the range [-4,4]

3.4.3. Management of Trees

To ensure a balance between functions, variables and random numbers in an individual, the first child of each node is either a function or a variable. Whereas, the second child of each node can be a variable, ERC or a function. We initialise the population using the well-known ramped-half-and-half method.

For a detailed presentation of the DGP algorithm, which includes discussions on how rainfall amounts are decomposed, and how the GA and GP parts of this hybrid algorithm operate, we refer the reader to [25].

3.5. Creating a Stochastic Equation

We introduce to our general model from Equation 4 the use of weights. This step is required to transform the deterministic solution to a stochastic solution. The motivation for including weights is to allow the probability of an event to be calculated for pricing and to assist in predicting the level of rainfall. Intuitively, certain parts of the year may be more dominated by κ or ϕ, due to the irregularities of annual rainfall. This allows the SMGP to estimate the most likely outcome at a particular point in time. We propose three variants as an extension to the previous model by using Equations 6, 7 and 8. Each variant specifies the weights differently based on how they interact with our model.

$$y_t = \omega_t \phi_t + (1 - \omega_t)\kappa_t + \epsilon_t, \tag{6}$$

$$y_t = \omega_t^{\phi} \phi_t + \omega_t^{\kappa} \kappa_t + \epsilon_t, \tag{7}$$

$$y_t = \omega_t(\phi_t + \kappa_t + \epsilon_t). \tag{8}$$

In these equations, ω is the weight in the interval [0,1] and ϵ_t is the error term. The motivation for three variations of ω is to promote different behaviours during evolution when estimating the value of y_t. Under all these approaches, there is a balance between ϕ and ω, which forms the basis for the stochastic equation that each individual represents. Under Equation 6, there is a direct tradeoff between ϕ and ω, where one of these two terms can contribute more from the other, or an equal weighting. Under Equation 7, there exists two separate weights, which allows for estimating the independent effect in their respective amounts. Finally, under Equation 8, the combined effect is controlled by one weight.

Through the estimation of ω, we are looking for the optimal value of ω that minimises the RMSE of the SMGP. A typical approach would be using a local search technique to optimise the value of ω throughout the evolutionary process. However, by doing so does not allow us to formulate a stochastic process. Since the end result would be a constant, and a deterministic model would be achieved. To create the stochastic nature of an equation for each individual, the goal is to estimate the weights by using a probabilistic approach. This allows us to perform a random walk on our rainfall values and to estimate a density that reflects each day in our testing set. Going back to the pricing problem, by calculating the probability that a rainfall event occurs under P, we can translate this into the risk-neutral measure of Q.

Algorithm 1 shows the SMGP algorithm, which is described in details within the following sections.

Algorithm 1 Overview of the SMGP algorithm creating the stochastic behaviour.

```
Initialise $\omega$.
Set $S$ (sample size).
for Generation $g = 1, ..., G$ do
    Evaluate each individual of the population.
    Sort population on fitness (RMSE).
    $\omega* \leftarrow$estimateWeights(Predictions$_g \in S$) (Algorithm 2).
    $\omega \leftarrow$updateWeights($\omega_g* \in S, \omega_{g-1} \in S$) (Algorithm 3).
end for
indi* $\leftarrow$ Best individual from training.
Error $\leftarrow$ predictWeights($\omega$, indi*) (Algorithm 4).
```

3.5.1. Sampling and Estimating the Weights

In order to estimate the value of ω to produce a stochastic equation, we specify that the weight follows a beta distribution, given by Equation 9:

$$f(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)}, \tag{9}$$

where $\alpha,\beta > 0$ are both shape functions and $B(\alpha,\beta)$ is the normalising constant. The benefit of the beta distribution is being a continuous probability distribution that is strictly bounded in the interval [0,1]. This property is suitable given we are bounding ω in the same interval, without the need to truncate other distributions within the same range. By sampling ω randomly via the beta distribution we are able to transform Equations 6, 7 and 8 into a stochastic process.

In order to estimate the weights for a given day, we first initialise the weights to be equal to 1, and start updating the weights after the first generation. To estimate the weights, we calculate the percentage difference for each day away from the expected value of rainfall for a set of individuals in the population. Then a beta distribution is fitted to those percentages based on the Maximum Likelihood Estimation of the parameters α and β. The mean of the estimated beta distribution is the weight for that day, that is calculated by $\frac{\alpha}{\alpha+\beta}$. We also keep track of whether the percentage is increasing or decreasing, where our prediction is less than or greater than the expected rainfall amount respectively for our random walk purposes. This is summarised in Algorithm 2.

For Equation 7, we estimate ω^{ϕ} first before estimating the effect of ω^{κ} on the modified values. For Equations 6 and 8, ω can be estimated based on the combined value of κ and ϕ.

3.5.2. Updating and Evaluating the Weights

As each individual evolves, we need to update the weights. Firstly, we estimate the new weights for the day, given the procedure listed above and then we decide whether we choose to accept or reject the new α and β. This is done via Monte Carlo simulation using inversion sampling to generate a uniform selection over our new distribution, taking into account our previous and current values of α and β. We evaluate whether the new values of α and β lead to an improvement in fitness, which is the RMSE of the difference between predicted r_t and actual rainfall amount $\bar{r}_t$, across the set of individuals, otherwise we keep the old α and β. We choose this method because of the possible shapes that can be generated using the beta distribution, where the range can be extremely high. This affects the generalisation of weights throughout evolution. By updating the prior belief with the additional information resulting from the evolution of our SMGP, the weights should converge. With respect to the three Equations 6, 7 and 8, all three are handled in the same manner.

Algorithm 2 Estimating weights for producing stochastic equations

```
$S \leftarrow$ sample size for calculating weights.
Set $\omega_t^1 = 1, \ \forall t = 1, \dots, T$.
Set $\omega_t^2 = 1, \ \forall t = 1, \dots, T$.
for Generation $g = 1, ..., G$ do
    Evaluate each individual of the population.
    Sort population on fitness (RMSE).
    for all $i \in S$ do
        for all $t \in T$ do
            if Prediction$_i^t$ < Actual$_t$ then
                increasingWeights $\leftarrow \frac{\text{Actual}_t - \text{Prediction}_i^t}{\text{Prediction}_i^t}$.
            else
                decreasingWeights $\leftarrow \frac{\text{Prediction}_i^t - \text{Actual}_t}{\text{Actual}_t}$.
            end if
        end for
    end for
    $\alpha^1, \beta^1 \leftarrow$ fitBetaDistribution(increasingWeights).
    $\alpha^2, \beta^2 \leftarrow$ fitBetaDistribution(decreasingWeights).
    for all $t \in T$ do
        $\omega_t^1 = \frac{\alpha_t^1}{\alpha_t^1 + \beta_t^1}$.
        $\omega_t^2 = \frac{\alpha_t^2}{(\alpha_t^2 + \beta_t^2)}$.
    end for
end for
```

As previously mentioned, we keep track of whether the weights increase or decrease the predicted rainfall value, by comparing the actual level of rainfall for that day with the amount predicted. In situations where too much rainfall is predicted, then a weight update is required to reduce the predicted rainfall amount and vice versa. If the rainfall predicted is to be increased, then the inverse of ω is used, as shown by Equation 10:

$$\omega_t = \begin{cases} \omega_t & \text{if} \quad r_{actual} < r_{predicted} \\ \frac{1}{\omega_t} & \text{otherwise.} \end{cases} \tag{10}$$

By producing weights in this manner, we are able to predict the extremes of rainfall in both directions. To avoid excessively large values being generated, we separate the weights according to whether they were under or over estimated. From understanding the data and previous experimentations, we expect weights for the positive shift to be no less than 0.3. Additionally, we would expect the full range from 0 to 1 being used to reduce the rainfall level. The process is summarised in Algorithm 3.

3.5.3. Sampling Future Weights

Up until the final generation, we are merely trying to estimate the best weights for the predictions produced. This is based on the evolutionary process of ϕ and κ, by fitting ω to learn on a daily basis how to achieve y. In order to evaluate the predictive performance in the testing set and to have a stochastic process for pricing, we propose a Markovian approach to sample the weights. By creating a Markov chain, we can produce a random walk with the final result after simulations being a density for each day.

Firstly, the weights calculated for each day are combined into a daily basis. For each day, we sum the respective PDFs to generate a mixture of beta distributions. This gives an indication of the expected weights for a particular period of time. Although we do not expect the same pattern to always occur, often it is the case that the possibility is witnessed in the past. We do this for both sets of increasing and decreasing weights. We perform inverse transform sampling to randomly sample values of weights directly from the cumulative distribution function (CDF) of our new distribution. Figure 5 shows the PDFs and the CDF resulting from the summation of beta distributions for a given day.

Algorithm 3 Updating weights based on new information for stochastic equations

```
Initialise bestFitness.
for Generation $g = 1, ..., G$ do
    estimateWeights (Algorithm 2).
    for $t \in T$ do
        Compute new density for time $t$ from additional information.
        Draw $N$ samples from proposed density.
        Draw $N$ samples from prior density.
        for $n \in N$ do
            if $\text{Predicted}_t^s < \text{actual}_t \ \forall s \in S$ then
                $\text{Predicted}_t^s \leftarrow \frac{\text{Predicted}_t^s}{\omega_t}$.
            else
                $\text{Predicted}_t^s \leftarrow \text{Predicted}_t^s * \omega_t$.
            end if
        end for
        Compute newFitness.
        if newFitness < bestFitness then
            Accept proposed density for $t$.
            bestFitness = newFitness.
        else
            Reject and use prior density for $t$.
        end if
    end for
end for
```

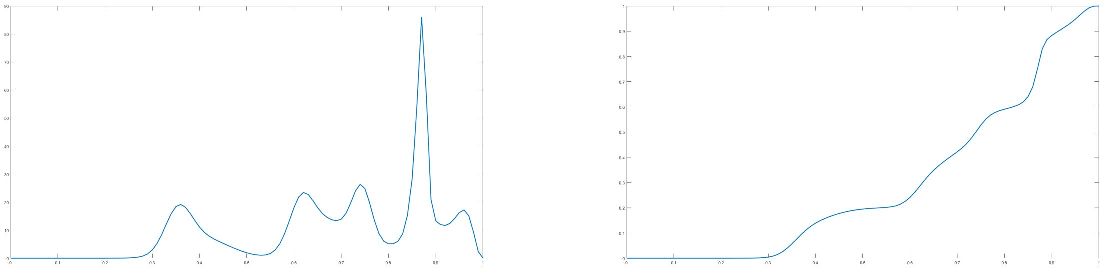
Our Markov chain determines two aspects. Firstly, whether we sample from the increasing or decreasing weights. Secondly, samples from a particular area of that mixture. In order to determine the states, we calculate the transitional probabilities of moving from increasing to decreasing (denoted as $P(d|i)$), decreasing to increasing (denoted as $P(i|d)$), or staying within the same state. Ultimately, this is a two state Markov chain similar to MCRP. However, since we usually have longer periods where we stay in the same state, we also consider the length of the under or over prediction. From previous experimentations, the GP spends a sufficient period of time either under or over predicting. In a minority of cases, there is a frequent switching behaviour. Therefore, we also incorporate a long-run effect that decays geometrically based on the transitional probabilities of switching from either state. This is given by:

$$P(X = x) = p(1 - p)^{x-1}, \tag{11}$$

where x is the day in the current run and p represents the probability of being in either state. Therefore, we are more likely to have longer runs sampling from the increasing weights or the decreasing weights. Once a state is decided, the probability of choosing which part of the mixture to sample from is calculated. The parts of the mixtures are directly linked to the partitions provided by the decomposition part of the DGP. Therefore, the probability is calculated conditioning on the previous day's state, namely high (ω_h), medium (ω_m) or low (ω_l). The rationale is to link ω to how our decomposition perceives the range of values we expect. The motivation is that in the rainfall low state and over predicting, we expect a lower weight than normal to decrease our rainfall amount. For example, going from 200 down to 50 requires a weight much lower compared to going from 350 down to 200.

3.5.4. *Modifying Predictions According to Weights*

Once the weights have been calculated, the final procedure is to transform the rainfall predictions by GP according to the beta distributed randomly sampled weights (Algorithm 4). This is performed by a random walk of our Markov process, where for each day we sample the respective weights calculated from the previous algorithms. The objective is to determine whether the predicted value is under or over estimated, given previous days' states and previous decomposition threshold. Based on the randomly sampled state, the output of GP is modified by either multiplying



(a) PDF

(b) CDF

Figure 5: The PDF and CDF of weights for Jan-01-2011.

or dividing by the weight (decreasing and increasing respectively). The final output's error is then computed via the median of all Markov chains. Furthermore, the final output's predictions will represent a density, indicating the uncertainty in future predictions.

The computational steps for predicting the weights can be summarised as follows:

- Sum probabilistic densities of ω for each day in a year.
- Calculate transitional probabilities.
- Calculate the renewal process.
- Extract densities for $\omega_h, \omega_m, \omega_l$.
- Calculate probabilities for $\omega_h, \omega_m, \omega_l$.
- Run Markov chain (Algorithm 4).
- Calculate median result.

Algorithm 4 Markov chain for estimating the beta adjusted predictions

```
for Iteration $i = 1, \dots, N$ do
    for $t \in T$ do
        Sample state.
        Sample weight given current state and decomposition level.
        if State = increase then
            $\text{predictions}_t^i \leftarrow \frac{\text{Predicted}_t^s}{\omega_t}$.
        else
            $\text{predictions}_t^i \leftarrow \text{Predicted}_t^s * \omega_t$.
        end if
    end for
end for
Error $\leftarrow$ Calculate median of predictions.
```

4. Experimental Set-Ups

4.1. Overview of the Experimental Process

We provide two elements for our experimentation. Our main goal is to price rainfall derivative contracts, but to ensure we reduce issues of mispricing [1], we also analyse the predictive performance.

First, we focus on evaluating the rainfall predicting performance of the proposed SMGP algorithm. During preliminary experiments, we tested all three variants of the algorithm (Equations 6 - 8) on a *validation set*, which was part of the training set (i.e., not part of the testing set used to measure predictive performance), and found that Equation 8 obtained the best mean rank. In addition, in our preliminary experiments we also tested another variant of the algorithm, where we assumed the weights to be constant ($\omega = 1$); this was the equivalent of Equation 4. However, we found that the best performing algorithm on the validation set continued being the SMGP that was using Equation 8. Hence, from this point on, whenever we mention SMGP, we will be referring to the variant with Equation 8.

All algorithms are trained from Jan-01-2001 until Dec-01-2010, and then tested on the unseen test set of Jan-01-2011 until Dec-31-2011. Tests take place on all 42 cities, which are presented in Section 4.4. Reported results are the average RMSE over 50 individual runs. It should also be mentioned that we use a tuning tool called iRace [40] to determine the optimal parameter configuration for all algorithms. Note that iRace had access only to the training set, not the testing set. We present the tuned parameter configurations of the algorithms in Section 4.3.

In the second step of our experiments, we present the theoretical prices produced by all algorithms. We expect the methodology with the lowest RMSE on the testing set to price the closest to the rainfall amount upon the contract's maturity. As we explained earlier, rainfall derivatives are currently traded as monthly contracts for the period from March until October; therefore, there exists 8 contracts per year per city. As a result, we will be comparing the pricing performance of the SMGP equation and the other algorithms on 336 different data points (8 monthly contracts × 42 cities). Currently, the complete pricing data is unavailable and only a few prices for future contracts exists, listed in [2] (these prices are for Detroit, Jacksonville and New York). To overcome this issue we provide an analysis on the theoretical prices produced by all algorithms and we compare them against the actual event of rainfall. We know from the literature that the pricing of a derivatives contract is based on the forecasted level of rainfall over a specified period of time, as defined by each individual contract. In addition, according to derivatives theory [41], prices *must* converge to the actual value of the underlying asset.[5] It is thus imperative to estimate this monthly value as accurately as possible, as this forms the price of a contract. Therefore, if we compare the predicted rainfall amounts against the actual rainfall amounts for each monthly contract, we can observe how far away the proposed prices are.

4.2. Benchmark Methods

In order to test the predictive performance of SMGP, we compare our proposed algorithm against five machine learning methods that are capable of performing regression, including: RBF, SVR, M5 rules, M5 model trees and k-nearest neighbour. Also included is Decomposition GP (DGP), which has been shown to be very effective in rainfall prediction, and the most commonly used method in rainfall derivatives, namely Markov chain extended with rainfall prediction (MCRP). MCRP splits the model into two parts, where it first captures the occurrence pattern (wet or dry X_t), and then the rainfall intensities. The estimated amount of rainfall is given by $R_t = r_t \cdot X_t$. The occurrence process X_t is the order of Markov chain that best fits a city's data based on the Akaike information criterion [43]. The rainfall amount r_t is a randomly drawn value either from the gamma or mixed exponential distribution, whichever explains the data better [44, 45, 46]. As the transitional probabilities and distribution parameters are estimated for each day, a truncated Fourier series is estimated via Maximum Likelihood Estimation (MLE), as suggested by [44]. Finally, as SMGP contains an autoregressive component, we compare against an AutoRegressive Integrated Moving Average (ARIMA) model.

Furthermore, we will examine SMGP's pricing performance. In addition to the previous eight algorithms, SMGP will also be compared to another benchmark method, namely Burn Analysis (BA). We use BA as it is the most frequently used benchmark in financial applications. It calculates prices under P based on the cost and payout of the same contract in the previous year. It computes the expected outcome over the accumulation period $I(\tau_1, \tau_2)$ with an additional risk premium that may occur. Therefore, $Q = P$ and the MPR is zero. The BA cannot price contracts on a daily basis, but it acts as a reasonable benchmark.

4.3. Tuning Parameter Configurations

As mentioned earlier, we use the iRace package [40] to tune the parameters of all our algorithms, including SMGP, all five machine learning algorithms, DGP and the ARIMA model. The general tuning procedure is outlined

[5] The only exception to this statement is the grains market, due to transportation costs [42].

as follows. Firstly, 10 cities that are not included in our main experiments (i.e. these 10 cities are not part of the 42 cities for which we will be presenting our results) are used for the tuning procedure with 65 years worth of data required for each city. Next, we break the data sets into 20 years with 5 years overlap between two consecutive sets, with the final year being the validation set used for determining the optimal parameter set. The 20 years is then used to construct the data into a training set of 10 years, with the final year being the validation check. 20 years is required, because we allow all algorithms to observe rainfall values 10 years ago and the final year is always the validation set to preserve the temporal nature of the data. iRace iteratively considers all tuning data sets, automatically evaluating many different parameter configurations. When the tuning tool finished, the best possible parameter set configuration, based on all tuning data sets is returned. The optimal parameter configuration for SMGP and DGP can be found in Table 3. The optimal parameters for all benchmark machine learning algorithms are found in Table 4 and the optimal ARIMA model returned was ARIMA(1,0,2). The optimal number of neighbours found for KNN was 8.

Table 3: The optimal configuration of SMGP and DGP, as found by iRace.

GP Parameters	SMGP	DGP
Max depth of tree	11	8
Population size	1200	1000
Crossover	83%	99%
Mutation	36%	30%
Terminal/Node bias	52%	64%
Elitism	4%	3%
Number of generations	35	70

Table 4: Optimal parameters using iRace for the four benchmark non-linear models: SVR, RBF, M5R and M5P for daily (top) and accumulated (bottom) rainfall

SVR		RBF		M5R and M5P		
SVM Type	epsilon-SVR	Minimum SD	5.6855	minInstance	6	2
Cost	9.0438	NumClusters	3	Regression tree	yes	yes
Gamma	0.4364	Ridge	4.2139	Unpruned	yes	no
Kernel Type	RBF			Unsmoothed	no	no
Epsilon	0.4909					

4.4. Data

The daily rainfall data used includes a total of 20 cities from around Europe and 22 from around the United States of America (USA). The data was retrieved from NOAA NCDC[6]. The 20 European cities are: Amsterdam (Netherlands), Arkona (Germany), Basel (Switzerland), Bilbao (Spain), Bourges (Germany), Caceres (Spain), Delft (Netherlands), Gorlitz (Germany), Hamburg (Germany), Ljubljana (Slovenia), Luxembourg (Luxembourg), Marseille (France), Oberstdorf (Germany), Paris (France), Perpignan (France), Potsdam (Germany), Regensburg (Germany), Santiago (Portugal), Strijen (Netherlands), and Texel (Netherlands). The 22 USA cities are: Akron, Atlanta, Boston, Cape Hatteras, Cheyenne, Chicago, Cleveland, Dallas, Des Moines, Detroit, Jacksonville, Kansas City, Las Vegas, Los Angeles, Lousville, Nashville, New York City, Phoenix, Portland, Raleigh, St Louis, and Tampa.

Using machine learning methods effectively requires a modification to the data to align it with the problem domain of rainfall derivatives. Following [22] we use a sliding window accumulation method, given by:

$$r_{t_s} = \sum_{t=t_s}^{t_e} r_t, \tag{12}$$

[6] https://www.ncdc.noaa.gov/

Table 5: The average RMSE (in inches of rainfall) for cities in the USA for all machine learning algorithms. Values highlighted in bold represent the best algorithm for each city.

Cities	SMGP	DGP	MCRP	ARIMA	SVR	RBF	M5R	M5P	KNN
Akron	**1.39**	1.88	2.27	2.15	2.15	2.31	2.56	3.14	2.33
Atlanta	**1.66**	2.02	2.63	2.83	2.03	2.03	2.39	2.24	2.38
Boston	1.64	1.72	2.63	1.93	1.76	**1.51**	1.81	1.94	2.17
Cape Hatteras	**1.22**	1.45	1.68	4.50	3.93	3.96	4.17	4.48	4.95
Cheyenne	1.29	1.32	4.81	1.51	1.42	1.30	1.33	**1.28**	1.81
Chicago	**0.91**	1.37	1.71	2.53	3.22	2.84	3.35	2.82	3.52
Cleveland	**2.21**	2.98	3.89	3.03	3.16	2.99	3.10	2.98	2.71
Dallas	**1.35**	1.79	4.21	2.45	1.79	1.97	2.35	2.46	2.19
Des Moines	**1.36**	2.07	2.96	1.53	1.94	2.09	2.19	2.02	3.00
Detroit	**1.56**	2.11	2.86	2.11	2.06	2.22	2.23	2.23	2.18
Jacksonville	**1.64**	2.08	3.01	2.42	2.03	2.09	3.82	3.63	4.08
Kansas	**1.44**	1.89	2.36	3.59	1.77	1.71	1.89	1.84	2.50
Las Vegas	0.48	0.63	0.42	0.51	0.34	**0.32**	0.76	0.73	0.40
Los Angeles	**1.04**	1.10	1.21	1.24	1.08	1.20	1.25	1.30	1.97
Louisville	**2.84**	3.95	5.05	3.53	3.98	3.88	4.19	4.13	3.67
Nashville	**1.42**	1.94	3.21	2.11	2.10	2.08	2.33	2.81	2.11
New York	**3.17**	4.35	5.73	3.82	4.63	4.48	4.65	4.51	5.40
Phoenix	0.45	0.47	0.60	0.46	**0.44**	0.45	0.56	0.47	0.50
Portland	**1.46**	2.12	2.53	1.51	1.95	1.85	2.31	2.18	4.63
Raleigh	**1.54**	2.13	2.79	2.03	2.08	2.07	2.42	2.24	2.74
St Louis	**1.85**	2.32	3.24	2.13	2.28	2.49	2.77	2.60	2.39
Tampa	**2.07**	2.40	3.87	2.54	2.64	2.77	3.01	3.76	4.58

where r_t is the accumulated amount of rainfall for a given day, with the day varying over a contract period from t_s till t_e.

This is consistent with pricing a contract, whereby the price of a contract is the total amount of rainfall within a specified period of time, otherwise known as the contract period. For this paper, we use the modal month length of 31 days, consistent with earlier works [22, 32]. We do not look for an optimum period to accumulate to help with prediction, because our problem domain is set out as the accumulated rainfall amounts over the contracts that are currently traded.

5. Results

Here we outline the results of experiments with the following methods: SMGP, DGP, MCRP, SVR, RBF, M5R, M5P, and KNN. We do not include BA in the error tables for rainfall prediction, but it is included later for pricing, as it is not a predictive technique.

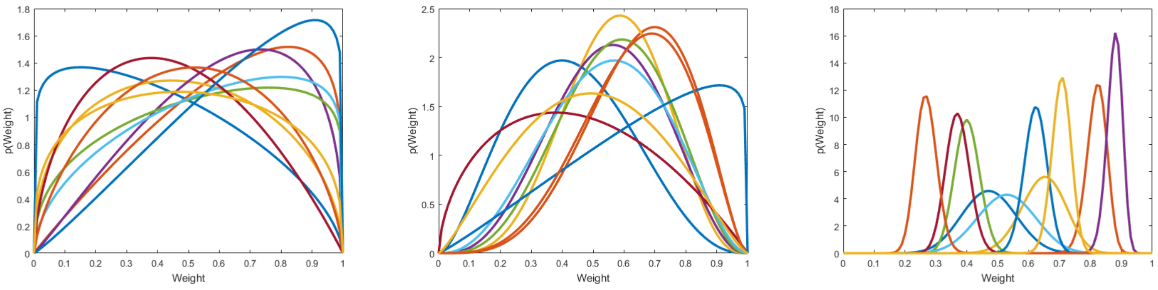
5.1. Predictive Error

We present the findings for all algorithms in Tables 5 (USA) and 6 (Europe). One of the clear observations from these tables is the consistency of the SMGP (mean rank 1.33 across all 42 cities), which has the lowest RMSE (shown in bold) for 18 out of 22 cities in Table 5 and 16 out of 20 cities in Table 6. This indicates that the use of weights has a positive effect on our model. One of the key aspects of the SMGP is modifying the predicted value through the weights to take into account the irregular pattern observed in rainfall. Furthermore, the SMGP provides lower mean ranks across all 42 cities compared to the baseline model of DGP and two powerful blackbox (with non-interpretable models) techniques of RBF and SVR, which have mean ranks of 3.54, 3.44 and 3.73, respectively. SMGP also outperforms all other methods including the most popular financial benchmark of MCRP.

Table 6: The average RMSE (in inches of rainfall) for cities in Europe for all machine learning algorithms. Values highlighted in bold represent the best algorithm for each city.

Cities	SMGP	DGP	MCRP	ARIMA	SVR	RBF	M5R	M5P	KNN
Amsterdam	**1.61**	1.76	3.41	1.99	1.79	1.76	2.17	2.18	1.93
Arkona	**1.31**	1.78	2.49	2.11	2.19	1.93	2.23	2.18	2.26
Basel	1.52	1.66	2.52	1.59	**1.36**	1.44	1.73	1.86	2.00
Bilbao	**1.05**	1.48	1.73	2.48	2.19	2.20	2.75	2.67	5.96
Bourges	**1.29**	1.55	2.89	1.55	1.33	1.32	1.60	1.69	1.88
Caceres	**1.05**	1.27	1.67	1.93	1.31	1.29	1.87	2.29	2.32
Delft	**1.42**	2.00	2.31	2.22	2.10	2.21	2.18	2.19	2.37
Gorlitz	1.48	2.06	3.14	1.55	**1.44**	1.63	1.98	1.71	2.05
Hamburg	**1.28**	1.71	2.11	1.83	1.77	1.68	1.94	2.25	1.77
Ljubljana	**1.98**	2.51	2.52	2.83	2.09	2.31	3.00	2.94	4.77
Luxembourg	**1.50**	1.81	2.16	2.01	1.91	1.85	2.25	2.08	1.88
Marseille	**1.63**	2.04	2.20	2.98	2.05	1.93	2.35	2.16	2.73
Oberstdorf	**2.42**	3.14	4.02	3.24	3.14	3.07	3.39	3.49	3.52
Paris	**1.01**	1.15	1.41	1.15	1.16	1.16	1.24	1.20	1.33
Perpignan	**3.07**	3.80	4.40	3.91	4.22	3.80	3.83	4.01	4.50
Potsdam	**1.22**	1.68	2.06	1.65	1.54	1.43	1.83	1.78	1.91
Regensburg	**1.07**	1.43	1.90	1.48	1.37	1.40	1.68	1.48	2.00
Santiago	2.12	2.73	2.46	5.45	3.11	**2.02**	4.03	3.63	7.00
Strijen	1.59	1.73	2.22	2.12	1.57	**1.50**	1.65	1.52	1.83
Texel	**1.27**	1.85	2.66	2.04	1.86	1.92	2.14	1.97	2.16

From the computational aspect of ω for the weights evolved by SMGP, Figure 6 shows the weights of SMGP converge. Each colour represents a different year for the same day. On the first generation, the initial beta distributions are fitted for the selected individuals for each day based on randomly generated individuals. Future generations show how the more evolved the SMGP individuals get, the more distinct the weights become. By the fifth generation, we can see the weights begin to converge on certain areas, but there is still no real defined areas and the weights still have potential to shift based on the evolution of the SMGP trees. At the twentieth generation, the majority of days has converged on distinct areas. The areas by this stage are well defined, and the last five generations fine tunes the weights and puts more emphasis on the evolution of the SMGP trees to maximise the predictive performance without overfitting.



(a) Generation 1

(b) Generation 5

(c) Generation 20

Figure 6: The convergence of SMGP's weights over the first 20 generations for 1^{st} March for Chicago.

On the focus of the SMGP, the best improvement can be seen in Chicago and Des Moines with a decrease in RMSE by 34% over the second-rank algorithm of RBF and DGP, but overall we have an average decrease of approximately

Table 7: The Friedman test statistic and Holm post-hoc test with the best performing algorithm as the control method (SMGP) based on the RMSE of rainfall prediction. Values in bold represent a significant result at the 5% level, which occurs when the p-value is smaller than the Holm score.

Friedman test statistic		**1.2208x10^{-38}**	
Approach	Ranks	*p*-value	Holm score
SMGP	1.34	-	-
RBF	**3.44**	**4.5489x10^{-4}**	**0.0500**
DGP	**3.54**	**2.2844x10^{-4}**	**0.0250**
SVR	**3.73**	**6.2276x10^{-5}**	**0.0167**
ARIMA	**5.17**	**1.6110x10^{-16}**	**0.0125**
M5P	**6.17**	**7.1580x10^{-16}**	**0.0100**
M5R	**6.79**	**8.7333x10^{-20}**	**0.0083**
KNN	**7.29**	**2.7794x10^{-23}**	**0.0071**
MCRP	**7.52**	**4.7074x10^{-25}**	**0.0063**

22%. In 18 cases, the percentage increase is greater than 25% over RBF and DGP. More importantly, compared to MCRP we are able to improve the predictive accuracy by approximately 60%. This shows a large decrease in predictive error and highlights the disadvantages of the MCRP approach. As MCRP exhibits such high error in comparison to SMGP, we expect our new method to be able to price contracts at the CME more accurately.

We use the Friedman hypothesis test to determine whether or not there are any statistically significant results at the 5% significance level, when comparing the 8 algorithms as a whole. Table 7 shows the Friedman statistic of 1.1608x10^{-37}, which is less than the 5% significance level and shows that one or more algorithms statistically outperformed another. Therefore, we apply the Holm post-hoc test by using the SMGP (the best method) as the control method, in order to determine whether or not SMGP obtained a significantly better result than each of the other 7 algorithms. The results are displayed within Table 7. We observe that the SMGP statistically outperformed all other algorithms at the 5% significance level.

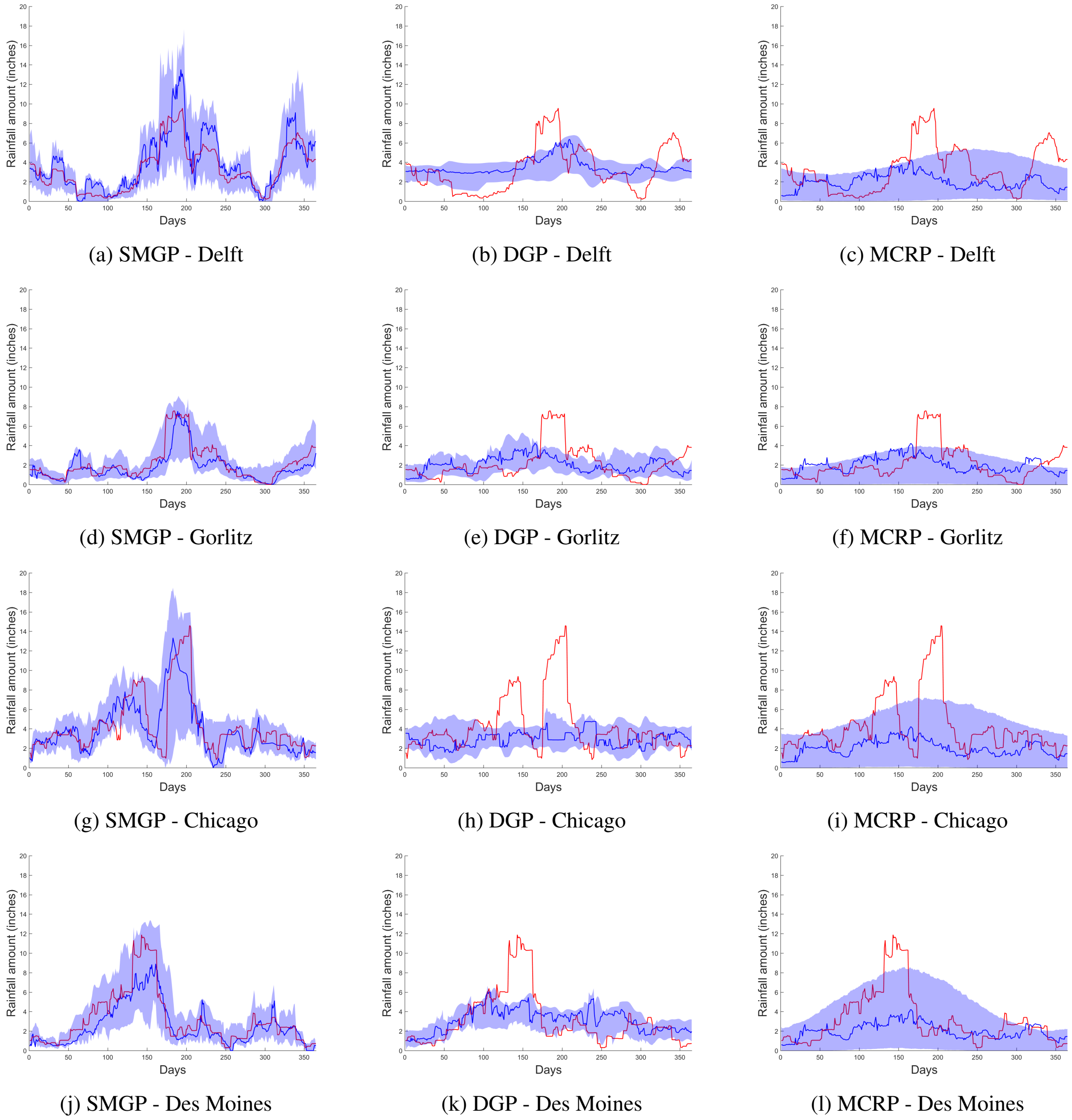
Furthermore, we show in Figure 7, four cities and the effect that the stochastic equation evolved by the SMGP has on rainfall predictions. The left column shows the 95% credible interval (shaded range),[7] the median observation (dark blue) and the actual rainfall (red) for the SMGP. The central column shows the results for the DGP after MCMC [32] and the right column shows the results for MCRP. The four cities used in Figure 7 present results broadly similar (from a qualitative perspective) to the results for the other cities. Hence, we focus on these four cities to simplify the discussion.

For the SMGP, the first observation is that all points are covered within the credible intervals. This indicates that the stochastic equations evolved by the SMGP can adequately predict rainfall pathways. The second observation is that the fluctuations around the median values take into account different parts of the year, where we observe very diverse and inconsistent rainfall periods. The third observation is that the DGP predictions prior to the modification of the weights by the SMGP algorithm are reasonably flat, whereas the weights are creating a more dynamic effect. Therefore, the use of weights indicates that the GP is capable of producing rainfall equations that better represent the behaviour of rainfall. One remarkable aspect is that during the most volatile periods, our stochastic equations are capable of mimicking well the true rainfall behaviour.

The central column of Figure 7 shows the extrapolation of predictions from the DGP using MCMC to estimate a density for each day. It is possible to visualise where the improvements are realised within the SMGP. The construction of the 95% credible interval shows that the peaks and troughs of the time series are not represented adequately. Additionally, none of the four data sets show that the DGP is able to cover the minimum and maximum of the rainfall amount. This is a concern for our model when we consider pricing, because the posterior median probability is not contained within the interval, which results in the probability of pricing a derivative to be zero. Thus, it is likely to reflect in poor pricing and it causes a loss of confidence in our model.

[7] Similar to Monte Carlo methods, we show the credible interval of predictions, as it would not be possible to clearly visualise each individual run. Therefore, we show the median prediction for each time point, as we evaluate the same GP model but through the random sampling of weights, which generates a spread of results.

The right column of Figure 7 shows the credible interval and median predictions for MCRP. The intervals of MCRP are almost capable of predicting all of the minimum and the maximums of rainfall. However, the wide variation of predictions possible for each day causes concerns and shows that predictively MCRP is very weak. It produces a substantial number of pathways not representative of the rainfall process.



(a) SMGP - Delft

(b) DGP - Delft

(c) MCRP - Delft

(d) SMGP - Gorlitz

(e) DGP - Gorlitz

(f) MCRP - Gorlitz

(g) SMGP - Chicago

(h) DGP - Chicago

(i) MCRP - Chicago

(j) SMGP - Des Moines

(k) DGP - Des Moines

(l) MCRP - Des Moines

Figure 7: The 95% credible interval (shaded range) of rainfall for the year from Jan-01-2011 until Dec-31-2011 for SMGP, DGP and MCRP, to be used to estimate pricing over all pathways. The median observation is shown in dark blue and the actual rainfall level in red.

For completeness, we include the SMGP trees from Figure 7 in their equation format (Equations 13 - 16) and not the raw tree for space considerations. From each model we have separated into seasonal, low rainfall, medium rainfall and high rainfall. The seasonal element tree corresponds to ϕ from our initial model equation presented in Equation 4. Note, the subscript of x represents the seasonal length. The low, medium and high make up the autoregressive component of κ from Equation 4.

Delft GP tree

$$\begin{aligned}
\text{Seasonal} &= 833.71 - 121.80cos(0.06x_{1060}) - 131.59cos(\log(0.0807x_{827})) + 4.03sin(0.041x_{649}) \\
\text{Low rainfall} &= \sqrt{R_{t-9}R_{y-9} + 1214.74R_{y-3}} \\
\text{Medium rainfall} &= -\sqrt{858.05R_{t-5}} + \frac{\sqrt{R_{t-10}}}{\sqrt{(R_{y-7})^{-1.75}}} - \frac{169.90R_{t-5}}{32.86} - \frac{\sqrt{R_{y-2}}}{829.42} + R_{t-10} \\
\text{High rainfall} &= \frac{R_{y-9}(\sqrt{R_{y-3}} + (R_{y-1})^2)}{\frac{(R_{y-5})^{-2.5}}{(R_{y-5})^{-0.25}}} + \frac{(R_{y-7})^{0.25}}{R_{y-3} + \sqrt{y-3}}
\end{aligned} \tag{13}$$

Gorlitz GP tree

$$\begin{aligned}
\text{Seasonal} &= 549.31 - 321.80cos(0.021x_{372}) - 131.59cos(\log(2.66x_{1020})) + 4.03sin(0.244x_{1062}) \\
\text{Low rainfall} &= \frac{2.42R_{y-9}R_{y-8}}{R_{t-2}R_{y-7}} \\
\text{Medium rainfall} &= \sqrt{R_{t-1}(R_{y-9} - R_{y-7})} + R_{y-7} \\
\text{High rainfall} &= \left(\log(R_{y-4}\sqrt{R_{y-5}R_{y-4} + (R_{y-5})^2})\right)^{2.5}
\end{aligned} \tag{14}$$

Des Moines GP tree

$$\begin{aligned}
\text{Seasonal} &= 792.11 + 3.19sin(0.01x_{230}) - \log|47.21cos(0.928 - x_{519})| + \frac{104.03cos(0.093x_{980})}{-47.21sin(0.60x_{1858})} \\
\text{Low rainfall} &= \sqrt{R_{t-1}R_{y-2}} \\
\text{Medium rainfall} &= \frac{R_{y-4} + R_{t-2} + R_{y-8} - R_{t-4} + \frac{R_{t-10}}{R_{t-8}} + R_{y-6}R_{t-9}}{\sqrt{R_{t-1}R_{t-4}}} \\
\text{High rainfall} &= \frac{R_{t-10}R_{y-7}(R_{y-7} + R_{t-10})((R_{y-6})^{-}1.5(R_{y-6} - R_{t-8}))}{\log\left|(R_{t-6}R_{t-10})^{2.75} + \sqrt{R_{t-8}(R_{t-4} - R_{t-10} - 234.23)}\right|}
\end{aligned} \tag{15}$$

Chicago GP tree

$$\begin{aligned}
\text{Seasonal} &= 809.72 + 121.80cos(0.06x_{1060}) - 131.59cos(log(0.0807x_{827})) + 4.03sin(0.041x_{649}) \\
\text{Low rainfall} &= (R_{t-2})^{0.75} + (R_{y-7})^2 \\
\text{Medium rainfall} &= \sqrt{783.38R_{y-7}} \\
\text{High rainfall} &= \sqrt{\frac{R_{y-1}(R_{t-10} - R_{y-3})}{R_{y-5} - R_{t-10} + R_{t-1} - R_{t-11} + R_{y-3}}} + \sqrt{R_{y-5}R_{y-3} + \frac{R_{t-10} + R_{t-1}(R_{y-5} + R_{t-1})(R_{y-5}R_{y-3})}{2R_{t-11} + R_{t-1}}}
\end{aligned} \tag{16}$$

where, the subscript of x represents the seasonal length, R_{t-n} represents the accumulated level of rainfall n time period(s) ago and R_{y-n} represents the accumulated level of rainfall n year(s) ago. The latter two variables use the definition presented earlier in Section 3.4.1.

Looking at the trees generated, one aspect we notice is that the seasonal length varies and is not consistent to a single frequency as we would expect with a Fourier transformation. This backs our earlier observation that rainfall

Table 8: The median absolute deviance (in inches) for cities in the USA, based on the predicted rainfall against the accumulated rainfall for each of the eight contracts that are traded for rainfall derivatives. Values in bold represent the best median absolute deviance for each city.

Cities	SMGP	DGP	MCRP	ARIMA	SVR	RBF	M5R	M5P	KNN	BA
Akron	**0.95**	1.93	3.25	2.19	2.78	2.61	2.70	1.92	2.45	2.10
Atlanta	0.82	1.65	**0.54**	0.94	0.89	1.26	1.59	1.32	1.53	1.43
Boston	**0.75**	0.8	1.77	2.22	1.52	1.32	1.46	1.67	1.08	2.25
Cape Hatteras	2.15	2.91	2.53	3.51	3.74	3.67	3.06	3.48	4.64	**0.26**
Cheyenne	0.60	**0.53**	0.93	1.19	1.35	1.99	1.58	3.11	0.78	1.48
Chicago	**0.48**	0.82	1.97	1.59	1.21	1.35	1.12	1.62	1.00	3.83
Cleveland	1.96	3.63	4.91	3.92	4.11	4.05	3.97	3.62	5.93	**1.91**
Dallas	**0.45**	1.93	1.23	1.22	2.24	2.94	2.28	2.21	2.99	1.37
Des Moines	**0.93**	1.35	1.06	1.20	1.30	1.44	1.35	1.98	1.37	2.44
Detroit	**0.59**	1.93	3.38	3.41	3.30	2.99	2.67	3.74	2.25	1.22
Jacksonville	1.09	1.20	1.33	1.25	**0.84**	0.89	1.45	1.33	3.85	1.62
Kansas	**0.33**	0.96	1.43	1.23	1.16	1.23	1.51	1.02	1.68	1.18
Las Vegas	0.05	0.17	0.14	0.40	0.05	0.25	0.14	**0.04**	0.36	0.17
Los Angeles	**0.02**	0.12	0.14	0.12	0.19	0.20	0.04	0.19	2.43	0.08
Louisville	**1.57**	1.77	3.91	2.45	2.32	2.25	2.91	2.65	2.14	1.92
Nashville	**0.50**	1.08	2.03	1.53	1.67	1.60	1.79	1.90	1.75	1.55
New York	**0.91**	1.35	3.64	2.72	2.00	1.57	2.27	1.42	3.31	1.90
Phoenix	**0.03**	0.23	0.23	0.33	0.08	0.26	0.11	0.09	0.13	0.21
Portland	**0.34**	0.86	0.49	0.64	0.57	0.62	1.11	0.59	5.76	0.68
Raleigh	1.44	1.27	1.62	2.34	1.15	**1.03**	2.82	1.42	1.64	1.16
St Louis	**1.27**	1.80	1.55	2.15	1.94	1.74	1.87	1.92	1.80	1.84
Tampa	**0.76**	1.90	3.38	2.55	2.08	2.29	2.51	2.84	4.48	1.61

seasonality is irregular and does not have a reoccurring pattern. For the autoregressive aspect of low, medium and high rainfall equations, there appears to be a good mix of rainfall parameters and the provided functions. We observe that for cities like Gorlitz, a greater selection of previous years' worth of parameters were chosen over shorter term parameters (more y's than t's), showing that previous years' values carry more information and hint towards a longer-term reoccurring pattern. On the other hand, both U.S. cities have a stronger mix of long run and short parameters, with an almost equal number of y's and t's in their final equations, indicating some reoccurring pattern, but does depend on more recent behaviour to indicate future rainfall levels.

To sum up, there are large benefits from estimating the irregularities in seasonal effect and randomly sampling according to an underlying Markovian process. The key benefits of the method are that it is computationally less expensive and is effective. It requires fewer generations and runs to estimate a density (reflected by the GP parameters), as well as, no estimation required through MCMC for each day. This translates to efficiency gains between 2-3 times compared to the DGP. Moreover, the predictive error is consistently reduced on the testing set over all other approaches by around 22%.

5.2. Pricing Performance

Regarding pricing performance, we fit each density (P) with the NIG distribution by using the well-known expectation-maximisation algorithm to estimate the four parameters. The risk-neutral density follows a Lévy process, so that we are able to shift the distribution (Q) according to the MPR (θ) through the Esscher transform: NIG(α,β, γ, δ) = NIG(α,$\beta + \theta, \gamma, \delta$). Once it is performed, the expected level of rainfall of the new distribution becomes our risk-neutral prices.

From looking at Tables 8 and 9, we can observe that SMGP ranks first more often (28 out of 42 cities) based on the median deviance. This demonstrates that SMGP is capable of predicting rainfall amounts more consistently than other techniques for the key dates we are interested in. Interestingly, we can observe that SMGP is able to consistently outperform DGP (second in mean rank) in 37 cities. This further demonstrates the improvements that are realisable by

Table 9: The median absolute deviance (in inches) for cities in Europe, based on the predicted rainfall against the accumulated rainfall for each of the eight contracts that are traded for rainfall derivatives. Values in bold represent the best median absolute deviance for each city.

Cities	SMGP	DGP	MCRP	ARIMA	SVR	RBF	M5R	M5P	KNN	BA
Amsterdam	**0.65**	0.74	1.64	1.63	1.55	1.03	1.41	1.36	1.96	1.76
Arkona	1.18	0.98	1.32	2.59	2.56	3.03	1.95	1.70	3.24	**0.73**
Basel	0.98	1.63	0.98	1.95	1.75	2.28	**0.97**	2.05	6.82	1.32
Bilbao	**0.81**	1.67	0.93	1.73	1.92	2.02	2.18	1.66	1.31	1.78
Bourges	**0.34**	0.88	1.04	0.73	0.88	0.99	0.74	0.79	1.68	0.86
Caceres	1.27	0.48	0.68	1.93	3.44	3.94	2.83	4.46	1.44	**0.39**
Delft	**0.80**	1.41	1.04	1.47	1.42	1.45	1.68	2.44	1.13	1.22
Gorlitz	**0.63**	0.92	1.08	1.30	1.07	1.03	0.98	1.17	1.13	0.70
Hamburg	**1.01**	1.34	0.87	2.21	1.89	2.18	2.17	3.16	1.97	1.19
Ljubljana	**0.88**	2.20	2.33	2.04	1.98	1.51	2.51	2.98	3.91	1.47
Luxembourg	0.92	1.04	1.08	1.94	1.02	1.39	1.68	**0.62**	1.46	1.21
Marseille	**0.79**	1.87	1.16	2.49	1.60	1.80	2.31	1.87	2.23	1.85
Oberstdorf	**0.89**	1.98	2.60	1.50	1.47	1.55	1.87	1.39	1.92	1.65
Paris	**0.43**	0.95	0.98	1.52	1.04	1.05	1.53	1.20	1.19	0.81
Perpignan	**0.74**	1.14	1.39	1.24	1.08	1.22	0.99	1.19	0.90	1.13
Potsdam	0.80	0.90	1.31	0.92	0.88	0.62	1.76	1.21	1.51	**0.56**
Regensburg	0.94	0.96	1.11	1.94	1.16	1.32	1.64	**0.87**	1.81	1.23
Santiago	1.73	0.97	**0.76**	2.34	1.77	2.22	2.52	1.86	7.60	2.23
Strijen	**0.56**	0.67	1.15	2.42	1.02	1.24	1.37	0.97	1.36	1.18
Texel	**0.59**	1.40	2.15	1.28	1.25	1.27	2.04	1.17	1.72	1.35

SMGP. By comparison, the second best algorithm regarding the number of wins is BA with 5. However, even though BA ranked second in terms of wins, it was often the worst performer.

In order to determine the effectiveness of SMGP for the periods that correspond to rainfall derivatives contracts, we use the Friedman test. Similarly to the previous comparison, we determine whether there is any significant difference among the different algorithms at the 5% significance level. The results of the Friedman test can be found in Table 10, which also includes the mean ranks based on the full set of results. As our Friedman test statistic is significant at the 5% level (p-value is 1.4262x10^{-62}), we use the Holm post-hoc test to compare the control (best) algorithm against each of the others.

We observe from Table 10, that SMGP is the best performing algorithm with a mean rank of 3.13 across all 42 cities. We witness a large ma8rgin in mean ranks between the first ranked approach (SMGP) and second place (DGP), which has a mean rank of 4.72. In comparison to the other machine learning methods, this demonstrates the consistent decrease SMGP has in predictive error, as observed earlier in Tables 5 and 6.

The results in this work, show that more accurate pricing is possible based on improvements in modelling the underlying variable. As we have discussed, we are able to derive more accurate theoretical prices without taking into account the MPR (assuming MPR = 0). From a pricing perspective this is of great benefit, because the initial contract prices generated by SMGP are more often closer to the true value of rainfall. This should reduce the volatile swings of price changes the nearer a contract gets to maturity. As the accuracy of the final price is increased, more certainty is provided for investors.

To summarise our findings, we found that our proposed SMGP method has significantly reduced the rainfall predictive error. Moreover, the SMGP led to the largest consistent error reduction, with on average 22% decreases in predictive error across all data sets, compared to RBF and DGP. For pricing accuracy, we observe in the level of rainfall prediction for each contract period, the SMGP comprehensively outperforms all other techniques. These results are very significant for the field, which increases the confidence and accuracy of pricing for rainfall derivatives.

Table 10: The mean rankings of all algorithms, the Friedman test statistic with the best performing algorithm (SMGP) being the control method for the Holm post-hoc test. Values in bold represent a significant difference at the 5% significance level, which occurs when the p-value is smaller than the Holm score.

Friedman statistic	**3.1137x10^{-66}**		
Algorithm	Mean rank	*p*-value	Holm score
SMGP (control)	3.13	-	-
DGP	4.72	**2.2471x10^{-13}**	0.0500
BA	4.89	**5.6955x10^{-15}**	0.0250
SVR	4.94	**8.1231x10^{-20}**	0.0167
RBF	5.83	**2.0923x10^{-24}**	0.0125
MCRP	5.93	**1.1316x10^{-26}**	0.0100
ARIMA	6.42	**1.1316x10^{-27}**	0.0083
M5P	6.69	**3.2444x10^{-28}**	0.0071
M5R	6.72	**2.1321x10^{-44}**	0.0063
KNN	7.22	**1.3259x10^{-50}**	0.0056

6. Conclusion

To conclude, this paper presented a novel GP algorithm for deriving pricing equations within rainfall derivatives, named Stochastic Model GP (SMGP). Through SMGP, we transformed the idea of deterministic white-box models to stochastic white-box models. This allowed us to estimate a daily density directly through GP. To achieve the stochastic nature, we formulated a general model with the addition of weights that followed a beta distribution, which is randomly sampled over time. We examined different variants of the SMGP algorithm, and we found that a single weight affecting the combination of the autoregressive and seasonal components showed the best performance, compared to a tradeoff approach and two weights affecting each component independently.

The rainfall prediction results showed that the SMGP was the most suitable algorithm, which *significantly outperformed all other machine learning algorithms on all data sets*. It achieved the lowest predictive error and is favourable for rainfall derivatives, based on the correlation between predictive error and the pricing accuracy [1, 33]. Whilst we observed evidence that this statement is true, we were unable to fully test this, because of the unavailability of daily prices. However, we noticed that the SMGP predicted the actual rainfall for each contract more accurately than all other algorithms. The results achieved contribute significantly both to the literature and to the practice of rainfall derivatives. The methodology is able to provide more certainty for future events by a more accurate predictive model.

Future work includes adopting Bayesian inference techniques for having a formal definition of a beta process with a filtration process, in order to further improve the estimation and predictive nature of the weights. When pricing data becomes available, more attention can be given to calculating the market price of risk and understanding the relationship between the underlying variable and the prices of contracts. This would help SMGP as the weights can be extended to account for the Esscher transform. Finally, the dynamics of the market price of risk over time can be estimated for daily pricing for all contracts.

[1] A. Alexandridis, A. Zapranis, Weather Derivatives: Modeling and Pricing Weather-Related Risk, New York, NY: Springer New York, 2013.

[2] R. Carmona, P. Diko, Pricing precipitation based derivatives, International Journal of Theoretical and Applied Finance 08 (07) (2005) 959–988.

[3] B. L. Cabrera, M. Odening, M. Ritter, Pricing rainfall futures at the CME, Journal of Banking & Finance 37 (11) (2013) 4286 – 4298.

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

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

[6] A. Agapitos, M. O´ Neill, A. Brabazon, Genetic programming for the induction of seasonal forecasts: A study on weather derivatives, in: Financial Decision Making Using Computational Intelligence, Springer, 2012, pp. 159–188.

[7] A. Agapitos, M. O´ Neill, A. Brabazon, Evolving seasonal forecasting models with genetic programming in the context of pricing weather-derivatives, in: Applications of Evolutionary Computation, Springer, 2012, pp. 135–144.

[8] A. K. Alexandridis, M. Kampouridis, S. Cramer, A comparison of wavelet networks and genetic programming in the context of temperature derivatives, International Journal of Forecasting 33 (1) (2017) 21 – 47.

[9] A. Alexandridis, M. Kampouridis, Temperature forecasting in the concept of weather derivatives: A comparison between wavelet networks

and genetic programming, in: L. Iliadis, H. Papadopoulos, C. Jayne (Eds.), EANN, Vol. 383 of CCIS, Springer Berlin Heidelberg, 2013, pp. 12–21. doi:10.1007/978-3-642-41013-0_2.

[10] D. S. Wilks, Multisite generalization of a daily stochastic precipitation generation model, Journal of Hydrology 210 (1998) 178–191.

[11] M. Ritter, O. Muhoff, M. Odening, Minimizing geographical basis risk of weather derivatives using a multi-site rainfall model, Computational Economics 44 (1) (2014) 67–86.

[12] M. Cao, A. Li, J. Z. Wei, Precipitation modeling and contract valuation, The Journal of Alternative Investments 7 (2) (2004) 93–99.

[13] M. Odening, O. Musshoff, W. Xu, Analysis of rainfall derivatives using daily precipitation models: opportunities and pitfalls, Agricultural Finance Review 67 (1) (2007) 135–156.

[14] R. W. Katz, Precipitation as a chain-dependent process, Journal of Applied Meteorology and Climatology 16 (7) (1977) 671–676.

[15] N. Q. Hung, M. S. Babel, S. Weesakul, N. K. Tripathi, An artificial neural network model for rainfall forecasting in bangkok, thailand, Hydrology and Earth System Sciences 13 (8) (2009) 1413–1425.

[16] J. Wu, J. Long, M. Liu, Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm, Neurocomputing 148 (2015) 136 – 142.

[17] Mislan, Haviluddin, S. Hardwinarto, Sumaryono, M. Aipassa, Rainfall monthly prediction based on artificial neural network: A case study in tenggarong station, east Kalimantan - Indonesia, Procedia Computer Science 59 (2015) 142 – 151, international Conference on Computer Science and Computational Intelligence (ICCSCI 2015).

[18] A. Danandeh Mehr, Month ahead rainfall forecasting using gene expression programming, American Journal of Earth and Environmental Sciences 1 (2) (2018) 63–70.

[19] A. Danandeh Mehr, V. Nourani, V. Karimi Khosrowshahi, A hybrid support vector regressionfirefly model for monthly rainfall forecasting, International Journal of Environmental Science and Technology.

[20] H. Weerasinghe, H. Premaratne, D. Sonnadara, Performance of neural networks in forecasting daily precipitation using multiple sources, Journal of the National Science Foundation of Sri Lanka 38 (3) (2010) 163–170.

[21] O. Kisi, J. Shiri, Precipitation forecasting using wavelet-genetic programming and wavelet-neuro-fuzzy conjunction models, Water Resources Management 25 (13) (2011) 3135–3152. doi:10.1007/s11269-011-9849-3.

[22] S. Cramer, M. Kampouridis, A. A. Freitas, A. K. Alexandridis, An extensive evaluation of seven machine learning methods for rainfall prediction in weather derivatives, Expert Systems with Applications 85 (2017) 169 – 181.

[23] S. Cramer, M. Kampouridis, A. A. Freitas, A. Alexandridis, Predicting rainfall in the context of rainfall derivatives using genetic programming, in: Computational Intelligence for Financial Engineering and Economics, 2015 IEEE Symposium Series on, 2015, pp. 711–718. doi:10.1109/SSCI.2015.108.

[24] S. Cramer, M. Kampouridis, A. A. Freitas, Feature engineering for improving financial derivatives-based rainfall prediction, in: Proceedings of 2016 IEEE Congress on Evolutionary Computation, IEEE Press, Vancouver, 2016.

[25] S. Cramer, M. Kampouridis, A. Freitas, A genetic decomposition algorithm for predicting rainfall within financial weather derivatives, in: Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO '16, ACM, New York, NY, USA, 2016, pp. 885–892.

[26] S. Cramer, M. Kampouridis, A. A. Freitas, Decomposition genetic programming: An extensive evaluation on rainfall prediction in the context of weather derivatives, Applied Soft Computing 70 (2018) 208 – 224.

[27] F. E. Benth, J. Š. Benth, Modelling and pricing derivatives on precipitation, World Scientific, 2012, Ch. 8, pp. 179–195.

[28] R. C. Noven, A. E. D. Veraart, A. Gandy, A lévy-driven rainfall model with applications to futures pricing, Advances in Statistical Analysis 99 (4) (2015) 403–432.

[29] I. Rodriguez-Iturbe, D. R. Cox, V. Isham, Some models for rainfall based on stochastic point processes, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences 410 (1839) (1987) 269–288.

[30] L. Le Cam, A Stochastic Description of Precipitation, in: J. Neyman (Ed.), Fourth Berkeley Symposium on Mathematical Statistics and Probability, 1961, pp. 165–186.

[31] E. Waymire, V. K. Gupta, I. Rodriguez-Iturbe, A Spectral Theory of Rainfall Intensity at the Meso-β Scale, Water Resources Research 20 (1984) 1453–1465. doi:10.1029/WR020i010p01453.

[32] S. Cramer, M. Kampouridis, A. A. Freitas, A. Alexandridis, Pricing rainfall based futures using genetic programming, in: EvoBAFIN, EvoStar, Springer-Verlag Berlin, 2017.

[33] S. Jewson, C. Ziehmann, A. Brix, Weather Derivative Valuation, Cambridge University Press, 2010.

[34] B. Jenson, J. Nielsen, Pricing by no arbitrage, in: D. Cox, D. Hinkley, O. Barndorff-Nielsen (Eds.), Time Series Models: In econometrics, finance and other fields, Chapman & Hall/CRC, Taylor & Francis, 1996.

[35] F. Esscher, On the probability function in the collective theory of risk, Scandinavian Actuarial Journal 1932 (3) (1932) 175–195.

[36] H. Gerber, E. Shiu, Option pricing by esscher transforms., Insurance Mathematics and Economics 16 (3) (1995) 287.

[37] H. Bhlmann, F. Delbaen, P. Embrechts, A. N. Shiryaev, On esscher transforms in discrete finance models, ASTIN Bulletin: The Journal of the International Actuarial Association 28 (02) (1998) 171–186.

[38] E. Kremer, A characterization of the esscher-transformation, ASTIN Bulletin: The Journal of the International Actuarial Association 13 (01) (1982) 57–59.

[39] F. Esche, M. Schweizer, Minimal entropy preserves the lvy property: how and why, Stochastic Processes and their Applications 115 (2) (2005) 299 – 327.

[40] M. López-Ibáñez, J. Dubois-Lacoste, T. Stützle, M. Birattari, The R package (irace) package, Iterated Race for automatic algorithm configuration, Tech. rep., IRIDIA, Université Libre de Bruxelles, Belgium (2011).

[41] J. Hull, Options, futures, and other derivatives, 6th Edition, Pearson Prentice Hall, Upper Saddle River, NJ [u.a.], 2006.

[42] K. Guo, T. Leung, Understanding the non-convergence of agricultural futures via stochastic storage costs and timing options, Journal of Commodity Markets.

[43] H. Akaike, A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (1974) 716–723.

[44] J. Roldán, D. A. Woolhiser, Stochastic daily precipitation models: 1. a comparison of occurrence processes, Water Resources Research 18 (5) (1982) 1451–1459.

[45] T. A. Buishand, Some remarks on the use of daily rainfall models, Journal of Hydrology 36 (1978) 295–308.

[46] D. Wilks, Interannual variability and extreme-value characteristics of several stochastic daily precipitation models, Agricultural and Forest Meteorology 93 (3) (1999) 153 – 169.

Glossary

arbitrage Risk free profit.. 1

arbitrage-free pricing The main pricing method for rainfall derivatives, based on the generalisation of the Black-Scholes model.. 2

Burn Analysis A technique to replicate previous historical events, to project with some level of risk to a future point in time.. 3

derivative A contract between 2 or more parties, where the value is determined on the underlying variable.. 1

hedge To protect against unfavourable market conditions.. 1, 4

indifference pricing Where a buyer/seller of a contract is indifferent between a range of two prices.. 2

Lévy process A stochastic process with independent, stationary increments, where successive displacements are random and independent and statistically i.i.d. over different time periods.. 5

Market Price of Risk The additional return or risk premium expected by investors for being exposed to undertaking an unprotected risk.. 4

martingales A martingale is a sequence of values of a random variable, where at a particular time in the realised sequence, the expectation of the future value is equal to the present observed value.. 4

risk-neutral The derivatives price is the discounted expected value of the future payoff.. 2

risky world The derivatives price has arbitrage opportunities.. 2

Theoretical prices The derivatives price for incomplete markets assuming a market price of risk of 0.. 5