Market Risk Assessment of a trading book using Statistical and Machine Learning

Noureddine Lehdili*[,1], Harold Guéneau[1,2], and Pascal Oswald[1]

[1]Natixis, Enterprise Risk Management, Market and Counterparty Risk Modelling
[2]AiYO

2019

Abstract

The internal models of banks are questioned by regulators that request to upgrade and improve the underlying quantitative risk methodologies and to implement others new quantitative rules. In this context, we are interested in how banks to complete the probate process of their internal models used to compute the regulatory capital. To address this question, it worth to focus on the more challenging aspect of market risk measurement task such as the practical implementation of the Value-at-Risk (VaR) and the Expected Shortfall (ES) models of the trading portfolio of the bank. Given the important number of risk factors behind the trading portfolio, these risk models are implemented using several proxies and simplistic assumptions for the performance purposes.

In this paper, we roll out Gaussian Processes (GP), a powerful algorithm for both regression and classification, to tackle the revaluation performance problem of the whole derivatives securities trading portfolio. To do this, the GP algorithm must learn the valuation task of all pricing models by training it directly on a data set generated by sophisticated pricing models. Then, the trained GP algorithm takes over to carry out the repeated revaluation of the derivatives security's portfolio in a fast and efficient way. The numerical tests show that the Gaussian Process Regression (GPR) can drastically improve the computing time whilst ensuring an excellent level of accuracy. In addition, the speed-up of the calculation of the VaR and the expected shortfall within the GPR framework is independent of the size and the composition of the trading portfolio, which is spectacular. The GPR is therefore a powerful algorithm that can be apply to risk measurement by financial institutions, in order to improve the accuracy and to speed-up current methods.

1 Introduction

Managing risk has always been a central and important activity of banking but since the global financial crisis it has increased in importance, and there is a more and more focus on how risks are being detected, quantified, monitored, reported and managed. Since, the internal models are questioned by regulators and the implementation of the new regulations is underway. Within the market risk management framework and the Fundamental Review of the Trading Book (FRTB),

*Corresponding author: noureddine.lehdili@natixis.com

market risk methodologies have most notably become more complex. Over the past thirty years, an abundant research ([7], [13], [15], [24], [28], [27], [34], [14], [1], [3], [2]), both in academia and financial industry, has focused on the developments in banking and risk management and the current and future challenges. In parallel, machine Learning, a branch of artificial intelligence, has received a lot of interest in recent years thanks to the Big Bata in business application, with many solutions already implemented and many more being studied and explored (([17], [26], [29], [11]). In finance, there are many applications (([10], [18], [12], [23]): pricing, fraud detection, credit scoring, portfolio management. They concern both retail banks and financial markets. For this, data has always been used, but until then the trend has been oriented towards the design of complex models with stochastic processes. A recent study of the Financial Stability Board ([17]) outlines that the machine learning can allow the building of more accurate risk models by identifying complex and nonlinear patterns within large datasets. According to the same study, the applications of machine learning in regulation and risk management constitute a new supplementary field of scientific research has opened where practically everything remains to be done. it is expected that this technology will have implications for financial services industry, and it will help in the transformation of the risk management function at banks.

Even there is an intensive research on market risk modeling to develop more rigorous and sophisticated risk management tools and models, machine learning algorithms are not yet really deployed in the context of classical quantitative finance and risk management problems. Nevertheless, since the last three years research papers have been published on the options pricing topics. The Artificial Neural Network (ANN) techniques are used to value financial options and to calculate implied volatilities with the aim of accelerating the corresponding numerical methods ([35], [9]). Others unsupervised and supervised algorithms, as Gaussian Mixtures and Gaussian Process, are also applied to the yield curve forecasting and derivatives pricing ([36], [8]) and this allows to speed-up and reduce computation times for calculation of the vanilla and exotic options under advanced models. On the other hand, within the market risk framework, it is established in ([16]) that volatility forecasting in the financial market can be improved by using the ANN models. On the same topic, Zhang et al. ([19]) have been proposed an interesting approach founded on the GARCH model and Extreme Machine Learning (ELM) algorithm to estimate volatility. The proposed non-linear random model, called GELM, predicts the volatility of target time series using GELM-RBF and extrapolating the predicted volatilities allows for the calculation of Value-at-Risk with improved performance in terms of accuracy and efficiency. The GELM model is built on a stochastic mapping method that doesn't require the Gaussian likelihood for estimation and is a not linear data driver model. Despite these various research axes, no machine learning method does not been imposed herself in the financial risk measurement framework.

In parallel, another line of research explores the use of Gaussian processes and Bayesian optimization in finance. Indeed, the Gaussian Process (GP) is a generalization of a Gaussian random vector and can be a stochastic process on continuous function where the covariance matrix can be replaced by a kernel function. Two interesting applications founded on the Gaussian Process Regression (GPR) are recently published (for more details, see [22] and [37]). The first one is dedicated to the fast derivative pricing and hedging where the market is often modelled with a limited set of parameters. This is adapted for statistical learning since the number of input parameters used in training set is small which easily allows the machine to learn the actual result function. The main advantage put forward by the authors ([22]) is the speed up time-consuming calculations and in the same time exotic option prices, sensitivities and hedge parameters are learned accurately. The second contribution ([37]) is devoted to the Value-at-Risk and expected shortfall calculations of a portfolio of derivatives. This work is directly based on the research paper ([22]) insofar as the repeated revaluation of the derivatives portfolio is carried out as a linear combination of each

instrument price according the GPR approach. Numerical tests presented in ([22]) highlight on an example of portfolios composed of vanilla and barrier options, the GPR algorithm leads to identical results as of the full revaluation and outperforms Taylor expansion approach. So, if we refer to these results, it is obvious that applications for risk management and regulatory capital calculations are promising. Therefore, in the sequel we are interested to deep the quantitative analysis and to explore the alternative directions to improve the pricing and computation of the value-at-risk and the expected shortfall of a whole trading book. It can be very useful since within the Fundamental Review of Trading Book (FRTB) where the regulatory requirements are became very cost to implement since the full revaluation of the derivatives portfolios is needed at different time horizons, up to 120 trading days in the future (Basel Committee on Banking Supervision, [30]).

In this paper, we will focus on one side on the classical problem of the valuation and the risk management of derivatives and on the other side on the Value-at-Risk and Expected Shortfall measures and the VaR back-testing. Contrary to the approach followed by the authors of [22] and [37] where the option pricing is carried out in the unitary manner, we deploy the machine learning technique GPR with the whole trading portfolio where the algorithm is directly trained on a set of the portfolio values without necessarily being interested to the unitary pricing of each instrument composing this portfolio. As the training sets must be limited for the calibration purpose a parallelization step, in terms of risk factors, is used if necessary for the training stage of the GPR algorithm. Then, the we explore the second topic closely linked to the Value-at-Risk, namely the Back-testing Value-at-Risk Models. based on the back-testing background, we deploy a machine learning algorithm to classify market risk VaR exceptions into an expected downside market move captured by the VaR measure and into loss arises from out-of-model risk factors that negatively impact the portfolio value (Model Issues).

The rest of this paper is organized as follows: Section 2 details the main principles of the market risk management. We focus on the value-at-Risk and Expected Shortfall and their numerical computing methods: historical and Monte Carlo Simulations. In parallel, a parametric approach initially introduced by Rockafellar and Uryasev ([33]) is also as an alternative method to simultaneously compute the VaR and ES. Section 3 introduces the Gaussian process and describes the main principles of the machine learning techniques. The section 4 discusses the numerical results and robustness of our new approach with numerical tests carried out on some stylized portfolios. Section 5 draws the conclusions and open further discussions.

2 Market Risk: Value-at-Risk and Expected Shortfall

Market risk can be defined as the risk of losses in on and off-balance-sheet positions arising from unexpected changes in market prices (e.g. such as security prices) or market rates (e.g. such as interest or exchange rates). Market risks, in turn, can be classified into interest-rate risks, equity risks, exchange rate risks, commodity price risks, and so on, depending on whether the risk factor is an interest rate, a stock price, or whatever. Market risks can also be distinguished from other forms of financial risk, most especially credit risk (or the risk of loss arising from the failure of a counterparty to make a promised payment) and operational risk the risk of loss arising from the failures of internal systems or the people who operate in them). The market risk framework of the Basel accord ([30]) consists of an internal models' approach (IMA) and a standardised approach (SA). To capture, tail risk better, the revised framework also saw a shift in the measure of risk under stress from the Value-at-Risk (VaR) to Expected Shortfall (ES). The Value at risk is the maximum likely loss over some target period the most it expected to lose over that period, at a specified probability level. In other terms, Value-at-risk is defined as the loss level that will not be

exceeded with a certain confidence level during a certain period. For example, if a bank's 10-day 99% VAR is 30 million euros, this means that there is only a 1% chance that losses will exceed 30 million in 10 days.

In the sequel, the Value V_t of the portfolio (or the position) will be expressed in terms of a function of time and a set of m underlying risk factors Z_t, namely $V_t = f(t, Z_t)$ where Z_t denotes the risk factor $(Z_{t,1}, Z_{t,2}, ..., Z_{t,m})^T$. Then, the time series of risk factors changes is defined by $X_t := Z_{t+h} - Z_t$. Typically, historical risk factor time series are available, and it is of interest to relate the changes in these underlying risk factors to the changes in portfolio value. Within this framework, since the risk factor values Z_t are known at time t, the loss $L_t(h)$ is determined by the risk factor changes X_t:

$$\begin{aligned} L_t(h) &:= -(V_{t+h} - V_t) \\ &= -(f(t+h, Z_{t+h}) - f(t, Z_t)) \\ &= -(f(t+h, Z_t + X_t) - f(t, Z_t)) \end{aligned} \tag{1}$$

Up to now, the Value at Risk measure is used to estimate the market risk to which a bank is exposed, and also for determining, the bank's minimum capital required to cover this risk. For a given time horizon h (e.g., one day or two weeks), the quantitative risk measure Value at Risk at the $100(1-\beta)$ percent confidence level is defined as follows:

$$\begin{aligned} VaR_\beta &= -inf\{Prob(v|\Delta V_t \leq v) > \beta\} \\ &= -inf\{Prob(v|L \geq -v) > \beta\} \end{aligned} \tag{2}$$

Where ΔV_t represents the profit or loss (random variation between futures and present value, namely $\Delta V_t = V_{t+h} - V_t = -L_t(h)$ of a portfolio for a given horizon h. The parameter β denotes the confidence level (e.g. 95%, 99%). Thus, the Value at Risk is the $(1-\beta)$ quantile of the return distribution which in most cases must be specified.

Even though the Value-at-Risk is widely used as a risk measure within the financial industry, it possesses several limitations. Its major weakness is its ignorance of the extreme left tail of the distribution of V_t beyond level β. Potentially, this can lead to careless risk measurement and management. A more accurate risk measure would be the Expected Shortfall (ES), an average value of the loss when it exceeds the quantile β. For continuous distributions, this can be expressed as follows:

$$ES_\beta = E(-\Delta V_t | \Delta V_t \leq -VaR_\beta) = \frac{1}{1-\beta} \int_0^{1-\beta} VaR_\alpha \, d\alpha \tag{3}$$

Where VaR_α is the Value at Risk for confidence level α, which changes from 0 to β.

According to John Hull [20], where the VaR asks the question "how bad can things get?", the Expected Shortfall asks, "if things do get bad, what is our expected loss?". Indeed, Expected Shortfall (ES), in contrast to VaR, gives information about the losses, which occur when the confidence level is exceeded and thus can potentially evaluate the extreme losses in the distribution tail.

As mentioned above, the Basel Committee of Banking Supervision (BCBS published the FRTB text in 2016 ([30]) with many important rules related to Basel IV that will have a huge impact on trading book definition, risk measurement, assessment and reporting at banks worldwide. The adaptation and the implementation of this text is planned with a deadline of December 31, 2019.

Among the main methodological elements of FRTB finale rule is the replacement of the Expected shortfall (ES) as risk measure by the currently Value at Risk, as well as applying varying liquidity horizons based on the various risk levels of the assets involved. In addition, the Basel committee proposes that the confidence level is to be changed from 99% to 97.5%. Two daily VaR's for both confidence level 99% and 97.5% are to be calculated and a daily Expected Shortfall for confidence level 97.5% (for more details, see [30]). Only, the numerical implementation of these new risk indicators for large portfolios of complex derivatives can be more expensive in terms of computing times. Indeed, whatever the numerical method chosen (historical simulations or Monte Carlo simulations), the repeated full revaluation of a whole derivatives portfolio under different risk scenarios and for different time horizons can be turn out more time consuming. For this, financial institutions must be aware to the computing time optimization aspects and they have to find alternatives to improve the trade-off between speed and accuracy. This paper shows that this is possible by using the machine learning techniques. But before, the main numerical methods to compute the Value-at-Risk (VaR) and the Expected Shortfall (ES) are reminded.

2.1 Risk model design using simulations

The main numerical methods to implement the risk models and to compute the Value at Risk and the Expected shortfall model are the two simulation methods called the historical simulations and Monte Carlo simulations.

Historical simulations approach is a non-parametric Value-at-Risk method which assumes that the set of possible future scenarios is fully represented by the past returns on a specific historical window ([13], [32], [5]). This method is founded on the main hypothesis which assumes that the future is adequately represented by recent history. The historical simulations approach has some interesting advantages due to its simplicity. The HS approach does not require estimation of statistical parameters like volatilities and correlations and most importantly does not make any assumptions about the probability distributions therefore fat tails of the return distributions. More precisely, it does not impose restrictive assumptions and distribution constraints on the portfolio returns. The method can also be applied practically to any type of financial portfolio and uses full valuations ([4], [25]). In addition, its numerical implementation is easy, and it gives a clear understanding of the potential losses to managers. Nevertheless, the historical simulations method has some disadvantage to be strongly dependent to the data, their availability, and their quality. The steps followed to compute VaR and Expected shortfall using the historical simulations are:

1. Calculate the returns or price changes of all assets in the trading portfolio between each time interval $[t, t+h]$
2. Apply the estimated shocks $X_t := Z_{t+h} - Z_t$ to the current mark-to-market value V_t of the asset and revalue the trading portfolio
3. Sort the series of the simulated profit and loss (P&L) of the trading portfolio from the lowest to the highest value
4. Compute the simulated values of VaR and Expected Shortfall corresponding to the desired confidence level

Monte Carlo simulations approach is the most well-known methodology when there is a requirement to develop a sophisticated and powerful Value at Risk and Expected Shortfall computing framework. This technique simulates future risk factor values for all linear and non-linear instruments with respect to their dependency inputs. Then, the repeated revaluation of the portfolio for

each simulation is carried out taking account of potential changes in the simulated market risk factors. The main difference between those two methods is that historical simulations use the historical market price changes as predictors of the future whereas Monte Carlo simulations randomly generate numbers to construct thousands hypothetical market changes.

The one main issue with the Monte Carlo simulation approach is the computational time since it needs a lot of resources, particularly with huge portfolio associated to a large of market risk factors. As a result, the numerical implementation turns out to be expensive ([25]). According to P. Jorion ([25]), another limitation is model risk, which arises because of inappropriate assumptions about the estimating models and stochastic processes. If these parameters are not specified properly, VaR estimates will be misleading.

The steps followed to compute VaR with Monte Carlo Simulations are very similar to that of followed by Historical Simulations except the first step of the algorithm, instead of using the historical data for the price (or returns) of the asset and assuming that this return (or price) can re-occur in the next time interval, we generate a random variable that will be used to estimate the return (or price) of the asset at the end of the analysis horizon.

2.2 Parametric VaR and Cornish-Fisher adjustment

In hereafter a closed formula, called delta-gamma method, is presented for calculating Vale at Risk for a derivatives portfolio of N options with portfolio weights $m_i, \forall i = \{1, 2, \ldots, N\}$. This formula is founded on the Cornish-Fisher approximation ([6]). Indeed, if we notice by V_t the value of a portfolio composed of options with the same underling S_t, then the profit or loss $\Delta V_t = V_{t+h} - V_t$ for a given horizon h can be expressed as follows:

$$\Delta V_t \approx \Delta S_t R_t + \frac{1}{2}\Gamma S_t^2 R_t^2 \tag{4}$$

Where $R_t = \frac{S_{t+h}-S_t}{S_t}$ can be interpreted as a rate of return of the underlyin asset S_t. The delta (Δ) and the gamma (Γ) are given by:

$$\begin{aligned} \Delta &= \sum_{j=1}^{N} m_j \Delta_j \\ \Gamma &= \sum_{j=1}^{N} m_j \Gamma j \end{aligned} \tag{5}$$

The variable δV_t is non-normal. According to Pichler and Selitsch ([31]), in the case of only one risk factor the distribution is a noncentral χ^2- distribution. So, the first three moments of this distribution can be derived from the equation (**) assuming that R_t is normal with the mean of zero and a volatility of σ:

$$\mu_V = E[\Delta V_t] = \frac{1}{2}\Gamma S_t^2 \sigma^2 h \tag{6}$$

$$E[(\Delta V_t)^2] = \Delta^2 S_t^2 \sigma^2 h + \frac{3}{4}\Gamma^2 S_t^4 \sigma^4 h^2 \tag{7}$$

$$E[(\Delta V_t)^3] = \frac{9}{2}\Delta^2\Gamma S_t^4\sigma^4h^2 + \frac{15}{8}\Gamma^3 S_t^6\sigma^6h^3 \tag{8}$$

According to Hull ([21]), the mean μ_V and the standard deviation σ_V of the derivatives portfolio are obtained in the following way:

$$\begin{aligned} \mu_V &= E[\Delta V_t] \\ &= \frac{1}{2}\Gamma S_t^2\sigma^2 h \end{aligned} \tag{9}$$

$$\begin{aligned} \sigma_V^2 &= E[(\Delta V_t)^2] - E[\Delta V_t]^2 \\ &= \Delta^2 S_t^2\sigma^2 h + \frac{1}{2}\Gamma^2 S_t^4\sigma^4h^2 \end{aligned} \tag{10}$$

Then, the skewness coefficient of the probability distribution of ΔV_t, ξ_V is defined as:

$$\begin{aligned} \xi_V &= \frac{E[(\Delta V_t - \mu_V)^3]}{\sigma_V^3} \\ &= \frac{3\Delta^2\Gamma S_t^4\sigma^4h^2 + \Gamma^3 S_t^6\sigma^6h^3}{\sigma_V^3} \end{aligned} \tag{11}$$

Now the parametric method, presented hereafter to compute the Value-at-Risk of the portfolio, is built on the Cornish-Fisher expansion around the quantile of a normal distribution and the estimation of the required quantile of the distribution of ΔV_t. Indeed, using the first three moments of ΔV_t, on the Cornish-Fisher expansion estimates the β-percentile of the distribution of ΔV_t, θ_β as:

$$\theta_\beta = \alpha_\beta + \frac{1}{6}(\alpha_\beta^2 - 1)\xi_V \tag{12}$$

Where α_β is the β-percentile of the standard normal distribution. Therefore, the Value at Risk of the derivatives portfolio is given as follows:

$$\begin{aligned} VaR_\beta &= (\alpha_\beta + \frac{1}{6}(\alpha_\beta^2 - 1)\xi_V)\sigma_\beta \\ with : \alpha_\beta &:= \Phi^{-1}(1-\beta) \end{aligned} \tag{13}$$

Now, the univariate case can be easily extended in the case of m risk factors. In this case, the portfolio can be expressed as follows:

$$V_t = \sum_{i=1}^{m} V_i(X_i) \tag{14}$$

And the change in the portfolio value is given by the following formula:

$$\begin{aligned} \Delta V_t &\approx \sum_{i=1}^{m} \Delta_i X_i R_i + \frac{1}{2} \sum_{i=1}^{m} \Gamma_i X_i^2 R_i^2 \\ where : R_i &= \frac{X_i(t+h) - X_i(t)}{X_i(t)} \end{aligned} \tag{15}$$

As in the univariate case, the moments of the distributions of ΔV_t must be determined first to compute analytically the Value-at-Risk. Once again, according to Pichler and Selitsch ([31]), the moments of the P&L distribution of the derivatives portfolio can be derived as follows:

$$\begin{aligned} \mu_V &= E[\Delta V_t] \\ &= \frac{1}{2} tr[\Gamma\Sigma] \end{aligned} \tag{16}$$

$$\begin{aligned} \sigma_V &= E[(\Delta V_t)^2] - E[\Delta V_t]^2 \\ &= \delta^T \Sigma \delta + \frac{1}{2} tr[\Gamma\Sigma]^2 \end{aligned} \tag{17}$$

Then, the skewness coefficient of the P&L distribution of ΔV_t, ξ_V, given by:

$$\begin{aligned} \xi_V &= \frac{E[(\Delta V_t - \mu_V)^3]}{\sigma_V^3} \\ &= \frac{3\delta^T \Sigma[\Gamma\Sigma]\delta + tr[\Gamma\Sigma]^3}{\sigma_V^3} \end{aligned} \tag{18}$$

Where:

$$\Gamma = \begin{pmatrix} \Gamma_{1,1} & ... & 0 \\ ... & ... & ... \\ 0 & ... & \Gamma_{m,m} \end{pmatrix} \tag{19}$$

$$\Sigma = \begin{pmatrix} \sigma^2(R_1)h & ... & Cov(R_1, R_m)h \\ ... & ... & ... \\ Cov(R_m, R_1)h & ... & \sigma^2(R_m)h \end{pmatrix} \tag{20}$$

$$\delta = \begin{pmatrix} \Delta_1 X_1 \\ \Delta_2 X_2 \\ ... \\ \Delta_m X_m \end{pmatrix} \tag{21}$$

In the same way as of the univariate case, the Value at Risk of the derivatives portfolio is given as follows:

$$\begin{aligned} VaR_\beta &= (\alpha_\beta + \frac{1}{6}(\alpha_\beta^2 - 1) * \xi_V)\sigma_V \\ with : \alpha_\beta &:= \Phi^{-1}(1-\beta) \end{aligned} \tag{22}$$

2.3 Parametric model characterizing VaR and ES

In this subsection an alternative approach is proposed for simultaneously calculating the VaR and ES of large portfolios of trading derivatives. Its numerical implementation is carried out using a semi-analytical formula. For this purpose, a result of Rockafellar and Uryasev ([33]) is used. The key point of their approach is a characterization of the VaR and the ES as a solution of a convex optimization problem of a target function chosen in an ingenious way. The target function is built out of the loss $L_t(h)$ of the portfolio of trading derivatives and its density distribution $g(.)$. Formally, the target function is given as:

$$F_\beta(x) := x + (1-\beta)^{-1} \int_x^{+\infty} (L_t(h) - x) g(L_t(h)) dL_t(h) \tag{23}$$

Where the function $g(.)$ is the distribution function of the loss $L_t(h)$. This is a convex function and then has a global minimum. Rockafellar and Uryasev showed that the minimum of this function is obtained at $\alpha = VaR_\beta(L)$ and that the value of the minimum is equal to the $F_\beta(VaR_\beta(L)) = ES_\beta(L)$. Nevertheless, for the practical implementation, the density distribution $g(.)$ need to be estimate and a Gaussian kernel smother is applied to a limited number of simulated or observed data $\{L_t^{(i)}(h)\}_{1 \leq i \leq n}$. For example, the Monte Carlo simulation can be used to simulate the risk factors underlying the portfolio and calculate these data.

Formally, the shape of the density distribution $g(.)$can be approximated by its kernel density estimator as follows:

$$\hat{g}_l(x) = \frac{1}{nl} \sum_n^{i=1} \frac{1}{l} K\left(\frac{x - L_t^{(i)}}{l}\right) \tag{24}$$

Where $K()$ is called kernel function that is generally a smooth, symmetric function such as a Gaussian and $l > 0$ is called the smoothing bandwidth that controls the amount of smoothing. In principle, the kernel density estimator (KDE) smoothest each data point $L_t^{(i)}(h)$ into a small density bumps and then sum all these small bumps together to obtain the final density estimate. Several types of kernel functions can be used. However, a very common choice is the Gaussian kernel function, $K()$, as it is intuitive and provides useful properties such as smoothness. The parametric form of this function is given as follows:

$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}} \tag{25}$$

The smoothing bandwidth l plays a key role in the quality of the Kernel density estimator. It has a strong influence on the resulting estimate. In general, the L_2-risk function, called the mean integrated squared error (MISE(.)) is used as the optimality criterion to select parameter l and in the case of Gaussian basis functions the optimal choice is given by minimizing the MISE, namely:

$$l = \left(\frac{4\hat{\sigma}^2}{3n}\right)^{\frac{1}{5}} \approx 1.06 \hat{\sigma}^2 n^{-\frac{1}{5}} \tag{26}$$

Where $\hat{\sigma}$ is the standard deviation of samples observed data $\{L_t^{(i)}(h)\}_{1 \leq i \leq n}$ for our framework.

3 Gaussian Process Regression in Machine Learning

Hereafter the main notions and techniques of Gaussian Process Regression used in machine learning are introduced. In regression and classification problems, they are used for interpolation, and extrapolation, and pattern discovery for which the choice of the kernel function is central. In supervised learning, we observe some inputs x_i and some outputs y_i. We assume that $y_i = f(x_i)$, for some unknown function $f(.)$. The optimal approach is to infer a distribution over functions given the data, $p(f|X, y)$, and then to use this to make predictions given new inputs, i.e., to compute:

$$p(y^*|x^*, X, y) = \int p(y^*|f, x^*)p(f|X, y)df \tag{27}$$

Gaussian Process Regression (GPR) is one of machine learning algorithms, which explore the idea to learn from data using statistical modelling and then to carry out a prediction. In the case where the data are labeled, namely each observation has a value, or a class associated to, it can be used to learn the relationship between the data (the input of the model), and the labels (the output of the model). Consider a training set of observations $(X, y) = \{(x_i, y_i)|i = 1, 2, ..., n\}$ where each x_i is an input vector of dimension x, and y_i is the corresponding output. The goal is to find the relation between inputs and outputs. In Gaussian process regression, this can be expressed as:

$$y_i = f(x_i) + \epsilon_i \tag{28}$$

Where $x \mapsto f(x)$ is a Gaussian process and $\epsilon_i \sim N(0, \sigma_n^2)$ are (i.i.d) random variables representing the noise in the data. A Gaussian process (GP) extends multivariate Gaussian distributions to infinite dimensionality. More precisely, a Gaussian process is a collection $\{f(x), x \in X\}$ such that for any $n \in N$ and $x_1, x_2, ..., x_n$, the random vector $(f(x_1), f(x_2), ..., f(x_3))$ has a joint multivariate Gaussian distribution. In other terms, a Gaussian process be considered as a Gaussian distribution over functions (considering functions as infinitely long vectors containing the value of the function at every input).

Therefore, the Gaussian process is characterized by its mean function and its covariance function:

$$\mu(x) = E[f(x)] \tag{29}$$

$$\begin{aligned} k(x, x') &= cov\left(f(x), f(x')\right) \\ &= E[(f(x) - \mu(x))(f(x') - \mu(x')] \end{aligned} \tag{30}$$

The covariance function $k(.,.)$ is called kernel function and it is a central concept in the analysis of Gaussian processes. The popular choice, in machine learning, of the kernel function is the squared exponential:

$$k(x, x') = \sigma_f^2 exp\left(\frac{-(x - x')^2}{2l^2}\right) \tag{31}$$

A Gaussian process is a random stochastic process where correlation is introduced between neighboring samples. The covariance matrix K has larger values, for points that are closer to each

other, and smaller values for points further apart. This because the points are correlated by the difference in their means and their variances. If they are highly correlated, then their means are almost same, and their covariance is high. In the sequel, one assumes a zero-mean function, $\mu = 0$. For a sample $(X, f) = \{(x_i, f_i)|i = 1, 2, ..., n\}$ generated from $f(x)$, then $f \sim N(0, K(X, X))$, where $K(X, X)$ is the covariance matrix, defined as:

$$K(X, X') = \begin{pmatrix} k(x_1, x_1) & ... & k(x_1, x_n) \\ ... & ... & ... \\ k(x_1, x_n) & ... & k(x_n, x_n) \end{pmatrix} \tag{32}$$

The kernel function $K(X, X)$ must have the following properties:

1. A symmetric matrix: $k(x_i, x_j) = k(x_j, x_i), \forall i, j \in \{1, 2, ..., n\}$
2. A positive definite matrix $K(X, X)$: the kernel function induced by $k(., .)$ for any set of inputs is a positive definite matrix.

A finite dimensional subset of the Gaussian process distribution results in a marginal distribution that is a Gaussian distribution $y \sim N(0, K(X, X'))$ with mean vector $\mu(x) = 0$ and covariance $K(X, X)$.

Now, we recall hereafter a well-known result of a conditional Gaussian distribution which will used directly for the main result behind the Gaussian process regression. Let us consider a Gaussian random vector defined as follows:

$$\begin{pmatrix} X \\ X^* \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_x \\ \mu_{x^*} \end{pmatrix}, \begin{pmatrix} \Omega_{x,x} & \Omega_{x^*,x} \\ \Omega_{x,x^*} & \Omega_{x^*,x^*} \end{pmatrix}\right) \tag{33}$$

Then, the marginal distributions of X and X^* are given by $X \sim N(\mu_x, \Omega_{x,x})$ and $X^* \sim N(\mu_x^*, \Omega_{x^*,x^*})$ and we have $covariance(X, X^*) = \Omega_{x,x^*}$ The conditional distribution of X^* given $X = x$ is a multivariate normal distribution:

$$X^*|X = x \sim N(\mu_{x^*|x}, \Omega_{x^*|x}) \tag{34}$$

where:

$$\begin{aligned} \mu_{x^*|x} &= E[X^*|X = x] \\ &= \mu_{x^*} + \Omega_{x^*,x}\Omega_{x,x}^{-1}(x - \mu_x) \end{aligned} \tag{35}$$

And:

$$\begin{aligned} \Omega_{x^*|x} &= \sigma^2[X^*|X = x] \\ &= \Omega_{x^*,x^*} + \Omega_{x^*,x}\Omega_{x,x}^{-1}\Omega_{x,x^*} \end{aligned} \tag{36}$$

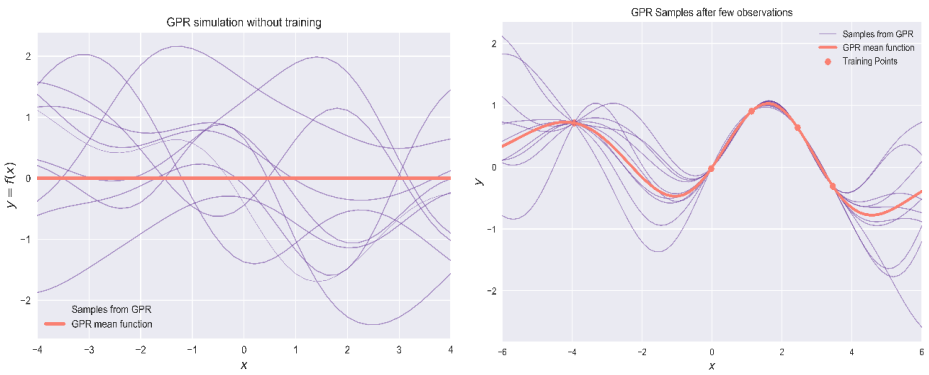
To illustrate the inference principle of the Gaussian principle regression, in the figure below, we will sample teen different function realizations from a Gaussian process with exponential quadratic prior without any observed data. More precisely the function realizations y of a function f are

drawn at random from a prior distribution at a finite, but arbitrary, set of points $x : y = f(x)$. We do this by drawing correlated samples from a 41-dimensional Gaussian $N(0, K(X, X))$ with $X = [X_1, X_2, .., X_n]$.

This prior is taken to represent our prior belief over the kinds of functions we expect to observe, before saying any data. Note that:

$$\frac{1}{n}\sum_{i=1}^{n} f_i(x) \rightarrow \mu(x), \ as \ n \rightarrow \infty \tag{37}$$

Now, given a set of five observed inputs and corresponding output values $(x_1, f(x_1))$, $(x_2, f(x_2))$, ..., $(x_5, f(x_5))$, and a Gaussian process prior on, $f \sim N(0, K(X, X))$, we would like to compute the posterior over the value $f(x^*)$ at any input x^*. The figure 1 shows teen realizations from the posterior distribution based on five observations from a sine function. We notice that the sample functions from the posterior pass close the observed values, by vary a lot in regions where there are no observations. This shows that the uncertainty is reduced near the observed values.



(a) Without training (b) After training

Figure 1: Simulations from a GP before and after training

3.1 Estimation Method using Gaussian Process Regression

One of the primary goals computing the posterior is that it can be used to make predictions for unseen test cases. Therefore, consider a matrix X^* consisting of n^* test inputs, each of dimension d, and denote the corresponding unknown vector of function values by f^*. Then, the joint distribution of training output y and f^* is Gaussian, namely:

$$\begin{pmatrix} y \\ f^* \end{pmatrix} \sim N\left(0, \begin{pmatrix} K(X,X) + \sigma^2 I & K(X,X^*) \\ K(X^*,X) & K(X^*,X^*) \end{pmatrix}\right) \tag{38}$$

Where:

$$K(X^*, X) = [k(x^*, x_1), k(x^*, x_2), ..., k(x^*, x_n)] \tag{39}$$

And:

$$K(X^*, X^*) = k(x^*, x^*) \tag{40}$$

Note that $y = f(x)$ when the there is no noise ($\sigma_n^2 = 0$). This happens when we have clean observations of the underlying process f. We are of course interested in the conditional probability $p(y^*|X, f, y)$: given the data, how likely is a certain prediction for y^*. According to equations 34 to 36, the probability follows a Gaussian distribution:

$$y^*|y \sim N(K^*K^{-1}y, K^{**} - KK^{-1}K_*^T) \tag{41}$$

Our best estimate for y^* is the posterior mean $E[f(x^*)|f(x)]$ of this distribution is given as follows:

$$\begin{aligned} \bar{y^*} &= E[f(x^*)|f(x)] \\ &= K^*K^{-1}y \\ &= K^*K^{-1}f(x) \end{aligned} \tag{42}$$

And the uncertainty in the Gaussian process regression is captured in its variance:

$$var(y^*) = K^{**} - KK^{-1}K_*^T \tag{43}$$

The posterior mean $E[f(x^*)|f(x)]$ can be interpreted in two ways. If the last the two terms of $K^{-1}f(x)$ are grouped, the posterior mean be expressed a linear combination of the kernel functions values, namely:

$$\begin{aligned} E[f(x^*)|f(x)] &= \sum_{i=1}^{n} \omega_i k(x^*, x_i) \\ \omega &= K^{-1}f(x) \end{aligned} \tag{44}$$

This means that the posterior mean can be computed without explicitly inverting the matrix K, by solving the equation $K\omega = f(x)$ instead. Similarly, by grouping the first two terms $K(X^*, X)K^{-1}$, the the posterior mean can be written as a linear combination of the observed function values:

$$\begin{aligned} E[f(x^*)|f(x)] &= \sum_{i=1}^{n} \phi_i f(x_i) \\ \phi_i &= K(X^*, X)K^{-1} \end{aligned} \tag{45}$$

Now, if the prior mean function is no-zero, we can still use the previous derivation by noting that if $f \sim N(\mu, K(X, X))$, then function $h = f - \mu$ is a zero-mean Gaussian process if $h \sim N(0, K(X, X))$. Hence, if the observations are deducted from the values of f, we can subtract the prior mean function values to get observations of h, do the inference on h, and finally once we obtain the posterior on $h(x^*)$, we can simply add back the prior mean $\mu(x^*)$ to the posterior mean, to obtain the posterior on f.

3.2 Training Gaussian process algorithms

The practical implementation of the Gaussian process regression is dependent how well we choose the covariance function. For this, the parameters l, σ_n, σ_f must appropriately selected. Set $\theta = \{l, \sigma_n, \sigma_f\}$ the parameters of the model. The usual way of selecting parameters is to maximize the log marginal likelihood function $l(\theta) = ln(p(y|\theta))$ given by:

$$\begin{aligned} l(\theta) &= ln(p(y|\theta) \\ &= -\frac{1}{2}y^T(K + \sigma_n^2 I)^{-1} - \frac{1}{2}log(|K + \sigma_n^2 + I|) - -\frac{n}{2}log(2\pi) \end{aligned} \tag{46}$$

The first term in the equation 46 can be interpreted as a data-fit term, the second term is a complexity penalty and the last term is a normalizing constant. The derivatives of the log marginal likelihood with respect to the hyperparameters are given by:

$$\begin{aligned} \frac{\partial l(\theta)}{\partial \theta_j} &= \frac{1}{2}y^T K^{-1}\frac{\partial K}{\partial \theta_j}K^{-1}y - \frac{1}{2}tr\left(K^{-1}\frac{\partial K}{\partial \theta_j}\right) \\ &= \frac{1}{2}tr\left(\omega\omega^T - K^{-1}\frac{\partial K}{\partial \theta_j}\right) \end{aligned} \tag{47}$$

$where\ \omega = K^{-1}y$

One drawback of Gaussian processes is that it scales very badly with the number of observations n. Indeed, the complexity of computing the marginal likelihood in the equation 46 is dominated by the need to invert the K matrix. The log determinant of K is easily computed as by-product of the inverse. Standard methods for matrix inversion of positive definite symmetric matrices require time $0(n^3)$ for inversion of an n by n matrix. Once the matrix K^{-1} is determined, the computation of the derivatives in the equation 38 requires only time $0(n^2)$ per hyperparameter. Thus, the computational overhead of computing derivatives is small, so the optimization problem in the equation 46 can be easily solved using gradient-descent, conjugate gradient or quasi-Newton algorithms.

4 Applications of the Gaussian process regression

In this section, we present two main financial applications of the Gaussian process algorithm. More precisely, we roll out this algorithm to tackle the revaluation performance problem and the risk measurement of the whole derivatives securities trading portfolio.

4.1 Derivatives trading portfolio valuation

The value V_t of a derivative trading portfolio, for a given date t, can be written as a linear combination of a finite number m of the sub portfolios of derivatives securities rearranged according the risk factors nature. Each derivative security is written on an underlying asset characterized by their own risk factors. More formally, the portfolio value can be expressed as follows:

$$V_t = \sum_{i=1}^{m} V_i(x_i) \tag{48}$$

In the sequel, we consider for illustration purposes that our portfolio is composed by the equity derivatives. In this case, each risk factor $X_i, i = 1, 2, ..., m$ can be characterized by the spot price S_i, the implied volatility σ_i and the interest rate r, namely $X_i = (S_i, \sigma_i, r)$. As explained in the introduction of this paper, the main goal is the compute the Value-at-Risk and the Expected Shortfall. Therefore, we postulate that the pricing models are available and, if needed, used only for the valuation purposes. Then, the GP algorithm learns the valuation task of all pricing models by training it directly on a data set generated by sophisticated pricing models. Then, the trained GP algorithm takes over to carry out the repeated revaluation of the derivatives security's portfolio in a fast and efficient way. To do this, for each risk factor $X_i, i = 1, 2, ..., m$ we construct a training set (X, y). The input matrix X consists of d combinations of market parameters S_i, σ_i, r. Random parameters combinations are sampled uniformly over $[S_{i,min}, S_{i,max}] \times [\sigma_{i,min}, \sigma_{i,max}] \times [r_{min}, r_{max}]$.

$$\begin{aligned} E[V_i(x_i^*)|\bar{V_l}] &= \sum_{j=1}^{d} \phi_{i,j} V_{i,j}(x_{i,j}) \\ with\ \phi_{i,j} &= \sum_{l=1}^{m} k(x_i^*, x_{i,l}) a_{il}^{(i)} \end{aligned} \tag{49}$$

$$\begin{aligned} \phi_i &= K_i^* K_i^{-1} \\ with\ K_i^{-1} &= (a_{jk}^{(i)})_{1 \leq j,k \leq d} \end{aligned} \tag{50}$$

$$\begin{aligned} \bar{V_l} &:= [V_i(x_{i,1}), V_i(x_{i,2}), ..., V_i(x_{i,d})]^T \\ K_i^* &= [k(x_i^*, x_{i,1}), k(x_i^*, x_{i,2}), ..., k(x_i^*, x_{i,d})]^T \end{aligned} \tag{51}$$

We consider hereafter a numerical example where the derivative trading portfolio of 100 vanilla options and barrier/american options written on four underlying's. Table 1 displays the composition portfolio according to the underlying and the option type.

Portfolio Composition			
Underlying	Vanilla Option	Barrier Option	Americal Option
S1	10	10	5
S2	10	10	5
S3	10	10	5
S4	10	10	5

Table 1: Number of options in the portfolio

More details of the portfolio composition (level and of barriers, options types, etc.) are displayed in Annex 1 of this paper. For illustration purposes and with no loss of generality, we start with a numerical example where the risk factor is only the spot prices $S_i, i = 1, 2, 3, 4$. Within this set-up, four sets of tests are considered by varying the size of the training set ($d = 5, 10, 15, 20$). Tables 3,4 display the training set (X, y) of the GPR algorithm for $d = 5, 10$ but for $d = 15, 20$ the training sets are stored in Annex 2 of this paper.

Market Data / Underlying			
Underlying	Spot Price	Interest Rate	Volatility
S1	100	2%	40%
S2	105	2%	20%
S3	90	2%	50%
S4	110	2%	30%

Table 2: Market Data / Underlying

Training Set (5 training observations)					
Spot	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 4	Global Portfolio
1	260	1 289	1 010	737	3 296
35	180	860	777	560	2 377
69	289	513	726	492	2 021
103	568	382	810	592	2 351
137	957	439	979	811	3 186

Table 3: Training set of the Gaussian process regression (size=5)

For each training test, the value of each of the four sub portfolios and the global portfolio is calculated by the usual pricing models used according the options composing the portfolio. It worth to outline that the vanilla and barrier options are valued with the Black & Scholes model and for the American options valuation we use the same model, but a trinomial tree method is needed for its practical implementation. Nevertheless, whatever used models to evaluate the portfolio the GPR algorithm will apply in the same way; even better, it will be more efficient.

Training Set (10 training observations)					
Spot	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 4	Global Portfolio
1	737	260	1 289	1 010	3 296
16	653	207	1 098	886	2 843
31	578	180	909	795	2 462
46	519	194	732	742	2 187
61	492	246	580	724	2 042
76	501	335	466	735	2 036
91	541	454	401	769	2 165
106	606	599	381	822	2 408
121	694	764	395	891	2 744
136	803	945	435	973	3 156

Table 4: Training set of the Gaussian process regression (size=10)

The training set is then fed to the algorithm that fits the Gaussian process algorithm. This allows to estimate the hyperparameters and inverting the kernel covariance matrix. Once the learning stage is carried out, the model is now ready for prediction. The valuation performance of the

GPR algorithm is quantified, for both the training set and a test, set using the two following main indicators:

$$MAE = \max_{1 \leq i \leq n} \{|V_i^* - V_i^{GPR}|\} \tag{52}$$

$$AAE = \frac{1}{n} \sum_{i=1}^{n} |V_i^* - V_i^{GPR}| \tag{53}$$

The test set is built by sampling uniformly new data $\{S_i^*, i = 1, 2, 3, 4\}$ over interval $[S_{i,min}, S_{i,max}]$. Results of the numerical test devoted to the valuation task of the GPR model, both in terms of accuracy and computation time is given in Figure 3.

Numerical results of the empirical tests dedicated to the repeated full revaluation of the trading portfolio are reported in Tables 3, 4 and Graphs 6, 7. These tests focus on the accuracy valuation and the computing time.

GPR valuation of the derivative portfolio		
Spot Price	GPR Price	Black Price
37	2555.62	2555.64
43	2453.47	2453.47
49	2372.80	2372.79
55	2313.27	2313.29
61	2274.56	2274.59
67	2256.69	2256.73
73	2260.23	2260.20
79	2284.84	2284.85
85	2329.53	2329.53
91	2392.87	2392.87
97	2473.47	2473.51
103	2570.10	2570.07
109	2681.05	2681.15
115	2805.50	2805.39

Table 5: The GPR valuation of the derivative portfolio with the training size=10

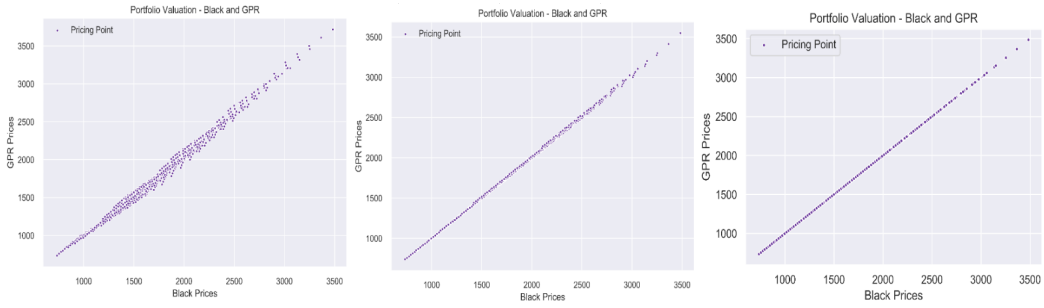


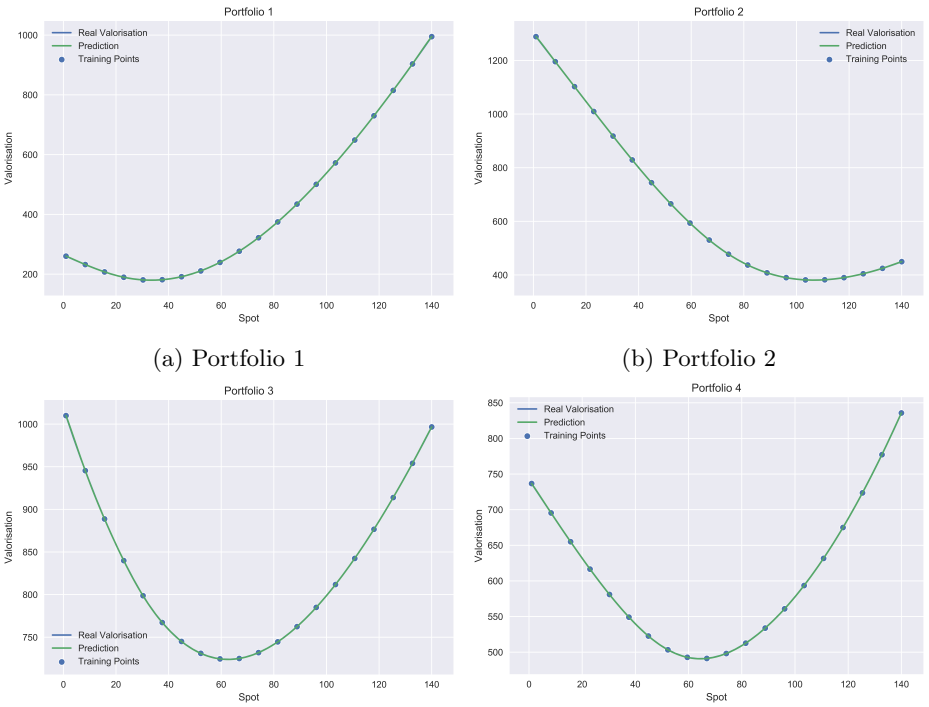
Figure 2: Out—of-sample prediction of the portfolio values (10000 points) with a GPR algorithm trained on 5, 10, and 20 points

		Portfolio 1			Portfolio 2			Portfolio 3			Portfolio 4			Global Portfolio		
	Number of Training Points	5	10	20	5	10	20	5	10	20	5	10	20	5	10	20
Precision	R2	0,98	1,00	1,00	0,99	1,00	1,00	0,95	1,00	1,00	0,96	1,00	1,00	0,97	1,00	1,00
	AAE	17,91	2,53	0,05	17,52	2,11	0,06	12,37	1,02	0,04	11,49	1,14	0,03	59,47	8,59	0,41
	MAE	73,51	16,52	0,98	65,48	11,82	1,03	38,15	5,38	0,57	37,46	5,78	0,46	275,96	66,06	3,92
Training computation time	Training Data (s)	1,42	2,76	5,04	1,32	2,57	5,16	1,42	2,780	5,52	1,33	2,546	5,07	5,50	10,65	20,80
	Model Training (s)	0,05	0,07	0,13	0,05	0,07	0,18	0,03	0,061	0,39	0,03	0,116	0,21	0,15	0,32	0,92
	Total Training	1,47	2,83	5,17	1,37	2,64	5,35	1,45	2,84	5,91	1,36	2,66	5,29	5,65	10,98	21,72
Model prediction for n test points (speed-up including training)	1	0,19	0,09	0,05	0,19	0,10	0,05	0,21	0,10	0,05	0,19	0,10	0,05	0,19	0,10	0,05
	10	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0
	100	17	9	5	19	10	5	18	10	5	19	11	6	18	10	5
	1 000	193	100	55	193	100	49	197	100	48	196	100	50	195	97	51
	10 000	1 933	1 003	549	1 931	1 000	494	1 965	1 001	481	1 956	1 002	505	1 946	971	506
	20 000	3 866	2 006	1 098	3 863	2 000	989	3 931	2 002	963	3 912	2 004	1 009	3 893	1 941	1 013
	1 000 000	193 320	100 283	54 876	193 129	100 001	49 448	196 542	100 092	48 128	195 598	100 225	50 463	194 649	97 052	50 629

Figure 3: Gaussian process algorithm performance with respect to the classical valuation models

As expected, the larger training set, the powerful the model and hence the more accurate the predictions. In addition, the numerical tests highlight that the Gaussian process regression (GPR) drastically improve the computing time in the case of the repeated revaluation which will be, for example, very useful for the value-at-Risk and ES computations.

The scatter-graph, Graph 2, is plotted using the 4 sub-portfolios values computed with Black & Scholes model and the Gaussian process algorithm. In a perfect estimation, we would observe a straight line: the closer the dots to the bisecting (45 degree) line the better the fit. We notice that the left scatter plot of Graph 2 corresponding of 10000 out-of-sample predictions of the global portfolio with the Gaussian process algorithm trained on 20 points hardly deviates from this straight line.



(a) Portfolio 1

(b) Portfolio 2

(c) Portfolio 3

(d) Portfolio 4

Figure 4: Gaussian process algorithm fit for the sub portfolios value (training set size =10)

The Graph 4 compares the sub portfolios values computed with the Gaussian process algorithm to that of obtained using the classical pricing model (here the Black & Scholes model). The GP algorithm is trained on 10 points what seems to suffice to a good accuracy.

As explained above, adding the volatility as risk axis should not be a problem. To bring to light this, we consider the same numerical example as of the Table 1. Then, the GPR algorithm will be trained on data sets $[S_{i,min}, S_{i,max}]x[\sigma_{i,min}, \sigma_{i,max}]$ with size data set equal to 10x10 for $i = 1, 2, 3, 4$ and the spot price and volatility levels are displayed in Table 6.

Training set of the Gaussian Process Regression				
Spot 1	Spot 2	Spot 3	Spot 4	Volatility
50	53	45	55	10,00 %
61	64	55	67	18,89 %
72	76	65	79	27,78 %
83	88	75	92	36,67 %
94	99	85	104	45,56 %
106	111	95	116	54,44 %
117	123	105	128	63,33 %
128	134	115	141	72,22 %
139	146	125	153	81,11 %
150	158	135	165	90,00 %

Table 6: Training set of the Gaussian Process Regression (size =10)

The GPR algorithm is used to evaluate each of portfolios 1,2,3, and 4 for spot price and volatility levels that obviously have not served for the training step. As expected, numerical results of the Figure 5 show that the GPR algorithm replicates efficiently the results of the pricing model.

Portfolio 1

Vol/spot	63	87	97	117	133
15%	79	177	255	464	683
31%	189	327	412	618	819
45%	293	464	556	766	963
63%	440	648	750	972	1172
77%	538	767	876	1107	1311

Portfolio 2

Vol/spot	56	81	95	126	151
15%	580	380	325	339	433
31%	681	575	554	590	672
45%	800	751	753	814	900
63%	974	986	1012	1110	1211
77%	1089	1134	1175	1296	1409

Portfolio 3

Vol/spot	51	78	99	123	129
15%	518	367	350	398	422
31%	588	523	544	625	653
45%	677	669	718	819	851
63%	811	866	949	1075	1111
77%	901	991	1094	1236	1275

Portfolio 4

Vol/spot	73	99	114	132	161
15%	405	453	499	585	850
31%	505	587	665	791	1054
45%	613	733	828	969	1238
63%	777	941	1054	1212	1494
77%	886	1075	1199	1367	1659

GPR Valuation

Portfolio 1

Vol/spot	63	87	97	117	133
15%	78	176	253	462	680
31%	189	326	411	617	818
45%	293	463	555	766	963
63%	440	647	749	971	1171
77%	538	768	876	1108	1313

Portfolio 2

Vol/spot	56	81	95	126	151
15%	576	378	323	337	430
31%	679	573	553	589	670
45%	799	750	752	813	899
63%	973	985	1012	1109	1209
77%	1089	1134	1176	1297	1410

Portfolio 3

Vol/spot	51	78	99	123	129
15%	516	365	348	397	420
31%	587	522	543	624	652
45%	676	668	718	819	850
63%	810	865	948	1074	1110
77%	902	992	1094	1236	1275

Portfolio 4

Vol/spot	73	99	114	132	161
15%	402	449	494	581	842
31%	504	586	663	789	1049
45%	613	733	828	969	1235
63%	776	940	1054	1211	1490
77%	889	1078	1203	1370	1661

Financial Model Valuation

Figure 5: GPR and Financial model prices as a function of spot price and volatility

We could have done a similar numerical test when adding the interest rates as a risk factor and the time to maturity, but we would therefore have found exactly the similar results confirming the performance of the Gaussian Process Algorithm.

4.2 VaR and ES computation of a trading portfolio

The Value-at-Risk and ES methods, described above, will be tested hereafter analyzing the accuracy and the speed-up of the Gaussian process algorithm. For the sake of simplicity, we consider the same portfolio of the above numerical example composed by a long position on of 100 vanilla options and barrier/ American options written on four underlying's. The detailed composition of the portfolio is displayed in Table 1 and 17. Then, we use the innovation of the GPR algorithm to compute efficiently the Value-at-Risk and the Conditional Value-at-Risk (or Expected Shortfall) of the such portfolio.

Based on 10,000 Monte Carlo simulations of the risk factors , the distribution of the changes in profit and loss (P&L) is estimated using the Gaussian process regression and it is compared to that of the P&L computed with the classical pricing model (this concerns the Black & Scholes model). Then, The Value-at-Risk with and the ES are respectively computed with 90%, 95%, and 99% as confidence levels. In the current setting, the Gaussian process algorithm is assumed to be trained

on the training sets reported in Tables 2, 3 and Tables 17, 18. In parallel, the market data displayed in Table 1 are used for our numerical tests and the covariance matrix, needed to carry out the MC simulations, is built using the following correlation levels:

$$\Sigma = \begin{pmatrix} 100\% & 60\% & 30\% & 10\% \\ 60\% & 100\% & 50\% & 30\% \\ 30\% & 50\% & 100\% & 50\% \\ 10\% & 30\% & 50\% & 100\% \end{pmatrix} \tag{54}$$

The initial valuation of the 4 sub portfolios and the global portfolio corresponding of the market data displayed in the Table 1 are given as follows:

Initial Valuation of Sub-Portfolios	
Portfolio	Valuation
Portfolio 1	538
Portfolio 2	381
Portfolio 3	766
Portfolio 4	628
TOTAL	2312

Table 7: Initial Valuation of Sub-Portfolios

Figure 6 are obtained with 10,000 simulations of P&L portfolio each and from using the full revaluation and the Gaussian process regression. The risk factor simulations reflect the changes of the spot prices over a horizon of one day. More precisely, it compares the MC simulated P&L distribution of the trading portfolio under the full revaluation with those obtained from the GPR algorithm varying the size of the training set (size = 3, 5, 10, 20). It is obvious that the numerical test highlights the perfect accuracy level of the GPR algorithm estimation.

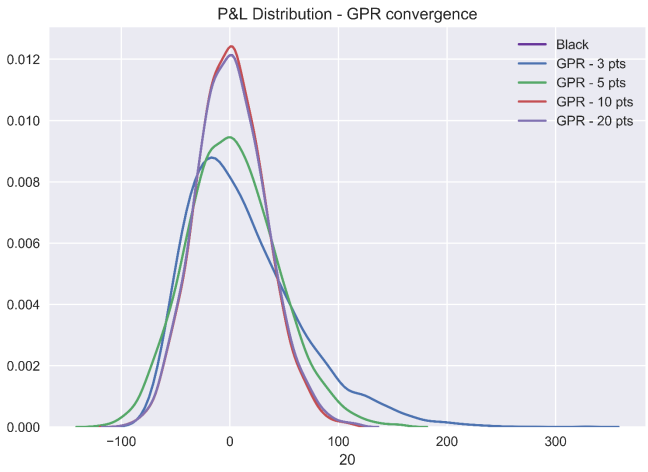


Figure 6: the MC simulated P&L distributions of the trading portfolio (size = 5,10,20)

Global Portfolio				
Indicator	B&S Valuation	GPR 20 pts	GPR 10 pts	GPR 5 pts
VaR 90%	40	40	40	50
VaR 95%	51	51	51	64
VaR 99%	69	69	72	88
VaR 99.9%	88	88	97	113
ES 90%	53	53	54	67
ES 95%	62	62	64	78
ES 99%	78	78	82	98
Speed-Up	3h30 - benchmark	x455	x920	x1974

Table 8: Accuracy and performance (speed-up) of the Gaussian process algorithm

As expected, for an appropriate choice (in this case, training set size = 20) of the training set size, results, in terms of accuracy and performance (speed-up) of the Gaussian process algorithm are spectacular. The numerical test of Table 4 show that the Gaussian process regression (GPR) can drastically improve the computing time whilst ensuring an excellent level of accuracy. In addition, for the Value-at-Risk Calculation purposes, the accurate estimation of the P&L distribution can be carried out with the GPR algorithm trained on a smaller number of prices. Empirical tests show that five spot prices appropriate chosen, for example, in $[S_0 - 2.33\sigma_s\sqrt{h},\ S_0 + 2.33\sigma_s\sqrt{h}]$ suffices to a have a good convergence (cf. Table 5 & GPR approach with 5 pts).

VaR and ES (Rockafellar & Uryasev Approach) and GPR algorithm						
	Value at Risk		Expected Shortfall		GPR Approach with 5pts	
Confidence Level	GPR-20pts	GPR-5pt	GPR-20pts	GPR-5pt	VaR MC	ES
90%	40	40	55	55	40	54
95%	52	52	64	64	52	63
99%	72	72	81	80	71	79

Table 9: VaR and ES computation Combining Rockafellar & Uryasev Approach and GPR algorithm

As the low cost of the implementation of the GPR algorithm also incite to estimate accurately the P&L distribution of the linear and non-linear portfolio, the Rockafellar & Uryasev approach constitutes an efficient parametric method, compared to the Monte Carlo simulations, to compute simultaneously and semi-analytically the VaR and the Expected Shortfall of the derivatives portfolio. This is confirmed by the numerical results of the Rockafellar & Uryasev approach displayed in Table 9.

We outline that whatever the pricing models used to price the global portfolio, once the Gaussian process learned the valuation task of all pricing models by training it directly on a limited data set, it takes over to carry out the repeated revaluation of the derivatives security's portfolio in a fast and efficient way.

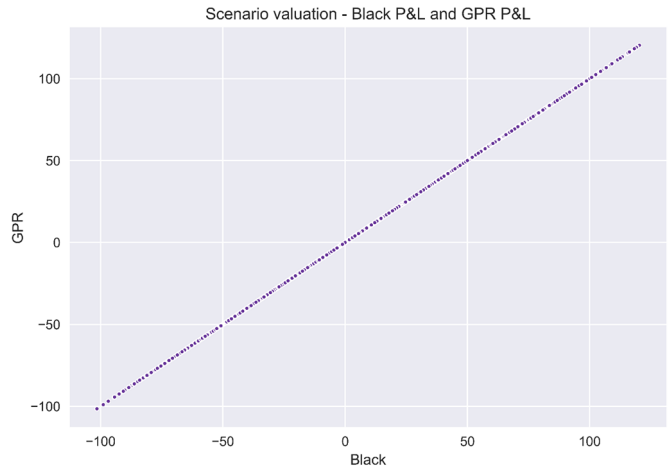


Figure 7: Out—of-sample prediction of GPR P&L versus Black P&L (10,000 simulations)

The scatter-graph 7 is devoted to plot 10,000 out-of-sample predictions of profit and loss (P&L) of the global portfolio computed with Black & Scholes model and the Gaussian process algorithm. Once again that the GP estimation is perfect since the Gaussian process algorithm trained on 20 points hardly deviates from to the bisecting (45 degree) line.

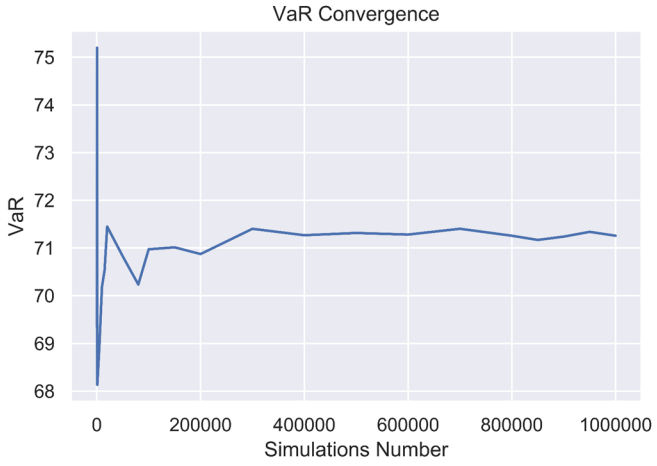


Figure 8: Monte Carlo Simulations convergence of the Value-at-Risk 99% with one day horizon

The Figure 8 shows the convergence of Monte Carlo simulations of the VaR and ES computations by varying the simulations number from 250 to 1,000,000. In this case, the Value-at-Risk converges to 71. In addition, the 99% VaR with one million of simulations is achieved with at least one minute on our machine. Of course, the speed-up of the calculation of the VaR and the expected shortfall within the GPR framework is independent of the size and the composition of the trading portfolio, which is spectacular!

4.3 Greeks and Non-Linear Approximation VaR

This section is devoted to test the GPR algorithm performance to compute the Greeks (Delta and Gamma) of the Non-linear derivatives portfolio described above. To do this, theses sensitivities are computed by the finite difference shocking the GPR price of the whole portfolio. Then, these indicators are compared to that of obtained directly with the classical model.

Portfolio	Financial Model	GPR Algorithm
1	9.82	9.82
2	-0.25	-0.26
3	2.88	2.88
4	5.51	5.50

Table 10: Delta of the portfolios 1,2, 3, and 4

Portfolio	Financial Model	GPR Algorithm
1	0.07	0.10
2	0.11	0.15
3	0.06	0.08
4	0.07	0.09

Table 11: Gamma of the portfolios 1,2, 3, and 4

As expected, the numerical results confirm conclusions in term of accuracy and performance of the Gaussian Process Regression (GPR) already obtained above for the valuation purposes. These Greeks can be used to compute the Delta-Gamma Value at Risk of the non-linear derivatives according the methodology described in the subsection 2.2.

Delta-Gamma Value at Risk			
Confidence interval	Financial Model	GPR Algorithm	MC-GPR
90%	42	42	40
95%	54	54	51
99%	76	76	69

Table 12: Delta-Gamma Value at Risk of the non-linear derivatives portfolio describe in Table 1

Once again, the numerical tests confirm that the GPR algorithm is performant to compute Greeks and Value at Risk.

5 Conclusion

Since the recent financial crises, banking and financial regulations have been strengthened and the internal models of banks are questioned by regulators that request to upgrade the risk models and to implement the new quantitative risk tools. Even though the challenges are enormous in dealing

with the new methodologies, banks try hard to fitful their obligations complying with regulatory requirements. In the same time, it is obvious that much more substantial means are needed if the challenge is to be met. As a first attempt, banks have recourse to Monte Carlo simulations to implement their risk models which are time consuming. These main limitations are largely due to repeatedly revalue in real time the large portfolios of linear and non-linear derivatives. To adequately tackle these limitations, we are interested to the issue related to the effective implementation of the risk models and to its performance problematic. Therefore, we have rolled out the statistical and machine learning tools, a powerful algorithm for both regression and classification, to carry out the repeated valuation of a large portfolio composed by linear and non-linear derivatives. To be more precise, we have applied a Bayesian non-parametric technique, called Gaussian Process Regression (GPR). To test the accuracy calculation and the performance of the GPR algorithm, we have considered a non-linear derivatives portfolio of 100 vanilla and barrier/American options written on an equity asset. Then, the GPR algorithm is trained once directly on a data set generated by the pricing models. The numerical tests show that the GPR algorithm takes over the pricing models to carry out the repeated valuation of a large portfolio in a fast and efficient way. It is worth to notice that no need to learn to the GP algorithm to separately revalue each derivative security composing the portfolio. This characteristic allows to make the speed-up of the calculation of the VaR and the expected shortfall independent of the size and the composition of the trading portfolio, which is spectacular. For this, it seems to us that it is more profitable for the Banks to build their risk models using the powerful of the GPR techniques in terms of the calculation accuracy and the speed-up. We note that the Gaussian Process Regression can be used for a huge number of real-life applications in risk-management field. We will focus on other topics and applications in a forthcoming paper.

6 Appendices

6.1 Portfolio composition of the numerical test

In the subsection 4.1, we consider a numerical example with a derivative trading portfolio of 100 vanilla options and barrier/ American options written on four underlying's. The details of the portfolio composition (level and of barriers, strike, maturity, options types, etc.) are displayed in the following table.

Portfolio on S1			
Type	Strike	Barrier	Maturity (y)
c	82		4
c	60		2
c	126		2
c	131		2
p	68		2
c	124		2
c	81		4
c	125		4
p	89		1
p	123		1
AC	108		1
AP	62		9
AC	64		9
AC	119		6
AC	61		2
pui	121	92	3
cuo	138	98	8
cdo	88	84	4
cui	63	100	5
cdo	110	144	2
pdo	108	99	4
cui	82	83	3
pdo	136	128	6
pdi	67	110	5
cdi	102	101	9

Table 13: Composition of the derivatives trading portfolio on S1

Portfolio on S2			
Type	Strike	Barrier	Maturity (y)
p	124		7
p	110		5
p	68		3
p	112		3
c	76		6
c	122		6
p	115		8
c	138		9
p	61		2
c	136		2
AP	103		3
AP	115		5
AC	102		1
AP	99		4
AP	100		5
cdi	81	72	9
cdi	82	149	6
pdo	139	123	1
pui	73	89	9
pdo	106	100	2
cui	92	127	6
pui	84	105	6
pdi	85	106	2
pui	131	131	7
pdi	87	126	6

Table 14: Composition of the derivatives trading portfolio on S2

Portfolio on S3			
Type	Strike	Barrier	Maturity (y)
p	70		5
p	136		8
c	77		2
c	116		8
p	80		3
c	97		6
c	80		1
c	136		9
p	110		7
c	72		4
AP	115		4
AC	127		1
AP	138		1
AP	136		8
AP	107		4
cdo	131	109	5
pdi	81	76	5
cdo	123	124	9
pui	98	86	9
pdi	139	126	1
cdi	85	74	1
cdi	73	105	6
cuo	79	126	2
pdo	104	118	7
pui	128	137	9

Table 15: Composition of the derivatives trading portfolio on S3

Portfolio on S4			
Type	Strike	Barrier	Maturity (y)
p	130		4
c	93		2
c	79		8
c	102		4
c	71		8
c	122		3
c	124		1
p	107		2
c	139		3
c	63		3
AC	110		6
AC	74		1
AP	101		5
AC	128		2
AP	93		1
cdi	123	127	3
pdi	71	95	2
cuo	61	115	1
cdo	118	99	5
cuo	96	77	3
cdi	67	124	7
cui	111	139	1
cdi	73	132	5
cuo	76	133	1
pui	122	108	9

Table 16: Composition of the derivatives trading portfolio on S4

6.2 Training sets of the Gaussian process regression

The following tables display inputs, for the training step of the Gaussian process algorithm, used in the numerical example. Recall that tests are considered by varying the size of the training set (d=5,10,15,20).

Training Set (16 training observations)					
Spot	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 4	Global Portfolio
1	260	1 289	1 010	737	3 296
10	226	1 174	932	686	3 018
19	198	1 060	865	637	2 760
28	182	946	811	592	2 531
37	181	836	769	551	2 338
46	194	732	742	519	2 187
55	221	637	728	498	2 084
64	261	554	724	491	2 030
73	314	485	730	496	2 026
82	379	434	746	514	2 073
91	454	401	769	541	2 165
100	538	384	799	578	2 299
109	630	381	835	622	2 468
118	729	390	876	675	2 670
127	834	409	923	735	2 901
136	945	435	973	803	3 156

Table 17: Training set of the Gaussian process regression (size=16)

Training Set (20 training observations)					
Spot	Portfolio 1	Portfolio 2	Portfolio 3	Portfolio 4	Global Portfolio
1	260	1 289	1 010	737	3 296
8	233	1 200	948	697	3 078
15	210	1 110	893	658	2 872
22	191	1 022	846	621	2 680
29	182	934	805	587	2 507
36	180	848	773	556	2 357
43	188	766	750	529	2 232
50	204	688	734	508	2 135
57	229	617	726	495	2 067
64	261	554	724	491	2 030
71	301	499	728	494	2 023
78	349	455	738	505	2 046
85	403	422	753	522	2 099
92	463	399	772	545	2 178
99	528	385	795	573	2 282
106	599	381	822	606	2 408
113	674	384	853	644	2 554
120	752	393	886	687	2 719
127	834	409	923	735	2 901
134	920	429	961	787	3 097

Table 18: Training set of the Gaussian process regression (size=20)

References

[1] Rosasco L. Alvarez, M.A. and N.D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends R*, 2012.

[2] Claes A.G. De Ceuster M.J. Annaert, J. and H. Zhang. Estimating the spot rate curve using the nelson-siegel model: A ridge regression approach. *International Review of Economics and Finance*, 2013.

[3] J.O. Berger. Statistical decision theory and bayesian analysis. *Springer*, 2013.

[4] J Berkowitz and J. O'Brien. How accurate are value-at-risk. *Great Britain: Pearson Education Limited*, 2002.

[5] Alexender C. Market risk analysis : Value at risk. *West Sussex: John Wiley & Son*, 2008.

[6] Fisher R.A Cornish E.A. Moments and cumulants in the specification of distributions. *Review of the International Statistical Institute*, 1938.

[7] M. Galai Crouhy and D. a. R. M. The essential of risk management. *McGraw-Hill*, 2014.

[8] Rajiv Sambasivan & Sourish Das. A statistical machine learning approach to yield curve forecasting. *Department of Computer Science, Chennai Mathematical Institute*, March 2007.

[9] Robert Culkin & Sanjiv R. Das. Machine learning in finance: The case of deep learning for option pricing. *Preprint*, August 2017.

[10] Robert Culkin Sanjiv R. Das. Machine learning in finance: The case of deep learning for option pricing. 2017.

[11] C. DeBrusk and E. Du. Why wall street needs to make investing in machine learning a higher priority. *Oliver Wyman Report*, 2018.

[12] Stéphane Crépey & Mattew F. Dixon. Gaussian process regression for derivative portfolio modeling and application to cva computations. 2019.

[13] K Dowd. Beyond value at risk. *John Wiley & Sons*, 1998.

[14] Kevin Dowd. Introduction to market risk measurement. *Wiley Finance*, 2002.

[15] D.; Duffie and Pan J. An overview of value at risk. *Journal of Derivatives*, 1997.

[16] S. A. M. &. D. Enke. Volatility forecasting using a hybrid gjr-garch neural network model. *Procedia Computer Science*, 2014.

[17] Financial Stability Board (FSB). Artificial intelligence and machine learning in financial services - market developments and financial stability implications. 2017.

[18] Antonio Maffia & Yong Chao Sun Gang Mu, Teodor Godina. Supervised machine learning with control variates for american option pricing. *Foundations of computing and decision science*, 2018.

[19] Yan Songc Shuqi Qiuc Ran Xiaod & Fei Sud Heng-Guo Zhanga, Chi-Wei Sua. Calculating value-at-risk for high-dimensional time series using a nonlinear random mapping model. *Economic Modelling*, 2017.

[20] J. Hull. Var versus expected shortfall. *Risk Management and Financial Institutions*, 2006.

[21] J.C. Hull. Options, futures, and other derivatives. *Prentice–Hall*, 1999.

[22] Sofie Reyners & Schoutens Jan De Spiegeleer, Dilip B. Madan. Machine learning for quantitative finance: Fast derivative pricing, hedging and fitting. *University of Leuven, Department of Mathematics*, 2018.

[23] Thierry Roncalli Joan Gonzalvez, Edmond Lezmi and Jiali Xu. Financial applications of gaussian processes and bayesian optimization. 2019.

[24] P. Jorian. Value at risk. *McCraw-Hill*, 2001.

[25] P. Jorion. Value at risk: The new benchmark for controlling market risk. *McGraw-Hil*, 2001.

[26] McKinsey. Fintechnicolor: The new picture in finance, mckinsey report. 2016.

[27] J.P. Morgan. Inroduction to risk metrics. 1994.

[28] J.P. Morgan. Riskmetrics - technical document. 1996.

[29] OECD. Robo-advice for pensions, oecd report. 2017.

[30] Basel Committe on Banking Supervision. Minimum capital requirements for market risk (frtb). *Bank for International Settlements 2015*, 2016.

[31] S. Pichler and K Selitsch. A comparison of analytical var methodologies for portfolios that include options, working paper. *Vienna University of Technology*, 1999.

[32] Carmen Lopez Pilar Abad, Sonia Benito. A comprehensive review of value at risk methodologies. *The Spanish review of Value at Risk Methodologies*, 2014.

[33] Rockafellar R.T. and Uryasev S. Optimization of conditional value-at-risk. *Journal of Risk*, 2000.

[34] M.M. Saunders, A. ; Cornett. Financial institutions management: A risk management approach, 4th edition. *McGraw-Hill*, 2003.

[35] Cornelis W. Oosterlee Shuaiqiang Liu and Sander M.Bohte. Pricing options and computing implied volatilities using neural networks. September 2018.

[36] Mikhail Kanevski & Vadim Timonin. Machine learning analysis and modeling of interest rate curves. *ESANN 2010 proceedings, European Symposium on Artificial Neural Networks*, April 2010.

[37] Sascha Wilkens. Machine learning in risk measurement: Gaussian process regression for value-at-risk and expected shortfall. *BNP Paribas, Risk Analytics & Modelling*, 2019.