Leveraging Machine Learning and Deep Learning for Pricing Down-and-In Put Options

Mohammed AHNOUCH[1]I, Erwan LE SAOUT[2]I, Lotfi ELAACHAK[3]II,
Abderrahim GHADI[4]II

IUniversité Paris 1 Panthéon Sorbonne
IIFSTT Université Abdelmalek Esaadi, Tangier-Morocco

Abstract. This paper delves into the pricing of down-and-in put option, integral to the complexity of derivatives like autocallables[5] . These options are sensitive to the skew and the autocorrelation of the underlying process rendering flat volatility pricing inadequate for the purpose of pricing and risk management. The study initiates by elucidating the risk profile of these products, specifically focusing on their vanna and volga. It then advances to evaluate the effectiveness of machine learning and deep learning technique in pricing. Through detailed comparative analysis, the paper cotributes to sheding light on leveraging AI for pricing derivatives and equity linked annuities[6].

Keywords: AI and finance, Asset pricing and risk premia, Derivatives, Financial risk management

1 Introduction

1.1 Background and Motivation

Down-and-in put options are frequently embedded within more complex derivative structures, such as autocallables and exotic basket options, thereby contributing to the intricate risk profile of these financial instruments (Bouzoubaa & Osseiran, 2010). While first-order Greeks such as Delta, Gamma, and Vega are well-understood, the literature exhibits a noticeable gap regarding the analysis of higher-order and cross Greeks, such as Volga and Vanna, despite their prevalent application in the pricing of FX exotic derivatives (Ovejero, 2022; Shkolnikov, 2009). Most studies tackle the

[1] ma518@cantab.net
[2] erwan.le-saout@univ-paris1.fr
[3] lotfi1002@gmail.com
[4] a.ghadi.fstt@gmail.com

[5] An autocallable is a financial derivative offering high coupons and early termination options based on an underlying asset's performance. It carries the risk of reduced payout at maturity due to a short option, typically a down-and-in put, which activates if the underlying asset falls below a certain thresh

[6] A study focusing on Equity-Linked Annuities (ELAs) incorporating autocallable features is undertaken in (Lee & Ko, 2018)

greeks of barrier options from a computational perspective (Cuomo, Sica, & Toraldo, 2020; Guardasoni & Sanfelici, 2016; Jang, Wang, & Kim, 2019; Jeannin & Pistorius, 2010; Kim, Cao, Kim, & Zhang, 2023; Saporito, 2020; Yang, Ma, & Liang, 2018). In particular, the authors in (Chiarella, Kang, & Meyer, 2012) address the challenge of numerically computing barrier option prices under Heston stochastic volatility. They introduce a method of lines approach, capable of efficiently pricing both continuously and discretely monitored barrier options, including those with early exercise features, and computes the early exercise boundary for American barrier options in both scenarios. Based on Broadie and Kaya's work (Broadie & Kaya, 2006), the enhancement of stochastic volatility model simulations is achieved by integrating rejection sampling, conditional Monte Carlo, and antithetic variable techniques, thereby improving the accuracy, and reducing the variance in pricing Barrier Options and their Greeks, and offering applicability to more complex financial products (Yang et al., 2018). In the same vein, utilizing a two-dimensional partial integro-differential equation and Fourier inverse transforms, a closed-form integral solution for pricing barrier options within Heston and Bates frameworks is available (Guardasoni & Sanfelici, 2016) . It enables an efficient, accurate numerical scheme based on the boundary element method, extendable for computing Greeks. The author of (Saporito, 2020) presents an efficient Monte Carlo method for approximating path-dependent derivatives prices under multiscale stochastic volatility models, utilizing functional Itô calculus and Malliavin calculus to derive a first-order price approximation formula, requiring only Black–Scholes model simulations and applicable to various path-dependent derivatives.

Concurrently, it is possible to extend the Black-Scholes model to encompass local volatility models for pricing barriers products (Funahashi & Kijima, 2016), the local volatility is defined as univariate functions of the spot price. The pricing of barrier options within this framework is decomposed into the valuation of three distinct European options. Two of these options are underpinned by an underlying asset, the volatility of which is derived from a composite function. This function combines the original local volatility model with a secondary function that maps the spot price to its reflected counterpart below the barrier level, thereby engendering a new local volatility surface. To price European options under this modified diffusion process, the study employs an approximation method based on the Wiener-Itô Chaos expansion. The results are satisfactory. However, it is noteworthy that the method, although applicable to Constant Elasticity of Variance (CEV) stochastic volatility diffusions, does not universally extend to Dupire's Local Volatility.

The authors in (Euch & Rosenbaum, 2018) investigate the conceptualisation of a perfect hedge within a rough volatility model, establishing its theoretical feasibility, albeit without practical application to a designated financial derivative. The relevance of Local Stochastic Volatility (LSV) models over the Black Scholes model for hedging barrier options is affirmed through a comparative analysis of the profit and loss distributions from hedged positions linked to knock-out barrier options, a common instrument in the foreign exchange market, under both modelling approaches (Ning, Lee, & Langrene, 2017).

The study in (Umeorah, Mashele, Agbaeze, & Mba, 2023) utilises closed-form expressions for barrier option pricing, specifically down-and-out calls, within the Black-Scholes framework, comprising the vanilla option price minus a correction term derived via the reflection principle. It thereby facilitates the derivation of first-order Greeks in closed form, which are then employed to train a Multilayer Perceptron (MLP). The investigation is confined to scenarios with constant volatility, excluding considerations of volatility skew or term structure. The efficacy of MLPs is compared against Polynomial and Random Forest regressors, with the optimal neural network architecture encompassing three hidden layers, a consistent dropout rate of 10%, and node configurations of 512, 32, and 352, respectively for a total 905 nodes and 32 096 connections. Although the Neural Network's R-squared score marginally trails that of a fourth-degree Polynomial regressor, it requires tenfold the computational time for training. Further, this neural network architecture is retained to train on a higher-dimensional dataset that includes three additional entries representing delta, gamma, and vega.

The present paper addresses second-order derivatives involving volatility and delves into the perfect hedge within the shifted lognormal framework. Which exhibits the inherent non-linearity of the pricing function, underscoring the imperative need for advanced numerical methodologies to accurately value such securities. The study's concentration is exclusively on down-and-in put options, offering a detailed exploration in this specific context.

Furthermore, we explore the use of machine learning and deep learning methods to price this product. Traditional computational methods, whilst reliable, can be cumbersome and resource-intensive, particularly when analysing portfolios laden a plethora of derivative structures. Machine learning models, post-training, promise expeditious pricing and scalability (Géron, 2019), quintessential for real-time trading applications. What is more, machine learning algorithms are particularly adept at capturing the highly nonlinear and multivariable valuation map to price down-and-in options, a capacity that conventional quantitative models might not fully achieve.

Our methodology, while bearing resemblance to that outlined in (Umeorah et al., 2023), diverges notably in its application to a distinct product - specifically, Down-and-In puts instead of Down-and-Out calls. Furthermore, our approach introduces a volatility model that accounts for both the volatility smile and term structure[7]. It is pertinent to highlight that we adopt a parsimonious approach regarding the incorporation of additional nodes and layers, reserving more intricate architectures for models and products of greater complexity in forthcoming studies. This includes scenarios where the feature vector encompasses the entirety of the local volatility grid, model exotic parameters, and features of structured products. Regarding tree methods, our approach employs a boosting algorithm incorporating bagging (Fávero, Belfiore, & de Freitas Souza, 2023),

[7] It is important to acknowledge that this model's complexity is somewhat constrained, being regulated by a singular additional parameter.

which involves training numerous decision trees and subsequently amalgamating their outputs.

1.2 Problem Statement and Objectives

A down-and-in option exhibits heightened sensitivity to skew (Fagnant, 2016), particularly to the decay of skew over time. To incorporate the dynamics of skew while retaining the computational simplicity afforded by the Black-Scholes framework, one may utilise the shifted lognormal model as a convenient approximation (Privault, 2022). For this study, we make the simplifying assumption of disregarding rebate, interest rates and dividend yields. Accordingly, the model encompasses six key parameters: the spot price of the underlying asset, the strike price of the option, the barrier level, and two volatility parameters σ_0, serving as a proxy for skew, and σ_1, which can be interpreted as a representation of the overall level of the volatility surface—complemented finally by the option's time to maturity. Less complex models tend to generalize better in machine learning which is crucial to our study (Liu, Li, Shan, & Liu, 2024; Reunanen, Guyon, & Elisseeff, 2003; Ying, 2019)[8].

Once the non-linear aspect of the pricing function is established, our objective is to evaluate the efficacy of various machine learning techniques, notably ensemble methods such as eXtreme Gradient Boosting (XGBoost) (Chen et al., 2015), and Deep Neural Networks (DNNs), in approximating the function that maps observable features to the price of an option. This endeavour illustrates a specific instance of a broader academic inquiry: the approximation of multivariable, nonlinear functions via machine learning techniques.

2 Theoretical framework

2.1 Product description

Down-and-in option is a type of barrier option that becomes active when the price of the underlying asset falls below a certain level B, known as the barrier. If the price of the underlying asset does not fall below the barrier, the option expires worthless. Down-and-in options are used by investors to hedge against downside risk or to speculate on the price movements of the underlying asset (Hull, 2022).

In Autocallable notes with terminal down-and-in payoff (Fagnant, 2016)[9], the strike is at the money forward and the barrier is the factor that allows the structurer to control the price of the equity leg by setting the value of the structured coupon. Additionally,

[8] In comparison, Heston model has additional 5 features.

[9] The down-and-in put option in autocallables is typically linked to an index or the worst-of a basket of stocks which introduces an additional layer of complexity beyond the scope of our current study.

the put option is typically structured with a $\frac{1}{Strike}$ gearing to ensure that the maximum loss is limited to the notional amount.

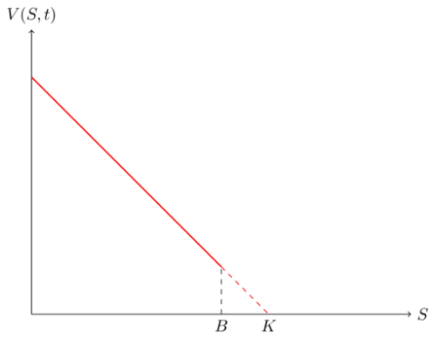


Figure 1 The payoff of a down-and-in put at maturity

The following formula elegantly decomposes a Down&In Put into a portfolio of simpler products: a vanilla Put, a Down&In Call Spread, and a Down&In Digit Put.

$$P_{D\&I}(K,B) = (K-B)PB_{D\&I}(B) + Put(B) - \left(Call_{D\&I}(B,B) - Call_{D\&I}(K,B)\right)$$

The Down&In Call Spread with zero inner value at the barrier level is insensitive to volatility. Elsewhere $Call_{D\&I}(K,B)$ and notably $Call_{D\&I}(B,B)$ both display low Vegas. The Down&In Digit Put component bears the discontinuity embedded in the Down&In Put and is sensitive to skew modelling.

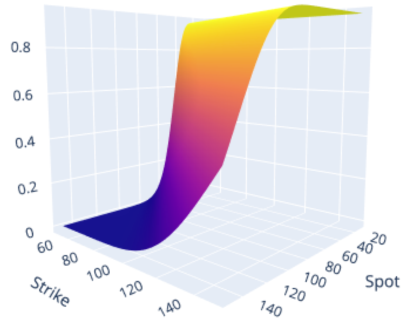


Figure 2 The price of a Down-and-In American digital put option.

The skew contribution to the price can be assessed by understanding the dynamics of the future skew over time[10]. Under the Dupire local volatility model, the skew sensitivity $\left.\frac{\partial\sigma(t,T,K)}{\partial K}\right|_{K=B}$ decays throughout the lifetime of the product. In practice, that translates to the following parametrisation:

$$\left.\frac{\partial\sigma(t,T,K)}{\partial K}\right|_{K=B} = \varphi(t)\left.\frac{\partial\sigma(0,T,K)}{\partial K}\right|_{K=B}$$

Which links future skew to spot skew. $\varphi(t)$ is a decreasing function in terms of t. Given that the local volatility model generates flatter smiles for longer maturities compared to homogeneous stochastic volatility models, the inverse of the values of φ is superlinear.

The skew correction applies to the Black and Scholes price which is (Shreve, 2004):

$$\begin{aligned}
&\mathrm{e}^{-(T-t)r}\mathbb{E}^*\left[(K-S_T)^+\mathbb{1}_{\{m_t^T<B\}} \mid \mathcal{F}_t\right] \\
&= \mathbb{1}_{\{m_0^t\leqslant B\}}\mathrm{Bl}_{\mathrm{put}}(S_t,K,r,T-t,\sigma) - S_t\mathbb{1}_{\{m_0^t>B\}}\Phi\left(-d_+^{T-t}\left(\frac{S_t}{B}\right)\right) \\
&+B\mathbb{1}_{\{m_0^t>B\}}\left(\frac{B}{S_t}\right)^{2r/\sigma^2}\left(\Phi\left(d_+^{T-t}\left(\frac{B^2}{KS_t}\right)\right)-\Phi\left(d_+^{T-t}\left(\frac{B}{S_t}\right)\right)\right) \\
&+\mathrm{e}^{-(T-t)r}K\mathbb{1}_{\{m_0^t>B\}}\Phi\left(-d_-^{T-t}\left(\frac{S_t}{B}\right)\right) \\
&-\mathrm{e}^{-(T-t)r}K\mathbb{1}_{\{m_0^t>B\}}\left(\frac{S_t}{B}\right)^{1-2r/\sigma^2}\left(\Phi\left(d_-^{T-t}\left(\frac{B^2}{KS_t}\right)\right)-\Phi\left(d_-^{T-t}\left(\frac{B}{S_t}\right)\right)\right), \\
&0\leqslant t\leqslant T.
\end{aligned}$$

Where $m_0^T = \min_{t\in[0,T]} S_t$, Φ the cumulative distribution function of standard gaussian and $d_\pm^T = \frac{\log\frac{\sigma_1 F_0+\sigma_0}{\sigma_1 K+\sigma_0}\pm\frac{1}{2}\sigma_1^2 T}{\sigma_1\sqrt{T}}$

It must be noted that the skew decay, in and of itself, is a broad topic. The future skew decay for longer maturities as observed in local volatility surfaces is unsatisfactory and was historically the real motive behind the introduction of local stochastic volatility model (Lipton, Gal, & Lasis, 2014; Ren, Madan, & Qian, 2007). As a matter of fact, all diffusion stochastic volatility models yield a long term ATMF[11] skew decay proportional to $\frac{1}{T}$ which corresponds to market observations but fall short of satisfying the

[10] The skew correction is the mean value of $Vega^{B\&S}(\tau,T,B)\times\left.\frac{\partial\sigma(\tau,T,K)}{\partial K}\right|_{K=B}$, where τ is the crossing time.

[11] At the money forward, given that we ignore rates and repo in the present study the forward and the spot are equal and so are the corresponding implied volatilities

ATMF skew decay shape for short term volatility smiles which is empirically measured to be proportional to $T^{H-\frac{1}{2}}$ where $H \in \left]0, \frac{1}{2}\right[$. This observation has led to the adoption of two-factor diffusion stochastic volatility models and subsequently to the introduction of rough volatility models, wherein 'H' denotes the Hurst parameter (Gatheral, Jaisson, & Rosenbaum, 2017). Local Stochastic volatility models , including Dupire model, are not dynamically consistent with such power law for short maturities (Fukasawa, 2017) nor are single factor stochastic volatility models such as Heston.

The prevailing notion of power law universality for major equity indices encounters substantial challenge in this analysis (Guyon & El Amrani, 2022) . It is observed that the asymptotic skew for extremely short maturities tends to converge to finite values, necessitating the modification of the power law skew curve to incorporate a cap within a three-week period. Consequently, this gives rise to a dichotomy in skew regimes, distinguishing between short and medium maturities. Subsequently, the authors engage in an empirical examination to find the model that most accurately aligns with the observed skew term structure. Their findings reveal that a relatively straightforward four-parameter term structure ATMF skew curve which pertains to a Markovian two-factor Bergomi model demonstrates superior efficacy in this context when compared to those generated by rough volatility (Bergomi, 2005, 2008).

Considering these elements, the methodology employed in our current research eschews the need for a layered approximation of this effect. Instead, it uses a model that not only exhibits a non-trivial local volatility surface but also directly offers analytical formulae for pricing Down-and-In barrier options.

2.2 Shifted lognormal volatility model.

The shifted lognormal model (also known as the displaced diffusion model) is a diffusion process whose volatility structure is a linear interpolation between the normal and lognormal volatilities (Lesniewski, 2014; Privault, 2022). The dynamics of the forward can be expressed as:

$$dF(t) = (\sigma_1 F(t) + \sigma_0) dW(t)$$

The volatility structure of the shifted lognormal model is given by the values of the parameters σ_0 and σ_1.

The model was first introduced in (Rubinstein, 1983) by leveraging existing Black Scholes formula to evaluate a constant weight portfolio consisting of a risky asset and a riskless security. It was then mainly used to generate smile in rates , credit models (Brigo & Mercurio, 2002; Dimitroff, Fries, Lichtner, & Rodi, 2016; Liu & Jackel, 2005; Mercurio & Pallavicini, 2005) and ,to a lesser extent, in equity (Chen & Lee, 2010; Chibane, 2013; Jackel, 2006; McCloud, 2010; ‘Shifted log-normal distribution and moments’, in press).

The shifted lognormal volatility diffusion generates a smile which is monotonous in terms of strikes and whose skew is bounded in absolute value (Lee & Wang, 2012). The implied volatility surface also displays an asymptotic value of volatility for higher strikes across maturities. All of which impedes fitting the model at will to different skew and smile scenarios. The authors suggest using a variant of the model for variance reduction purposes in Monte Carlo simulations.

The prices of calls and puts are given by the functions $Bl_{\mathrm{sln}}^{\mathrm{call}}(T, K, F_0, \sigma_0, \sigma_1)$ and $Bl_{\mathrm{sln}}^{\mathrm{put}}(T, K, F_0, \sigma_0, \sigma_1)$ which generalise both Black Scholes formula and Bachelier formulae corresponding to lognormal and normal models respectively:

$$Bl_{\mathrm{sln}}^{\mathrm{call}}(T, K, F_0, \sigma_0, \sigma_1) = \left(F_0 + \frac{\sigma_0}{\sigma_1}\right) \Phi(\mathrm{d}_+^{\mathrm{T}}) - \left(K + \frac{\sigma_0}{\sigma_1}\right) \Phi(\mathrm{d}_-^{\mathrm{T}})$$

The shifted lognormal model allows for a non-constant, strike-dependent implied volatility. This is achieved by a transformation of the spot price, strike price, and barrier price, which is given by $x \to x + \frac{\sigma_0}{\sigma_1}$. The skew is negative, and its absolute value is an increasing function of σ_0 the while σ_1 controls the level of the entire surface.

In line with (Bergomi, 2017), the shifted lognormal is a model in that there is a break-even constant volatilty σ^* which links the theta of the model to its Gamma . The P&L is the product of the Gamma and the difference between the realised and the breakeven variance. Such a model is arbitrage free.

The perfect vega hedge of the volatility surface in the shifted lognormal model requires two vanilla puts/calls: Although the volatility surface is not flat, it is still completely determined by two parameters σ_0 *and* σ_1. In Local Volatility model, the constitution of such a hedge requires the computation of the hedge ratio at every single point of the volatility grid. The feasibility of such a hedge as well as the weights obtained by recourse to Malliavin calculus are discussed in (Henry-Labordere, 2013; Kettani & Reghai, 2020)

2.3 Sensitivity Analysis

In this section, unless otherwise specified, the characteristics of the down-and-in put option with continuous monitoring and without early exercise feature along with its perfect vega hedge are set as follows:

Option Type	**Strike**	**Barrier**	**Maturity (years)**	**F0**	$\boldsymbol{\sigma_0}$	$\boldsymbol{\sigma_1}$
Down-and-In Put	55	40	1	60	5	16%
Vanilla Put 1	45	N/A	1	60	5	16%
Vanilla Put 2	65	N/A	1	60	5	16%

Below are the curves and surfaces of the value of the portfolio consisting in the Down-and-In put with its vega and delta hedge as function of different features. The Down-and-In put displays negative Volgas, second order derivative, in terms of both σ_0 and σ_1. The gamma of the portfolio is zero hence we perceive a third order behaviour in Figure 4 . Under a perfect delta and vega hedge the expectation of the future gamma, conditional to any value of the underlying, is nil (Henry-Labordere, 2013).

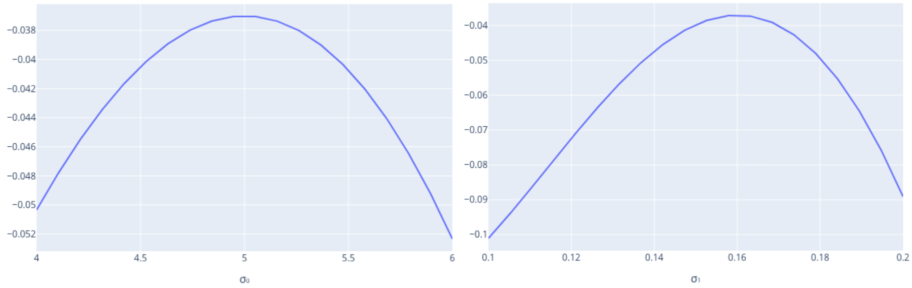


Figure 3 Negative $volga_0$ (left) and $volga_1$ (right)

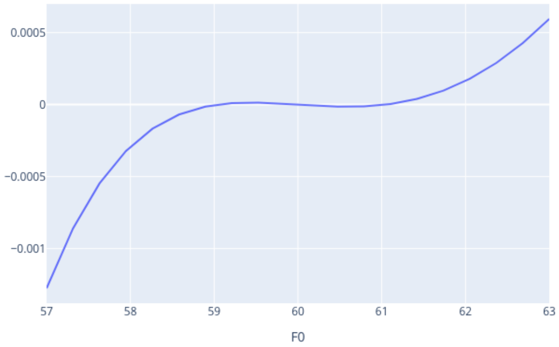


Figure 4 Nil Gamma, third order sensitivity displayed.

In each of Figure 5 and Figure 6, The surface to the left is that of a vanilla put and that to the right pertains to the down-and-in put both struck at 55. Both $Vega_0$ and $Vega_1$ display a similar dependence on spot and σ_0 and σ_1 respectively.

The figures hint at $Vanna_0$ and $Vanna_1$ terms, cross derivatives in volatility and Spot, which change sign and are zero around the strike of the down-and-in put. The vegas of the down-and-in put display a more pronounced dirac spike behavior for low sigma values. $Vega_0$ is a proxy for sensitivity to the skew. The latter, although having a term structure, is controlled by one single parameter σ_0.

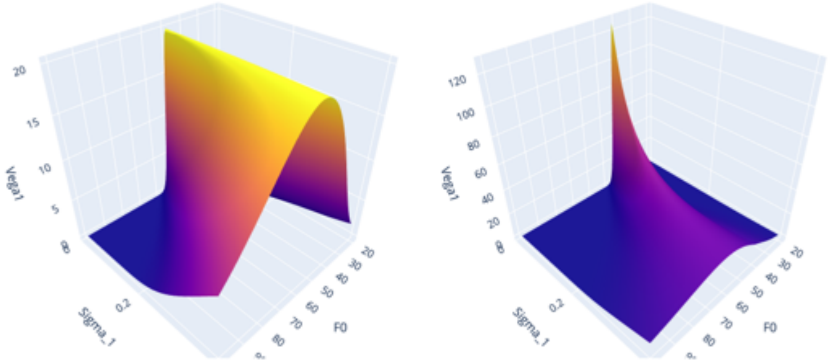


Figure 5 $Vega_1$in terms of σ_1 and initial Spot F_0 (Vanilla Put to the left and DI put to the right)

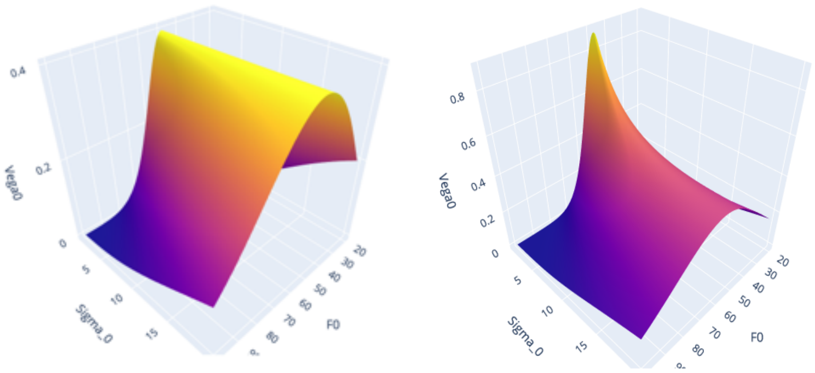


Figure 6 $Vega_0$in terms of σ_0 and initial Spot F_0 (Vanilla Put to the left and DI put to the right)

The change of the sign of Vanna makes hedging the barrier complicated. In practice, it is limited in absolute value when compared to Volga. The latter is negative and contributes positively to the P&L of the seller. However, the Vanna term becomes important when the DI put option is embedded in an autocallable. In this case, the Vanna term is amplified by the presence of the survival indicator function which further complicates the rebalancing of the vega hedge (Guennoun, 2019).

To ascertain the accurate pricing of prospective hedging costs, it is imperative that the model is crafted to embed, from its inception, a proper dynamic for implied volatilities that aligns with the historically observed patterns.

Expressed in an alternative manner, an apt model necessitates the integration of a theta that balances the vanna and volga. This aspect holds significantly greater importance than the model's capacity to precisely replicate the current smile surface (Cont, 2008).

We have established that pricing down-and-in options is a complex task. An effective model must provide realistic dynamics for input variables and an accurate function that maps these variables to a fair price. This is crucial, among other things, for generating the correct theta. We will now move forward with using machine learning to estimate the precise relationship between parameters and fair prices. Temporarily setting aside the task of accurately modelling the joint distribution and dynamics of underlying assets, we will provide justification for this approach.

2.4 Role of Machine Learning in Finance

The necessity for utilising machine learning algorithms in the pricing of financial derivatives is twofold. Machine learning offers the prospect of deriving complex relationships between derivative prices and their various features, which are otherwise arduous to model using traditional, tractable equations. By training these algorithms on expansive sets of historical data, one can potentially unearth subtle, non-linear relationships that would remain elusive to conventional financial models.

Once trained, these machine learning models are highly efficient to execute, thereby affording even smaller market participants, particularly those on the sell side, the capacity to adeptly manage their positions in increasingly sophisticated financial products. This is particularly pertinent for small to middle entities such as pension funds and insurance companies, which may lack the expertise to otherwise engage with these complex instruments effectively.

However, it must be acknowledged that the training phase represents a considerable bottleneck in the deployment of such machine learning models (Reis & Housley, 2022). Given the inherently non-linear nature of derivative pricing, a voluminous dataset, often comprising millions of data points, is requisite for adequately training the model. In this context, the shifted lognormal model, employed in the current study, serves as a compromise between simplicity and exhaustivity. Notably, it incorporates a feature designed to control the skewness of the implied volatility, a parameter to which down-and-in put options are notably sensitive.

2.5 Mathematical context

The Universal Approximation Theorem (UAT) posits that a feed-forward neural network with merely a single hidden layer can approximate any continuous function to an arbitrary degree of accuracy (Voigtlaender, 2023). This extraordinary capability is conditional upon the neurons in the hidden layer employing a non-constant, bounded, and monotonically increasing continuous activation function. The theorem is most pertinent to functions defined on compact subsets of R^n . However, it is crucial to recognise that while the theorem elucidates the theoretical potential of neural networks as general function approximators, it does not offer specific guidance on the optimal architecture or parameterisation, leaving these questions to empirical investigation(Scarselli & Tsoi, 1998).

In approximating non-convex functions, neural networks confront a complex landscape replete with intricate topological features like multiple peaks, valleys, and even discontinuities where the function is not differentiable. These discontinuities pose a particularly nettlesome challenge, given that neural networks primarily rely on gradient-based optimisation techniques (Deng, Wang, Qin, Fu, & Lu, 2022). At these points, the gradient ceases to exist, making it arduous for the network to learn the function's behaviour in these regions. Even minor inaccuracies near these discontinuities could lead to disproportionate errors in the approximation.

Adding to the complexity, the so-called 'curse of dimensionality' implies that the volume of the space under consideration grows exponentially with the number of variables. This necessitates an ever-greater corpus of data for effective approximation, particularly for non-convex, multi-variable functions (Magai & Ayzenberg, 2022). The implications of this are far-reaching, especially in instances where data collection is prohibitively expensive or time-consuming. Furthermore, the network's architecture consists of local approximators like activation functions, and the challenge lies in ensuring that these local approximations coalesce to represent the global behaviour of the function accurately. Such a task becomes increasingly intricate in regions of high non-linearity or rapid changes in the function.

In the current study, we have retained Deep Neural Networks (DNNs) and Extreme Gradient Boosting (XGBoost) for function approximation. These algorithms offer substantial benefits in capturing complex relationships within data but are not without their challenges, such as overfitting and increased computational requirements. However, their potential to deliver heightened accuracy, especially in high-dimensional and non-linear scenarios, makes them compelling choices.

3 Methodology

3.1 Data Collection and Preprocessing

The table below summarises the ranges of each feature that is selected randomly and plugged into the modified Black-Scholes formula for shifted lognormal to generate one million data points which split into two thirds for training and one third for testing.

FEATURE	RANGE
SPOTS	50 to 150 (20 values)
Σ0	2 to 20 (20 values)

Σ1	0.05 to 0.5 (20 values)
STRIKES	20 to 120 (20 values)
BARRIERS	10 to 100 (20 values)
MATURITIES	0.01 to 5 (20 values)

Table 1 Features selection intervals.

These ranges ensure a comprehensive sampling of each feature, providing a robust foundation for the study.

Data is generated within a 6-dimension hypercube, following a uniform distribution, thus sidestepping the complexities of correlated feature behaviour between the spot, σ_0, and σ_1. The consequence of such a procedure is twofold. It simplifies the learning process by focusing solely on the mapping between the feature vectors and the derivative prices (Mishra, 2022). Moreover, it alleviates model risk related to changes in feature interdependencies—for instance, a sudden shift from positive to negative correlation as illustrated in Figure 7. This methodology confers additional virtues, such as the avoidance of localised model biases and the enhancement of robustness in the model's generalisation capabilities (Hua et al., 2021; Jin, Liu, Ma, Aggarwal, & Tang, 2022; Liu, Wang, Li, & Fu, 2021; Shen et al., 2020).

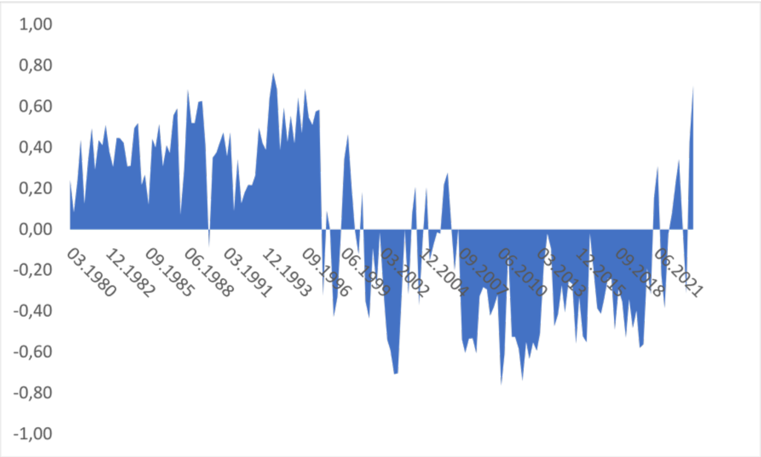


Figure 7 Correlation of S&P 500 and US Treasury returns(‘Markets adjust to “higher for longer”’, 2023)

By disrupting intrinsic data patterns, the process effectively mitigates the risks associated with overfitting (Reis & Housley, 2022). Moreover, the stochastic nature of the sampling serves as a countermeasure against the neural network algorithm's sensitivities to the sequence and arrangement of training data, thereby circumventing pitfalls such as local minima entrapment. Complementarily, the approach is computationally efficient, dynamically adjustable, and inherently conducive to data augmentation, noise tolerance, and objective post-training evaluation.

Nevertheless, there are limitations inherent to this approach. The most salient among these is the risk that the generated data may not be fully representative of real-world conditions, particularly if the underlying distribution for generation deviates markedly from the actual distribution. Additionally, a poorly calibrated random sampling mechanism could inadvertently focus the model on anomalous data points, thereby skewing its learning towards outliers. Thus, while the methodology has demonstrable merits, it also necessitates a considered implementation to ensure that the resultant model is both robust and applicable.

Next in the machine learning pipeline, the MinMaxScaler was retained for feature normalisation, as it provides an efficacious avenue for scaling the features into a prescribed range, typically between 0 and 1. Such scaling is pivotal for various machine learning algorithms, particularly those predicated upon distance metrics like k-NN, as well as gradient-based algorithms, facilitating improved performance and accelerated convergence (Géron, 2019). Furthermore, the MinMaxScaler offers interpretability advantages, conferring a uniform range across features that augments the interpretive clarity of model coefficients. It also enhances numerical stability by constricting the feature values within a defined range, thereby mitigating the risk of numerical overflows, a nontrivial consideration in computational settings.

3.2 Machine Learning Techniques[12]

The application of Deep Neural Networks (DNNs) and Extreme Gradient Boosting (XGBoost) for the purpose of pricing down-and-in options presents a compelling case. Introduced in the 1980s, DNNs have evolved to comprise multiple layers with a high degree of freedom, thereby demonstrating an aptitude for capturing complex, non-linear relationships. XGBoost, a tree-based ensemble method introduced in 2014 (Chen et al., 2015), is renowned for its high-performance and capability to model intricate patterns(Sagi & Rokach, 2021). The algorithm uses boosting[13] to convert weak learners into a strong predictive model.

There remains an intriguing possibility for XGBoost to outperform DNNs under certain conditions (Güvenç, Çetin, & Koçak, 2021). XGBoost often demonstrates resilience against overfitting, particularly with its capacity for automatic feature selection and regularisation, which could offer a distinctive advantage in situations where data is sparse or particularly noisy. Furthermore, the interpretability of XGBoost models could be beneficial for risk assessments and regulatory compliance, facets not to be overlooked in financial contexts. Hence, the decision to employ both DNNs and XGBoost not only capitalises on their respective strengths but also furnishes a robust

[12] The tests were conducted on an NVIDIA P100 GPU on the Kaggle cloud.

[13] Each new tree takes a step in the direction that minimizes the loss function, akin to how gradient descent works for differentiable functions. This "gradient" step is why the method is termed "gradient boosting" (Wade, 2020)

methodological framework that could accommodate the nuanced complexities inherent in option pricing.

4 Technical specifications and Results

4.1 XGboost

In the initial XGBoost Regressor configuration, several hyperparameters are specified to fine-tune the model's performance. The model seeks to minimise the squared error between predicted and actual values—a conventional choice for regression tasks. The number of boosting rounds or trees to be built is set at 20. The `max_depth` parameter is configured to limit the maximum depth of each decision tree to 7, which helps control the model's complexity and mitigate the risk of overfitting. The `learning_rate` parameter is set at 0.5, a high value that scales the contribution of each tree; this parameter effectively regularises the model by preventing rapid convergence to a suboptimal solution. The `early_stopping_rounds` parameter is employed and set to 20 rounds, serving as an additional regularisation technique. Later 50 000 data points were taken from our input data to run grid search for optimal hyperparameters.

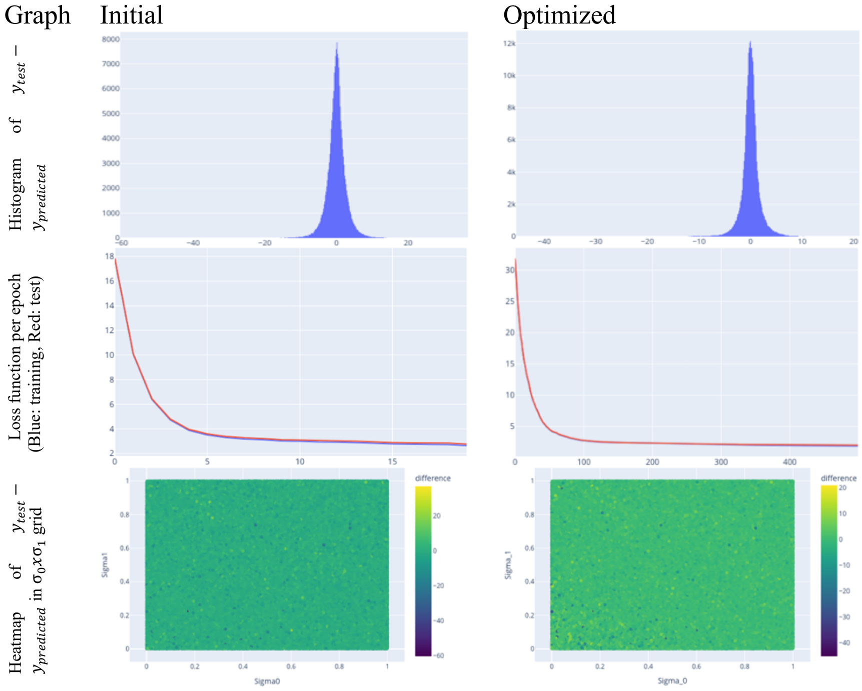


Figure 8 XGBoost metrics

The loss function values for both the training and test datasets are nearly identical, it is generally indicative of a well-generalised model. Moreover, the Root Mean Square

Error hovers around 2.7 in both iterations. Despite the overall consistency, it is noteworthy that a minuscule number of outliers exist within a large dataset comprising 330,000 data points. Importantly, a graphical representation on $\sigma_0 x\, \sigma_1$ grid demonstrates that these anomalous points are not confined within any region of the features' vector space.

4.2 DNN

For our deep neural networks (DNN), the learning rate was set at 0.01 and the model was trained for a maximum of 300 epochs to ensure both convergence and computational efficiency. To mitigate fluctuations in the loss function on the test set, the batch size was incrementally from 64 to 1024. We employed the Adam optimizer[14] , LeakyRelu activation function[15] and dropout as a regularisation technique. This combination of hyperparameters and algorithms was chosen to bolster the DNN's capacity for robust and stable learning.

In a chronological order of model development, various configurations of Deep Neural Networks (DNNs) were explored, each differing in the number of nodes per layer. Initially, an additional hidden layer was introduced with the intent of enhancing the model's performance; however, this led to a conspicuous instance of overfitting. Consequently, the third iteration of the DNN was designed with three layers, albeit with a greater number of nodes per layer compared to the inaugural DNN. Notably, this did not induce any appreciable decrease of the Root Mean Square Error. To mitigate the propensity for overfitting, the subsequent version was adjusted to have fewer nodes per layer. In a final attempt to exploit the Universal Approximation Theorem, a DNN with a singular hidden layer comprising 10 nodes was employed, while extending the epoch count to 1000 from the initial setting of 300 and setting a cap on the error equal to 1 bp $=10^{-4}$. This computational endeavour necessitated an extensive training duration of nearly three hours after 1000 epochs.

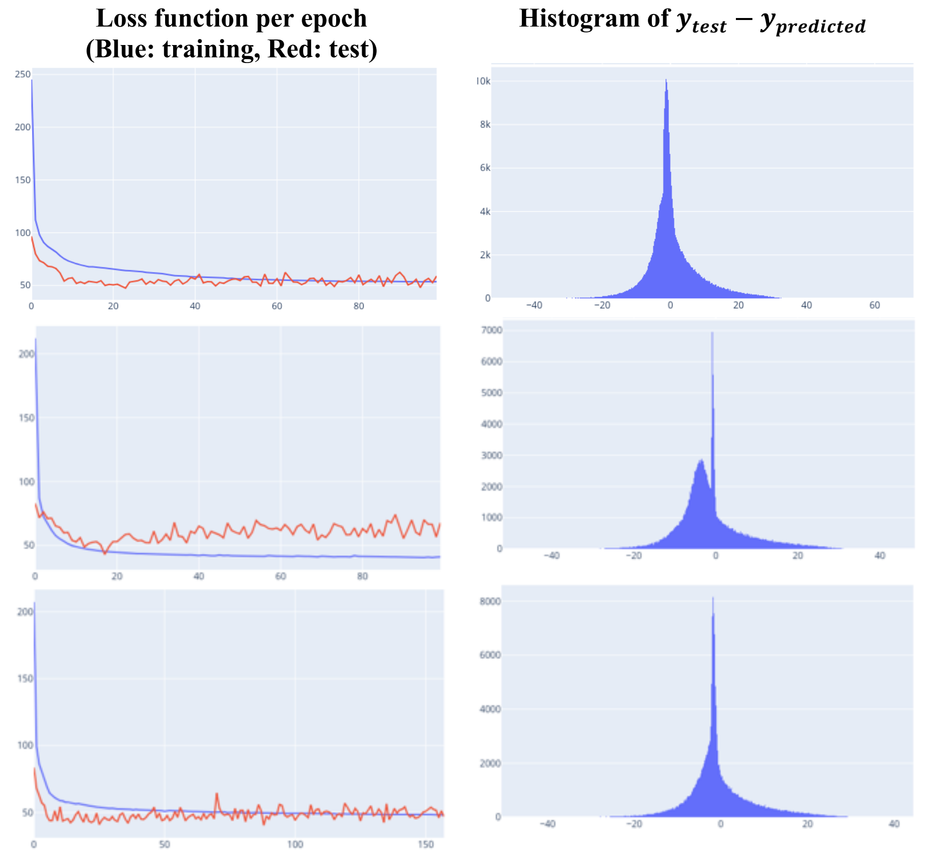
Architecture	**RMSE**	**Training time**
1-30-20-10-1	33.87	20 min 34
1-50-40-20-5-1 (overfit)	33.82	18 min 30
1-50-30-10-1	33.85	19 min 47
1-24-12-6-1	33.95	18 min 08
1-10-1 (1000 epochs)	33.83	2 h 57

[14] A first-order gradient-based optimisation algorithm, particularly popular for its adaptive learning rate and efficient handling of sparse gradients

[15] The Leaky ReLU activation function is designed to solve the "dying ReLU" problem by allowing a small, non-zero gradient when the input is less than zero. Its ability to permit negative values through a small constant α enables it to capture and approximate highly non-linear functions in euclidian spaces.

The stability of RMSE metric, which serves as a quantifiable representation of the model's prediction errors, suggests that further refinements are warranted to enhance the model's predictive accuracy. Moreover, it is noteworthy that the distribution of the residuals—specifically the difference between the test and predicted values—exhibits a slight bias towards negative values.

A vacillating trajectory in the validation loss (red) may signify a range of challenges or phenomena within the training procedure. One noteworthy possibility is that of overfitting, particularly manifest when the training loss consistently diminishes whilst the validation loss demonstrates intermittent fluctuations.


Loss function per epoch (Blue: training, Red: test)

Histogram of $y_{test} - y_{predicted}$

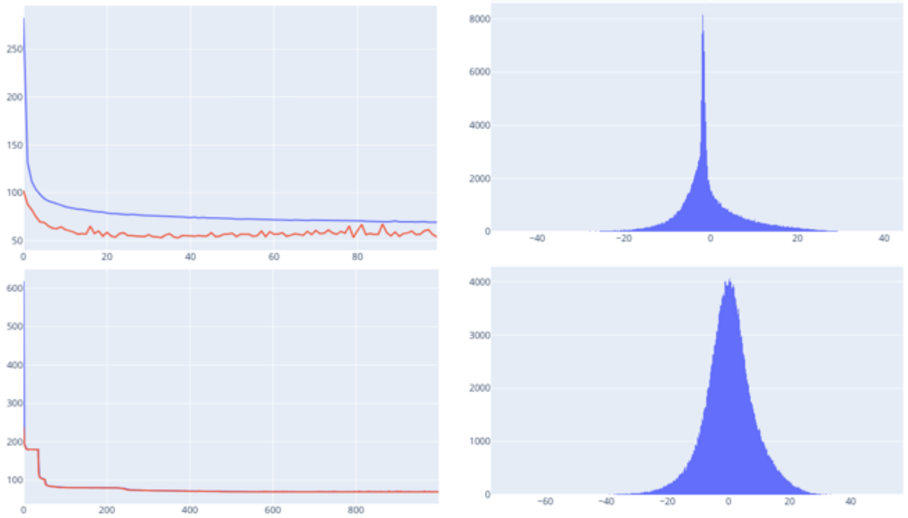


Figure 9 Metrics for different DNN architectures

4.3 Performance Comparison

In the context of our empirical investigation, it was evident that XGBoost markedly surpassed the Deep Neural Network (DNN) in performance. While both XGBoost and DNNs possess distinct advantages and limitations contingent upon the particularities of the problem at hand (Elbrächter, Grohs, Jentzen, & Schwab, 2022), XGBoost demonstrates a proclivity for excelling with structured data and the ability to manage missing values[16].

Several factors contribute to the superiority of XGBoost in approximating nonlinear multivariable functions. XGBoost models allows for straightforward tuning and explanation of model decisions (Sagi & Rokach, 2021). Besides, the built-in regularisation techniques—L1 (Lasso) and L2 (Ridge)—significantly mitigate the risk of overfitting. In terms of simplicity, XGBoost typically requires fewer hyperparameter adjustments compared to DNNs, facilitating the rapid development of robust models (Wade, 2020).

The results of the present investigation concur with the established consensus in the specialized literature (Grinsztajn, Oyallon, & Varoquaux, in press; Konstantinov & Utkin, 2023; Shwartz-Ziv & Armon, 2021). The paper (Grinsztajn et al., in press) shows that whilst deep learning exhibits prowess in text and image processing, its predominance in managing tabular data, especially of a medium scale (approximately 10,000 samples), is not as pronounced. Notably, tree-based models continue to lead in this area, irrespective of their expedited processing capabilities. This research undertakes an examination to elucidate the factors contributing to this discrepancy. Furthermore, an

[16] In contrast, DNNs are adept at capturing intricate patterns and tend to perform exceptionally well in high-dimensional or unstructured data scenarios, such as image and text analysis.

additional paper (Shwartz-Ziv & Armon, 2021) assesses the comparative performance and calibration requisites of deep learning frameworks and XGBoost with tabular data. The investigation demonstrates the superior performance of XGBoost in diverse datasets, achieved with comparatively minimal tuning. The paper subsequently concludes that an ensemble of deep learning models and XGBoost demonstrates improved performance over XGBoost alone.

5 Conclusion

In the present study, we have elucidated the behaviour of second-order Greeks in down-and-in put options, employing a model that accommodates skewness. Our findings demonstrate a marked superiority of the eXtreme Gradient Boosting (XGBoost) algorithm over Deep Neural Networks (DNNs) for the specified task of pricing this barrier option.

For future scholarly endeavours, there are several compelling avenues to explore. Primarily, the application of more sophisticated models for data generation would be pertinent, allowing for a richer feature set that could lead to more nuanced results. Alternate neural network architectures could also be scrutinised for their efficacy in contrast to traditional DNNs in feed-forward settings. Most crucially, a promising approach consists in incorporating transfer learning in the DNN process, enhancing the efficacy of pricing complex derivatives, or applying more sophisticated models by reusing and fine-tuning pre-trained layers (Neufeld & Sester, 2023). The integration of automatic differentiation frameworks ,such as Autograd in Pytorch (Mishra, 2022), holds significant promise for the real-time hedging of financial instruments without requiring additional training. By leveraging existing work in this domain (Huge & Savine, 2020), we can extend our model's utility to not just pricing derivatives, but also to crafting optimal hedging strategies, thus providing a more holistic financial risk management solution.

6 Bibliography

Bergomi, L. 2005, March 1. *Smile Dynamics II.*

Bergomi, L. 2008. Smile dynamics III. *Available at SSRN 1493308*. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1493308

Bergomi, L. 2017, January 1. *Local-Stochastic Volatility: Models and Non-Models. Available at SSRN 2902130.*

Bouzoubaa, M., & Osseiran, A. 2010. *Exotic Options and Hybrids: A Guide to Structuring, Pricing and Trading*. John Wiley & Sons.

Brigo, D., & Mercurio, F. 2002. Displaced and mixture diffusions for analytically-tractable smile models. *Springer Finance* (pp. 151–174). Berlin, Heidelberg: Springer Berlin Heidelberg.

Broadie, M., & Kaya, Ö. 2006. Exact simulation of stochastic volatility and other affine jump diffusion processes. *Operations research*, 54(2): 217–231.

Chen, R.-R., & Lee, C.-F. 2010. Displaced Log Normal and Lognormal American Option Pricing: A Comparison. In C.-F. Lee, A. C. Lee, & J. Lee (Eds.), *Handbook of Quantitative Finance and Risk Management* (pp. 439–446). Boston, MA: Springer US.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., & Zhou, T. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4): 1–4.

Chiarella, C., Kang, B., & Meyer, G. H. 2012. The evaluation of barrier option prices under stochastic volatility. *Computers & mathematics with applications* , 64(6): 2034–2048.

Chibane, M. L. D. de L. L. 2013, August. Hybrid smiles made fast. *RiskPrévisualiser les détails de la publication; London volume*. search.proquest.com. Retrieved from https://search.proquest.com/openview/aa081c4c7c4e92d91c35f05277277c80/1?pq-origsite=gscholar&cbl=32048

Cont, R. 2008. Frontiers in quantitative finance. Wiley, Hoboken. Retrieved from http://www.untag-smd.ac.id/files/Perpustakaan_Digital_1/CREDIT%20RISK%20Frontiers%20in%20Quantitative%20Finance%20Volatility%20and%20Credit%20Risk%20Modeling%20-%20(2008).pdf

Cuomo, S., Sica, F., & Toraldo, G. 2020. Greeks computation in the option pricing problem by means of RBF-PU methods. *Journal of computational and applied mathematics*, 376: 112882.

Deng, C., Wang, Y., Qin, C., Fu, Y., & Lu, W. 2022. Self-directed online machine learning for topology optimization. *Nature communications*, 13(1): 388.

Dimitroff, G., Fries, C. P., Lichtner, M., & Rodi, N. 2016, March 19. *Lognormal vs Normal Volatilities and Sensitivities in Practice*. *Available at SSRN.*

Elbrächter, D., Grohs, P., Jentzen, A., & Schwab, C. 2022. DNN expression rate analysis of high-dimensional PDEs: Application to option pricing. *Constructive Approximation. An International Journal for Approximations and Expansions*, 55(1): 3–71.

Euch, O. E., & Rosenbaum, M. 2018. PERFECT HEDGING IN ROUGH HESTON MODELS. *The annals of applied probability: an official journal of the Institute of Mathematical Statistics*, 28(6): 3813–3856.

Fagnant, C. 2016. *Pricing Exotic Options*.

Fávero, L. P., Belfiore, P., & de Freitas Souza, R. 2023. Boosting and bagging. *Data Science, Analytics and Machine Learning with R* (pp. 403–427). Elsevier.

Fukasawa, M. 2017. Short-time at-the-money skew and rough fractional volatility. *Quantitative Finance*, 17(2): 189–198.

Funahashi, H., & Kijima, M. 2016. Analytical pricing of single barrier options under local volatility models. *Quantitative Finance*, 16(6): 867–886.

Gatheral, J., Jaisson, T., & Rosenbaum, M. 2017, September 30. *Volatility Is Rough*.

Géron, A. 2019. *Machine Learning avec Scikit-Learn: Mise en oeuvre et cas concrets*. Dunod.

Grinsztajn, L., Oyallon, E., & Varoquaux, G. in press. Why do tree-based models still outperform deep learning on typical tabular data?

Guardasoni, C., & Sanfelici, S. 2016. Fast Numerical Pricing of Barrier Options under Stochastic Volatility and Jumps. *SIAM journal on applied mathematics*, 76(1): 27–57.

Guennoun, H. 2019, May 14. *Understanding Autocalls: Real Time Vega map. Available at SSRN 3387810.*

Güvenç, E., Çetin, G., & Koçak, H. 2021. Comparison of KNN and DNN classifiers performance in predicting mobile phone price ranges. *Advances in Artificial Intelligence Research*, 1(1): 19–28.

Guyon, J., & El Amrani, M. 2022. Does the term-structure of equity at-the-money skew really follow a power law? *SSRN Electronic Journal*.

Henry-Labordere, P. 2013, March 7. *Vega Decomposition of Exotics on Vanillas: A Monte-Carlo Approach*.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., & Zhao, H. 2021, May 2. *On Feature Decorrelation in Self-Supervised Learning*. *arXiv [cs.LG]*. Retrieved from http://arxiv.org/abs/2105.00470

Huge, B., & Savine, A. 2020. Differential machine learning. *arXiv preprint arXiv:2005.02347*. Retrieved from https://arxiv.org/pdf/2005.02347

Hull, J. C. 2022. *Fundamentals of Futures and Options Markets, Global Edition*. Pearson Higher Ed.

Jackel, P. 2006. Hyperbolic local volatility. jaeckel.org. Retrieved December 8, 2023, from http://www.jaeckel.org/HyperbolicLocalVolatility.pdf

Jang, H., Wang, J., & Kim, J. 2019. Equity-linked security pricing and Greeks at arbitrary intermediate times using Brownian bridge. *Monte Carlo Methods and Applications*, 291–305.

Jeannin, M., & Pistorius, M. 2010. A transform approach to compute prices and Greeks of barrier options driven by a class of Lévy processes. *Quantitative Finance*, 10(6): 629–644.

Jin, W., Liu, X., Ma, Y., Aggarwal, C., & Tang, J. 2022, June 15. *Feature Overcorrelation in Deep Graph Neural Networks: A New Perspective*. *arXiv [cs.LG]*. Retrieved from http://arxiv.org/abs/2206.07743

Kettani, O., & Reghai, A. 2020. Local Volatility Model. In O. Kettani & A. Reghai (Eds.), *Financial Models in Production* (pp. 15–37). Cham: Springer International Publishing.

Kim, H.-G., Cao, J., Kim, J.-H., & Zhang, W. 2023. A Mellin transform approach to pricing barrier options under stochastic elasticity of variance. *Applied Stochastic Models in Business and Industry*, 39(2): 160–176.

Konstantinov, A. V., & Utkin, L. V. 2023, February 13. *Multiple Instance Learning with Trainable Decision Tree Ensembles*. *arXiv [cs.LG]*. Retrieved from http://arxiv.org/abs/2302.06601

Lee, H., & Ko, B. 2018. Valuing equity-indexed annuities with icicled barrier options. *Journal of the Korean Statistical Society*, 47: 330–346.

Lee, R., & Wang, D. 2012. Displaced lognormal volatility skews: analysis and applications to stochastic volatility simulations. *Annals of finance*, 8(2–3): 159–181.

Lesniewski, A. 2014. Option Smile and the SABR Model of Stochastic Volatility. Retrieved from https://lesniewski.us/papers/presentations/MIT_March2014.pdf

Lipton, A., Gal, A., & Lasis, A. 2014. Pricing of vanilla and first-generation exotic options in the local stochastic volatility framework: survey and new results. *Quantitative Finance*, 14(11): 1899–1922.

Liu, C., Wang, L., Li, K., & Fu, Y. 2021. Domain Generalization via Feature Variation Decorrelation. *Proceedings of the 29th ACM International Conference on Multimedia*, MM ’21 (pp. 1683–1691). New York, NY, USA: Association for Computing Machinery.

Liu, J., Li, D., Shan, W., & Liu, S. 2024. A feature selection method based on multiple feature subsets extraction and result fusion for improving classification performance. *Applied soft computing*, 150.

Liu, Y., & Jackel, P. 2005. Options On Credit Default Index Swaps. jaeckel.org. Retrieved December 8, 2023, from http://www.jaeckel.org/OptionsOnCreditDefaultIndexSwaps.pdf

Magai, G., & Ayzenberg, A. 2022. Topology and geometry of data manifold in deep learning. *arXiv preprint arXiv:2204.08624*. Retrieved from https://arxiv.org/pdf/2204.08624

Markets adjust to ‘higher for longer’. 2023, December 4. Retrieved January 15, 2024, from https://www.bis.org/publ/qtrpdf/r_qt2312a.htm

McCloud, P. L. D. de L. 2010, January 20. Putting the smile back on the face of derivatives. *RiskPrévisualiser les détails de la publication; London volume*. search.proquest.com. Retrieved from https://search.proquest.com/openview/6b61f14459da33dcd6013e4ece99bc8c/1?pq-origsite=gscholar&cbl=32048

Mercurio, F., & Pallavicini, A. 2005. Swaption skews and convexity adjustments. *SSRN Electronic Journal*.

Mishra, P. 2022. *PyTorch Recipes: A Problem-Solution Approach to Build, Train and Deploy Neural Network Models*. Apress.

Neufeld, A., & Sester, J. 2023. A Deep Learning Approach to Data-Driven Model-Free Pricing and to Martingale Optimal Transport. *IEEE transactions on information theory / Professional Technical Group on Information Theory*, 69(5): 3172–3189.

Ning, W., Lee, G., & Langrene, N. 2017. *Hedging barrier options through a log-normal local stochastic volatility model*. (Australian Government - Bureau of Metrology; CSIRO; NSW Government - Office of Environment and Heritage; Visual Interfaces for Serious Simulations, Trans.). Presented at the 22nd International Congress on Modelling and Simulation: Managing Cumulative Risks through Model-Based Processes, MODSIM 2017 - Held jointly with the 25th National Conference of the Australian Society for Operations Research and the DST Group led Defence Operations Research Symposium, DORS 2017, Modelling and Simulation Society of Australia and New Zealand Inc. (MSSANZ).

Ovejero, J. M. 2022, November 22. *Vanna-Volga pricing for single and double barrier FX options*.

arXiv [q-fin.PR]. Retrieved from http://arxiv.org/abs/2211.12652

Privault, N. 2022. *Introduction to Stochastic Finance with Market Examples*. CRC Press.

Reis, J., & Housley, M. 2022. *Fundamentals of Data Engineering*. 'O'Reilly Media, Inc.'

Ren, Y., Madan, D., & Qian, M. Q. 2007. Calibrating and pricing with embedded local volatility models. *RISK-LONDON-RISK*.

Reunanen, J., Guyon, I., & Elisseeff, A. 2003. Overfitting in making comparisons between variable selection methods. jmlr.org. Retrieved January 15, 2024, from https://www.jmlr.org/papers/volume3/reunanen03a/reunanen03a.pdf

Rubinstein, M. 1983. Displaced diffusion option pricing. *The Journal of finance*, 38(1): 213–217.

Sagi, O., & Rokach, L. 2021. Approximating XGBoost with an interpretable decision tree. *Information sciences*, 572: 522–542.

Saporito, Y. F. 2020. Short Communication: Pricing Path-Dependent Derivatives under Multiscale Stochastic Volatility Models: A Malliavin Representation. *SIAM Journal on Financial Mathematics*, 11(3): SC14–SC25.

Scarselli, F., & Tsoi, A. C. 1998. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural networks: the official journal of the International Neural Network Society*, 11(1): 15–37.

Shen, Z., Cui, P., Liu, J., Zhang, T., Li, B., & Chen, Z. 2020. Stable Learning via Differentiated Variable Decorrelation. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20 (pp. 2185–2193). New York, NY, USA: Association for Computing Machinery.

Shifted log-normal distribution and moments. in press.*Cross Validated*. Retrieved December 8, 2023, from https://stats.stackexchange.com/questions/156277/shifted-log-normal-distribution-and-moments

Shkolnikov, Y. 2009. Generalized Vanna-Volga Method and its Applications. *SSRN Electronic Journal*.

Shreve, S. E. 2004. *Stochastic Calculus for Finance II: Continuous-Time Models*. Springer Science & Business Media.

Shwartz-Ziv, R., & Armon, A. 2021, June 6. *Tabular Data: Deep Learning is Not All You Need*. *arXiv [cs.LG]*. Retrieved from http://arxiv.org/abs/2106.03253

Umeorah, N., Mashele, P., Agbaeze, O., & Mba, J. C. 2023. Barrier Options and Greeks: Modeling with Neural Networks. *Axioms*, 12(4): 384.

Voigtlaender, F. 2023. The universal approximation theorem for complex-valued neural networks. *Applied and computational harmonic analysis*, 64: 33–61.

Wade, C. 2020. *Hands-On Gradient Boosting with XGBoost and Scikit-learn: Perform Accessible Machine Learning and Extreme Gradient Boosting with Python*.

Yang, Y., Ma, J., & Liang, Y. 2018. The research on the calculation of Barrier Options under stochastic volatility models based on the exact simulation. *IAENG International Journal of Applied Mathematics*, 48(3): 349–361.

Ying, X. 2019. An Overview of Overfitting and its Solutions. *Journal of physics. Conference series*, 1168: 022022.