Nonparametric option hedging

João A. Bastos   Raquel M. Gaspar [*]

Abstract

We introduce a novel nonparametric methodology for option hedging. Unlike alternative approaches, our method makes no assumptions about the relationship between option prices and underlying explanatory variables. This approach relies on accumulated local effects given by ensembles of decision trees. Although we employ a gradient boosting machine for the ensemble, any method for creating decision tree ensembles can be considered. By simulating synthetic option prices generated via the Black-Scholes model, we observe that our method accurately reproduces the Greeks as predicted by this model. Then, we conduct an empirical study using American call and put options on individual stocks. Our findings reveal deviations from the binomial model when applied to real market prices.

Keywords: American option; Hedging; Nonparametric model; Accumulated local effect; Gradient boosting machine

[*] Corresponding author. ISEG, Universidade de Lisboa and CEMAPRE/REM Research Center. Rua do Quelhas 6, 1200-078 Lisboa, Portugal. E-mail: Rmgaspar@iseg.ulisboa.pt

1 Introduction

The first statistical models for option pricing, such as those by Black and Scholes (1973), Cox and Ross (1976), Cox et al. (1979), and Heston (1993), assume some data generating processes to derive closed-form analytical solutions for option prices. However, many of these assumptions are quite stringent and are not aligned with empirical regularities observed in financial markets. For instance, the Black-Scholes model provides simple analytical formulas not just for European option prices but also for their sensitivities with respect to the input parameters. These sensitivities, commonly referred to as “Greeks”, are used for hedging the risk associated with holding options in a portfolio. Despite its convenience, the model has several limitations. In particular, it assumes a log-normal distribution and constant volatility for the underlying asset returns, both of which are not empirically observed. Furthermore, it cannot be used for pricing and hedging American options and other exotic options.

These limitations can be addressed through data-driven approaches, particularly machine learning models, that do not depend on stylized assumptions about the stochastic processes governing asset price dynamics. Indeed, the use of machine learning methods has become well-established in the option pricing literature, with roots in the influential works of Malliaris and Salchenberger (1993) and Hutchinson et al. (1994). These methods are good at uncovering complex and nonlinear relationships, making them valuable tools for pricing options and other derivatives, even when closed-form pricing equations are available.

Among the diverse machine learning algorithms, the feedforward neural network trained with back-propagation (Rumelhart et al., 1986) emerges as a widely used model for predicting option prices (e.g., Anders et al., 1998; Garcia and Gençay, 2000; Andreou et al., 2010). However, alternative machine learning models, such as support vector machines (Vapnik, 1995)), have demonstrated higher accuracy compared to neural networks in option pricing (Wang, 2011; Park et al., 2011). Moreover, Ivaşcu (2011) compares multiple machine learning algorithms for predicting option prices, showing that tree-based ensembles like random forests (Breiman, 2001) and gradient boosting machines (Friedman, 2001) outperformed neural networks.

Neural networks can also be used to hedge the risk of option positions. Hutchinson et al. (1994) and Garcia and Gençay (2000) show that neural networks effectively track the error of a delta-hedged portfolio. To achieve this, they analytically compute the derivative of the model's output with respect to the underlying asset price. While this approach is technically feasible – albeit cumbersome – for deep neural networks, it becomes impractical for tree-based machine models that approximate the pricing function using piece-wise constant functions.

A common misconception in the finance literature is the designation of feedforward neural networks as nonparametric models. In fact, Hutchinson et al. (1994) specifically label their neural network models for option pricing as "nonparametric." Parametric models are characterized by having a fixed number of parameters that do not depend on the amount of training data. Feedforward neural networks are composed by layers containing neurons, and the number of parameters in the model is determined by the connections between the neurons in each layer. Of course, there are some architectural decisions to be made, such as deciding on the number of layers and the number of neurons in each layer. But once these decisions are made, the number of parameters is fixed. Or, in other words, the functional form between inputs and outputs is predefined before training. Therefore, feedforward neural network are *highly parameterized* models. Nonparametric models, on the other hand, do not have a fixed number of parameters and the functional form between inputs and outputs is not predefined by the analyst, but is instead defined by the training data. Decision tree-based models like random forests and gradient boosting machines are examples of nonparametric models because their complexity and number of parameters depend on the data itself.

In this paper, we propose propose the first fully nonparametric approach for hedging options. We refrain from making assumptions about the underlying relationship between option prices and their determinants. In other words, the market data exclusively dictate how option prices depend on the inputs. First, we use an ensemble of decision trees to learn the relationship between option prices and explanatory variables. This is a nonparametric model, wherein the functional form between option prices and the explanatory variables is not predetermined but entirely dictated by the data. Furthermore, the number of parameters is not fixed.

Second, we need to understand how the pricing function depends on the inputs in order to obtain the Greeks. Taking the derivative of the model's output with respect to the input variables is unfeasible given the size of the models and the fact that tree-based models approximate the pricing function using piece-wise constant functions. Our solution to this problem is based on "accumulated local effects" (ALE), as introduced by Apley and Zhu (2020). This is a model-agnostic technique that was originally introduced as a tool for visualizing how the output of a "black-box" model depends on its inputs. However, we are not interested in the visualization tool *per se* but in the actual ALE *values* provided by the algorithm. ALE values provide the dependence of the output of a "black-box" model on its inputs, apart from an unknown constant. Luckily, for this concrete application we can use finance theory to calibrate the ALE values such that they represent option prices. Indeed, any reasonable option pricing model must satisfy some constraints. As the price of the underlying asset approaches zero, the price of a call option must also approach zero. Similarly, as the time to expiration nears zero, the option must tend toward the terminal payoff, and so forth. Using this constraints we are able to *calibrate* the ALE values. Finally, we obtain approximations for the option Greeks by computing first and second differences of the ALE values.

We illustrate our method using two datasets. First, we simulated a set of synthetic option prices generated via the Black-Scholes model. This allowed us to evaluate our approach in a controlled environment. We found that our method accurately reproduced the Greeks predicted by this model. Next, we assessed our approach using a dataset comprising real market prices of American options on individual stocks. Our findings unveiled significant deviations from the the binomial model when applied to real market data.

The remainder of the paper is structured as follows: the next section outlines our approach for hedging options. Following that, Section 3 provides a description of the datasets. Section 4.1 reports the results obtained for the synthetic Black-Scholes prices. It will become apparent to the reader that our method effectively reproduces the Black-Scholes model. In Section 4.2, we present the results concerning the real market data, examining the disparities observed in comparison to the Black-Scholes model. Section 5 provides some concluding remarks.

2 Methodology

Suppose we have trained a model to predict option prices for calls C and puts P, considering inputs like the underlying asset price S, strike price K, volatility σ, time to maturity τ, and interest rate r. Here, $\boldsymbol{X}$ represents a set of regressors, $f(\boldsymbol{X})$ represents a trained model, and n denotes the number of training observations.

2.1 Hedging Method

2.1.1 Accumulated local effects

Accumulated local effects (ALE) plots (Apley and Zhu, 2020) were originally proposed as a visual representation of how a regressor influences the model predictions. It indicates the type of relationship between the dependent variable and the regressors. For example, if the true relationship is linear, such plot will actually show a linear dependence. An ALE plot shows how the model output varies as a function of a regressor, keeping other covariates constant (*ceteris paribus* effect). ALE plots are the state-of-the-art approach for visualizing relationships between a target variable and the regressors. Indeed, as pointed out by Apley and Zhu (2020), partial dependence plots (Friedman, 2001) and individual conditional expectation plots (Goldstein et al., 2015) are not appropriate when the regressors are not independent, which is the case here.

Suppose we want to understand how regressor X_j affects the model's output. First, we divide its range using a grid with M bins. Let $\{Z_k\}_{k=0}^{M}$ denote the set of X_j values that define the boundaries of these bins. For instance, the first bin encompasses all X_j values between Z_0 and Z_1, while the last bin comprises all X_j values between Z_{M-1} and Z_M. Typically, the Z_k values are selected as the

(k/M)-quantiles of the empirical distribution of X_j, where Z_0 is chosen slightly below the smallest observation, and Z_M equals the largest observation.

Let $\mathcal{I}_k$ denote the set of indices corresponding to the observations where $Z_{k-1} < X_j \leq Z_k$. The count of observations in each bin is represented as n_k, where $k = 1, \ldots, M$. The average local effect of X_j within a specific bin, while holding all other regressors $\boldsymbol{X}_{\backslash j}$ constant, is given by:

$$\frac{1}{n_k} \sum_{i \in \mathcal{I}_k} \left[f(Z_k, \boldsymbol{X}_{\backslash j,i}) - f(Z_{k-1}, \boldsymbol{X}_{\backslash j,i}) \right]. \tag{1}$$

The sum in Equation 1 loops over all observations in a given bin. For each of these observations, we obtain the difference between the model predictions with X_j equal to the upper limit of the bin, Z_k, and X_j equal to the lower limit of the bin, Z_{k-1}. We divide this sum by the number of observations in that bin, n_k, to obtain the average local effect of X_j on the model's output.

Now, let $k(X_j)$ represents the index of the bin where a specific value of X_j is located, with $k(X_j) = 1$ if X_j falls within the first bin, $k(X_j) = 2$ if it lies within the second bin, and so on. The "accumulated local effect" at value X_j is simply the sum of the local effects from the first bin up to the bin where X_j is located:

$$f_A(X_j) = \sum_{k=1}^{k(X_j)} \frac{1}{n_k} \sum_{i \in \mathcal{I}_k} \left[f(Z_k, \boldsymbol{X}_{\backslash j,i}) - f(Z_{k-1}, \boldsymbol{X}_{\backslash j,i}) \right]. \tag{2}$$

This sum accumulates the local average effects up to a given value of X_j. The plot of $f_A(X_j)$ as a function of X_j provides a visualization of the dependence of the on X_j across its range. The reader may notice that the values $f_A(X_j)$ initiate at zero and subsequently ascend or descend based on the sign of the average local effects. That is, $f_A(X_j)$ estimates the dependency of the model's output on X_j apart from a constant factor.

2.1.2 Calibrating the accumulated local effects

Accumulated local effects estimate the relationship between the target variable and the covariates apart from an unknown constant. In their original paper, Apley and

Zhu (2020) suggest subtracting the mean ALE value from the individual values so that the ALE plot is centered at zero,

$$f_A(X_j) \leftarrow f_A(X_j) - \frac{1}{M}\sum_{k=1}^{M} f_A(X_j). \tag{3}$$

For our objectives, this solution is not adequate as we aim to compare the relationship between option prices as provided by the model and its respective inputs. Therefore, to implement our proposed hedging method, we require the calibration of the ALE values.

Fortunately, concerning option contracts, finance theory guides us regarding the appropriate vertical positioning of the ALE values. Indeed, any reasonable option pricing model, must satisfy the following conditions. When the underlying asset price tends to zero, the call option price must tend to zero as it is written on an asset without value:

$$\text{when} \quad S \to 0, \quad C \to 0. \tag{4}$$

On the other hand, when the underlying asset price tends to infinity, the put option price must tend to zero since it will not be exercised:

$$\text{when} \;\; S \to +\infty, \;\; P \to 0. \tag{5}$$

The two equations above allow us to calibrate ALE values when we examine their variation with respect to the underlying asset price.

When the time to maturity approaches zero, the call price approaches the terminal pay-off:

$$\text{when} \;\; \tau \to 0, \quad C \to \max(S-K,0), \tag{6}$$

and likewise for the put price:

$$\text{when} \;\; \tau \to 0, \;\; P \to \max(K-S,0); \tag{7}$$

The two equations above allow us to calibrate ALE values when we examine them as a function of time to maturity.

When the volatility of the underlying asset approaches zero – and the underlying asset price does not move – the call price approaches the underlying asset price minus the discounted value of the strike price:

$$\text{when} \quad \sigma \to 0, \quad C \to \max(S - Ke^{-r\tau}, 0). \tag{8}$$

Likewise, for put options we have:

$$\text{when} \ \ \sigma \to 0, \ \ P \to \max(Ke^{-r\tau} - S, 0) \ . \tag{9}$$

These equations allow us to calibrate ALE values when we examine them as a function of the volatility of the underlying asset.

2.1.3 Calculating Greeks from ALE values

The Greeks are a set of risk metrics that measure the option price's sensitivity to shifts in the underlying asset (the Delta and Gamma), asset volatility (the Vega), and the time-value decay (the Theta). When a parametric model is being used, they are obtained from the derivatives of the pricing function with respect to its inputs. Naturally, the accuracy of the Greeks is only as good as the model used to compute them. In contrast, in our nonparametric approach there is no closed-form pricing function. What we do have is option prices – given by ALE values – evaluated at a finite number of input values. Therefore, we approximate the derivatives by finite differences. Let us assume that we have a long position on a call option written on a stock. The following relations also hold for put options. Additionally, if we have a short position on an option, the sign of expressions should be flipped.

Delta (Δ) measures how the call price changes with movements in the price of the stock. We approximate it using the first difference of the ALE values with respect to the stock price,

$$\Delta \equiv \frac{\partial C}{\partial S} \approx \frac{f_A(S_{k+1}) - f_A(S_k)}{S_{k+1} - S_k}, \ \ k = 1, \ldots, M-1. \tag{10}$$

Vega ($\mathcal{V}$) measures the risk of changes in implied volatility or the forward-looking expected volatility of the underlying asset price. We approximate it using the first difference of the ALE values with respect to the implied volatility,

$$\mathcal{V} \equiv \frac{\partial C}{\partial \sigma} \approx \frac{f_A(\sigma_{k+1}) - f_A(\sigma_k)}{\sigma_{k+1} - \sigma_k}, \ \ k = 1, \ldots, M-1. \tag{11}$$

Theta (Θ) measures the rate of change of the option price with respect to the passage of time t,

$$\Theta \equiv \frac{\partial C}{\partial t} \approx -\frac{f_A(\tau_{k+1}) - f_A(\tau_k)}{\tau_{k+1} - \tau_k}, \quad k = 1, \ldots, M-1. \tag{12}$$

As time passes, the time to maturity τ decreases and, hence, the minus sign in the first difference.

The gamma (Γ) of an option measures the rate of change of the option's delta with respect to the price of the stock. Therefore, it is the second derivative of the pricing function with respect to the stock price. We approximate the gamma by the second differences of the ALE values with respect to the stock price,

$$\Gamma \equiv \frac{\partial^2 C}{\partial S^2} \approx \frac{f_A(S_{k+1}) - 2f_A(S_k) + f_A(S_{k-1})}{(S_{k+1} - S_{k-1})/2}, \quad k = 2, \ldots, M-1\ . \tag{13}$$

The differences in (10)–(13), or their put equivalent expressions, give us first or second order difference values, that can be interpolated to obtain the general shape of the Greeks.

2.2 Nonparametric predictive model

The nonparametric model we consider is a "gradient boosting machine" (Friedman, 2001). Gradient boosting machines combine multiple base models to form a robust "committee" of models. Decision trees (Breiman et al., 1983; Quinlan, 1986) are used as the base models and consist of a series of if-then-else conditions based on the explanatory variables, which ultimately lead to a prediction. A decision tree can contain numerous branches, each with multiple sequential tests on the explanatory variables.

The prediction $\hat{Y}$ of the gradient boosting machine is obtained by summing the predictions of a set of B decision trees $\{f_b(\boldsymbol{X})\}_{b=1}^{B}$,

$$\hat{Y} = \sum_{b=1}^{B} f_b(\boldsymbol{X})\ , \tag{14}$$

where the initial tree, $f_1(\boldsymbol{X})$, is a standard decision tree trained on the original data. Subsequent decision trees, $\{f_b(\boldsymbol{X})\}_{b=2}^{B}$, are incrementally added to the committee. However, each new tree is trained on the errors produced by the trees already present in the committee. This process aims to rectify the errors made by the existing committee of trees. During each iteration, the tree to be added is determined by a gradient descent algorithm that minimizes the regularized loss function,

$$\sum_{i=1}^{n} L\left(Y_i, \hat{Y}_i^{(k-1)} + f_b(\boldsymbol{X}_i)\right) + \gamma T + \frac{1}{2}\lambda ||\boldsymbol{w}_b||^2, \tag{15}$$

where $L(\cdot)$ is the squared-error loss:

$$L\left(Y_i, \hat{Y}_i^{(b-1)} + f_b(\boldsymbol{X}_i)\right) = \left(Y_i - \hat{Y}_i^{(b-1)} - f_b(\boldsymbol{X}_i)\right)^2. \tag{16}$$

The last two terms in Equation (15) are regularization terms that aim to penalize complex trees, thereby preventing the committee from overfitting the training data. The parameter γ serves as a penalty on the number of terminal nodes in a tree, denoted by T, while λ serves as a penalty on the magnitude of the tree weights.

There are several efficient implementations of gradient boosting. A well-known implementation is eXtreme Gradient Boosting, also known as XGBoost, which is the one used here. This is probably the best 'off-the-shelf' algorithm for a wide range of predictive tasks. Indeed, about 60% of the winning solutions posted on Kaggle during 2015, and the best solutions in the KDD Cup 2015 used XGBoost (Chen and Guestrin, 2016). We optimized the "hyper-parameters" – parameters that are not learned in the training process – of our XGBoost model using grid-search[1].

[1] The grid values were: number of trees in the ensemble $\in \{100, 500, 1000, 1500\}$; maximum tree depth $\in \{4, 8, 16, 32\}$; learning rate $\in \{0.01, 0.025, 0.05, 0.075, 0.1\}$.

3 Datasets

We test the proposed hedging method using two distinct datasets: i) a simulated dataset where option prices are derived from the Black-Scholes equation, and ii) a dataset comprising real market option prices. Our objective is to show the method's performance under both a controlled simulated scenario and with real market data on options.

3.1 Black-Scholes simulated dataset

The Monte Carlo experiments focused on simulating call options on stocks and assuming that their prices are determined by the Black-Scholes formula for pricing European options. We also assumed the underlying asset does not pay dividends. Since there are no ex-dividend events during the life of the option, it is always optimal to exercise the option at expiry. In this scenario, the Black-Scholes formula can also be used to value American call options. Therefore, our simulated data can be interpreted as data from both European and American call options.

Variable	**Simulated process**
Spot price	$S \sim U(500, 1500)$
Volatility	$\sigma \sim U(0.1, 1)$
Time-to-maturity	$\tau \sim U(1/252, 2)$
Interest rate	$r \sim U(0.001, 0.05)$
Strike price	$K \sim S/z,\ z \sim N(1, 0.2)$

Table 1: Statistical distributions for generating Black-Scholes prices of non-dividend paying call options. $U(a, b)$ is the uniform distribution bounded by a and b, and $N(\mu, s^2)$ is the normal distribution with mean μ and variance s^2.

The option parameters are generated according to the processes outlined in Table 1. The data generation procedure is the following. A specified number of sets $\{S, \sigma, \tau, r\}$ is generated based on the processes described in Table 1. Then, for each set, four strike prices K are generated using the process specified in the last row of Table 1 – that is, the strike prices are constrained to be in the vicinity of the spot price S.

For each set $\{S, \sigma, \tau, r, K\}$, the call prices C_{BS} are calculated using the Black-Scholes formula. We generated a dataset with 100,000 observations. We make the assumption that the homogeneity of degree one in the spot and strike prices of the Black-Scholes formula partially holds, regardless of the true pricing model. Therefore, both spot and option prices are scaled by the strike price. That is, the machine learning models are trained to learn the relationship between C_{BS}/K and the inputs $\{S/K, \sigma, \tau, r\}$. After training, the predicted prices can be recovered in monetary units by multiplying them by the strike price.

3.2 Empirical dataset

The empirical dataset used in this study comprises information on American option contracts obtained through the Yahoo Finance API[2]. The dataset includes data for both call and put contracts. The underlying assets are stocks of four large companies: Amazon, AMD, Boeing, and Meta. The data collection period for the option prices spans from November 11, 2020, to February 12, 2021. Each option price was matched with the closest trading price of the corresponding underlying stock. None of these companies had dividend events during the options' lifespan at the time of data collection. Consequently, for call contracts, the empirical results can be directly compared to the findings from the simulation exercise. The volatility of the underlying assets was determined using the implied volatility obtained from the binomial options pricing model (Cox et al., 1979).

As in the case of the synthetic dataset, for the real market data we also consider as model outputs C/K or P/K, and as model inputs $\{S/K, \sigma, \tau, r\}$. To exclude highly illiquid contracts, the following criteria were applied. Observations with a maturity of fewer than 10 trading days until expiry or greater than 1 year were excluded. Contracts that were deep in-the-money or deep out-of-the-money were also omitted from the analysis. Finally, we only regarded implied volatilities lower than 80%. That is, the following restrictions were applied to our dataset: (i)

[2] https://pypi.org/project/yfinance/

$10/252 < \tau < 1$ years, (ii) $0.5 < S/K < 1.5$ and (iii) $0 < \sigma < 0.8$. Finally, during our sample period the term structured of interest rates proved to be rather flat, and given we are only dealing with options with up to one-year maturity, we used the 3-month Treasury Bill rate from the Federal Reserve Bank of St. Louis as a measure of the interest rate.

			Call options			
Company	Count	Spot price	Strike price	Option price	Maturity	Volatility
Amazon	14676	3215	3498	96.7	0.18	0.38
		(85)	(459)	(93)	(0.16)	(0.09)
AMD	10477	90	100	6.0	0.26	0.54
		(4)	(19)	(6)	(0.24)	(0.09)
Boeing	9278	212	246	10.9	0.29	0.52
		(13)	(46)	(12)	(0.25)	(0.10)
Meta	9941	271	304	11.3	0.27	0.41
		(9)	(51)	(13)	(0.24)	(0.09)

			Put options			
Company	Count	Spot price	Strike price	Option price	Maturity	Volatility
Amazon	11537	3225	2994	74.5	0.15	0.38
		(88)	(251)	(91)	(0.14)	(0.08)
AMD	6700	90	84	5.1	0.22	0.53
		(4)	(12)	(7)	0.22	0.09
Boeing	6050	211	192	10.4	0.24	0.53
		(13)	(28)	(13)	(0.22)	(0.09)
Meta	7692	271	245	9.5	0.25	0.43
		(8)	(30)	(13)	(0.23)	(0.10)

Table 2: Summary statistics of the dataset: number of observations; mean and standard deviation (in parenthesis) for the spot price; strike price; option price; time-to-maturity (years); and annual implied volatility. Prices are in US dollars.

Table 2 presents summary statistics for the calls and puts datasets, including the number of observations, mean spot price, mean strike price, mean option price, mean annual implied volatility, and mean time-to-maturity in years, with the appropriate standard deviations in parenthesis. All prices are denominated in US dollars. The number of observations varies across the companies, with Amazon having the highest number and Meta having the lowest. Our dataset has more calls than puts, for all underlyings. The mean annual implied volatility ranges from 0.38 to 0.54 across the companies. The mean time-to-maturity ranges from 0.15 (38 trading days) to 0.29 (73 trading days). For the call options, the mean spot prices are lower than the mean strike prices, indicating that these options are typically out-of-the-money. Conversely, for the put options the mean spot prices are greater than the mean strike prices, indicating that these options are also typically out-of-the-money.

Figure 1 complements this information presenting the empirical distributions of the inputs, for each of the underlyings. Full lines refer to data from calls, while dashed lines refer to data from puts. We observe a slight left skewness in the distribution of implied volatilities for Amazon, Boeing, and Meta, while there is no evident skewness in the case of AMD. When comparing call to put implied volatilities AMD puts present more concentrated volatilities than call, while the opposite happens in the Meta case. For Amazon and Boeing implied volatilities from calls and puts behave virtually the same. Concerning the time to maturity distributions, most traded options have a maturity of less than 3 months regardless of the company. Regrading moneyness, we observe almost “mirror” distributions for call and puts around the at-the-money region ($S/K \approx 1$). In both cases, there is a skew towards the out-of the money region. Amazon displays option contracts with more concentrated moneyness values compared to other companies.

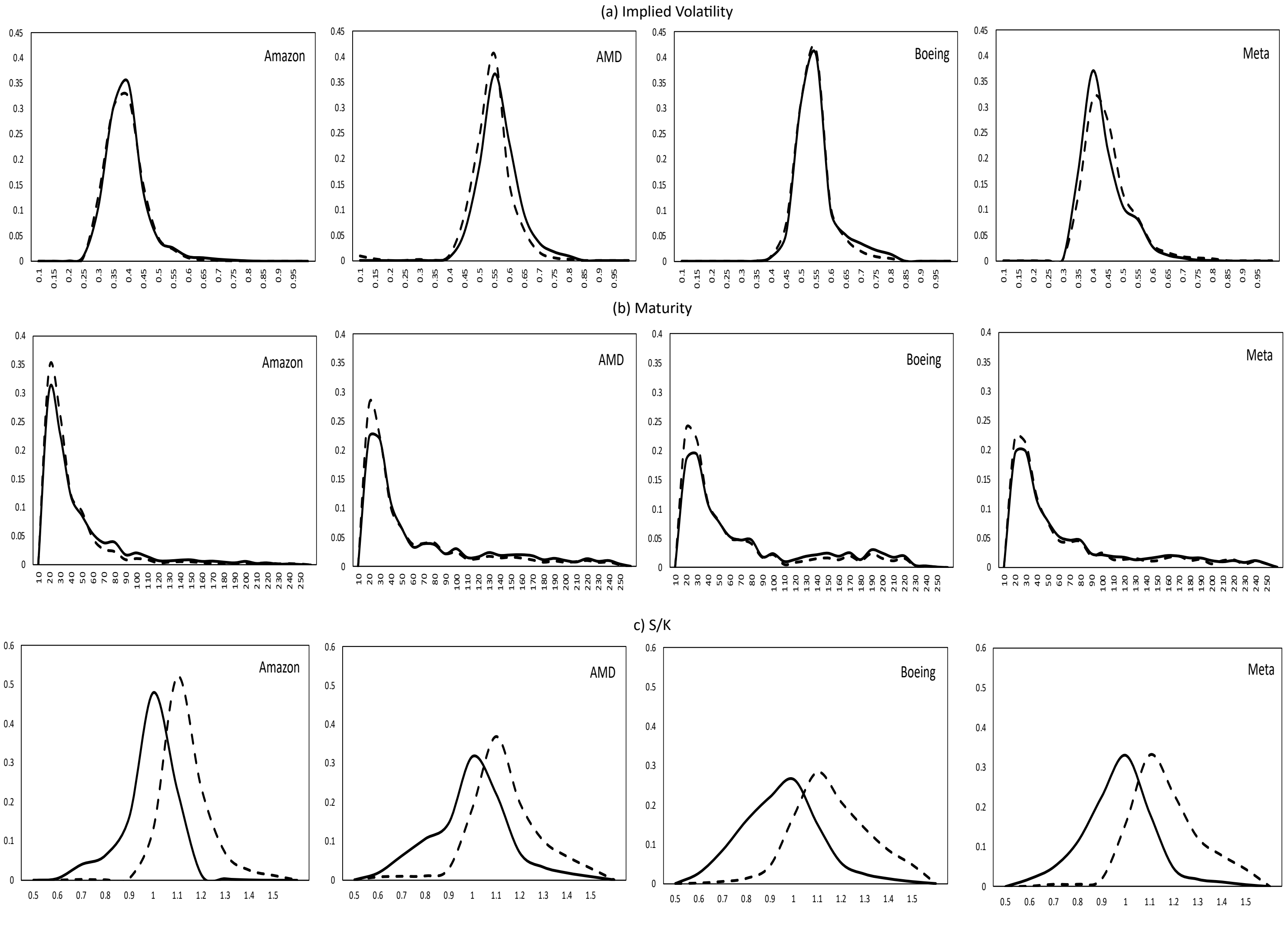

Figure 1: Distribution of inputs in the empirical dataset: (a) implied volatility σ, (b) maturity τ, (c) moneyness S/K. Full black lines refer to call options, dashed black lines refer to put options.

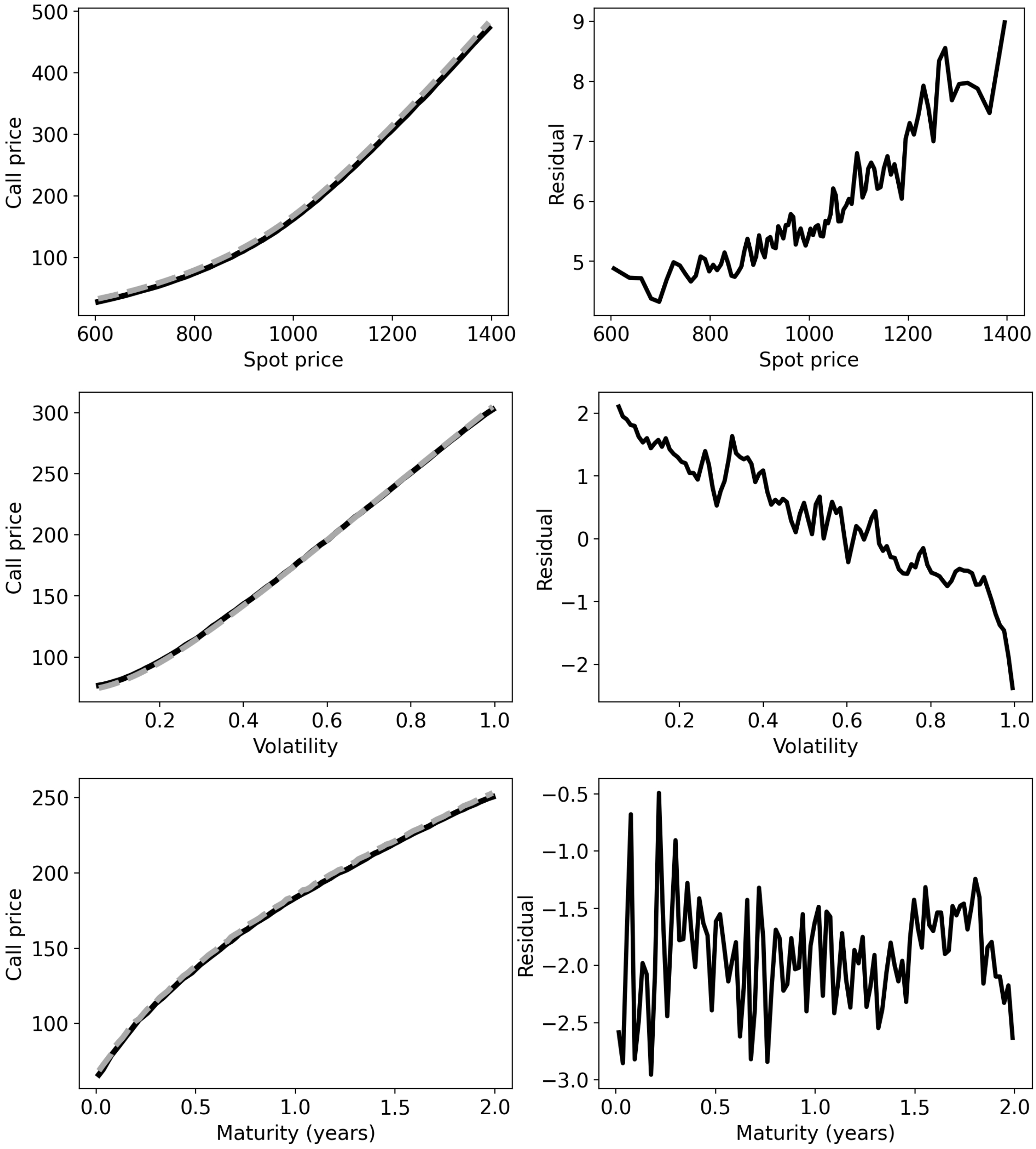

Figure 2: Simulation results. Left images: call price as a function of the spot price, volatility and time-to-maturity. The solid black line represents the dependence given by the nonparametric model; the dashed gray line gives the dependence given by the Black-Scholes model. Right images: Difference between Black-Scholes model and the nonparametric model.

4 Results

4.1 Simulation Results

Figure 2 shows the dependence of call prices in terms of spot price, volatility and maturity. On the left-hand-side images we compare the theoretical values given by the Black-Scholes model (drashed grey lines) with those obtained by the nonparametric model (solid black line). On the right-hand-side we plot the corresponding residuals. We observe that our approach perfectly replicates the prices generated by the Black-Scholes equation. Indeed, the residuals amount to only a few percent of the price levels.

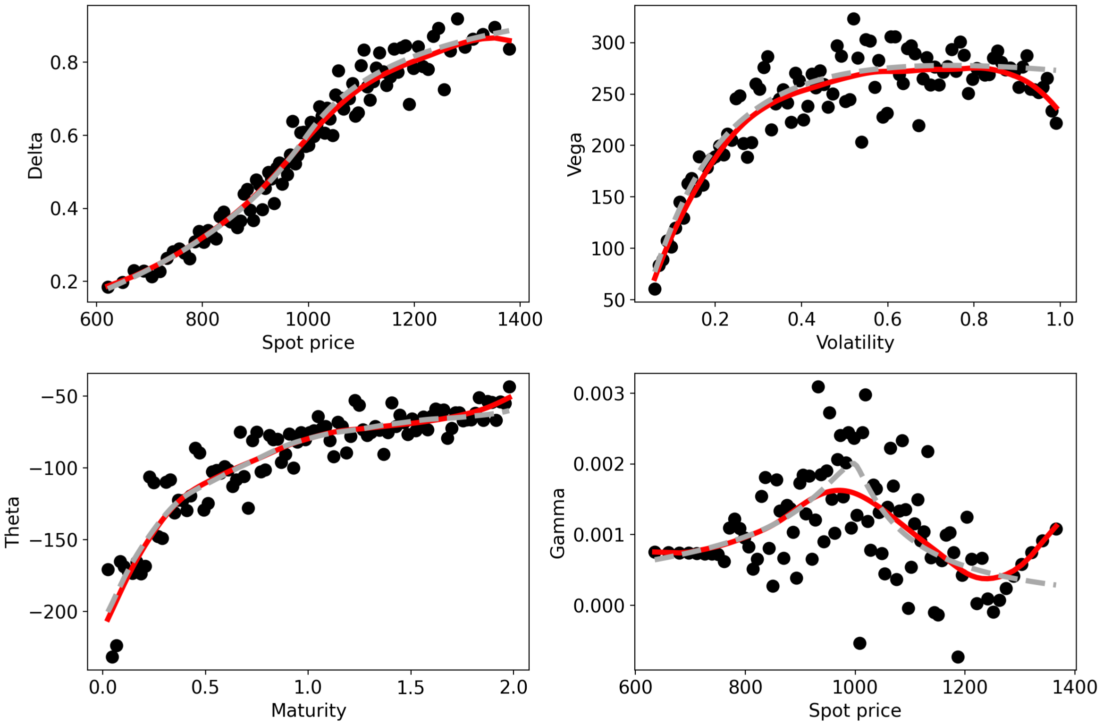

Figure 3: Simulation results. Greeks as a function of the corresponding input variable. The black dots represent ALE differences while the red line represents the smooth nonparametric sensitivity; the dashed gray line gives the dependence given by the Black-Scholes model.

Figure 3 presents the theoretical Black-Scholes Greeks (dashed grey line), together with the nonparametric ALE differences (black dots) and their LOESS smoother (red line). We can observe an almost perfect alignment between the Black-Scholes Greeks and the Greeks derived from our nonparametric approach, except in the case of Gamma. Since Gamma relies on second-order differences, it is very sensitive to sampling variability in the ALE values and the quantiles used for their calculation. Nonetheless, the nonparametric Gamma manages to capture the correct overall pattern of the theoretical Black-Scholes Gamma. This issue could be mitigated by assigning lower weight to out-of-the-money options, as their first differences with respect to the underlying asset price exhibit greater variability, as evident in the plot for the Delta.

The results above, under a totally controlled environment, are highly encouraging for our method. In the subsequent section, we evaluate its performance using real market data.

4.2 Empirical Results

Real market option prices are known to deviate from the Black-Scholes model. In particular, there is no constant volatility across different options contrasts written on the same asset. In this section, we present the results obtained for a set of American calls and puts on four individual stocks: Amazon, AMD, Boeing and Meta. Because these options may be exercised before maturity we use the binomial model as benchmark. Also, recall that the volatility parameter used for each option corresponds to the binomial implied volatility. Despite this, unlike the simulation experiment – where the Black-Scholes prices were the “true” prices – in this section the binomial model prices do not represent the “true” prices. So we compare nonparametric prices and Greeks given by our method with prices and Greeks given by the binomial model, without knowing the ground truth. To the extent that our approach is fully data driven, it is more likely to capture the “empirical truth”. Indeed, as shown below, we can explain some of the discrepancies between our approach and the binomial model by examining the distribution of returns for the stocks under analysis and comparing them to equivalent Gaussian returns.

In Figure 4, the black solid lines show the empirical return distributions of the

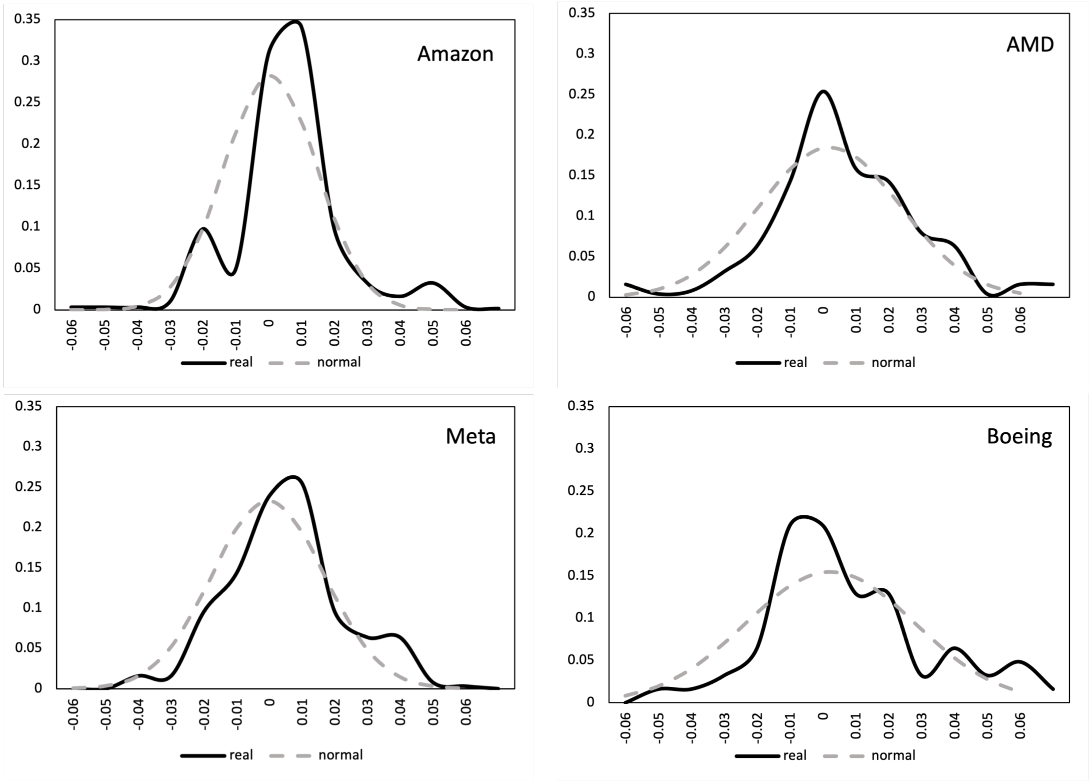

Figure 4: Distributions of daily stock returns during the period covered by the data. Black solid lines represents the empirical return distributions. Gray dashed lines illustrate the Gaussian return distributions (with means and standard deviations equal to the respective sample values).

four stocks under analysis. The gray dashed lines illustrate the Gaussian return distributions with means and standard deviations equal to the respective sample values. It is clear from Figure 4 that the empirical return distributions of the four stocks diverge from the Gaussian distribution assumed by both the binomial and Black-Scholes models. Although, the skewness and kurtosis differ across stock, there seems to be a slight skew to the right in the returns of Amazon, AMD and Meta, while Boeing presents left skewness. Despite this, all stocks present heavier right tails. Even though skewness and kurtosis differ across stock returns, Amazon, AMD, and Meta returns have a slight right skew, while Boeing's returns exhibit left skewness. Nevertheless, all stocks have heavier right tails.

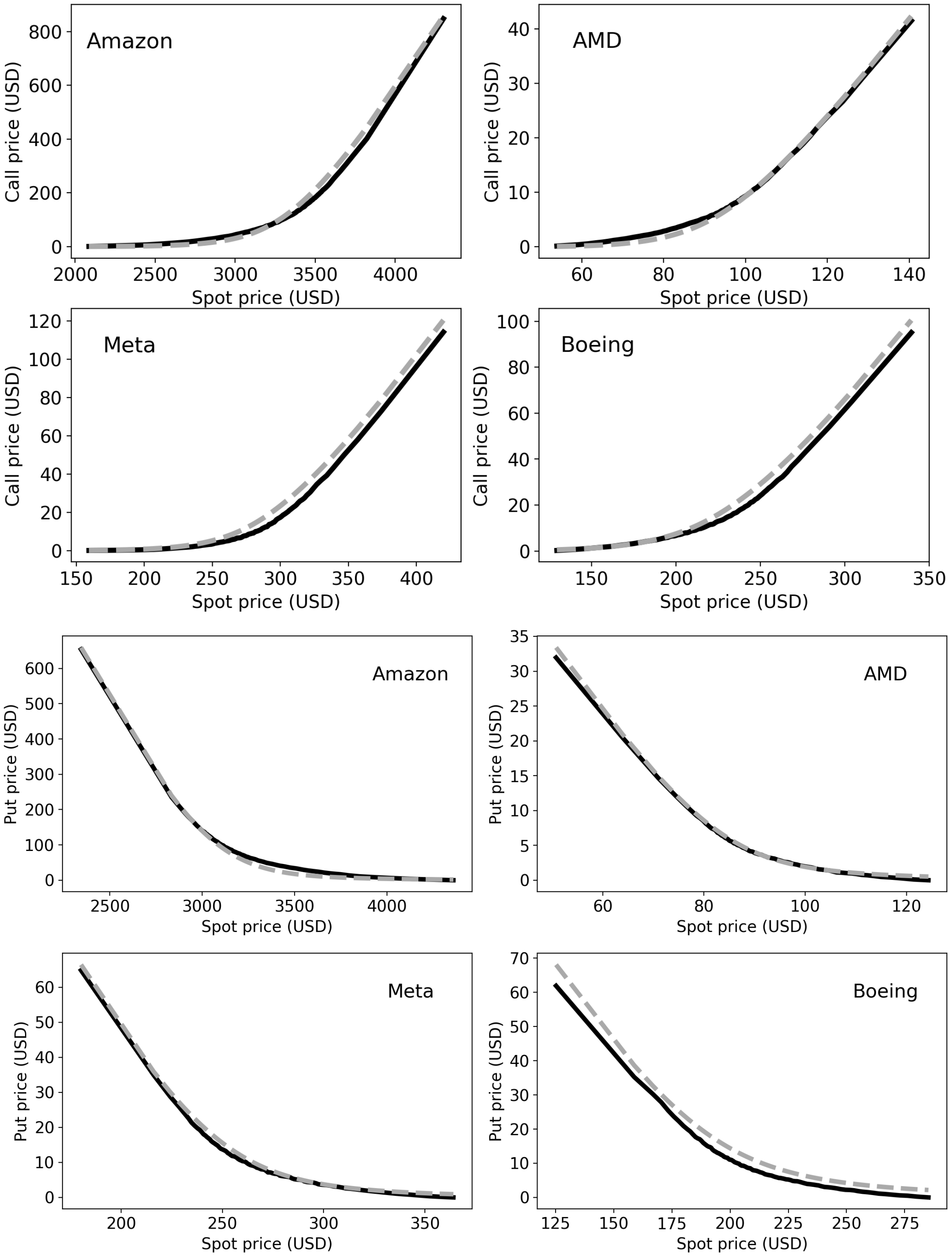

Figure 5: American call and put prices as a function of the spot price. The solid black line represents the dependence given by the machine learning model; the dashed gray line gives the dependence given by the binomial model.

4.2.1 Results for options prices

Figures 5 to 7 present eight plots each. In these figures, the top four plots display the results for call options (one for each company), while the bottom four plots depict results for put options (also one for each company). The black solid lines present results given by our nonparametric method, while the gray dashed lines depict results for the binomial models. Figure 5 indicates that for Meta and Boeing, the nonparametric call prices tend to be slightly lower than those of the binomial model for higher stock price values. The nonparametric method also suggests lower prices for put options written on Boeing stock. Overall, there are no substantial discrepancies between the two pricing methods.

Figure 6 illustrates how call and put prices change as a function of the volatility. In this case, we observe substantial discrepancies between our nonparametric method and the binomial model. Two consistent effects are observed across both call and put options for all four underlying stocks. The first effect is that nonparametric prices rise with volatility but consistently remain below those provided by the binomial model. While "empirical volatility" is hard to quantify, empirical returns are more likely to cluster around the center of their distribution than normally distributed returns with the same variance, as shown in Figure 4. Therefore, our speculation is that the market does not price correctly the fat tails or the likelihood of extreme returns. The second effect is that an S-shaped pattern emerges in the way our nonparametric method models how market prices change with volatility. That is, the variation in option prices with volatility is less pronounced for "uncommon" volatility values. This S-shaped pattern might be explained by the saying that "stock markets go up by escalator and go down by elevator." In other words, higher volatilities often coincide with lower stock prices, implying that high volatility call options might not hold as much value. Similarly, this holds true for puts, albeit for a different reason. When stock prices are low, put options become deep in-the-money. However, facing, on average, higher volatility implies increased chances of ending out-of-the-money, thereby reducing the value of put options as well.

Figure 7 illustrates how call and put prices depend on time-to-maturity. Here,

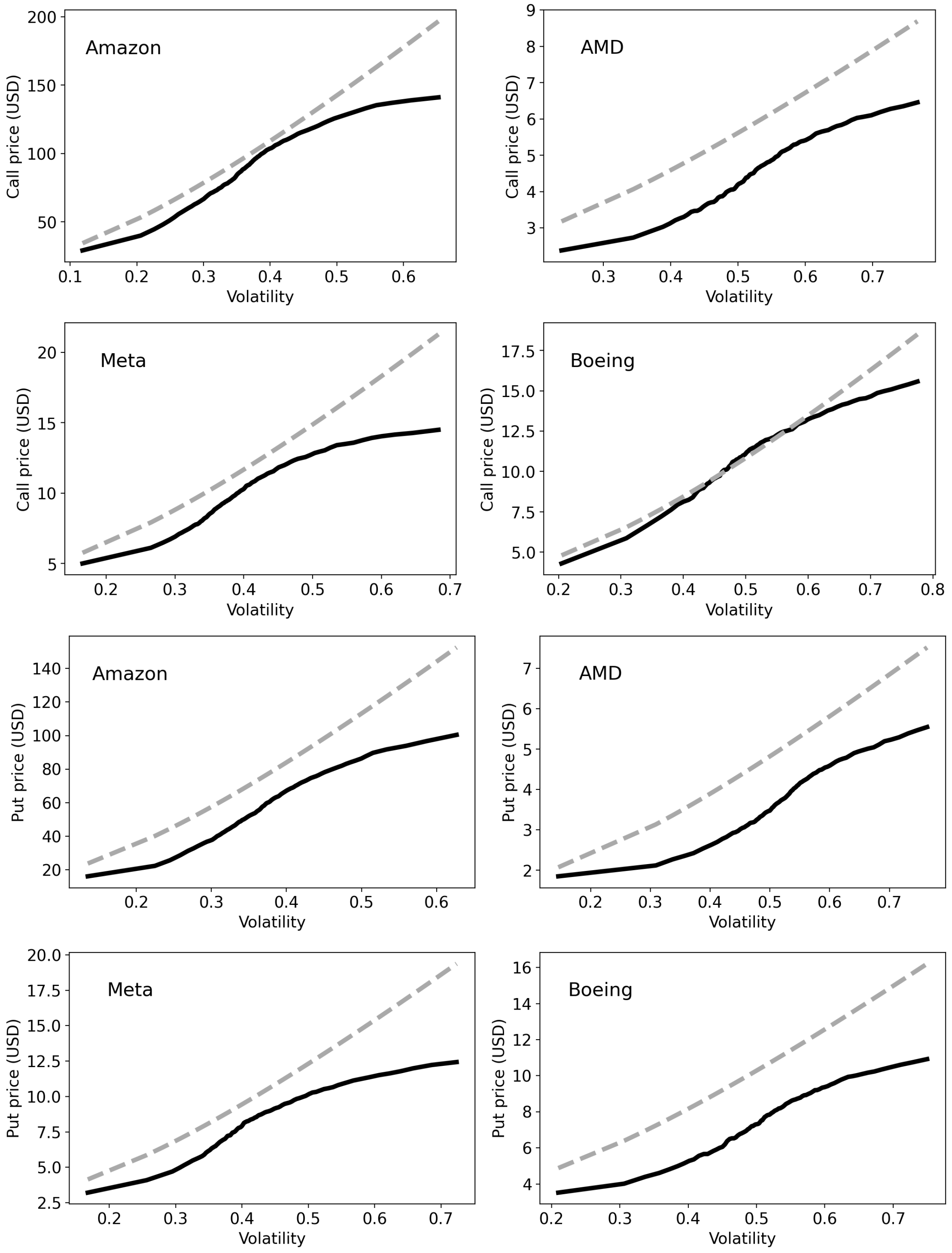

Figure 6: American call and put prices as a function of the volatility. The solid black line represents the dependence given by the nonparametric method; the dashed gray line gives the dependence given by the binomial model.

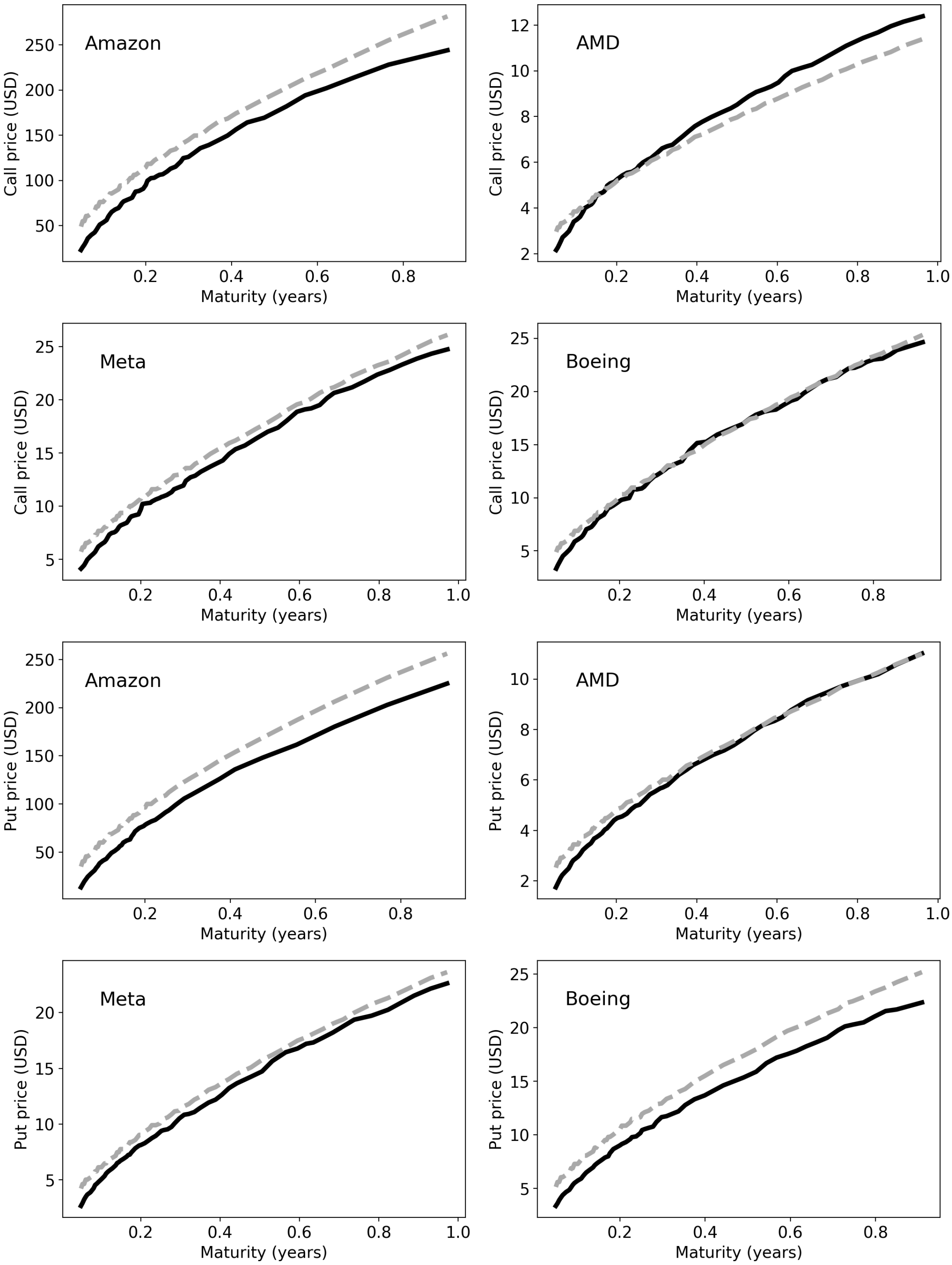

Figure 7: American call and put prices as a function of the time-to-maturity. The solid black line represents the dependence given by the nonparametric method; the dashed gray line gives the dependence given by the binomial model.

we notice that the shape of this dependency is consistent across both models. With the exception of call options written on AMD and Boeing stocks and put options for AMD stocks, where there's almost a perfect match, the nonparametric method indicates slightly lower prices for options across all maturities.

4.2.2 Results for options Greeks

Next, we analyze the results for the option Greeks, derived from the first and second differences of the options prices. Again, the figures present eight plots each – the top four plots display the results for call options (one for each company), while the bottom four plots depict results for put options (also one for each company). The black dots present results given by our nonparametric method, while the gray dashed lines depict results for the binomial models. The red line represents the smoothed nonparametric Greeks obtained via a LOESS regression.

Figure 8 shows how the Delta depends on the underlying asset price. For call options the nonparametric Delta has a more pronounced S-shaped pattern than the Delta given by the binomial model. This is attributed to the lower Delta values around the at-the-money region, rather than substantial deviations for the in-the-money or out-of-the-money options. On the other hand, regarding put options, there is substantial agreement between the two pricing methods. The observed differences to the binomial model can be explained by referencing the empirical distribution of returns in Figure 4, alongside the statistics of our options sample in Figure 1. Figure 4 shows that the empirical returns around or slightly above the mean are more common than what a Gaussian distribution would suggest, irrespective of the stock. This accounts for the more pronounced S-shaped patterns observed for delta. The disparities between the nonparametric and the binomial Deltas are particularly noticeable for calls. This is because our calls sample is moderately biased toward out-of-the-money options, with higher S values. Conversely, for put options, our sample is biased toward lower values of S, as depicted in Figure 1 (c). Finally, there is greater variability in the individual Delta values for option written on Meta and Boeing stocks. This may be attributed to the dispersion of the sample moneyness. In-the-money options are relatively less traded,

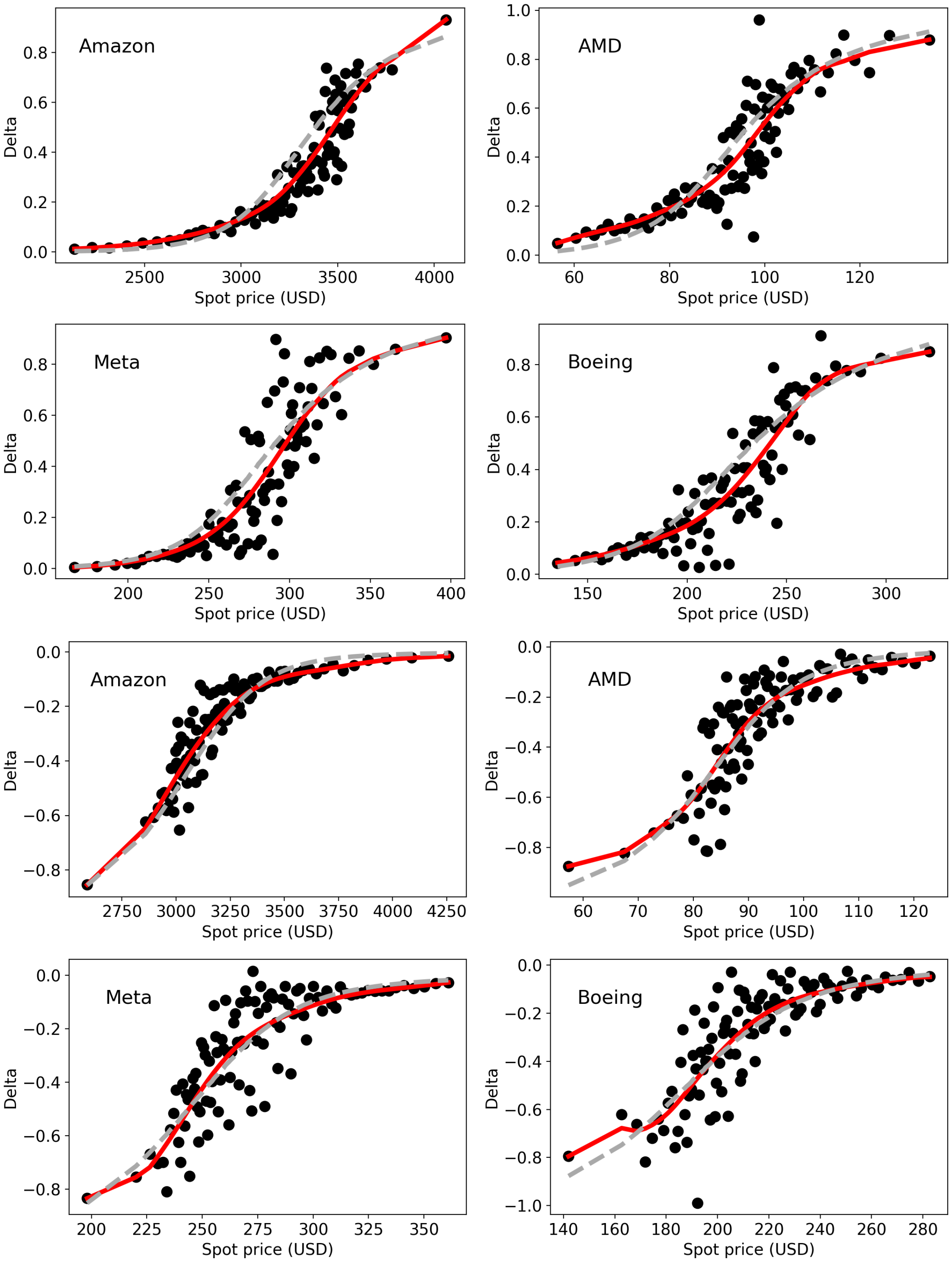

Figure 8: Delta for American calls and puts as a function of the spot price. The black dots represent ALE differences while the red line represents the smoothed nonparametric sensitivity. The dashed gray line gives the dependence given by the binomial model.

resulting in fewer instances of high S values for calls or low S values for puts. This is a result of the sample dispersion in terms of moneyness. ITM options are relatively less traded which imply less occurrences of high S values for calls or low S values for puts.

In Figure 9 we show how the Vega depends on the volatility of the underlying asset. Unlike the Vega provided by the binomial model, which increases monotonically, the Vega derived from our nonparametric method displays a hump-shaped pattern for all stocks, both for calls and puts. The hump-shaped pattern of the nonparametric Vega is somewhat expected, given the S-shaped pattern in Figure 6. Of course, the hump-shaped pattern observed in the nonparametric Vega aligns with the S-shaped pattern in Figure 6.

Figure 10 shows how the Theta depends on time-to-maturity. The nonparametric and binomial Theta values are relatively close to each other for nearly all stocks, including both calls and puts. The exception lies in calls written on AMD stocks, where the nonparametric Theta values are lower compared to the binomial Theta values. However, there is a degree of convergence as time-to-maturity increases. The increased dispersion in individual Theta values for short-maturity options might be attributed to "pin risk" (Golez and Jackwerth, 2012). This refers to the difficulty in pricing or hedging options that are near expiration, especially those that are at-the-money.

Finally, Figure 11 shows how the Gamma depends on the underlying asset price. Again, we note that since Gamma relies on second-order differences, it is more sensitive to sampling variability in the ALE values and the quantiles used for their calculation. Despite this, the LOESS smoother generates plausible results considering the limited observations in our dataset, closely approximating the Gamma values provided by the binomial model in most cases. It is important to note that we do not anticipate significant differences between these alternative pricing models here, as they provide a similar relationship between option prices and stock prices in Figure 5. Thus, the unexpected behavior of the nonparametric Gamma for low stock price values might be a statistical artifact resulting from the low number of in-the-money put options.

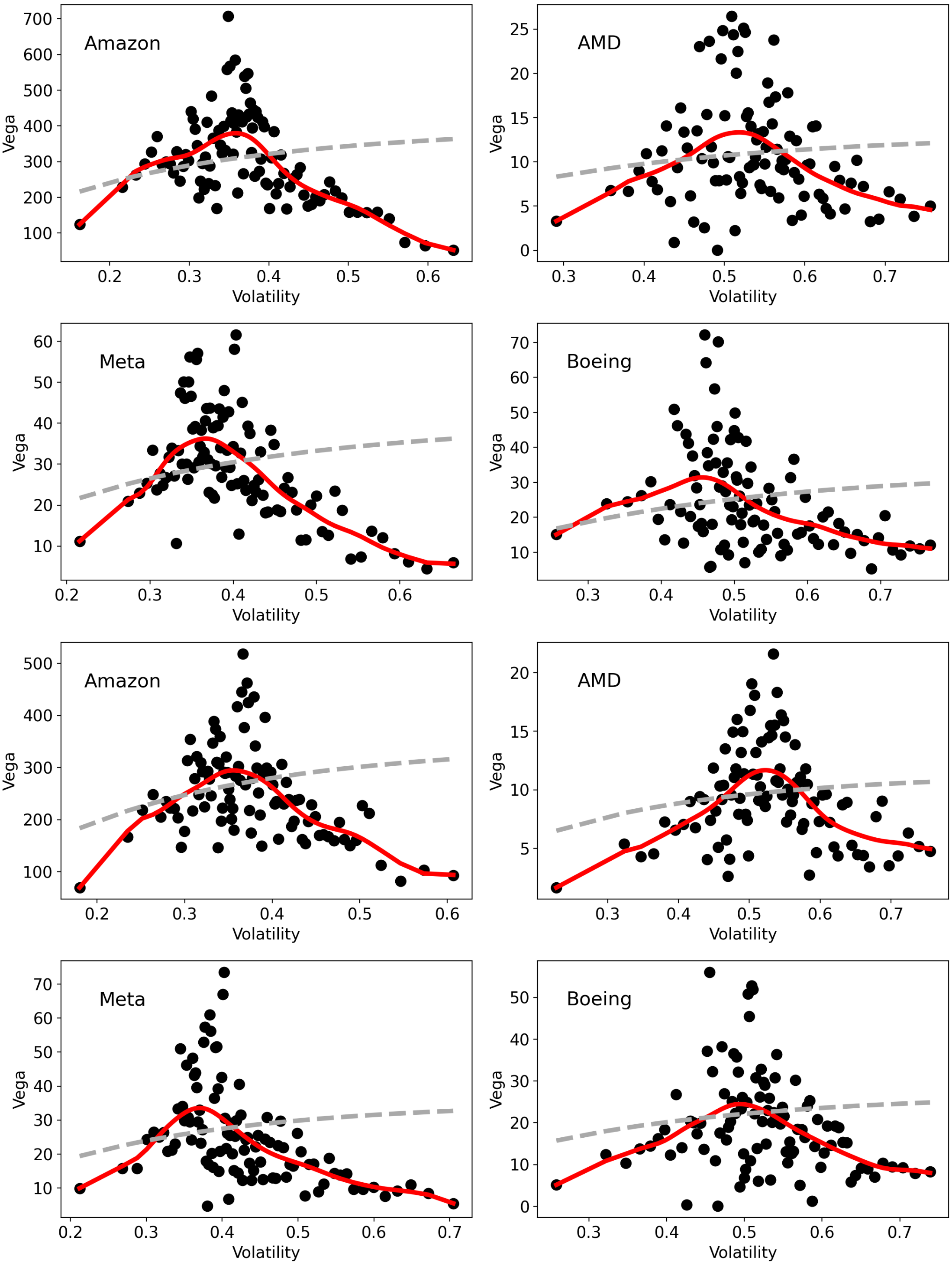

Figure 9: Vega for American calls and puts as a function of the volatility. The black dots represent ALE differences while the red line represents the smoothed nonparametric sensitivity. The dashed gray line gives the dependence given by the binomial model.

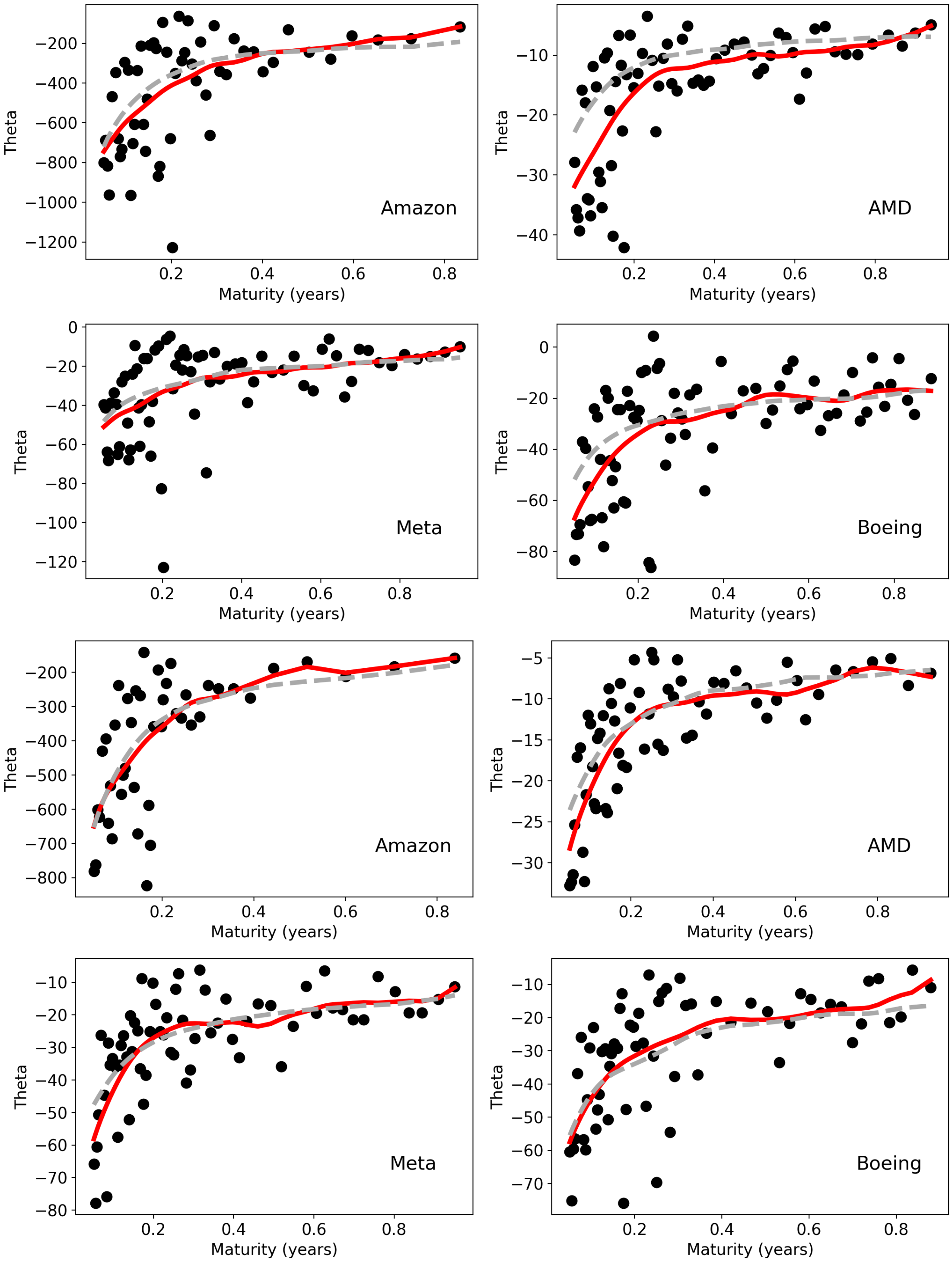

Figure 10: Theta for American calls and puts as a function of the volatility. The black dots represent ALE differences while the red line represents the smoothed nonparametric sensitivity. The dashed gray line gives the dependence given by the binomial model.

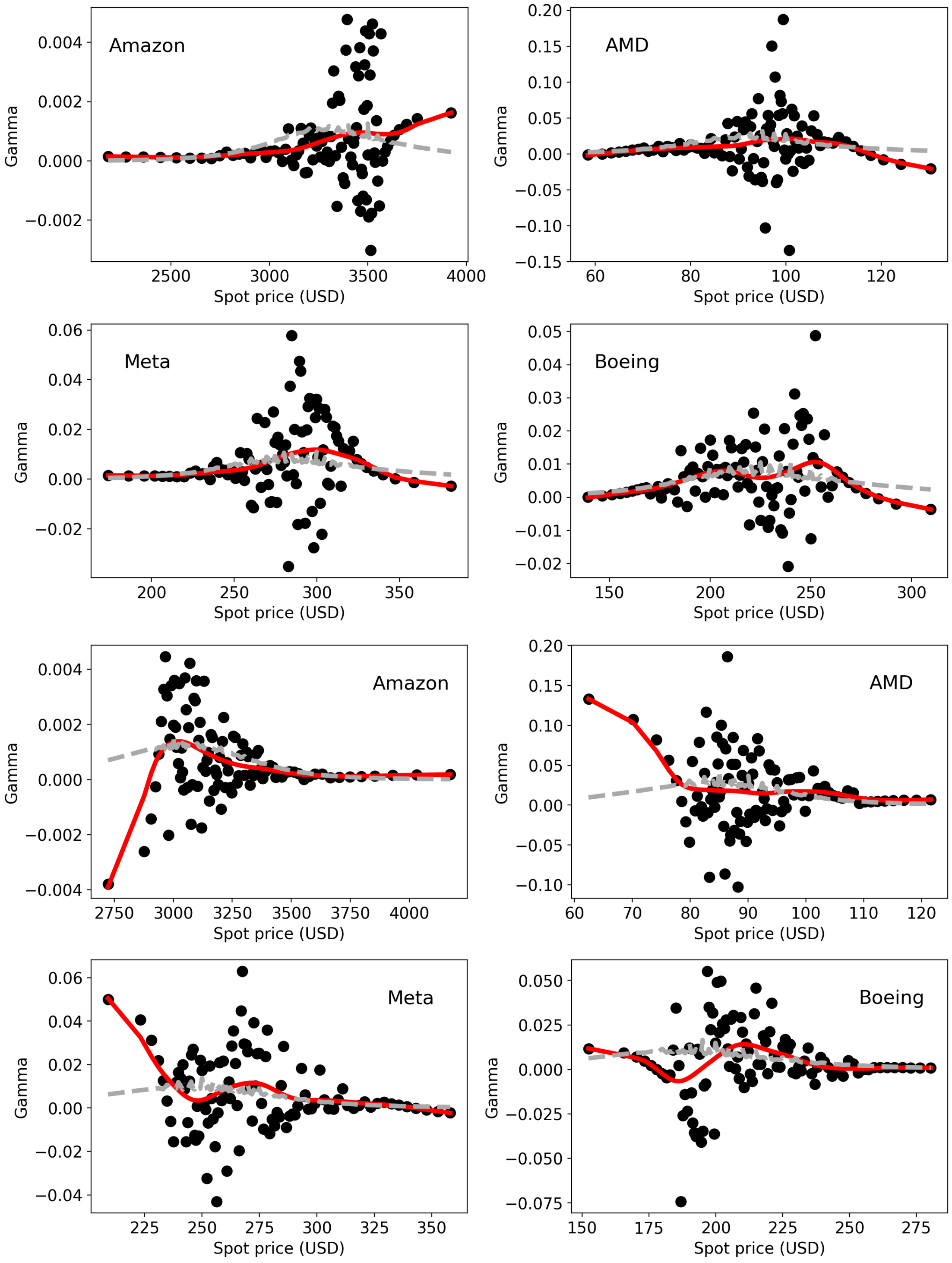

Figure 11: Gamma for American calls and puts as a function of the spot price. The black dots represent ALE differences while the red line represents the smooth nonparametric sensitivity. The dashed gray line gives the dependence given by the binomial model.

5 Conclusion

Our paper introduces a groundbreaking non-parametric method for option hedging that extends its application to American options. Our hedging method is based upon ALE values and contributes to the literature by being the first non-parametric approach to hedging. Here we take the stand that deep neural-networks are parametric models once their architecture is defined.

The application of our approach to two distinct datasets provides a comprehensive evaluation of its performance. The use of simulated Black-Scholes data establishes a strong foundation, demonstrating the method's proficiency under controlled conditions. Furthermore, the empirical dataset, featuring options from high-profile stocks such as Amazon, AMD, Meta, and Boeing, serves as a real-world litmus test, affirming the robustness and adaptability of our non-parametric approach.

As we use a machine learning XGBoost model to calibrate both datasets, our results also showcase the effectiveness these type of models in pricing models as previously suggested by Ivaşcu (2011). The model's ability to navigate the intricacies of American options, combined with its success in a diverse set of real-world scenarios, underscores its practical utility for financial practitioners.

In conclusion, our research marks a significant step forward in the realm of option hedging methodologies and proposes a alternative, *vis a vis* what seems to be the default machine learning approach in finance: the deep neural networks. markets.

However, there are still challenges to overcome, such as the need for large amounts of data and the potential for overfitting. Further research is needed to address these challenges and explore the potential of machine learning algorithms in pricing and hedging options.

References

Anders, U., O. Korn, and C. Schmitt (1998). Improving the pricing of options: A neural network approach. *Journal of forecasting 17*(5-6), 369–388.

Andreou, P. C., C. Charalambous, and S. H. Martzoukos (2010). Generalized parameter functions for option pricing. *Journal of banking & finance 34*(3), 633–646.

Apley, D. W. and J. Zhu (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society. Series B: Statistical Methodology 82*(4), 1059–1086.

Black, F. and M. Scholes (1973). The pricing of options and corporate liabilities. *Journal of political economy 81*(3), 637–654.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32.

Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1983). *Classification and Regression Trees*. Wadsworth, Belment CA.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Cox, J. C. and S. A. Ross (1976). The valuation of options for alternative stochastic processes. *Journal of financial economics 3*(1-2), 145–166.

Cox, J. C., S. A. Ross, and M. Rubinstein (1979). Option pricing: A simplified approach. *Journal of financial Economics 7*(3), 229–263.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Garcia, R. and R. Gençay (2000). Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics 94*(1-2), 93–115.

Goldstein, A., A. Kapelner, J. Bleich, and E. Pitkin (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *journal of Computational and Graphical Statistics 24*(1), 44–65.

Golez, B. and J. C. Jackwerth (2012). Pinning in the s&p 500 futures. *Journal of Financial Economics 106*(3), 566–585.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The review of financial studies 6*(2), 327–343.

Hutchinson, J. M., A. W. Lo, and T. Poggio (1994). A nonparametric approach to pricing and hedging derivative securities via learning networks. *The journal of Finance 49*(3), 851–889.

Ivaşcu, C.-F. (2011). Option pricing using machine learning. *Expert Systems with Applications 163*, 113799.

Malliaris, M. and L. Salchenberger (1993). A neural network model for estimating option prices. *Applied Intelligence 3*, 193–206.

Park, H., N. Kim, and J. Lee (2011). Parametric models and non-parametric machine learning models for predicting option prices: Empirical comparison study over kospi 200 index options. *Expert Systems with Applications 41*(11), 5227–5237.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning 1*, 81–106.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature 323*(6088), 533–536.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory.* New York: Springer-Verlag.

Wang, P. (2011). Pricing currency options with support vector regression and stochastic volatility model with jumps. *Expert Systems with Applications 38*(1), 1–7.