Check for updates

Abstract and Applied Analysis

WILEY

RESEARCH ARTICLE OPEN ACCESS

Data-Driven and Numerical Approaches With PDEs for Option Pricing

Nageena K. Frost | Choi-Hong Lai

School of Computing and Mathematical Sciences, University of Greenwich Old Royal Naval College, Park Row, London, SE10 9LS, UK

Correspondence: Nageena K. Frost (n.k.frost@greenwich.ac.uk)

Received: 14 April 2026 | **Revised:** 18 August 2026 | **Accepted:** 24 September 2026

Academic Editor: Mohammad Safi Ullah

Keywords: Black–Scholes equation | European options | historical volatility | Laplace transform | machine learning | pricing model | Thomas algorithm

ABSTRACT

This study examines the performance of various numerical and machine learning (ML) approaches in option pricing. The Black–Scholes equation is solved via finite difference (FD) methods and Laplace transform (LT), and its numerical inversions then used to price European options where the volatility of the underlying asset is calculated through the statistical measure of stock markets' historical data. In data driven approaches, calibrated via ML, some conventional and deep learning-based approaches are reviewed by utilising 50ETF European stock index options data. The efficiency and accuracy of the adapted methods is demonstrated by computing the conventional metrics for benchmark options. The results reveal that the numerical inversion techniques of LT with a direct solver such as Thomas algorithm are more efficient in pricing European options as compared to the classical time marching methods. When the dataset is sufficiently large the omission of some input parameters in a pricing model can still generate relatively accurate results using ML techniques.

1 | Introduction

Options are widely used by investors and traders in speculative investments or managing market risks through portfolio hedging. Finding the fair value of an option, or in other words, option's expected return, is a well-studied literature area contributing to a wide range of pricing models, operating under certain assumptions or economic settings. Financial markets react quickly to new data. Processing large assets may require extreme efficiency, performing enormous updates as quickly as a fraction of a millisecond. Accuracy also remains a crucial factor in measuring portfolio sensitivity to market changes. While analytical methods such as the closed-form Black–Scholes formula offer the efficient exact solution, they pose limitations on model parameters such as the assumption of constant volatility, restricting their ability to capture the real markets' dynamic behaviour. Researchers seeking efficiency and accuracy in option-pricing techniques often resort to numerical approaches which have also evolved to offer multiple variations across different use cases. For example, the explicit finite difference (FD) method provides an efficient but numerically unstable solution, whereas the implicit and Crank–Nicolson (C–N) FD methods offer stable but computationally expensive solutions [1, 2]. Seeking advanced pricing methods which are efficient and accurate and can contribute to minimising mispricing is the primary focus of this work.

Numerical solution of the partial differential equation (PDE) arising in the modelling of option pricing is proposed by Khodayari and Ranjbar [3], comprising a combination of radial basis functions (RBFs) and FD to approximate the spatial and time derivatives. The method is reported to be very fast and effective in approximating the exact solution of multi-asset European options. A combination of C—N and sextic B-spline in pricing European options and the use of a redefined cubic B-spline in

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Copyright © 2026 Nageena K. Frost and Choi-Hong Lai. *Abstract and Applied Analysis* published by John Wiley & Sons Ltd.

pricing Asian options have been reported to provide better results in terms of stability and convergence as compared to some of the existing B-spline-based methods [4, 5]. On the other hand, the use of a high-order compact FD method has been observed to provide the same rate of convergence as those obtained theoretically in pricing European options [6]. Analytical and numerical Laplace transform (LT) methods have also been known to provide efficient alternatives to standard FD in option pricing [7]. Use of the generalised finite integration method with the LT technique for pricing European options is reported to show good stability with minimal computational effort [8]. However, the accuracy of this approach remains questionable due to the slowness of the technique adapted in numerical inversion of LT. Their work proposes further research into developing a more elegant Laplace inversion technique. Applying LT to the Black–Scholes model converts the complex, time-dependent PDE to a simpler ordinary differential equation (ODE) in Laplace space, providing various computational and analytical advantages. Scholarly research also reports other techniques which can facilitate the derivation of an exact analytical solution with the conversion of nonlinear PDE to ODE [9] or an algebraic system. These techniques include the use of mapping methods such as wave transformation, multivariate generalised exponential rational integral function method, generalised projective Riccati equation method and creation of a semi-analytical framework through a differential transform method [10–12]. These have wide-ranging applicability in real physical systems and may also be suitable in simulating the stochastic processes in risk management and non-linear option-pricing frameworks.

One of the key parameters for pricing a financial asset is its volatility, which determines the riskiness of the asset in terms of variation in expected returns. Considering asset volatility to be constant is an underlying assumption of the Black–Scholes model. On the contrary, techniques that consider a time-varying estimate of the volatility have proved successful in providing more realistic estimates. The tractable option pricing model with time-varying risk aversion based on a flexible pricing kernel and the Heston–Nandi GARCH model has been reported to be highly accurate in reducing the root mean squared error (RMSE) of pricing errors by more than 50% [13]. A key advantage of their method over traditional stochastic volatility models is its ease of estimation. Neural networks have also been reported as an emerging approach for forecasting daily stock market volatility in risk management applications [14]. The current study reviews two historical volatility (HV) models applicable to predicting stock market volatility using historical data. In the results section of the IBM options example, this HV is used as the volatility input in the FD and LT models to determine the option prices.

In terms of accuracy, data-driven approaches utilising machine learning (ML) algorithms for option pricing are also becoming a vastly researched area due to technology empowered parameter calibration using data rather than assumed statistical and economical norms. A feed-forward deep learning neural network trained on simulated market data can be employed to price the options with very low errors and generalised to real-world problems with actual market data [15]. Other deep learning-based approaches such as long short-term memory (LSTM) have been reported as highly effective methods to learn financial time-series data [16]. A range of standard ML models including support vector machine (SVM), random forest and neural networks explored by Ivaşcu [17] have been shown to outperform the classical Black–Scholes model by a great margin, with best forecasting performance achieved with boosting models XGBoost and LightGBM. The performance enhancement of LSTM via realised skewness has shown to produce the best overall results compared with standard ML and benchmark models, specifically in moneyness states [18].

Although a considerable amount of scholarly research exists in the theory of option pricing, continued advancements in this area suggest that there is still a need to investigate methods which could improve predictive analysis for efficiency and accuracy. The work presented in this study proposes a new framework via a combination of numerical LT, its inversion and direct solvers, such as the Thomas algorithm or tridiagonal matrix algorithm (TDMA), which is highly efficient and stable. Some of the existing approaches around numerical LT involve a hybrid LT and FD method with an iterative algorithm based on smooth conditioning [19]. The iterative solvers excel in time-dependent simulations; however, with reduction into a stationary boundary-value problem through LT, the direct solvers eliminate time-stepping errors and offer cost-effective one-time factorisation enabling robust and high accuracy computation. The current work also provides a comprehensive comparison among inversion methods, including the ones known to be fast and stable, with fewer evaluations than just exploring the commonly favoured approach of Gaver–Stehfest [7, 19, 20]. In option pricing, Ahn et al. [21] acknowledge the use of TDMA for LU decomposition of a tridiagonal systema, but their proposed framework primarily focuses on a flexible inverse transform based on the quotient-difference algorithm. Some other scholarly studies [22] also report the use of TDMA in solving linear complementary problems of the Black–Scholes model but are structured around FD discretisation rather than LT and its inversion. Another novelty of the current work is the diverse application areas it examines under the proposed framework, including real stock markets, as well as the comparison of contemporary methods surrounding data ubiquity. Alongside providing an effective mechanism for managing extensive workloads, the suggested numerical framework can also support simultaneous computations across multiple networks, thus enabling parallel processing in a distributed computational environment. The ability of ML models to learn complex relationships from financial market data and outperform the restrictive assumptions-based traditional model in reducing pricing errors puts them at the forefront of popular choices. This work also highlights how the selected ML models may remain highly effective in producing relatively accurate results despite the omission of some important input parameters.

This work provides a background review of numerical techniques including FD and LT and its inversion and data-driven approaches for price predictions with focus on the concrete analysis of European call options. The presented analysis can be extended to other types of options with suitable adjustments to the equations and boundary conditions. The layout of the work is organised to include section II with the theory and approximation for the European call option by using the classical Black–Scholes model. Section III provides a description of the methodology including numerical solution methods; (i) FD

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

method with Euler forward time marching as the simplest and the quickest scheme, (ii) LT, its various inversions and an algorithm for the chosen direct solver method, (iii) some measures of HV and (iv) an outline of selective ML models to suffice for data-driven approaches. In Section 4, the effectiveness of the LT method using real stock market data is tested, and the performance of ML models trained on 50ETF options is examined using different metrics. Discussions on the effectiveness of the numerical and data-driven techniques are also included in this section. Finally, the key research findings are presented in the Conclusion section.

2 | The Model

A European option with price $v(s, t)$ satisfies the following second-order PDE.

$$\frac{\partial v}{\partial t}+\frac{1}{2}\sigma^2S^2\frac{\partial^2 v}{\partial S^2}+rS\frac{\partial v}{\partial S}-rv=0, S\in[0,\infty]. \tag{1}$$

With S the underlying asset price, σ the volatility, r the interest rate and t the time, Equation (1) represents the Black–Scholes equation of asset price movement. Both σ and r are assumed to be constant here. A European call option is a financial contract which gives its holder the right to buy the underlying asset at a fixed price, known as a strike price, at a prescribed time in the future, known as the expiry date.

In order to determine $v(s, t)$ uniquely from Equation (1) it is desirable to impose the boundary conditions in the domain $0\le S, 0\le t\le T$ (T the expiry date) defining the problem, in this case, European call option. This leads to the following formula

$$v(0,t)=0,\ v(S_\infty,t)=S_\infty-ke^{-r(T-t)},$$

where k is the strike price. The difficulty of S being on an infinite domain is usually resolved by choosing $S\in[0,L]$ where L is assumed according to the widely used criterion [4] as follows:

$$L=\max\left(Rk,\ ke^{\sqrt{2T\sigma^2\ln 100}}\right),$$

where R is chosen with the rule of thumb as 'three or four times the strike price'. The boundary conditions for a European call thus are as follows:

$$v(0,t)=0,\ v(L,t)=L-ke^{-r(T-t)}. \tag{2}$$

Domain truncation theorem: Let v_∞ denote the exact Black–Scholes solution on the infinite domain $(0, \infty)$, and v_L the solution obtained on $(0, L)$ with boundary condition

$$v(L,t)=L-ke^{-r(T-t)}.$$

Then,

$$||v_\infty-v_L||\le Ce^{-cL},$$

for positive constants C and c. Consequently,

$$v_L\to v_\infty,$$

as

$$L\to\infty.$$

Therefore, the truncation error introduced by selecting L such as three times the strike price is exponentially small and does not affect the overall convergence of the numerical scheme used for the solution approximation.

From the definition of the option type, the value of the option at expiry is the payoff of the option. Hence, the terminal or final condition is defined as follows:

$$v(S,T)=\max(S-k,0), \tag{3}$$

order to transform the model into initial time condition form, the time to maturity variable $\tau=T-t$ is introduced. Defining

$$V(S,\tau)=v(S,\ T-t),$$

the option price is re-expressed in terms of the remaining time to maturity. By the chain rule

$$\frac{\partial v}{\partial t}=-\frac{\partial V}{\partial \tau},$$

while the spatial derivatives remain unchanged. Equations (1)–(3) then take the form

$$\frac{\partial V}{\partial \tau}=\frac{1}{2}\sigma^2S^2\frac{\partial^2 V}{\partial S^2}+rS\frac{\partial V}{\partial S}-rV, \tag{4}$$

subject to initial condition

$$V(S,0)=\max(S-k,0), \tag{5}$$

and boundary conditions

$$V(0,\tau)=0,\ V(L,\tau)=L-ke^{-r\tau}. \tag{6}$$

3 | Methodology

An analytical or exact solution to Equations (4)–(6) can be obtained by using the Black–Scholes formula. This section presents two numerical approaches to compute the approximate solution: FD and LT and their inversion.

3.1 | FD

The key step underlying the FD method is to replace the differential operators occurring in Equation (4) with the difference operators in the domain $0\le x\le$ and $0\le t\le\tau$. Using the Euler forward method, which approximates the time derivative with forward difference and space (price) derivative with central difference, Equation (4) in the discretized form can be expressed as

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

follows:

$$V_i^{j+1} = \frac{\Delta t}{2}(\sigma^2 i^2 + ri)V_{i+1}^j + (1 - \Delta t(\sigma^2 i^2 + r))V_i^j + \frac{\Delta t}{2}(\sigma^2 i^2 - ri)V_{i-1}^j, \tag{7}$$

where $0 \le i \le N$, $0 \le j \le M$. N and M represent the number of equally spaced points on space and time axes respectively.

Rewriting Equation (7) in a simplified form as follows:

$$V_i^{j+1} = a_i V_{i+1}^j + b_i V_i^j + c_i V_{i-1}^j, \tag{8}$$

where

$$\begin{aligned} a_i &= \frac{\Delta t}{2}(\sigma^2 i^2 + ri) \\ b_i &= 1 - \Delta t(\sigma^2 i^2 + r) \\ c_i &= \frac{\Delta t}{2}(\sigma^2 i^2 - ri) \end{aligned} . \tag{9}$$

Then, the initial and boundary conditions read as follows:

$$\begin{aligned} V_i^1 &= \max(i - k, 0) \\ V_1^j &= 0,\ V_{N-1}^j = L - ke^{-r\tau_j} \end{aligned} .$$

This forward difference or explicit FD is stable on the condition defined by Courant–Friedrichs–Lewy (CFL) which prevents the numerical errors from blowing up, and relates the time step size Δt to asset price step size Δx imposing the following limitation

$$\Delta t \le \frac{\Delta x^2}{2\sigma^2}.$$

This condition forces the use of an extremely small step size for high volatility values, thereby increasing the computational time.

3.2 | LT and its Inversion

The second numerical approach used in this study to solve Equations (4)–(6) is the LT method. The method obtains the option prices in Laplace space by transforming the PDEs in the Black–Scholes model into a system of ODEs. The corresponding option values are then retrieved in time space through the numerical inversion of the method.

Applying LT to Equation (4) leads to the following:

$$\frac{1}{2}\sigma^2 S^2 \frac{d^2U}{dS^2} + rS\frac{dU}{dS} - (r + p)U = -V(S, 0), \tag{10}$$

subject to the boundary conditions

$$U(p, 0) = 0, U(p, L) = \frac{L}{p} - \frac{k}{p + r}, \tag{11}$$

where p is the transformation parameter. Using the explicit FD scheme for spatial discretisation, Equation (10) leads to

$$\frac{1}{2}\sigma^2 i^2 \Delta x^2 \frac{U_{i+1} - 2U_i + U_{i-1}}{\Delta x^2} + ri\Delta x \frac{U_{i+1} - U_{i-1}}{2\Delta x} - (r + p_j)U_i = -\max(i\Delta x - k, 0) \tag{12}$$

Rewriting the discretised Equation (12) in a simplified form leads to

$$a_{i1}U_{i+1} - a_{i2}U_i + a_{i3}U_{i-1} = b_i, \tag{13}$$

where

$$\begin{aligned} a_{i1} &= \frac{1}{2}(\sigma^2 i^2 + ri) \\ a_{i2} &= (\sigma^2 i^2 + r + p_j) \\ a_{i3} &= \frac{1}{2}(\sigma^2 i^2 - ri) \\ b_i &= -\max(i\Delta x - k, 0) \end{aligned} . \tag{14}$$

Equations (13) and (14) can be expressed in matrix form as follows:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & & & \\ a_{11} & a_{12} & a_{12} & & \\ & a_{21} & a_{22} & a_{23} & \\ & & \cdots & \cdots & \\ & & a_{n-1,1} & a_{n-1,2} & a_{n-1,3} \\ & & & 0 & 1 \end{bmatrix}$$
$$\boldsymbol{b} = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_n \end{bmatrix},\ \boldsymbol{U} = \begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_n \end{bmatrix} \text{ and } \boldsymbol{AU} = \boldsymbol{b} \tag{15}$$

In this work, Equations (13)–(15) are solved numerically via TDMA to obtain the value of $U(p, S)$ as a forward transform. A numerical Laplace inversion on this value is then performed to obtain the value of $V(S, \tau)$ in the original time space. This study explores three methods to perform Laplace inversion. These include the Gaver–Stehfest method, Weeks' method and de Hoog's method. The computational efficiency of the Laplace-space numerical solution may become critical due to the estimation of a large number of parameters; hence, an efficient inversion approach may be desirable in many cases. The numerical inversion of LT is generally an ill-posed problem due to the computation of p in various regions of the complex plane; hence, no single approach is optimal to suit the temporal behaviours [23].

3.2.1 | Gaver–Stehfest Method

The Gaver–Stehfest [24] method is a numerical inversion technique popular for its simplicity and ease of implementation. Due to the estimation of the time-domain function only along the real axis of the complex plane, the method is known to work well with low-frequency/non-oscillatory functions. This method has been reported in the literature for solving linear/nonlinear diffusion problems [25]. The approximate value of $V(S,\tau)$ is given as follows:

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$V(S,\tau) \approx \frac{\ln(2)}{\tau} \sum_{j=1}^{m} w_j U(\lambda_j, S), \tag{16}$$

where the weighting factor w_j is defined as follows:

$$w_j = (-1)^{j+\frac{m}{2}} \times \sum_{k=\frac{(j+1)}{2}}^{\min\left(j,\frac{m}{2}\right)} \times \frac{k^{\frac{m}{2}}(2k)!}{\left(\frac{m}{2}-k\right)!k!(k-1)!(j-k)!(2k-j)!}, \tag{17}$$

and the transformation parameters λ_j are defined as follows:

$$\lambda_j = j\frac{\ln(2)}{\tau},\ j = 1,\ 2,\ \ldots,\ m. \tag{18}$$

In Equation (18) m is an even number, usually chosen small in order to avoid truncation errors. A range of m between 6 and 20 is proposed by Cheng and Siduruk [26]. In the test case example included in section IV, some numerical experiments are performed to confirm the limitations on its range.

There are two special features that could be stressed regarding the computational efficiency of the LT method. First, the system of Equations (10)–(18) can be solved for several different times $t1$, $t2$, $t3$,… by calculating U only once for a set of λ_j, $j = 1, 2, \ldots, m$. Second, due to the parallelisation property of the method, for a specific time t, each $U(\lambda_j,t)$ can be solved independently for a λ_j, $j \neq k$, thus enabling parallel processing in a distributed computational environment.

3.2.2 | Weeks' Method

Based on a power series expansion of a Laguerre polynomial, the inverse LT in Weeks' method [27, 28] is given as follows:

$$V_N(S,\ \tau) \approx e^{\kappa\tau} \sum_{j=0}^{N-1} a_j e^{-b\tau} L_j(2b\tau), \tag{19}$$

where κ and b are free scaling parameters which depend on the maximum specified time $t_{\max}$ rather than τ and are chosen to optimise the convergence. Coefficients a_j contain the information specific to $U(p, S)$ and may be complex if $V(S, \tau)$ is complex. The Laguerre polynomial $L_j(x)$ of degree j can be defined using the Rodriguez relationship [29] as follows:

$$L_j(x) = \frac{e^x}{j!}\frac{d^j}{dx^j}\left(e^{-x}x^j\right). \tag{20}$$

The connection to $U(p, S)$ is established through the following equation:

$$U(p_j,\ S) = \frac{1-w}{2b} \sum_{j=0}^{N} a_j w^j, \tag{21}$$

with bilinear transformation w given as follows:

$$w = \frac{\kappa + b - p}{\kappa - b - p}.$$

3.2.3 | de Hoog's Method

The de Hoog method [30] is based around the approximation of the inverse transform via accelerated Fourier series inverse algorithms. The Fourier series transform is given as follows:

$$V(S,\tau) = \frac{e^{\gamma\tau}}{T}\left\{\frac{U(p)}{2} + \sum_{j=1}^{N} \mathrm{Re}\left[U\left(p + \frac{i\pi j}{T}\right)\cos\left(\frac{i\pi j}{T}\right)\right\}\right.$$
$$\left. -\mathrm{Im}\left[U\left(p + \frac{i\pi j}{T}\right)\sin\left(\frac{i\pi j}{T}\right)\right)\right]\Bigg\}, \tag{22}$$

where $i = \sqrt{-1}, T = 2\,t_{\max}$ and $t_{\max}$ is the maximum specified time. The parameter p is given as follows:

$$p = \alpha - \frac{\ln E}{2T},$$

where E is the desired error tolerance and α is a real number. A suitable choice of discretisation scheme can lead to a minimal discretisation error making the method robust for complex $U(p, S)$ in the evaluation of $V(S, \tau)$.

3.2.4 | A Note on Stability and Convergence

In the transformed Laplace space, TDMA is used in this work to solve the resulting tridiagonal system of equations. TDMA is a direct solver based on a 2-step LU decomposition in an efficient manner [31, 32]. In one-dimensional situations, the exact solution of the system is achieved by forward elimination and back-substitution in a fixed number of steps. The condition for the algorithm to be stable is as follows:

$$||a_{i2}|| > ||a_{i1}|| + ||a_{i3}||,$$

for all i in Equation (15). In the time domain inversion in the Gaver–Stehfest method, where $V_t(p)$ are continuous and non-oscillatory, the solution is efficient and highly accurate for the small number of transformation parameters but diverges as m increases. Working in a multi-precision environment has been shown to provide better accuracy and reduction in perturbation errors [33]. Time domain inversion via Weeks' or de Hoog's methods is not restricted to real values only, as in the case of the Gaver–Stehfest method and requires no limitations such as continuity and non-oscillatory $V_t(p)$. The accuracy of the methods increases with the increase in the size of transformation parameter p but grows much slower than that required by the other similar methods [23].

In this work, the numerical approximation of the Black–Scholes equation is performed using the explicit FD method, as shown in Equations (7)–(9), and is used for comparative purposes against the methodologies of Section 3 B. Crank–Nicolson and Implicit FD, although not shown explicitly in section III, have also been employed in the results section as the benchmark for comparison. The explicit FD method is easier to implement, computationally faster and convergent provided that the option value does not increase backward in time for each iteration. The

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

implicit FD method, which approximates the time derivative with backward difference, is; however, more stable than explicit FD but computationally more expensive. Research studies [19] show that the approximate option price obtained by mixed methods, such as FD and LT, is equivalent to the standard FD with a fully implicit differencing scheme. Since the fully implicit FD scheme is convergent, so is the numerical solution obtained by the mixed methods. The different measures considered to ascertain the stability and convergence of the commonly used Laplace inversions are discussed in various studies [23, 27, 34]. For the Gaver–Stehfest method this includes the constraint of the computational rounding error shrinking faster than the growth of weights in Equation (17) requiring arbitrary precision arithmetic scaling as digits $= 1.1 \times m$ and the local behaviour of the time-domain function satisfying logarithmic Dini condition [35]. In practice, the Dini criterion behaves as a quality control check on the hidden time-domain function. In Weeks' method the expansion coefficients a_j in Equation (19) are calculated using a boundary contour integral on a unit disk instead of real-axis differences which provides an unconditional stable inversion matrix for the coefficient calculation. The system convergence depends on the selection of optimal scaling values of the shift parameter κ and the stretching factor b. Based on how much is known about the underlying Laplace function, the three distinct approaches used for selecting these parameters include analytical methods, automated algorithms and ML techniques [27, 36, 37]. The numerical stability of de Hoog's method is preserved by selecting a large enough safety value for parameter p in Equation (22) forcing the rational poles away from the unit evaluation circle. The structural stability of the method is preserved by using the measures which dampen the aliasing error such as setting pT between 5 and 10. The stability trade-offs of the structural parameters can be found in [33].

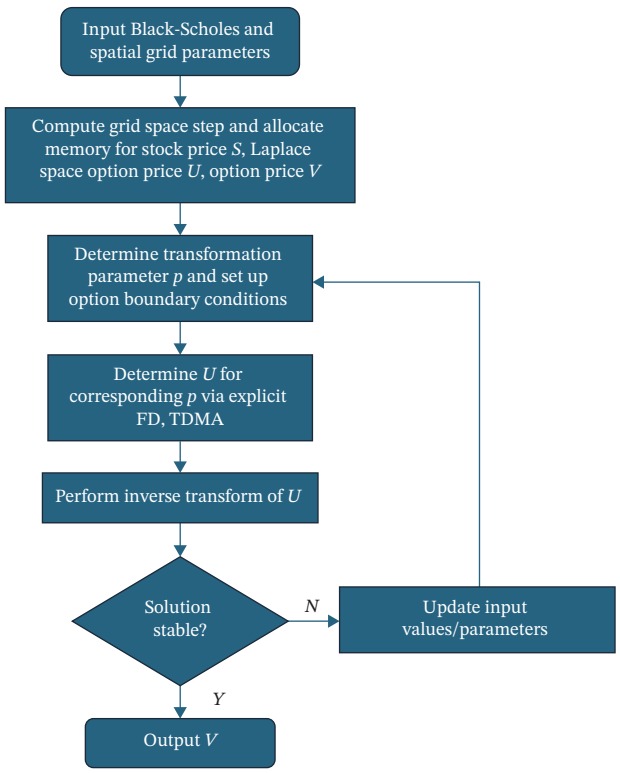


FIGURE 1 | Flow chart for the algorithm of the scheme used in numerical LT and its inversion.

ALGORITHM 1

i. Set up:

For a given p assemble coefficients a_1, a_2, a_3, Equation (14)

Set boundary conditions U_1, U_n

Define new parameters C and D and set

$U_1 = C_1$

$U_n = C_n$

$D_1 = 0$

ii. Forward sweep:

Loop: $i = 2$ to n–1

temp $= a_2 – a_3 \times D_{i-1}$

$_{Di} = a_1/\text{temp}$

$C_i = b_i + (a_3 \times C_{i-1})/\text{temp}$

iii. Back substitution:

Loop: i $= n$–1, 2

$U_i = D_i \times U_{i-1} + Ci$

A flow chart of the scheme used in numerical LT and its inversion is given in Figure 1 and the algorithm of obtaining the option prices in Laplace space via TDMA is given below (Algorithm 1).

3.2.4.1 | Stability, Convergence, Error, Complexity Analysis and Limitations of the Framework. Based on the workflow discussed above a rigorous stability and convergence justification can be established by viewing the method as a composition of three operators:

i. LT
ii. FD approximation of the transformed elliptic problem
iii. Numerical inverse LT.

3.2.4.2 | Mathematical Assumptions

1. The linear parabolic PDE describing option pricing can be written in the form

$$\frac{\partial V}{\partial t} = \mathscr{A}V + f,$$

where $\mathscr{A}$ is a second order uniformly elliptic operator in the form

$$\mathscr{A}V = a(S)V_{SS} + b(S)V_S - c(S)V,$$

with $a(S) \geq a_0 > 0$, $c(S) \geq 0$.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

2. Assume sufficient regularity

$$V \in C([0, T]; H^2(\Omega)),$$

so that second order FD truncation estimates exist. Here $H^2(\Omega)$ represents Sobolev space containing functions whose derivatives up to order two are square-integrable over the spatial domain Ω.

3. The continuous transformed problem has a unique solution for

$$\mathrm{Re}(p) \geq p_0 > 0.$$

4. The infinite asset domain is truncated to

$$\Omega = [0,\ L],$$

with boundary conditions whose truncation error satisfies $\varepsilon_B(L) \to 0,\ L \to \infty$.

3.2.4.3 | Stability. Applying LT in time

$$U(S,\ p) = \int_0^\infty e^{-pt} V(S,\ t)\ dt,$$

gives

$$pU - V_0 = \mathscr{A}U + \widehat{f}.$$

Hence,

$$(pI - \mathscr{A})U = V_0 + \widehat{f}.$$

For every transformation parameter $p = \lambda_j$,

$$(pI - \mathscr{A}),$$

is an elliptic operator.

Let

$$\mathrm{Re}(p) = (pI - \mathscr{A})^{-1}.$$

If $\mathscr{A}$ generates an analytic semigroup then

$$\|\mathrm{Re}(p)\| \leq \frac{C}{|p|},$$

for $\mathrm{Re}(p) \geq p_0 > 0$. Consequently

$$\|U\| \leq \frac{C}{|p|} \|V_0 + \widehat{f}\|.$$

Thus, the transformed problem is well conditioned whenever all Laplace parameters lie in the right-half plane. This explains the stability requirement $\mathrm{Re}(\lambda_j) > 0$. The numerical inversion methods used in this work all employ transform points satisfying this requirement.

For FD approximation, let $h = \Delta S$, then using central differencing

$$U''(S_i) = \frac{U_{i+1} - 2U_i + U_{i-1}}{h^2} + O(h^2)$$

and

$$U'(S_i) = \frac{U_{i+1} - U_{i-1}}{2h} + O(h^2).$$

Therefore

$$\mathscr{A}_h U = \mathscr{A}U + \mathscr{F}_h,$$

with

$$\|\mathscr{F}_h\| = O(h^2).$$

Hence, the FD scheme is consistent of order 2 in space.

The discretisation yields

$$\mathscr{A}_h(\lambda_j)U = b,$$

and

$$\mathscr{A}_h(\lambda_j) = \lambda_j I - \mathscr{L}_h,$$

where $\mathscr{L}_h$ is the discrete diffusion–convection operator. For sufficiently small mesh size and positive diffusion coefficient, $\mathscr{A}_h$ is a tridiagonal M-matrix $a_i > 0,\ c_i > 0,\ b_i > a_i + c_i$. Then

$$\mathscr{A}_h^{-1} \geq 0,$$

and

$$\|\mathscr{A}_h^{-1}\| \leq C,$$

uniformly in h. Hence

$$\|U\| \leq C\|b\|,$$

is the discrete stability estimate. With Thomas elimination being an exact LU factorisation for tridiagonal matrices no additional discretisation error is introduced beyond floating-point roundoff. It does not alter the order of convergence of the FD discretisation. Thus, the FD stability in the framework requires matrix stability and not time-step stability, which is a key difference from explicit Euler scheme.

3.2.4.4 | Convergence. Let

$$e_h = U - U_h.$$

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Subtracting the discrete and continuous equations yields

$$\mathscr{A}_h e_h = \mathscr{L}_h.$$

Thus

$$e_h = \mathscr{A}_h^{-1}\mathscr{L}_h.$$

Using stability

$$\|e_h\| \le \|\mathscr{A}_h^{-1}\|\,\|\mathscr{L}_h\| \le Ch^2.$$

Hence

$$\|U - U_h\| = O(h^2).$$

By the Lax equivalence principle, for linear well-posed problems, consistency plus stability implies convergence.

3.2.4.5 | Error. Let V be the exact option value and V_N denote the numerical inversion obtained from N Laplace evaluations. Then, the inversion error is

$$E_{\text{inv}} = V - V_N.$$

For Gaver–Stehfest inversion the error E_{GS} can be decomposed into

$$E_{\text{GS}} = E_{\text{trunc}} + E_{\text{cancel}} + E_{\text{round}}.$$

However, since

$$E_{\text{trunc}} = \sum_{j=m+1}^{\infty} w_j U(\lambda_j, S),$$

only m terms are retained and as m increases $E_{\text{trunc}} \longrightarrow 0$.

Also $\lceil w_j \rceil$ grows rapidly as m increases. The approximation contains large positive and negative terms whose difference may be $O(1)$. This introduces catastrophic cancellation. Consequently,

$$E_{\text{cancel}} \propto \kappa(m)\varepsilon_{\text{mach}},$$

where $\kappa(m)$ is an effective condition number. The implication of round-off error is that as m increases $\kappa(m) \to \infty$. Theoretical convergence therefore eventually becomes dominated by numerical noise. For sufficiently smooth non-oscillatory functions $E_{\text{GS}} \longrightarrow 0$ as $m \longrightarrow \infty$ in exact arithmetic. However, in floating-point arithmetic an optimal value m^* exists beyond which the error increases.

For Weeks inversion, the error decomposition can be represented as follows:

$$E_{\text{Weeks}} = E_{\text{series}} + E_{\text{parameter}}.$$

If $U(p)$ is analytic in the required region,

$$|a_j| \le Ce^{-\beta j}.$$

Consequently,

$$E_{\text{series}} = O(e^{-\beta N}),$$

which represents exponential convergence.

Weeks' method introduces free parameters κ and b whose poor choices may significantly degrade accuracy. Thus,

$$E_{\text{parameter}} = O(|\kappa - \kappa^*|) + O(|b - b^*|).$$

For de Hoog, the error can be written as follows:

$$E_{\text{DH}} = E_{\text{Fourier}} + E_{\text{acceleration}}.$$

Fourier inversion produces

$$E_{\text{Fourier}} = O(e^{-\gamma N}).$$

The continued-fraction acceleration of the Fourier series substantially reduces oscillatory truncation errors. As a result,

$$E_{\text{acceleration}} \ll E_{\text{Fourier}}.$$

Consequently,

$$E_{\text{DH}} = O(e^{-\gamma N}) + O(\varepsilon_{\text{mach}}),$$

where $\varepsilon_{\text{mach}}$ denotes machine precision. Thus, the overall error of the framework is

$$V - V_h^N = E_{\text{FD}} + E_{\text{inv}} + E_B + E_r,$$

where E_{FD} is the FD discretisation error, E_{inv} represents inverse LT error, E_B represents boundary truncation error and E_r represents round-off error.

Using the earlier stability analysis,

$$E_{\text{FD}} = O(h^2),$$

$$E_{\text{B}} = O(\varepsilon_B(L)).$$

Also,

$$E_r = O(\varepsilon_{\text{mach}}).$$

3.2.4.6 | Complexity. For each Laplace parameter λ_j

$$A(\lambda_j)U_j = b,$$

must be solved. The resulting matrix is tridiagonal. The cost for TDMA for forward elimination is $O(N)$ as well as for back substitution. Therefore

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$T_{\mathrm{TDMA}} = O(N).$$

Since the inverse transform requires m evaluations,

$$T_{\mathrm{LT}} = O(mN).$$

The storage requirement is as follows:

$$S_{\mathrm{LT}} = O(N).$$

Because only the three diagonals and solution vector must be stored this is a significant advantage over dense solvers which require $O(N^2)$ storage. Moreover, the Laplace parameters are independent. Meaning

$$U(\lambda_j) \perp U(\lambda_k),\ j \neq k.$$

Hence, all transformed systems may be solved simultaneously. With p processors,

$$T_{\mathrm{LT,parallel}} = O\left(\frac{\mathrm{mN}}{P}\right).$$

If $P \approx m$ then

$$T_{\mathrm{LT,parallel}} = O(N).$$

Thus, the method exhibits nearly ideal strong scalability.

Table 1 provides a short summary of the computational cost, memory and parallelisation comparison among the common approaches, where N is the number of spatial grid points, M is the number of time steps, m is the number of Laplace inversion parameters and P represents number of processors available for parallel execution.

In Monte Carlo, N_pM represents N_p paths with M timesteps per path.

3.2.4.7 | Limitations. The framework has the following notable limitations:

- Domain truncation can introduce an additional approximation error $\varepsilon B(L)$.
- Non-smooth payoffs such as $\max(S-K,\ 0)$ may reduce the observed second-order convergence near the strike price.
- Inverse LT methods are sensitive to parameter selection.
- Large transform parameters may lead to floating-point cancellation errors.
- Loss of diagonal dominance may weaken the TDMA stability guarantees.
- American, free-boundary, jump-diffusion and non-linear models require additional analysis.

3.3 | HV

HV is the statistical measure of asset price movement based on historical data. There are different ways of estimating HV from previous asset behaviour. Studies suggest that time-series historical models built on realised volatility can provide good forecasting performance [38]. The HV models considered here are the historical average and exponentially weighted moving average (EWMA). For FD and LT models where HV is used as the input in this work, it is estimated from realised volatility as the rolling input and is calculated using the annualised standard deviation for closed stock price returns over a given time period. It is demonstrated as below.

Assuming that the historical asset prices $S_{t1}, S_{t2},\ldots, S_{\mathrm{tn}}$ are available, the percentage price change in the market is calculated as follows:

$$U_i = \ln\left(\frac{S_{\mathrm{ti}}}{S_{\mathrm{ti-1}}}\right). \tag{23}$$

An average price change over a period of N days can be expressed as follows:

$$U_n = \frac{1}{N}\sum_{i=1}^{N} U_{n-i}. \tag{24}$$

Calculating the average variance from Equation (24) provides the estimated HV σ_d

TABLE 1 | Comparison of complexity.

Method	Computational cost	Memory	Parallelisation
Explicit FD	$O(NM)$	$O(N)$	Limited
Implicit FD	$O(NM)$	$O(N)$	Moderate
Crank–Nicolson	$O(NM)$	$O(N)$	Moderate
Binomial tree	$O(M^2)$	$O(M)$	Poor
Trinomial tree	$O(M^2)$	$O(M)$	Poor
Monte Carlo	$O(N_pM)$	$O(N_p)$	Excellent
Proposed framework	$O(mN)$	$O(N)$	Excellent

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\sigma_d = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(U_i - U_n)^2}. \quad (25)$$

Expressing σ_d in terms of annualised HV for the average number of trading days in a year leads to

$$\sigma = \sigma_d \times \sqrt{252}. \quad (26)$$

In the historical average model, the forecast is based on the past fluctuations in asset price, with equal weights given to all past returns. The optimal forecast of future volatility is thus described as follows:

$$\widehat{\sigma_{t+1}} = \frac{1}{t}\sum_{j=1}^{t}\sigma_j, \quad (27)$$

where $\widehat{\sigma_{t+1}}$ is the expected volatility at time $t+1$.

EWMA excels over some of the other HV models due to its adaptability to rapid current market changes without needing complex parameter estimation. In EWMA, the volatility forecast is based on geometrically declining weight where the immediate past observed volatility is replaced with the moving average forecast. The model is specified as follows:

$$\widehat{\sigma_{t+1}} = \omega\widehat{\sigma_t} + (1-\omega)\frac{1}{T}\sum_{j=t-T}^{t}\sigma_j, \quad (28)$$

where $0 < \omega < 1$. The most commonly used value of ω is 0.94. EWMA stands out due to its responsiveness to sudden market shocks in recent trading days and absorbing them effectively. The model also compensates for the ghost effect of volatility spikes observed in the rolling-window moving averages by gradually diminishing the impact of the long past. The model also benefits from the industry standardised benchmark of the decay factor ω. In the results section the historical average model will serve as a comparison against the EWMA.

3.4 | Data-Driven Approaches

It could be argued that the increased computational powers and efficiency of computers have assisted in processing the large volume of data, enabling the effective use of data-driven ML models. This section provides a brief overview of the commonly used approaches including neural networks, LSTM, SVM, *K*-nearest neighbours (KNN), random forest and AdaBoost.

Mimicking the function of biological neural systems, neural networks are computing systems containing interconnected neurons or nodes arranged in a layered architecture. A neural network contains an input layer with the number of nodes equal to the number of input variables, none, one or more hidden layers and an output layer consisting of nodes predicting the outcome or target. A cost function in a neural network is computed as a measure of performance by optimisation algorithms such as stochastic gradient descent (SGD) or Adam. Repetitive optimisation is carried out until the error is minimised.

In a feed-forward neural network, the information travels from input, through the hidden layers, to the output layer in one direction. In a recurrent neural network (RNN); however, the output is fed back to the input in hidden layers, making the input and output interdependent. The network is thus able to learn and memorise from the past information, yielding an architecture suitable for time-series forecasting or sequence processing. LSTM which is based on RNN consists of many components including input, forget and output gates, memory store and cell state for information transfer and long-term memory enhancement.

SVM is a standard ML algorithm applied to classification and regression problems. The aim of SVM is to find the function that can approximate the target from the training set with a tolerance ε with allowable positive and negative deviations of ε from the target. Kernel functions such as polynomial or radial basis function (RBF) are applied to solve for non-linearity where it exists. Like SVM, the KNN model can be applied to classification or regression problems. The model is attractive due to its simplicity and not having any underlying assumptions about the data. KNN regression determines the target value by calculating its distance to KNN in the training dataset and selecting the average value of these KNNs. Euclidean or other similar methods are used for distance calibration, and the size of *K*, usually chosen by cross-validation, is the one which minimises the least squared error.

The Random forest model is based on ensemble learning where a more powerful prediction is sought out by combining a single algorithm multiple times or several algorithms together trained on the same dataset. The building blocks of the random forest model are decision trees, which use a divide and conquer approach to narrow down the possibilities until an outcome is reached. The prediction method used in the random forest model samples individual trees and their attributes from the training set with bootstrap aggregation, otherwise known as bagging. This enforces variability among the trees, thereby constructing a forest that outperforms any of the individual models. The final result is obtained by taking the mean value of the integrated models. AdaBoost commonly uses decision tree as the underlying model and falls in the boosting category of ensemble learning. Boosting in ML works by combining multiple learners to create a strong learner in an attempt to improve the errors. In AdaBoost, the errors in the sequential model builds are used to increase the weights of incorrect predictions. The iterative process of weight adjustments and combining weighted outputs results in the improved performance of the model. Further details and the mathematical framework of the ML models described here can be found in the literature [14, 22, 39–44].

Instead of following the laws of physics, the standard ML models are designed to learn from data and find patterns. This makes them prone to accidentally output impossible prices, violating financial logic and leading to arbitrage. To satisfy the core no-arbitrage constraints such as monotonicity and convexity in strike and monotonicity in maturity, different model types require different strategies. For the ML models considered in this work, these are discussed alongside other model setups in the results section.

3.4.1 | Forecasting Performance of ML Models and Numerical Methods

To evaluate the performance of ML models in options price predictions, five commonly used metrics are chosen. These include *R*

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

squared (R^2), MSE, RMSE, mean absolute error (MAE), mean absolute percentage error (MAPE) and are defined as below:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}, \tag{29}$$

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2, \tag{30}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2}, \tag{31}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y}_i|, \tag{32}$$

$$\mathrm{MAPE} = 100\%\frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \widehat{y}_i}{y_i}\right|, \tag{33}$$

where $\widehat{y}_i$ is the model predicted value, y_i is the true value and N is the number of predictions. Other metrics to evaluate the error of numerical methods include the maximum absolute error and is defined as below:

$$\mathrm{MaxAE} = \max_{i=1,2,\ldots,n}|y_i - \widehat{y}_i|. \tag{34}$$

TABLE 2 | Computational examples with constant volatility.

Example	k	S_0	T	r	σ
1	4	5	1	0.05	0.3
2	9	10	1	0.06	0.1

4 | Results and Discussions

4.1 | Preliminary Test Case With Constant (Non-HV)

To examine the accuracy of the numerical methods, a test case is carried out using the computational examples for pricing European call options described by Higham [45]. The parameters used are presented in Table 2.

4.1.1 | Accuracy and Sensitivity

The computed numerical solutions are compared with the exact solution, and absolute errors and MAE are determined. The number of grid points in space is chosen as $n = 50$. The asset range is truncated at $L = 3k$, based on the criteria presented in Section 2. All numerical simulations are performed in C++ using double-precision arithmetic, providing approximately 15–16 decimal digits of accuracy. Call option prices for the two examples and errors for LT with Weeks' (LT-W), de Hoog (LT-DH) and Gaver–Stehfest (LT-GS) inversion are shown in Table 3. For LT inversion methods (prefix LT), m represents the number of transformation or evaluation parameters.

The results show that the LT inversion methods produce accurate and stable solutions with reasonably small values of m, making them suitable candidates for the pricing analysis considered here.

TABLE 3 | Comparison of European call option prices and absolute errors obtained using the finite difference (FD) method and Laplace-transform inversion methods.

E.g.	Method	Call option price	Error	MAE
1	LT-GS, $m = 2$	1.60579	2.82691e–01	5.6538e–03
1	LT-GS, $m = 4$	1.33494	1.12227e–02	2.24453e–04
1	LT-GS, $m = 6$	1.32427	5.32519e–04	1.06504e–05
1	LT-GS, $m = 10$	1.32334	2.39671e–04	4.79343e–06
1	LT-GS, $m = 12$	1.32334	2.36890e–04	4.73781e–06
1	LT-GS, $m = 14$	Spurious result	—	—
1	LT-W, $m = 10$	1.32398	8.8009e–04	1.76018e–05
1	LT-DH, $m = 10$	1.32398	8.72955e–04	1.74591e–05
2	LT-GS, $m = 2$	2.08134	5.38411e–01	1.07682e–02
2	LT-GS, $m = 4$	1.54434	1.40405e–03	2.80811e–05
2	LT-GS, $m = 6$	1.54000	2.92998e–03	5.85985e–05
2	LT-GS, $m = 10$	1.54100	1.93179e–03	3.86359e–05
2	LT-GS, $m = 12$	1.54098	1.95481e–03	3.90962e–05
2	LT-GS, $m = 14$	Spurious result	—	—
2	LT-W, $m = 10$	1.544218	1.280993e–03	2.56198e–05
2	LT-DH, $m = 10$	1.54421	1.280422e–03	2.56084e–05

Note: LT-GS, LT-W and LT-DH denote the Gaver–Stehfest, Weeks and de Hoog inversion techniques, respectively. Examples 1 and 2 correspond to the benchmark test cases defined in Table 2 with different volatility levels. For LT-GS, m denotes the number of inversion parameters. Errors are computed relative to the analytical Black-Scholes solution.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

It can also be noticed that the smaller values of σ (Example 2) produce larger errors in LT-GS, and the most accurate solution is obtained at just $m = 4$, leading to the conclusion that a larger value of m does not necessarily provide better accuracy when compared with the exact solution. The solution also diverges for any value of m greater than 12 for LT-GS.

After computing the numerical Laplace solution with the same p across all S in the space domain, the inverse transform computation via Weeks' and de Hoog's methods provide fast convergence towards the exact solution even with a minimal number of evaluations, for example, 4 and 10 (Figures 2 and 3). The two inversion methods also produced nearly identical solutions. Finally, a sensitivity analysis is performed to test whether $L = 3k$ is sufficient for European call options. With the number of grid points in space fixed at 50, call option prices for the two examples with $L = 2k$, $3k$, $4k$ and $5k$ are shown in Table 4. These numerical experiments produce negligible differences in the computed

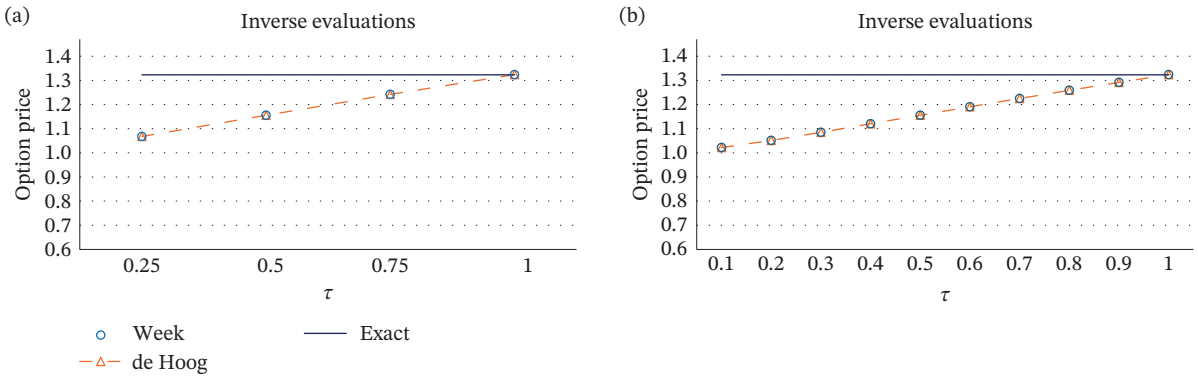


FIGURE 2 | Comparison of inverse Laplace transform evaluations obtained using Weeks' (LT-W) and de Hoog's (LT-DH) methods for pricing European call options. Results are shown for Example 1 ($\sigma = 0.3$) with (a) four evaluations and (b) 10 evaluations. LT-W and LT-DH denote Weeks' and de Hoog's inversion methods, respectively.

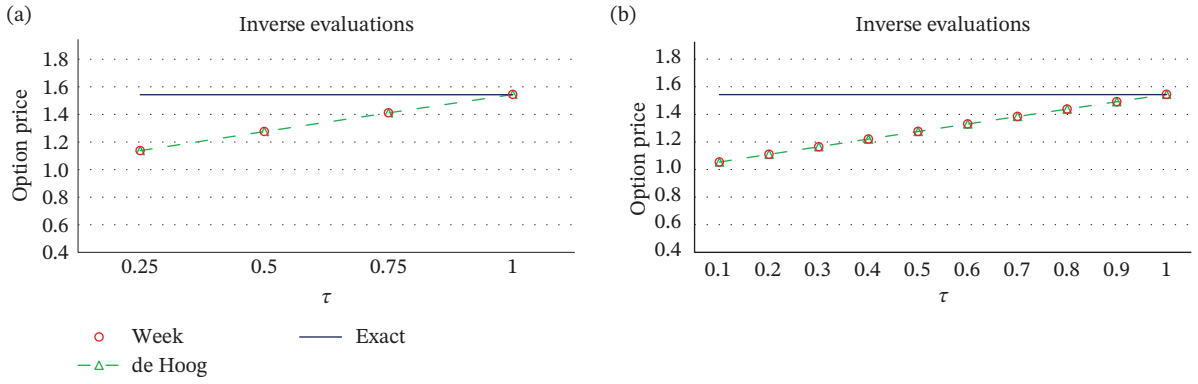


FIGURE 3 | Comparison of inverse Laplace transform evaluations obtained using Weeks' (LT-W) and de Hoog's (LT-DH) methods for pricing European call options. Results are shown for Example 2 ($\sigma = 0.1$) with (a) four evaluations and (b) 10 evaluations. LT-W and LT-DH denote Weeks' and de Hoog's inversion methods, respectively.

TABLE 4 | Sensitivity analysis of the computational domain size L for the European call option pricing problem.

E.g.	Method	$L = 2k$	$L = 3k$	$L = 4k$	$L = 5k$
1	LT-GS, $m = 10$	1.3225	1.3233	1.3238	1.320
1	LT-W, $m = 10$	1.3228	1.3239	1.3257	1.3226
1	LT-DH, $m = 10$	1.3228	1.3239	1.3257	1.3226
2	LT-GS, $m = 10$	1.5407	1.5410	1.5407	1.532
2	LT-W, $m = 10$	1.5418	1.5442	1.5439	1.5331
2	LT-DH, $m = 10$	1.5418	1.5442	1.5439	1.5331

Note: The option price is computed using the proposed Laplace-transform finite-difference framework while varying the artificial upper boundary L. The results are used to assess the influence of domain truncation on the numerical solution and to verify the suitability of the choice $L = 3$k, where k denotes the strike price. Examples 1 and 2 correspond to the benchmark test cases defined in Table 2.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 5 | Precision analysis of the numerical schemes considered for European call option pricing.

Example	n	LT-GS MAE	LT-W MAE	LT-DH MAE	FD-Exp	FD-Imp	FD-CN
1	10	2.2828e–03	2.2822e–03	2.2822e–03	1.6445e–01	6.2334e–03	6.2305e–03
	20	3.5264e–04	3.5241e–04	3.5239e–04	9.9913e–02	1.1046e–03	1.0977e–03
	40	4.9718e–05	5.0178e–05	4.9671e–05	5.4775e–02	3.2881e–04	3.2024e–04
	80	5.8710e–06	5.7873e–06	5.8460e–06	2.8550–02	1.1297e–04	1.0435e–04
	160	7.6755e–07	6.4471e–07	7.5492e–07	—	6.4286e–05	5.3911e–05
2	10	1.4979e–02	1.4977e–02	1.4977e–02	1.4729e–01	1.2163e–02	1.2162e–02
	20	3.4554e–04	3.4648e–04	3.4651e–04	1.0355e–01	5.6205e–03	5.61709e–03
	40	5.5298e–05	5.4712e–05	5.4698e–05	5.9200e–02	1.0939e–03	1.0913e–03
	80	7.2826e–06	7.0485e–06	7.0405e–06	3.1795e–02	3.2639e–04	3.2316e–04
	160	7.0274e–07	5.4780e–07	5.9584e–07	1.6443e–02	9.8622e–05	9.5129e–05

Note: The table compares the computed option prices and associated errors obtained using the finite difference (FD) method and Laplace-transform-based methods employing Gaver–Stehfest (LT-GS), Weeks (LT-W) and de Hoog (LT-DH) inversions. Errors are calculated relative to the analytical Black–Scholes solution, with smaller values indicating higher numerical accuracy.

TABLE 6 | Elapsed time (in seconds) for efficiency comparison of numerical schemes.

Example	n	LT-GS	LT-W	LT-DH	FD-Exp	FD-Imp	FD-CN
1	10	1.36e–05	0.0065143	0.0003898	0.0004925	0.0116713	0.0121992
	20	1.47e–05	0.0080428	0.000394	0.0019234	0.022890	0.0244780
	40	8.26e–05	0.0244788	0.0004543	0.0023628	0.0554319	0.0609216
	80	3.05e–05	0.117142	0.0010872	0.0053660	0.1582268	0.1653794
	160	0.0002473	1.5882563	0.003561	—	0.4846182	0.5140724
2	10	3.22e–05	0.008415	0.0009777	0.0010607	0.0259271	0.0232784
	20	5.44e–05	0.0212795	0.0022915	0.0028738	0.0469012	0.0457724
	40	6.86e–05	0.0516321	0.0021798	0.0047806	0.1097236	0.1260957
	80	5.06e–05	0.2874416	0.003451	0.0102774	0.3083839	0.3129165
	160	0.0001605	1.5245945	0.0038811	0.023020	0.9715414	0.9748101

option prices, confirming that $L = 3k$ is sufficient for the test cases considered. The sensitivity of space grid size is tested within the computation for efficiency and convergence, which are discussed next.

4.1.2 | Convergence and Efficiency

TDMA as a direct method provides the exact solution in a finite number of steps. It is; however, prone to rounding errors due to the recursive nature of the algorithm involving forward sweep and back substitution. The discretisation error of the model decreases as the step size is reduced, but the decreased step size can also lead to the greater accumulation of rounding errors. The precision of numerical solution to the tridiagonal linear system is assessed by MAE, Equation (32). The explicit FD solution (FD-Exp) is acquired for reference for various values of n while setting $m = 1000$ for Example 1 and 2000 for Example 2 to test the effects of space grid size on numerical solution. These temporal grid sizes are chosen to be compliant with the CFL condition stated in Section 2 A and have been adopted in Crank–Nicolson (FD-CN) and implicit FD (FD-Imp) for ease of comparison. L is fixed at $3k$. All computations are performed in C++ with an Intel(R) Core (TM) i5 CPU with 16 GB RAM and a 1.3 GHz processor. MAE and elapsed time in seconds for different values of n for Examples 1 and 2 are shown in Tables 5 and 6. Optimal values of the LT parameter, as suggested by the results presented in Table 3, are chosen for LT inversion methods. These results show that the solution precision increases with a smaller step size for the three inversion methods, and the computations remain highly efficient for both Gaver–Stehfest and de Hoog's inversions with the increased step size. It can also be seen that the numerical solution obtained via explicit FD is efficient and is comparable to Weeks' inversion but produces the greatest errors.

4.2 | HV

The performance of the numerical LT inversions is tested using real stock market data to price European call options. In our simulations, we use IBM stock options from 18–19 July 2005 with September expiration. The HV models of historical average

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

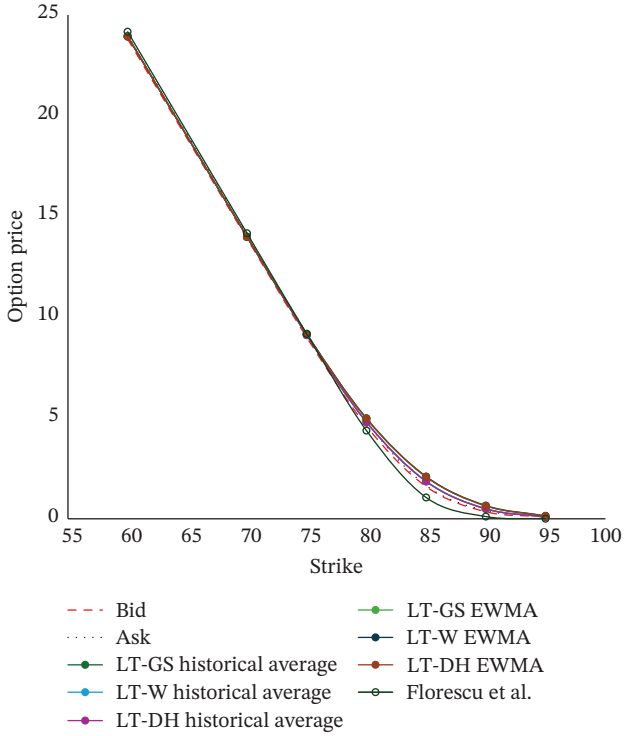


FIGURE 4 | Estimated IBM call option values and the bid-ask spread. LT solution is simulated under HV using HV models of historical average and EWMA. Results from literature (Florescu et al.) are shown for comparison.

and EWMA described in Section III-C are used for volatility calibration with a sample data set of daily stock prices from 19 October 2004 to 18 July 2005. Daily log returns are computed as shown in Equation (23), and HV is estimated using a rolling window of 188 trading days and is annualised using Equation (26). In EWMA, the weight parameter ω is set as 0.94, and initial variance is computed using a sample variance of the first 20 returns.

To determine the call option value, the parametric values used in the Laplace solution are chosen as $L = 3$ k and $m = 6$ in the Gaver–Stehfest inversion. The interest rate and stock price are 0.034 and 83.7, respectively. The estimated volatilities for historical average and EWMA come to be 0.1989 and 0.1646, respectively. A comparison among LT inversions using HV models of historical average and EWMA along with bid-ask spread and results reported by Florescu and Viens [46] is shown in Table 7. A plot of the values is shown in Figure 4. As can be seen, the performance of all inversion methods used with these HV models is remarkable in predicting the IBM call option values. The estimated values lie very close in range to the bid-ask spread except for strikes around 85–90. For near-the-money contracts, option prices are generally sensitive to the market's expectation of future volatility rather than realised HV. The assumed market conditions in this case did not resemble past behaviour. Second, the Black–Scholes model assumes constant volatility across all strikes as opposed to volatility smiles or skews exhibited by the observed market prices. This could contribute to an inadequate representation of implied volatility embedded in options around specific strikes.

TABLE 8 | ML models' input setup.

ML models setup	Input	Output
S1	S_0, k, r, T, σ_{implied}	p
S2	S_0, k, r, T, σ	p
S3	S_0, k, r, T	p
S4	S_0, k, r	p
S5	S_0, r	p
S6—moneyness	S_0, k, r, T, σ	p

4.3 | ML for Call Options

The option trading data used in the final example are SSE 50ETF European call stock index options traded during January–December 2020 in Mainland China, obtained from the Guotaian Financial Database [47]. The dataset consisted of 13,819 daily prices and was filtered to include strike price k, stock price S_0, time to maturity T, risk-free interest rates r, implied volatility σ_{implied} and HV σ as input variables and theoretical option price p as the output variable. The theoretical price of an SSE European option is calculated using the Black–Scholes formula, which was included in the dataset used in this work. The values of input variables range between 2.35 and 4, 2.553 and 3.634, 0.0048 and 2.6804, 0.1281 and 0.2686 for k, S_0, σ_{implied} and σ, respectively, while the time to maturity ranges between 0.6685 and 0 year. The risk-free interest rate used is 1.5%. An 80:20 split

TABLE 7 | Solution comparison for IBM call option values.

Strike	Bid	Ask	LT-GS historical average	LT-W historical average	LT-DH historical average	LT-GS EWMA	LT-W EWMA	LT-DH EWMA	Florescu and Viens [46]
60	23.8	24	23.9441	23.937	23.9348	23.9436	23.9298	23.9278	24.1654
70	13.9	14.1	13.9918	13.985	13.9821	14.0007	13.9871	13.9841	14.1654
75	9	9.2	9.11809	9.11272	9.10736	9.18324	9.17059	9.16436	9.1699
80	4.6	4.8	4.80792	4.80387	4.79302	4.99016	4.97948	4.96761	4.3799
85	1.6	1.65	1.86555	1.86112	1.84857	2.09978	2.08906	2.07543	1.0571
90	0.35	0.4	0.502154	0.497067	0.489555	0.65774	0.64604	0.63716	0.1123
95	0.1	0.15	0.092271	0.086307	0.083975	0.149891	0.138378	0.13508	0.0040

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 9 | Prediction performance of machine-learning models for setup S1, where the inputs are stock price S_0, strike price k, interest rate r, time to maturity T, implied volatility σ_{implied} and the output is the option price P.

Model	R^2	MSE	RMSE	MAE	MAPE
Neural network	1.0	4.3210e–05	0.0065	0.0048	0.0291
LSTM	0.99	0.00026	0.0161	0.0125	0.0715
SVM	0.96	0.00188	0.0433	0.0357	0.1924
KNN	1.0	0.00018	0.0135	0.0083	0.0601
Random forest	1.0	5.9754e–05	0.0077	0.0048	0.0342
AdaBoost	1.0	7.5182e–05	0.0086	0.0052	0.0384

Note: Performance is measured using MAE, RMSE, MAPE and R^2.

TABLE 10 | Prediction performance of ML models in test set in S2.

Model	R^2	MSE	RMSE	MAE	MAPE
Neural network	1.0	2.4626e–06	0.0015	0.0012	0.0068
LSTM	1.0	7.6467e–05	0.0087	0.0067	0.0381
SVM	0.97	0.00173	0.0416	0.0339	0.1816
KNN	1.0	0.000104	0.0102	0.0068	0.0446
Random forest	1.0	5.3691e–05	0.0073	0.0044	0.0319
AdaBoost	1.0	6.5378e–05	0.0080	0.0051	0.0354

TABLE 11 | Prediction performance of ML models in test set in S3.

Model	R^2	MSE	RMSE	MAE	MAPE
Neural network	1.0	8.1888–05	0.0090	0.0060	0.0395
LSTM	1.0	0.00024	0.0156	0.0123	0.0682
SVM	0.97	0.00165	0.0406	0.0331	0.1774
KNN	1.0	0.00013	0.0117	0.0075	0.0514
Random forest	1.0	6.5629–05	0.0081	0.0050	0.0353
AdaBoost	1.0	8.1521e–05	0.0090	0.0055	0.0394

TABLE 12 | Prediction performance of ML models in test set in S4.

Model	R^2	MSE	RMSE	MAE	MAPE
Neural network	0.96	0.00177	0.0421	0.0335	0.1839
LSTM	0.97	0.00170	0.0413	0.0330	0.1805
SVM	0.95	0.00232	0.0482	0.0397	0.2106
KNN	0.96	0.00195	0.0442	0.0337	0.1930
Random forest	0.95	0.00263	0.0512	0.0390	0.2239
AdaBoost	0.95	0.00271	0.0521	0.0395	0.2276

for the dataset is used for training and testing, respectively, in all ML models described in section III for training the models and performing validations.

After the initial trials with different configurations, the arrangement used in LSTM and neural networks include four hidden layers of 300 nodes each, with the ReLU function for activation

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 13 | Prediction performance of ML models in test set in S5.

Model	R^2	MSE	RMSE	MAE	MAPE
Neural network	0.15	0.04261	0.2064	0.1667	0.9011
LSTM	0.15	0.04271	0.2066	0.1681	0.9021
SVM	0.14	0.04317	0.2077	0.1659	0.9069
KNN	0.12	0.04659	0.2158	0.1710	0.9422
Random forest	0.17	0.04159	0.2039	0.1650	0.8902
AdaBoost	0.15	0.04264	0.2065	0.1682	0.9014

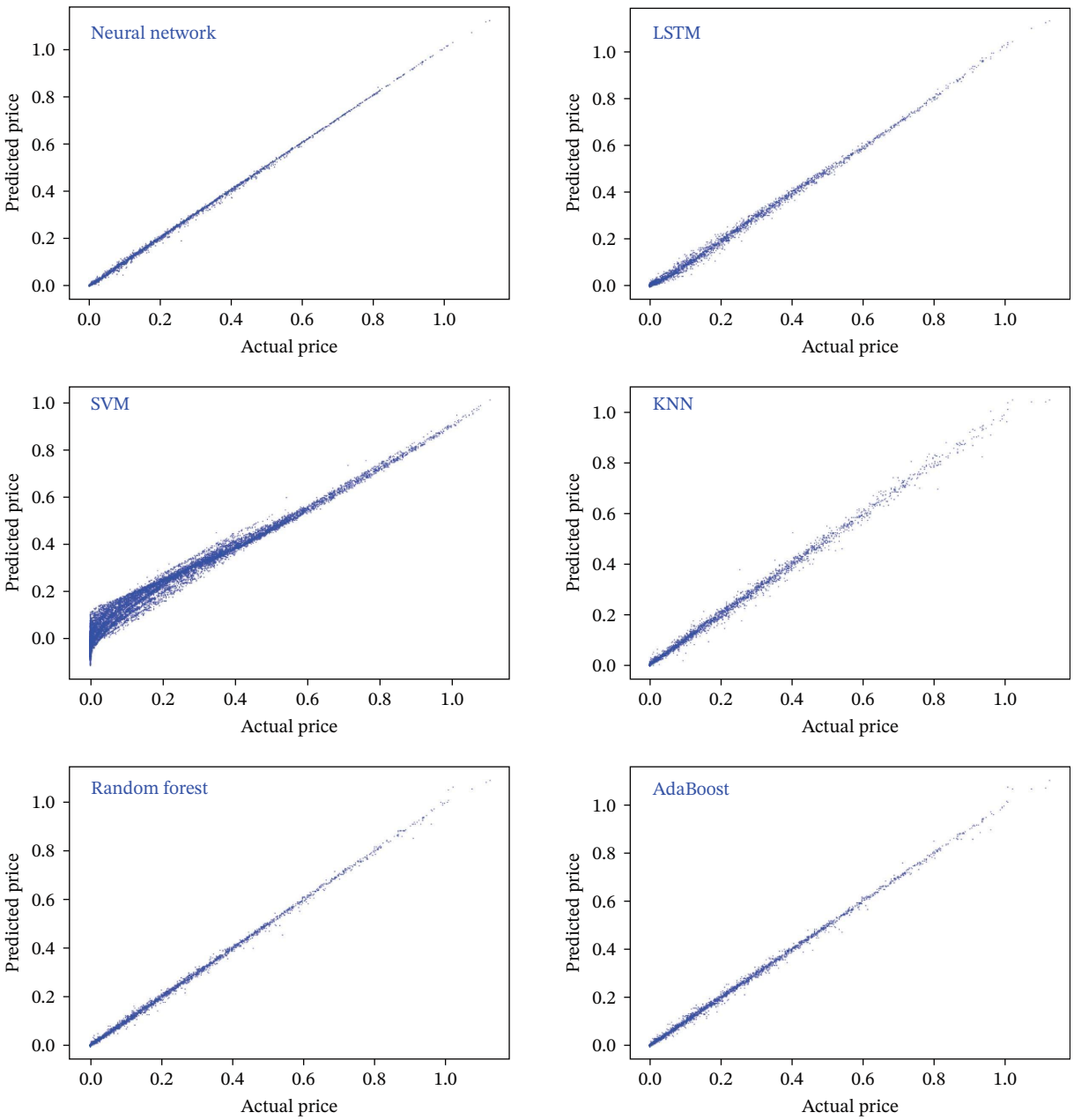


FIGURE 5 | Price prediction accuracy in test set of ML models in S1.

and Adam for optimisation. ReLU guarantees non-negative option prices and is ideal for fitting option pay offs as it is non-differentiable at zero due to its abrupt change in slope. RBF is used as the kernel function in the SVM. The RBF kernel relies on exponential Gaussian distributions; hence, the resulting price surface is smooth and infinitely differentiable. But despite its

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

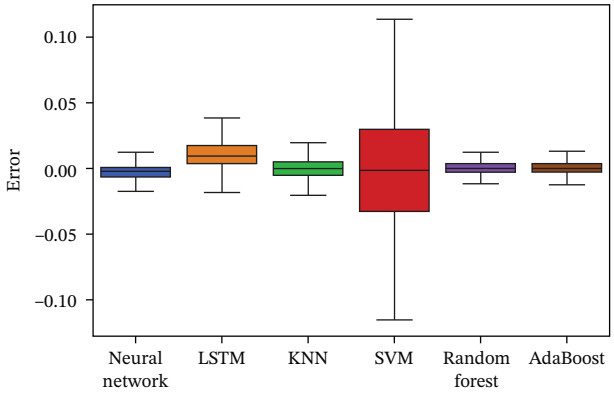


FIGURE 6 | Boxplots of errors between actual and predicted prices in test set of ML models in S1.

smoothness, the RBF kernel may create arbitrage violation risk, for example, when the input moves far away from the training data, resulting in negative option prices. In KNN, the value of K is chosen as 9 via hyperparameter tuning. The choice of K dictates the balance between statistical accuracy and strict economic validity. Finally, the number of estimators or tree depth in random forest and AdaBoost is chosen as 20. Tree depth can be useful in controlling the size and frequency of arbitrage violations on the pricing surface.

The pricing analysis is performed using six different setups to determine the effects of different inputs on the price prediction. These setups are described as S1, S2, S3, S4, S5 and S6—moneyness in Table 8. Setups 2–6 required feature selection and filtering the original data. The computation of metrics shown via

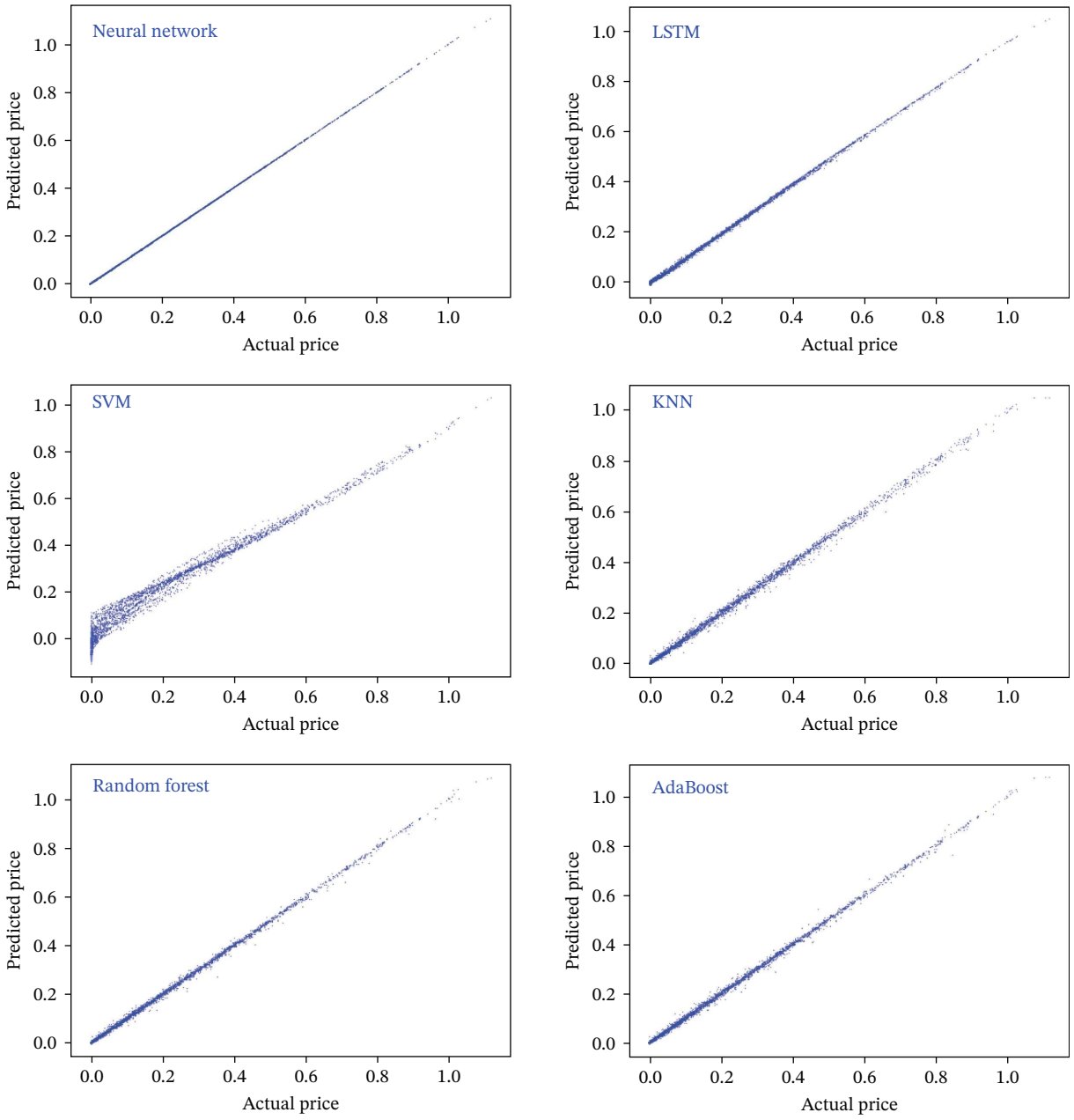


FIGURE 7 | Price prediction accuracy in test set of ML models in S2.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

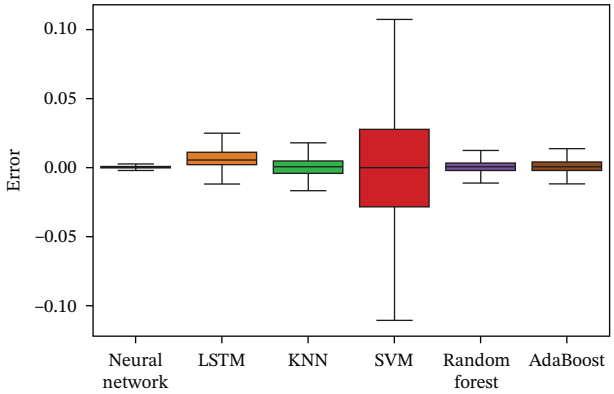


FIGURE 8 | Boxplots of errors between actual and predicted prices in test set of ML models in S2.

Equations (29)–(33) has been presented for each setup using out-of-sample data and the graphs of actual prices against the predicted prices in the test set. Boxplots of errors have also been presented in selective setups for each ML model.

4.3.1 | S1–S5

The numeric output of pricing errors of all ML models in these setups is presented in Tables 9–13. A value of R^2 close to or equal to 1 indicates a good fit of the model for predicting the dependent variable or the price using the input variables. All ML models in S1, S2, S3 and S4 show promising results. R^2 is 1 or close to 1, and other metrics are sufficiently small, with the largest RMSE observed of around 0.05 in S1–S4. The RMSE is expressed in the same unit as the target variable. In S5, the small value of

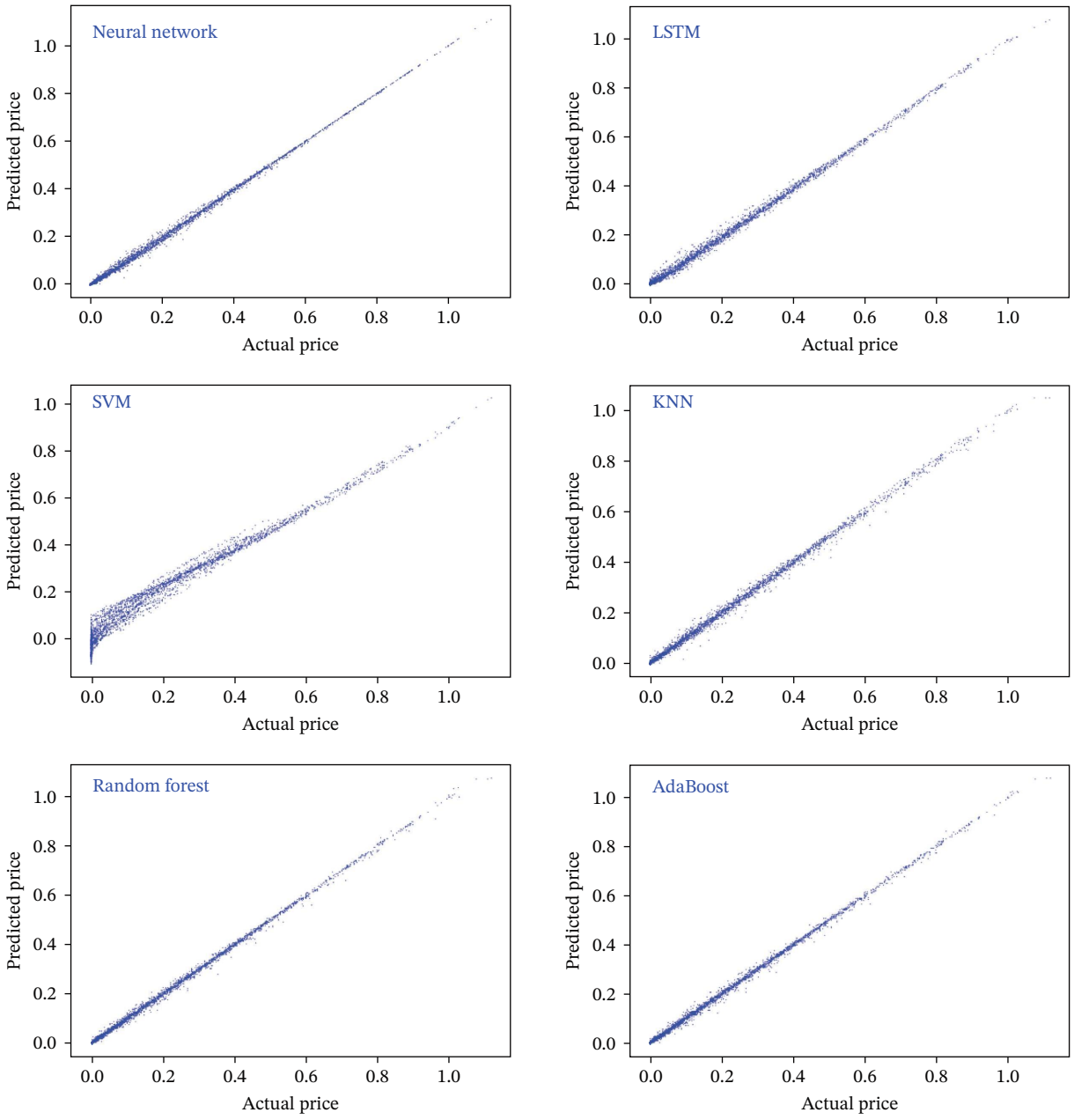


FIGURE 9 | Price prediction accuracy in test set of ML models in S3.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

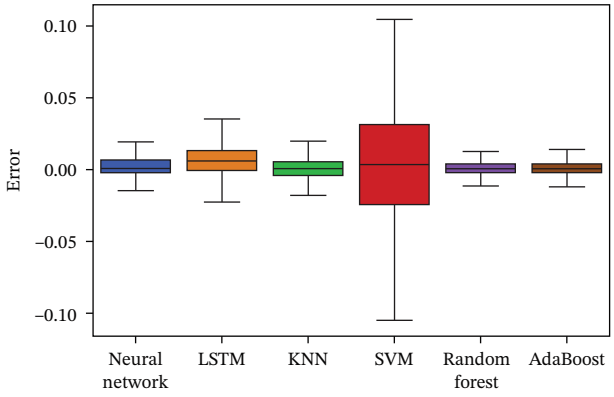


FIGURE 10 | Boxplots of errors between actual and predicted prices in test set of ML models in S3.

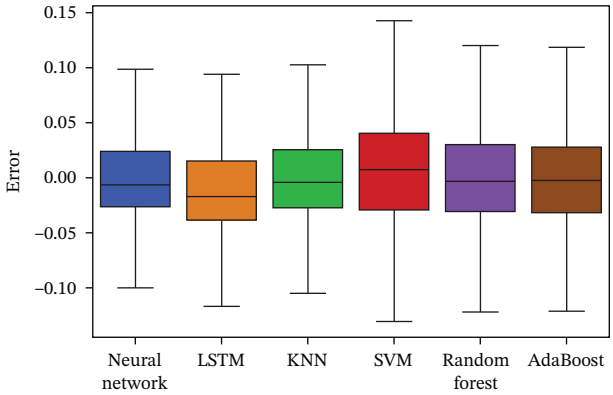


FIGURE 12 | Boxplots of errors between actual and predicted prices in test set of ML models in S4.

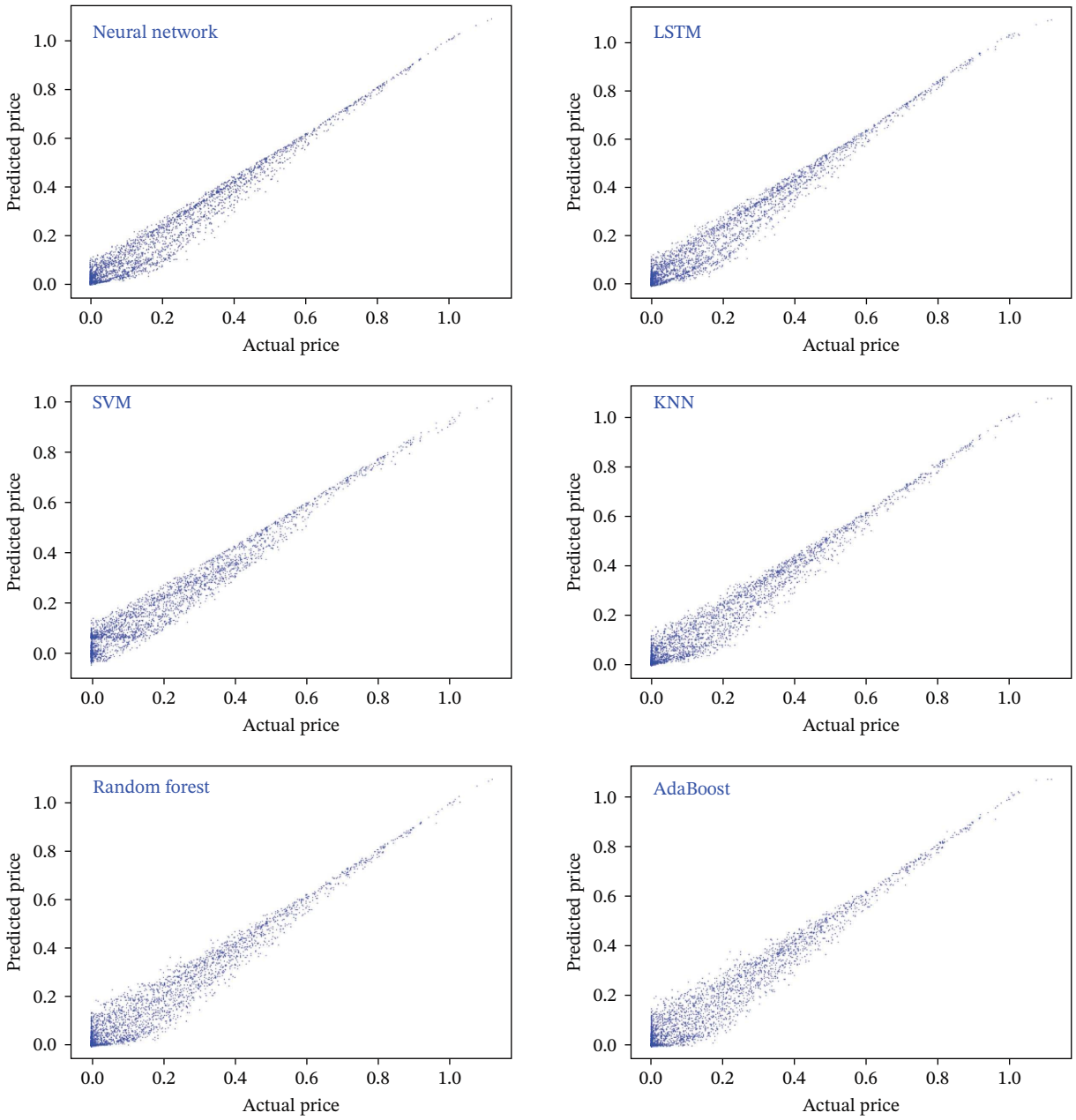


FIGURE 11 | Price prediction accuracy in test set of ML models in S4.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 14 | Performance of machine-learning models for moneyness-based option classification.

Moneyness	Model	R^2	MSE	RMSE	MAE	MAPE
ITM	Neural network	1.0	4.7988e–06	0.0021	0.0014	0.0665
	LSTM	0.91	0.00015	0.0124	0.0090	0.3779
	SVM	−2.35	0.00566	0.0752	0.0702	2.2863
	KNN	0.97	5.6410e–05	0.0075	0.0037	0.2281
	Random forest	0.99	2.0716e–05	0.0045	0.0024	0.1382
	AdaBoost	0.99	2.1891e–05	0.0046	0.0026	0.1421
ATM	Neural network	1.0	9.6613e–06	0.0031	0.0020	0.0263
	LSTM	0.9	0.00052	0.0229	0.0176	0.1947
	SVM	0.62	0.00201	0.0448	0.0370	0.3802
	KNN	0.98	0.00012	0.0109	0.0075	0.0928
	Random forest	0.98	8.9014e–05	0.0094	0.0065	0.0799
	AdaBoost	0.98	0.00010	0.0103	0.0066	0.0877
OTM	Neural network	1.0	3.2654e–05	0.0057	0.0050	0.0164
	LSTM	0.99	0.00013	0.0118	0.0087	0.0340
	SVM	0.82	0.00222	0.0472	0.0394	0.1361
	KNN	0.99	0.00015	0.0124	0.0093	0.0358
	Random forest	0.99	6.7424e–05	0.0082	0.0056	0.0236
	AdaBoost	0.99	9.3777e–05	0.0096	0.0063	0.0279
DITM	Neural network	−0.06	2.2727e–06	0.0015	0.0012	2.5721
	LSTM	−1.99	6.4138e–06	0.0025	0.0023	4.3209
	SVM	−21.29	4.7898e–05	0.0069	0.0068	11.8081
	KNN	0.46	1.1604e–06	0.0010	0.0003	1.8379
	Random forest	0.89	2.4414e–07	0.0004	0.0001	0.8430
	AdaBoost	0.66	7.2999e–07	0.0008	0.0002	1.4577

Note: Moneyness is defined as S/k, where S is the current stock price and k is the strike price. ATM, ITM, OTM and DITM denote at-the-money, in-the-money, out-of-the-money and deep-in-the-money options, respectively. Model performance is evaluated using MAE, MSE, RMSE, MAPE and R^2.

R^2 shown in Table 13 exhibits the inability of ML models to explain the proportional variations in prices due to just S_0 and r used as input. The plots of actual prices against the predicted prices in the test sets and the boxplots of errors in Figures 5–10 show that the deviations are more prominent in SVM in S1 and S2, while all other models' performance is somewhat comparable with each other. The deviations are also greater with the lower number of input parameters used as in the case of S4 (Figures 11 and 12). A slightly better performance in S2 while using HV is noticed as compared to S1 with implied volatility for this dataset. Overall, the neural network shows the best prediction results, followed by AdaBoost and random forest.

4.3.2 | S6—Moneyness

The chosen moneyness, S/k, criteria for call options in this study include: out of the money (OTM) if $0.8 \leq$ moneyness < 0.97; at the money (ATM) if $0.97 \leq$ moneyness ≤ 1.03; in the money (ITM) if $1.03 <$ moneyness ≤ 1.20 and deep in the money (DITM) if moneyness > 1.20. Filtering the options data based on these criteria led to ATM = 3276 samples, DITM = 177 samples, ITM = 4329 samples and OTM = 4295 samples.

For ITM options, the neural network produced the smallest MAPE, while the SVM produced the largest. This is also apparent in the R^2 metrics (Table 14). For DITM options, MAPE is; however, smallest for random forest. A relatively smaller sample size in DITM could be a contributory factor in the least accuracy obtained by the neural network and LSTM. For ATM and OTM options, the pricing performance of most of the models is relatively comparable with each other apart from SVM which shows the largest MAPE in relation to the other models.

5 | Conclusion

In this study, we have examined the ability of the techniques based around numerical LT inversion including Gaver–Stehfest, Weeks' and de Hoog's method, combined with a direct solver as well as data-driven approaches to value European call options

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1439795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

with real stock market and options data. The performance of the adapted numerical techniques with different forecasting models for predicting stock market volatility has been investigated. We have shown that these methods produce significantly accurate results and can be used in conjunction with statistical HV forecasting models. These methods also have proved to be more efficient in pricing European options as compared to the classical time marching FD schemes and are suitable candidates for enabling parallel processing in a distributed computational environment. In data-driven approaches, the standard ML models including SVM, KNN, random forest and AdaBoost and deep learning models constructed around feed-forward neural networks and RNN of LSTM were investigated for their applicability to price European options. It was found that these models generate plausible results with relatively small pricing errors even when the number of input parameters is reduced to just 3, as seen in setup S4. In standard ML models, which are easy to set up and configure as compared to the deep learning models, ensemble boosting and decision tree-based models of AdaBoost and random forest produced the smallest errors due to their adaptability to nonlinearities in the data, whereas SVM produced the largest. A reduced number of available samples can also obscure the performance in some metrics for both standard ML and deep learning-based approaches, as was in the case of DITM moneyness call options. Future research will focus on the integration of surrogate modelling and ML-based calibration with numerical LT solvers to further improve accuracy and computational efficiency. Future work may also consider American options, jump diffusion and Levy process models alongside integro-differential modelling for tracking asset price movement and capturing extreme market events.

Author Contributions

Nageena K. Frost: writing – reviewing and editing, conception and design, analysis, interpretation of the data. **Choi-Hong Lai**: reviewing, conception, acquisition of data.

Acknowledgments

This work was supported by the School of Computing and Mathematical Sciences, University of Greenwich, UK.

Funding

No funding was received for this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. S. Fadugba, C. Nwozo, and T. Babalola, "The Comparative Study of Finite Difference Method and Monte Carlo Method for Pricing European Option," *Mathematical Theory and Modeling* 2, no. 4 (2012): 2225–0522.

2. M. Kale, "Numerical Methods for Solving PDEs in Finance: Finite Difference Techniques for Option Pricing Models," *Advances in Consumer Research* 2, no. 4 (2025): 1533–1541.

3. L. Khodayari and M. Ranjbar, "A Computationally Efficient Numerical Approach for Multi-Asset Option Pricing," *International Journal of Computer Mathematics* 96, no. 6 (2019): 1158–1168.

4. P. Roul and V. M. K. Prasad Goura, "A Sixth Order Numerical Method and Its Convergence for Generalized Black–Scholes PDE," *Journal of Computational and Applied Mathematics* 377 (2020): 112881.

5. P. Roul and V. M. K. Prasad Goura, "An Efficient Numerical Method Based on Redefined Cubic B-Spline Basis Functions for Pricing Asian Options," *Journal of Computational and Applied Mathematics* 401 (2022): 113774.

6. P. Roul and V. M. K. Prasad Goura, "A New Higher Order Compact Finite Difference Method for Generalised Black–Scholes Partial Differential Equation: European Call Option," *Journal of Computational and Applied Mathematics* 363 (2020): 464–484.

7. A. Tagliani and M. Milev, "Laplace Transform and Finite Difference Methods for the Black–Scholes Equation," *Applied Mathematics and Computation* 220 (2013): 649–658.

8. Y. Ma, C. Z. Shi, and Y. C. Hon, "Generalized Finite Integration Method With Laplace Transform for European Option Pricing Under Black–Scholes and Heston Models," *Engineering Analysis with Boundary Elements* 164 (2024): 105751.

9. I. Iqbal, H. U. Rehman, T. Alrashdi, Y. Alrashedi, and W.-X. Ma, "Fractional Memory Effects in Nonlinear Waves: Dynamical Study of Solitons and Chaotic Attractors," *International Journal of Geometric Methods in Modern Physics* 24, no. 4 (2026).

10. I. Iqbal and H. U. Rehman, "Fractional Solitons and Climate's Secret Currents: Insights From the Generalized Fractional Kundu–Mukherjee–Naskar Equation," *Modeling Earth Systems and Environment* 12, no. 1 (2026): 69.

11. I. Iqbal, A.-M. Wazwaz, Y. Alrashedi, T. Alrashdi, and H. U. Rehman, "Dynamics of Soliton Solutions and Chaotic Behaviors of the Complex Coupled Higgs Equation Using Two Analytical Approaches," *Qualitative Theory of Dynamical Systems* 25, no. 3 (2026): 102.

12. N. Jafar, A. Jhangeer, H. U. Rehman, G. Akram, and I. Iqbal, "Enabling Technologies in Optical and Plasma Engineering Through Fractional Soliton Dynamics," *Ain Shams Engineering Journal* 17, no. 5 (2026): 104074.

13. P. R. Hansen, C. Tong, and S. Giglio, "Option Pricing With Time-Varying Volatility Risk Aversion," *The Review of Financial Studies* 39, no. 3 (2026): 875–924.

14. A. Petrozziello, L. Troiano, A. Serra, et al., "Deep Learning for Volatility Forecasting in Asset Management," *Soft Computing* 26, no. 17 (2022): 8553–8574.

15. R. Culkin and S. R. Das, "Machine Learning in Finance: The Case of Deep Learning for Option Pricing," *Journal of Investment Management* 15, no. 4 (2017): 92–100.

16. H. Zouaoui and M.-N. Naas, "Option Pricing Using Deep Learning Approach Based on LSTM-GRU Neural Networks: Case of London Stock Exchange," *Data Science in Finance and Economics* 3, no. 3 (2023): 267–284.

17. C.-F. Ivaşcu, "Option Pricing Using Machine Learning," *Expert Systems with Applications* 163 (2021): 113799.

18. Y. Liu and X. Zhang, "Option Pricing Using LSTM: A Perspective of Realized Skewness," *Mathematics* 11, no. 2 (2023): 314.

19. J. Ma, Z. Zhou, and Z. Cui, "Hybrid Laplace Transform and Finite Difference Methods for Pricing American Options Under Complex

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

Models," *Computers & Mathematics with Applications* 74, no. 3 (2017): 369–384.

20. S.-P. Zhu and J. Zhang, "Using Laplace Transform to Price American Puts," *Applications and Algorithms* 19 (2012): 447–469.

21. J. Ahn, S. Kang, and Y. H. Kwon, "A Laplace Transform Finite Difference Method for the Black–Scholes Equation," *Mathematical and Computer Modelling* 51, no. 3-4 (2010): 247–255.

22. M. Moradipour, "An Effective Algorithm to Solve Option Pricing Problems," *International Journal of Nonlinear Analysis and Applications* 12, no. 1 (2021): 261–271.

23. K. L. Kuhlman, "Review of Inverse Laplace Transform Algorithms for Laplace-Space Numerical Approaches," *Numerical Algorithms* 63, no. 2 (2013): 339–355.

24. H. Stehfest, "Numerical Inversion of Laplace Transforms," *Communications of the ACM* 13, no. 1 (1970): 47–49.

25. S. Zhu, P. Satravaha, and X. Lu, "Solving Linear Diffusion Equations With the Dual Reciprocity Method in Laplace Space," *Engineering Analysis with Boundary Elements* 13, no. 1 (1994): 1–10.

26. A. H.-D. Cheng and P. Siduruk, "Approximate Inversion of the Laplace Transform," *The Mathematic Journal* 4, no. 2 (1994).

27. P. Kano and M. Brio, "Weeks' Method for Numerical Laplace Transform Inversion With GPU Acceleration. Available on the MATLAB File Exchange," 2011, https://www.mathworks.com/matlabcentral/fileexchange/30965-weeks-method-for-numerical-laplace-transform-nversion-with-gpu-acceleration (Last Accessed 19 Jun, 2026).

28. J. A. C. Weideman, "Algorithms for Parameter Selection in the Weeks Method for Inverting the Laplace Transform," *SIAM Journal of Scientific Computing* 21, no. 1 (1999): 111–128.

29. G. Szego, "Orthogonal Polynomials," in *American Mathematical Society Colloquium Publications*, XXIII, (American Mathematical Society, 1939).

30. F. R. de Hoog, J. H. Knight, and A. N. Stokes, "An Improved Method for Numerical Inversion of Laplace Transforms," *SIAM Journal on Scientific and Statistical Computing* 3, no. 3 (1982): 357–366.

31. W. T. Lee, "Tridiagonal Matrices: Thomas Algorithm, MS6021, Scientific Computation," (University of Limerick, (Last Accessed 22 August 2025) http://www.industrial-maths.com/ms6021_thomas.pdf.

32. W. Ford, *Numerical Linear Algebra With Applications Using MATLAB* (Academic Press, 2015).

33. C. L. Defreitas and S. J. Kane, "The Numerical Inversion of the Laplace Transform in a Multi-Precision Environment," *Applied Mathematics* 13, no. 5 (2022): 401–418.

34. A. M. Cohen, "Numerical Methods for Laplace Transform Inversion," in *Numerical Methods and Algorithms*, 5, (Springer, 2007).

35. A. Kuznetsov, "On the Convergence of the Gaver–Stehfest Algorithm," *SIAM Journal on Numerical Analysis* 51, no. 6 (2013): 2984–2998.

36. M. Brio, P. O. Kano, and J. V. Moloney, "Application of Weeks Method for the Numerical Inversion of the Laplace Transform to the Matrix Exponential," *Communications in Mathematical Sciences* 3, no. 3 (2005): 335–372.

37. P. O. Kano, M. Brio, and J. Bailey, "Optimal Parameter Selection in Weeks' Method for Numerical Laplace Transform Inversion Based on Machine Learning," *Journal of Algorithms & Computational Technology* 15 (2021): 1–22.

38. S.-H. Poon and C. W. J. Granger, "Forecasting Volatility in Financial Markets: A Review," *Journal of Economic Literature* 41, no. 2 (2003): 478–539.

39. V. R. H. Pu, "*Pricing Options Using Deep Neural Networks From a Practical Perspective: A Comparative Study of Supervised and Unsupervised Learning*," (M. S. thesis, Department of Mathematics, Imperial College London, 2021).

40. A. Graves, N. Jaitly, and A. R. Mohamed, "Hybrid Speech Recognition With Deep Bidirectional LSTM," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, (IEEE, 2013): 273–278.

41. V. Vapnik, *The Nature of Statistical Learning Theory* (Springer Science & Business Media, 2013).

42. R. Goya, P. Chandra, and Y. Singh, "Suitability of KNN Regression in the Development of Interaction-Based Software Fault Prediction Models," in *International Conference on Future Software Engineering and Multimedia Engineering*, 6, (IERI Procedia, 2014): 15–21.

43. Q. Cheng, Z. Chunhong, and L. Qianglin, "Development and Application of Random Forest Regression Soft Sensor Model for Treating Domestic Wastewater in a Sequencing Batch Reactor," *Scientific Reports* 13, no. 1 (2023): 9149.

44. D. P. Solomatine and D. L. Shrestha, "AdaBoost RT: A Boosting Algorithm for Regression Problems," *IEEE International Joint Conference on Neural Networks* 2 (2004): 1163–1168.

45. D. J. Higham, *An Introduction to Financial Option Valuation: Mathematics, Stochastics and Computation* (Cambridge University Press, 2008).

46. I. Florescu and F. Viens, "Stochastic Volatility: Option Pricing Using a Multinomial Recombining Tree," (Department of Statistics, Purdue UniversityPreprint, 2006).

47. China Stock Market Accounting Research Database, 2024, https://cn.gtadata.com/.

4058, 2026, 1, Downloaded from https://onlinelibrary.wiley.com/doi/10.1155/aaa/1459795 by University Of Greenwich Stockwell Street Library, Wiley Online Library on [28/10/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License