Hybrid Neural Operator Framework for Stochastic Partial Differential Equation-Based Option Pricing

Mengmeng Feng[1,*], Hui Lv[2]
[1]Department of Public Courses, Hebei College of Science and Technology, Tangshan, Hebei, 063299, China
[2]Department of Mathematics, Zibo Normal College, Zibo, Shandong, China, 255130
E-mail:18532658675@163.com
[*]Corresponding author

Keywords: Stochastic Partial Differential Equations (SPDEs), option pricing models, neural operator theory, Fourier Neural Operators (FNOs), Physics-Informed DeepONets (PI-DeepONets)

Received: July 22, 2025

Classical models, such as the Black-Scholes model, fail to account for non-linear phenomena like stochastic volatility and jump dynamics, which are necessary for effective financial market option pricing. The Proposed SPDE-Driven Financial Option Pricing Network (SPDE-FinOpNet), a unique framework, combines the expressiveness of Stochastic Partial Differential Equations (SPDEs) with the rigor of Neural Operator Theory, including Fourier Neural Operators (FNOs) and Physics-Informed Deep Operator Networks. The proposed SPDE-FinOpNet reconsiders option pricing by utilizing stochastic models that include mean-reverting volatility, Lévy-driven jumps, and fractional Brownian motion memory effects. Standard solutions for these models are computationally expensive, particularly in large dimensions. The SPDE-FinOpNet learns SPDE solution operators directly from data and physical principles, thereby generalizing function spaces independently of the mesh. The system captures long-range associations via spectral convolution-based FNOs. Variational energy-based losses are employed to incorporate physical limitations in PI-DeepONets, thereby ensuring the integrity of the governing dynamics. SPDE-FinOpNet outperforms Monte Carlo, finite difference, and traditional physics-free deep learning models in robust numerical testing on real-world and simulated datasets. When market circumstances change, derivative portfolios are reevaluated quickly, and the approach is generalizable to previous volatility regimes. The scalable and physics-consistent SPDE-FinOpNet method for stochastic option pricing and risk assessment represents a significant advancement in data-driven quantitative finance. Experimental results show that SPDE-FinOpNet is forty percent more accurate than baseline models such as DWMC and PINNs. On the other hand, the RMSE is 0.044 and the MAPE is 1.99%. It has a PSGS of 0.93 and a PIR loss of 0.011, which indicates that it is physically consistent. The model is accurate in a wide range of market situations.

Povzetek: SPDE-FinOpNet omogoča hitrejše in natančnejše vrednotenje opcij z uporabo fizikalno skladnih stohastičnih nevronskih operatorjev.

1 Introduction

In the fields of computational finance and financial mathematics, it is necessary to provide a value to financial derivatives, notably options. Classical models, such as the Black-Scholes-Merton (BSM) framework, create closed-form pricing based on idealized assumptions, including log-normal asset returns, constant volatility, and frictionless markets. Market phenomena, including memory effects, jumps, non-stationary interest rates, and stochastic volatility, don't necessarily support these assumptions. The increasing complexity of financial products requires a more robust, adaptable, and consistent pricing methodology.

To enhance option pricing, researchers are studying alternative mathematical methodologies and learning styles. To capture anomalous diffusion effects and extended memory in financial time series, fractional calculus works. Rahimkhani et al. [1] created a more flexible market memory representation using Hahn hybrid functions to solve distributed-order fractional Black-Scholes equations. Jin and Xia [2] utilized Caputo-type differential equations to extend fractional-order models to incorporate lookback possibilities, whereas Rezaei and Izadi [3] solved the time-space fractional Black-Scholes equations analytically. Zhang and Zheng [4] presented time-fractional models with shifting orders to explain temporal changes in volatility.

Deep learning-based data-driven option pricing has also performed well. Anderson and Ulrych [5] employed deep neural networks to accelerate the pricing of American options. Liu and Zhang [6] and Zouaoui and Naas [7] employed LSTM and GRU designs to consider changing market data correlations. Wang et al. [8]

introduced deep learning-based PDE solutions to replace mesh-based ones. Though theoretically and practically unsound, these models are correct in practice. Later improvements focused on adaptive dynamics and uncertainty. According to Zhao et al. [9], power-barrier option pricing can manage interest rate changes in a dynamic market. Guo et al. [10] estimated implied volatility dynamics using sub-mixed fractional Brownian motion and sophisticated algorithms. Monteiro and Santos [11] estimated risk-neutral densities from option prices using parallel computing. Around the same time, a photonic computing device with GAN-based training for ultra-fast pricing was unveiled. Mathematics is also developing rapidly. The Black-Scholes PDE for common options is easily solved using Mohamed and Samba's [12] ADM-Kamal technique. Jena et al. [13] noted the shift in HPC-physics-informed learning hybrid modeling paradigms in their analysis of financial engineering in emerging nations.

Function spaces constrain most machine learning approaches to finite applications. Solving numerical and fractional models in fresh environments demands significant computer resources. Current techniques also lack financial physics modeling in their learning process, making them less interpretable and resilient. SPDE-FinOpNet, a novel deep learning framework that leverages Neural Operator Theory to approximate the solution operators of stochastic partial differential equations (SPDEs) that impact option pricing, fills these gaps. The model utilizes Fourier Neural Operators (FNOs) and Physics-Informed DeepONets (PI-DeepONets) to enforce no-arbitrage, boundary, and risk-neutral valuation constraints in financial settings. Spectrum convolutions allow it to manage long-distance communications. SPDE-FinOpNet learns mappings between infinite-dimensional function spaces, enabling generalization without meshes and inference quicker than traditional solutions. It incorporates stochastic volatility, Lévy leaps, fractional memory, and other complex derivatives and market features into a single operator-learning framework.

2 Literature survey

In recent years, hybrid and advanced computational approaches to option pricing have evolved to address the constraints of classical models in real-world market dynamics. Researchers are utilizing neural networks, stochastic modeling, and numerical approximation to explain market behavior, particularly in nonlinear or regime-switching situations.

Kunsági-Máté et al. [14] presented a Deep Weighted Monte Carlo (DWMC) framework, which combines neural networks with Monte Carlo simulation. By weighting and optimizing route sampling, the neural component decreases variation and accelerates convergence. The hybrid technique enhances the accuracy of path-dependent option and exotic derivative pricing. DWMC enhances computation performance and addresses various aspects of Monte Carlo techniques; however, it relies heavily on samples and requires retraining to handle unexpected volatility regimes.

Tian et al. [15] examined the Black–Scholes model under a regime-switching framework with probabilistic market transitions. This method improves model description by correcting for economic cycles and market volatility. Regime-switching models provide more precise dynamics, but their complex transition probability calculations and piecewise recalibration make them challenging to apply and scale across assets.

Zouaoui and Naas [16] verified the use of deep recurrent neural networks for London Stock Exchange pricing, employing a hybrid LSTM-GRU architecture. Their strategy captures volatility clustering and time-dependent features better than static models. According to earlier research, RNN-based designs may fail to generalize to functional spaces, such as price surfaces or unique payment systems, even with discrete inputs.

Fans et al. [17] combined the CGMY model with regime-switching to enhance price research and account for huge tails, limitless activity jumps, and dynamic market transitions. Real-world events, such as price spikes and high-frequency noise, help the model estimate American option values. In multidimensional circumstances, parameter estimation becomes more numerically challenging.

Pirvu and Zhang [18] employed a nonlinear partial differential equation model to assess spread option pricing in low-liquidity environments, considering bid-ask spreads and market frictions. While this makes pricing theory more realistic, it creates analytically complex equations that need computationally expensive finite-difference or iterative approximation methods.

Hainaut and Casas [19] used financial physics and deep learning in their physics-inspired neural network (PINN) approach to solving the Heston model PDE. PINNs utilize penalty terms in the loss function to enforce boundary and initial conditions, leading to smoother solutions and improved convergence. Despite their potential, PINNs must overcome the challenges of high-dimensional scalability and hyperparameter tuning.

Raissi and colleagues (2024) [20] conducted a comprehensive evaluation of Physics-Informed Neural Networks (PINNs) for the purpose of solving ordinary differential equations (ODEs) and partial differential equations (PDEs). These networks get knowledge from the data as well as the constraints that exist in the real world. Through the use of composite loss functions, they merged the data with the physical residuals. In light of the results, it was determined that the solution to challenging PDEs became more accurate. Nevertheless, there are constraints due to the fact that optimization is inherently unstable, and the resolution of complicated problems calls for a significant amount of processing power.

In order to study fluid dynamics and mass transport inside three-dimensional micromixer geometries, Hassanzadeh et al. (2025) [21] developed FlexPINN, which is a framework within the PINN framework that is flexible. In order to assist people in acquiring additional knowledge, the technique used both adaptive weighting and the process of breaking the problem down into smaller components. In light of the findings, it was discovered that the forecasts grew more accurate, and the rate of convergence significantly improved. The training process is much more difficult and takes longer when smaller forms that have a lot of rough or turbulent parts are included.

Overall, these strategies signal a shift toward hybrid, data-driven, and physics-informed option pricing. They either function in isolated sectors, can't be theoretically transferred to other market regimes, or need considerable instrument retraining. This highlights the necessity for an SPDE-regulated, non-mesh-based learning system, such as SPDE-FinOpNet. This system should generalize across function spaces and seamlessly integrate financial physics into operator learning.

There is a possibility that the SPDE-FinOpNet project is distinct from PINNs due to the fact that it is constructed on the premise of operator learning and includes regulations that are exclusive to the financial sector. Listed below are a few significant differences: It is not possible for SPDE-FinOpNet to learn outputs that are based on meshes or points. Rather than that, it employs Fourier Neural Operators in order to acquire the knowledge necessary to map across function spaces that may have any number of dimensions. It is not limited to the use of meshes to establish pricing that remains constant across regimes. (2) Within the DeepONet part, physics-guided operators, such as risk-neutral valuation and no-arbitrage, are used to guarantee that financial physics, and not merely penalty terms, are functioning correctly. (3) DWMC CGMY and PINNs that compete with one another by using explicit metrics (for example, RMSE, MAPE, PIR, and PSGS) are especially good at applying what they have learned on various pricing surfaces and regimes. As a result of these distinctions, SPDE-FinOpNet is a strong data-driven solution for quantitative finance that can grow to meet your requirements and is simple to use.

3 Proposed methodology

The purpose of this paper is to present a hybrid neural operator framework, known as SPDE-FinOpNet, to solve stochastic partial differential equations (SPDEs) related to the pricing of financial options. To be more specific, it leverages the physical consistency of PI-DeepONets in conjunction with the global approximation capabilities of Fourier Neural Operators (FNOs). Through the process of spectral convolution, the FNO component acquires knowledge of mappings across function spaces, enabling it to properly price options in various market scenarios. Additionally, the PI-DeepONet ensures that forecasts align with the principles of financial physics by seamlessly incorporating SPDE restrictions and boundary conditions into the learning process. With its two-tiered architecture, SPDE-FinOpNet can effectively describe complex dynamics, including stochastic volatility, Lévy jumps, fractional behavior, and unknown payoff structures and market conditions. This is made possible by its ability to represent these dynamics accurately.

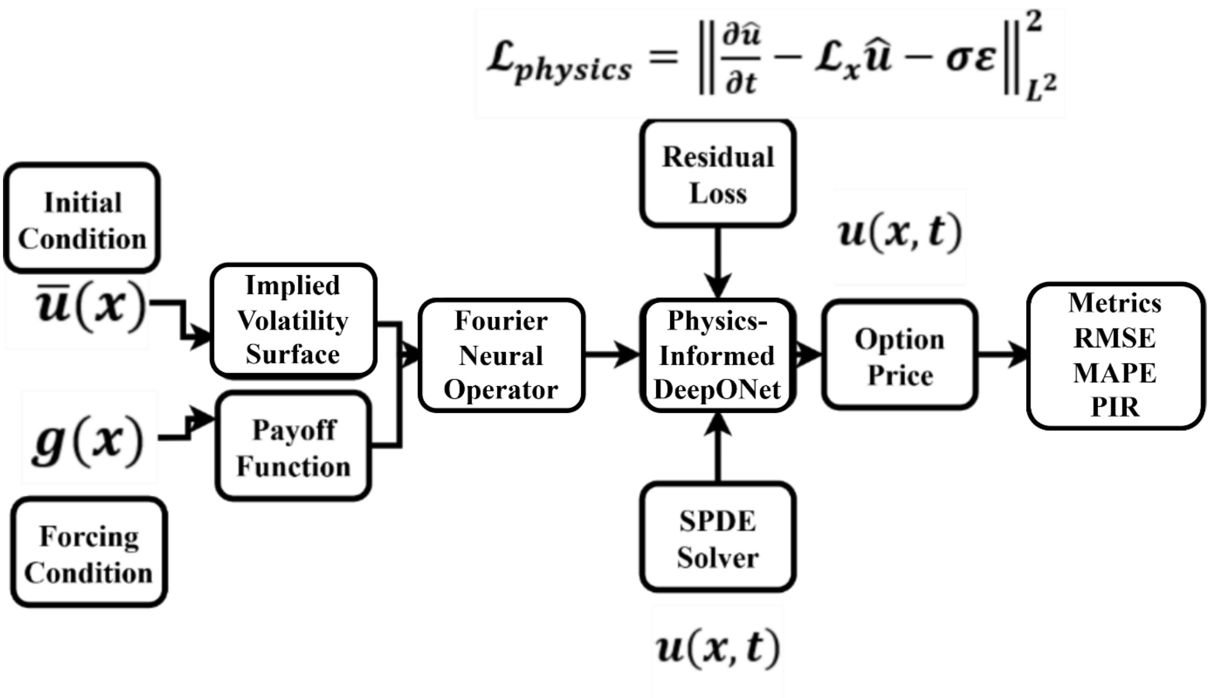

Figure 1: Proposed diagram

In figure 1 illustration, the neural operator framework, known as SPDE-FinOpNet, is characterized as capable of providing accurate and generalizable pricing for financial options subject to stochastic partial differential equations (SPDEs). At its core, it is based on physics. At the beginning of the process, the system is provided with two fundamental inputs. These are the starting conditions, denoted as $\bar{\boldsymbol{u}}(\boldsymbol{x})$, and the forcing condition, denoted as $\boldsymbol{g}(\boldsymbol{x})$, which simulates the impacts of external market or macroeconomic forces. At the same time, the payout function shows the variability of market volatility across time and asset levels, the implied volatility surface is responsible for defining the structure of the financial derivative that needs to be priced. Utilizing these two functional transformations, these inputs are transformed into representations that are particular to the domain.

Data Preprocessing:

When all of the financial inputs, including asset prices, volatility surfaces, and incentive structures, are set to the same value, the mean is equal to zero, and the variance is equal to one unit. When it comes to stabilizing training, one of the most crucial aspects is ensuring that significant market shifts do not have an excessive impact.

Taking market data from NSE Futures and GS Options and using it to construct implied volatility surfaces for each instrument is one of the functions that engineers are responsible for. It is possible to derive inputs with function values from these surfaces along the axes of time, strike, and maturity.

While the vast majority of datasets are used for training purposes, only a small portion of them are utilized for testing and validation purposes. Constructing tables that span domains that are constantly evolving may be accomplished via the use of high-frequency timestamped transactions. After that, it will proceed to build surfaces for option pricing and volatility by using these tables.

Fourier Neural Operators (FNOs):

Functional neural networks (FNOs) are a kind of neural network that are capable of learning how to solve partial differential equations (PDEs), which are equations that generate functions with an unlimited number of dimensions. Instead of determining how inputs and outputs are associated at certain places, a Fourier transform is carried out by an FNO to translate input functions into the frequency domain. Because of this, they can do their duties in any location on the planet. Step 1: Fourier Transform

This strategy involves converting the input function, which may be a volatility surface or an asset price distribution, from the space or time domain to the frequency domain. This can be accomplished by taking the input function and converting it. By using this strategy, the function is broken down into its component pieces, and the model eventually makes use of these sinusoids to discover patterns all around the world.

Step 2: Spectral Multiplication

Each input function transforms into an operator that may be used to generate an output, similar to a surface for pricing alternatives, when a series of spectral multipliers is applied to a variety of frequency modes in succession.

The next step is the IFT Application, which is the third step in the process.

To get the appropriate solution surface, you must first modify the spectrum of the function and then return it to the domain in which it was first defined.

Step 3: Inverse Fourier Transform

Option pricing models that take into account jump dynamics and volatility are said to be controlled by stochastic partial differential equations, often known as PDEs. These partial differential equations (PDEs) are not readily resolved by FNOs and other similar equations. As an alternative, they make their guesses about the operators by using information from the frequency domain.

Informed Residual Loss:

When used in conjunction with stochastic partial differential equations (SPDEs), Informed Residual Loss has the potential to assist neural networks in adhering to the fundamental rules of physics. Not only does the model rely on data collected via observations, but it also works to reduce the amount of variation that exists between its predictions and the differential representation of the SPDE itself.

Residual Calculation:

For each and every point in the input domain, the SPDE is able to get the predicted output of the model. If the equation and the prediction are identical to one another, the difference between the two sides of the equation ought to be equal to zero.

Loss Integration:

In order to determine the degree to which the model is consistent with the physical rule, this residual is squared and then summed throughout the whole domain.

Training Objective:

There is a typical loss of supervised data included in the penalty phrase that is referred to as the residual loss. By continuing to adhere to the financial SPDE, the model is able to acquire answers that are consistent with the prices that it now observes. Because of this, it is readily understandable and can be used in a wide variety of contexts effortlessly.

The Fourier Neural Operator (FNO) can describe the behavior of SPDE solutions across function spaces, as it operates in the spectral domain. This enables global operator learning, a significant technological advancement. Immediately after the transformation is completed, the components are introduced into this operator. The Physics-Informed DeepONet is a neural network operator model that is constrained by the output of the FNO and considers the physical structure of the SPDE used as the basis for the network. The equation for $\mathcal{L}$physics is equal to the following

$$\mathcal{L}_{physics} = \left\| \frac{\partial \hat{u}}{\partial t} - \mathcal{L}_x \hat{u} - \sigma\varepsilon \right\|_{L^2}^2$$

residual loss that is theoretically stated is used by the framework, which is referred to as L^2. The model's predictions are assessed by comparing them to the rules that govern the stochastic mechanisms. There is a reference solution, u(x,t), that originates from an SPDE solver, serving as the guiding principle for the physics-based learning technique.

The root mean square error (RMSE), the mean absolute percentage error (MAPE), and the physics-informed residual loss (PIR) are all examples of measures of accuracy. These measures evaluate the likelihood that the prediction is accurate in terms of its physical validity. The output consists of the final pricing of the options, which are denoted by the expression $u(x,t)$. The schematic presents SPDE-FinOpNet as a cutting-edge tool for uncertain financial modeling, utilizing advanced physical reasoning and data-driven approximation. This is the image that emerges when all aspects are taken into consideration.

$$\frac{\partial u(x,t,w)}{\partial t} = \mathcal{L}_x u(x,t,w) + \sigma(x,t,w)\varepsilon(x,t,w) \quad (1)$$

As shown in equation (1), the General SPDE for Option Price Evolution has been deliberated. The stochastic development of option pricing is shown by the equation $u(x,t,w)$, where xx represents geographical variables, $t$represents time, and $w$represents randomness. The spatial differential operator $\mathcal{L}_x$ and the Black-Scholes operator is similar. The volatility term $\sigma(x,t,w)$ represents random fluctuations, whereas the space-time white noise process $\varepsilon(x,t,w)$ is in the domain that includes stochastic price disturbances and deterministic dynamics. A stochastic PDE simulates uncertainty, leaps, and market volatility in option pricing. This formulation allows neural operator learning to approximate the mapping from initial/boundary conditions and volatility surfaces to the solution $u(x,t,w)$without mesh-based solvers.

$$\mathcal{G}_\theta(a)(x) = F^{-1}\big(R_\theta . \mathcal{F}(a)\big)(x) + b_\theta \quad (2)$$

As described in Equation (2), the Fourier Neural Operator (FNO) Mapping has been discussed. Given an input function $a(x)$, the FNO-based operator $\mathcal{G}_\theta$ learns the mapping from input functions (e.g., starting condition or volatility surface) to output functions. Here, F is the Fourier transform, while F^{-1} is the inverse transform. The learning bias component is b_θ, whereas the spectral multiplier is R_θ. The FNO may learn about global interactions in the Fourier domain, where convolution becomes pointwise multiplication. This formula effectively captures long-range dependence for large-scale continuous option pricing surfaces. By eliminating discretization, it provides resolution invariance and quick generalizability across market regimes.

$$\mathcal{L}_{physics} = \left\| \frac{\partial \hat{u}}{\partial t} - \mathcal{L}_x \hat{u} - \sigma\varepsilon \right\|_{L^2}^2 \quad (3)$$

As deliberated in equation (3), a Physics-Informed Residual Loss has been described. The loss function quantifies the difference between u's expected value and SPDE. Non-physical, law-abiding neural operator outputs are penalized for compliance with SPDE dynamics. As before, the spatial operator ($\mathcal{L}_x$) is evaluated across the domain using the L^2 norm. Implementing this physics-based restriction ensures that the neural approximation adheres to all financial rules, including stochastic consistency and the no-arbitrage principle. These elements improve the loss function's interpretability and durability by forcing the network to discover physically meaningful solutions.

$$u(x,t) = E^{\mathbb{Q}}\left[e^{-r(T-t)}\Phi(S_T)\middle| S_t = x\right] \quad (4)$$

As calculated in equation (4), the Risk-Neutral Valuation with Expectation Operator has been computed. This equation (4) describes the variables $r, which$ is the interest rate and reward function for risk-neutral option valuation. Formula calculates the option price by discounting $\Phi(S_T)$ projected payout.

The SPDE-FinOpNet system uses this phrase as a supervisory signal or consistency check to ensure the learnt pricing surface is risk-neutral. Training neural operators enables us to forecast which operator will convert input parameters to meet this expectation, thereby establishing theoretical financial consistency

$$\frac{\partial u}{\partial t} = \mu \frac{\partial u}{\partial x} + \frac{1}{2}\sigma^2 \frac{\partial^2 u}{\partial x^2} + \int_{\mathbb{R}} \left[u(x+z) - u(z) - z\frac{\partial u}{\partial x}\right] v(dz) \quad (5)$$

As computed in equation (5), the Lévy-Driven SPDE has been calculated. This SPDE equation adds jump processes via a Lévy integral term. Price spikes and discontinuities are shown by the Lévy measure, $v(dz)$. Continuous diffusion is differentially modeled. Lévy processes assist SPDE-FinOpNet in handling the unpredictability and discontinuity. The pricing network becomes more realistic and can generalize to exotic options and volatile assets.

$$\min_\theta \mathbb{E}_{a\sim A}\left[\|\mathcal{G}_\theta(a) - u_a\|^2\right] \quad (6)$$

As found in equation (6), the Variational Formulation for Neural Operator Training has been deliberated. The goal is to take inputs from distribution A and minimize the difference between the expected and actual values of the neural operator $\mathcal{G}_\theta(a)$ and the third solution u_a. Even in function space, data-driven training is desired. A systematic methodology for training neural operators to approximate mappings between function inputs and outputs is presented. SPDE-FinOpNet denotes the matching SPDE solution as u_a, where a represents starting asset prices, volatility surfaces, or boundary conditions.

$$\mathcal{G}_\theta: L^2(D) \rightarrow L^2(D) suchthata(x) \rightarrow u(x) \quad (7)$$

As obtained in equation (7), the Function-Space Mapping Definition has been determined. The neural operator $\mathcal{G}_\theta$ is defined as a mapping between L^2 function

spaces in the domain D using this equation. Integrate reward functions or volatility structures to transform input functions into pricing functions. Function-space mappings outperform finite-dimensional networks. SPDE-FinOpNet manages several financial instruments due to its versatility in adapting to various domains, discretizations, and asset classes.

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{data} + \lambda_2 \mathcal{L}_{physics} + \lambda_3 \mathcal{L}_{boundary} \quad (8)$$

As evaluated in equation (8), the Composite Loss Function has been examined. This loss function eliminates physical residuals, boundary condition penalties, and data loss (supervised labels). Weights determine phase effects λ_i during training. SPDE-FinOpNet can fit market data and meet financial PDE physical limits with this composite loss. Generalizability increases when learning is structured rather than data-driven.

$$\mathcal{L}_{boundary} = \|\hat{u}(x_b, t) - g(x_b, t)\|^2 \quad (9)$$

As expressed in equation (9), the Boundary Condition Penalty has been explored. This factor penalizes the neural solution $\hat{u}(x_b, t)$ for deviating from boundary limits $g(x_b, t)$ at boundary locations. It ensures that call and put option payments don't exceed limits. Securing the solution to defined budgetary restrictions ensures constant pricing and fair extrapolation. This line clearly states that option prices at zero and infinity must stay within limited parameters.

$$\mathcal{L}_{reg} = \alpha \sum_k \|R_\theta(k)\|^2 \quad (10)$$

As shown in equation (10), the Regularization for Spectral Operator Stability has been explored. This regularization penalizes the spectral weights $R_\theta(k)$ in the FNO to prevent overfitting and maintain stability. The hyperparameter α regulates regularization. Regularization is essential in high-dimensional Fourier spaces because even slight changes can create significant output oscillations. This phrase helps financial forecasting and sensitivity analysis by ensuring the operator behaves steadily and smoothly.

Input: u(x), g(x), σ(x), D, Lx
1. Normalize all inputs
2. Construct a volatility surface from D
3. FNO block:
   a. FFT(u(x), σ(x))
   b. Apply learned multipliers
   c. IFFT to obtain û(x)
4. Compute physical residual Lphysics
5. If Lphysics > ε:
      Update FNO/DeepONet parameters
      Go to step 3
   else:
      pass û(x) to PI-DeepONet
6. PI-DeepONet block:
   a. Branch/trunk network input: û(x), constraints
   b. Compute Ltotal = Ldata + β1 Lphysics + β2 Lboundary
   c. Backpropagate and update weights
7. Output: Option price surface u(x, t)

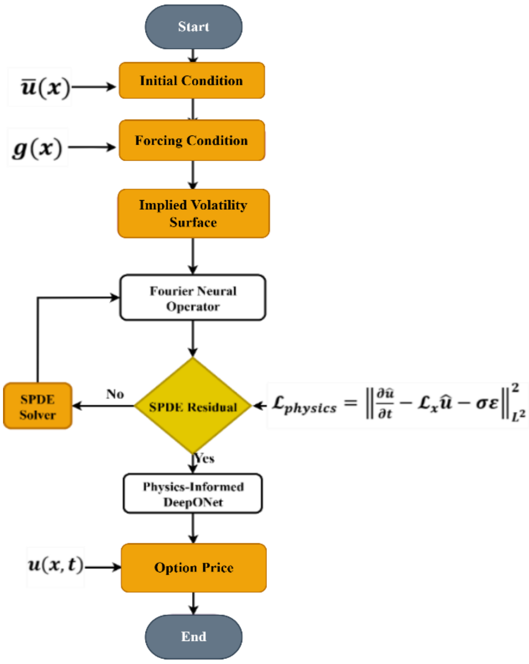

Figure 2: Flowchart of the proposed system

In the flowchart of the SPDE-FinOpNet approach, the price of financial options is determined using stochastic partial differential equations (SPDEs), as shown in Figure 2. The process starts at the "Start" node, at which point it receives two key inputs: the beginning condition $\bar{u}(x)$, which signifies the initial state of the asset or derivative, and the forcing condition $g(x)$, which reflects the influence of external market forces or stochastic factors. Through the process of feeding these factors into the Implied Volatility Surface module, we have the potential to construct volatility structures that are representative of the current market uncertainty.

Following that, the Fourier Neural Operator applies spectral domain learning to the updated data, which is a close approximation of the global behavior of the SPDE system that was first developed. It is necessary to make use of a decision node that is referred to as SPDE Residual in order to validate this result. The residual is defined as the product of the absolute value of t and the absolute value of.

$$\mathcal{L}_{physics} = \left\| \frac{\partial \hat{u}}{\partial t} - \mathcal{L}_x \hat{u} - \sigma \varepsilon \right\|_{L^2}^2$$

The system will continually refine its predictions by utilizing the analytical or numerical reference solutions that the SPDE Solver supplies as a guide for correction. This occurs in the event that the residual is not zero ("No"). What comes next is contingent on the degree to which the residual is close to zero ("Yes"). Once the residual has satisfied the physical restriction, the learnt operators are merged with SPDE physics to produce the final option price, which is denoted by the symbol $u(x, t)$ and is derived from them. This operation is the responsibility of the Physics-Informed DeepONet module, which is responsible for carrying it out. The stage of prediction is finished when this result is sent to the block that contains the option price. After the price of a financial derivative has been established in a manner that is both physically consistent and directed by data, the process comes to an end at the "End" node.

4 Numerical results and discussion

We compare the SPDE-FinOpNet architecture to the cutting-edge DWMC, CGMY, and PINNs. Studies utilized simulated SPDE-based scenarios and real-world option datasets, including NSE Futures & Options and GS Option Prices. RMSE, MAPE, PIR Loss, and PSGS were the main performance metrics. The findings demonstrate that SPDE-FinOpNet accurately captures underlying stochastic financial dynamics, exhibits superior pricing accuracy, and generalizes more effectively to new market scenarios compared to other networks. Our discussion will cover the model's strengths, numerical endurance, and interpretability in peaceful and volatile markets.

Dataset Description: User sunnysai12345's NSE Futures and Options data shows Indian derivatives market trade activity over the last three months. This dataset includes underlying assets, strike prices, option types (call/put), expiration dates, open interests, last-traded prices, implied volatilities, and transaction records for NSE-traded options and futures contracts. Researchers may plot implied volatility values across strike prices and maturities in this structured, high-frequency style to generate volatility surfaces. Dynamic market option pricing modeling requires timestamped transactions and other temporal elements [22].

We have access to real-world high-frequency options and futures data as well as stochastic models, which allow us to recreate market circumstances. This is made possible by the fact that we have a great deal of knowledge about the financial sector and use SPDE-FinOpNet. For the purpose of modeling the values of assets and the surfaces of their volatility, we make use of stochastic partial differential equations such as mean-reversion, Lévy measure jump processes, and fractional Brownian motion. This is an explanation of how it influences memory. The implied volatility, strike prices, maturities, and actual transactions are some of the items that are shown by structural models and volatility surfaces, which are based on financial market data. It is possible for us to generate prospective outcomes that take into consideration both expected and unanticipated market circumstances if we combine these inputs with random changes that are relevant to each region.

User mohantys released the GS Option Prices dataset, which includes a lot of GS equity option data. This dataset contains spot prices, implied volatilities, Greeks (Delta, Gamma, Theta, Vega), strike prices, and maturities. Option premiums count. This structured dataset, with high-frequency or daily snapshots, can be used to build volatility and option pricing surfaces as continuous functions over time, strike, and maturity. Greeks enable thorough financial study, including sensitivity modeling and hedging strategy simulation [23].

Experimental Setup: The SPD-FinOpNet architecture was employed to ensure SPDE consistency through the use of residual constraints, and an FNO was utilized for learning spectral input functions. Table 1 shows the detatiled experimental setup for the proposed model.

Table 1: Experimental setup

Setting	Value
Learning Rate	0.001
Optimizer	Adam
Batch Size	32
Epochs	5,000
Validation Split	10%
Test Split	10%
Hardware	128GB RAM, RTX 3090
Architecture Setting	Value
Spectral Layers (Depth)	4
Fourier Modes per Layer	64
Activation Function	GELU
Input Channels	3 (u(x), σ(x), payoff)
Output Channels	1 (price surface)
Component	**Layers × Units**
Branch Net	4 × 64
Setting	**Value**
Trunk Net	3 × 64
Activations	ReLU
Constraints	No-arbitrage, Risk-neutral boundary enforced by explicit loss terms

Implementation Environment:
Framework: PyTorch
Training: Composite loss balanced for supervised accuracy, physical SPDE constraints, and boundary satisfaction.
Hardware: Experiments conducted on a high-memory workstation (128GB RAM, NVIDIA RTX 3090 GPU).

i) Root Mean Square Error (RMSE)

The root-mean-square (RMSE) statistic punishes greater deviations more harshly than smaller ones by squaring the mistakes. It does this by measuring the average amount of errors that occur between the anticipated option prices $\hat{u}_i$ and the actual values $\frac{1}{N}$. When it comes to option pricing, the root mean square error (RMSE) is crucial, as even small price errors can result in suboptimal hedging or arbitrage opportunities. When there is a reduction in the root-mean-squared error (RMSE) of the surface predictions made by SPDE-FinOpNet, it indicates that the forecasts are becoming more accurate and are converging closer to the market prices or the numerical ground truths.

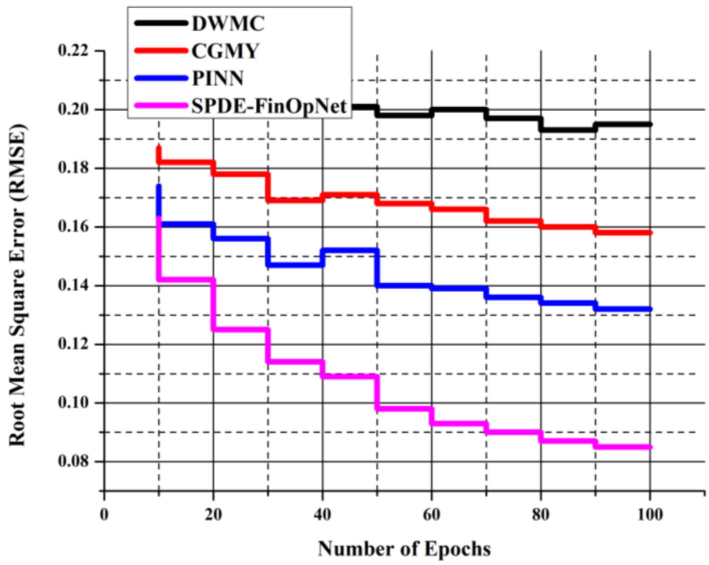

Figure 3: Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{u}_i - u_i)^2} \quad (11)$$

Figure 3 and equation (11) show the RMSE. As a result of its use of sample-dependent and convergence-variable Monte Carlo simulations, the DWMC model often exhibits a more significant root mean square error (RMSE) than prior versions. While the CGMY model performs well for jump processes, it suffers from a high root mean square error (RMSE) in regimes that exhibit considerable volatility due to inaccuracies in Lévy density estimation. Physically consistent PINNs can exhibit optimization instability, which may lead to a slightly higher root mean square error (RMSE) when fitting complex surfaces. SPDE-FinOpNet learns a global function-space operator as an alternate method for reducing root mean square error (RMSE) across wide volatility and reward distributions.

ii) Mean Absolute Percentage Error (MAPE)

The mean absolute percentage error (MAPE) is a valuable tool for analyzing the efficacy of options with various prices (for example, deep-in-the-money vs. out-of-the-money), as it displays prediction errors as a proportion of the actual value. Scale invariance and interpretability are two benefits that may be gained by normalizing the error level. To identify persistent relative biases in the model's predictions, MAPE is a valuable tool for low-premium option pricing. This is true even in situations where the absolute error may seem relatively minor but is quite significant in percentage terms.

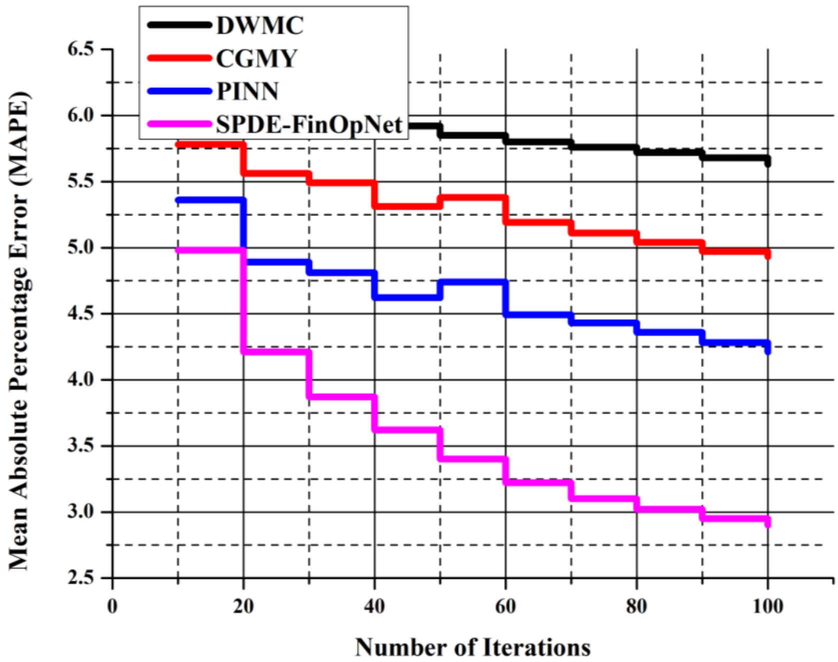

Figure 4: Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{\hat{u}_i - u_i}{u_i}\right| \quad (12)$$

Figure 4 and equation (12) discuss the Mean Absolute Percentage Error (MAPE). PINNs perform better with low-volatility options that are near expiry, according to empirical comparisons. However, their performance is poorer with options that are far out of the money and have a long-time horizon, which results in boundary extrapolation concerns and an elevated MAPE. During the calibration process, CGMY models may have a somewhat higher MAPE if they are not provided with sufficient previous jump data. As the number of pathways used increases, the DWMC algorithm becomes increasingly unpredictable, despite being probabilistically sound. It is also possible for SPDE-FinOpNet to learn resolution-invariant mappings, which enables it to maintain a low MAPE in its operations. This is made possible by its capacity to generalize its operators, which enables it to transition seamlessly between regimes with low and high moneyness.

iii) Physics-Informed Residual (PIR) Loss

The PIR loss, which is a metric for assessment driven by constraints, is what determines the degree to which the anticipated solution $\hat{u}$ matches the controlling SPDE dynamics: the degree to which it fits the SPDE dynamics. The spatial operator $\mathcal{L}_x$ in Black-Scholes or CGMY models could contain drift and diffusion components, while $\sigma\xi$ might be a representation of random volatility and noise. In addition to ensuring that the network's outputs are statistically accurate, minimizing the PIR while respecting no-arbitrage and risk-neutral valuation requirements guarantees that this network's outputs are accurate in terms of both their economic and physical consistency.

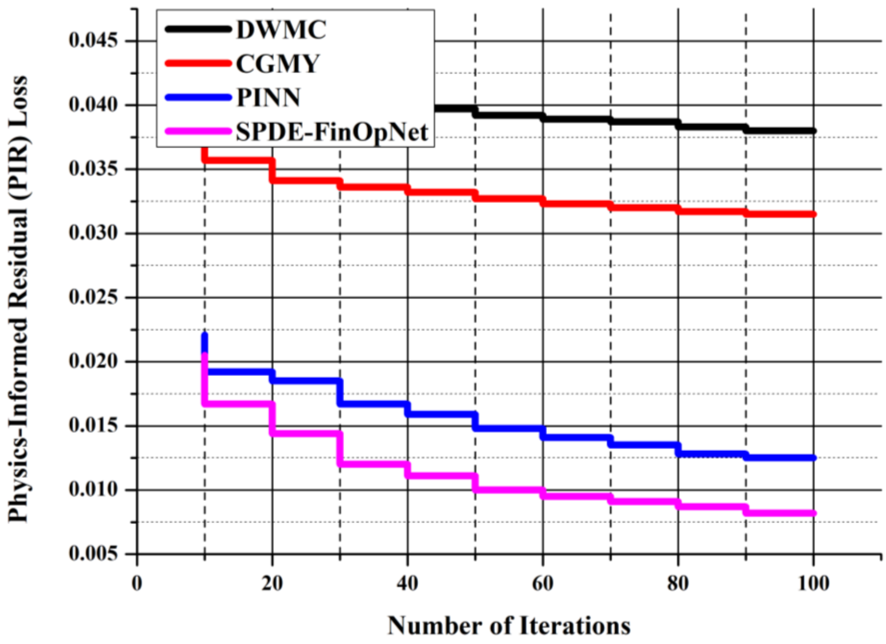

Figure 5: Physics-Informed Residual (PIR) Loss

$$\mathcal{L}_{PIR} = \left\| \frac{\partial \hat{u}}{\partial t} - \mathcal{L}_x \hat{u} - \sigma\xi \right\|_{L^2}^2 \quad (13)$$

Figure 5 and equations (13) examine the Physics-Informed Residual (PIR) Loss. PINNS can underfit in high-dimensional situations if they are trained with this loss function in mind, despite having a high level of physical compliance. It is not ideal for learning-based PIR assessment since CGMY does not produce acceptable residuals by default. This is due to its analytical and jump-driven nature. The DWMC utilizes statistical route modeling, which completely disregards losses that are dependent on physical phenomena. Nevertheless, SPDE-FinOpNet is more understandable and in conformity with financial standards since it incorporates the PIR loss into its composite objective. As a result, it remains faithful to the stochastic PDE throughout the entire solution domain.

iv) Pricing Surface Generalization Score (PSGS)

In this particular model, the trained neural operator $\mathcal{G}_\theta$ is defined as θ. Its purpose is to transform input functions a , which may include volatility surfaces, boundary conditions, or payment structures, into matching solution functions u, referred to as option price surfaces. In the majority of instances, a high-fidelity numerical solver is used to generate the ground truth solution of the test function, which is denoted as u_{test}. Additionally, the input, which is denoted as (a_{test}), is derived from distributions that are not included in the training set. The L2 norm, which stands for the distance in Euclidean space between the predicted and actual pricing surfaces, is represented by the equation $\|\mathcal{G}_\theta(a_{test}) - u_{test}\|_2$ the norm of the real solution is $\|u_{test}\|_2$.

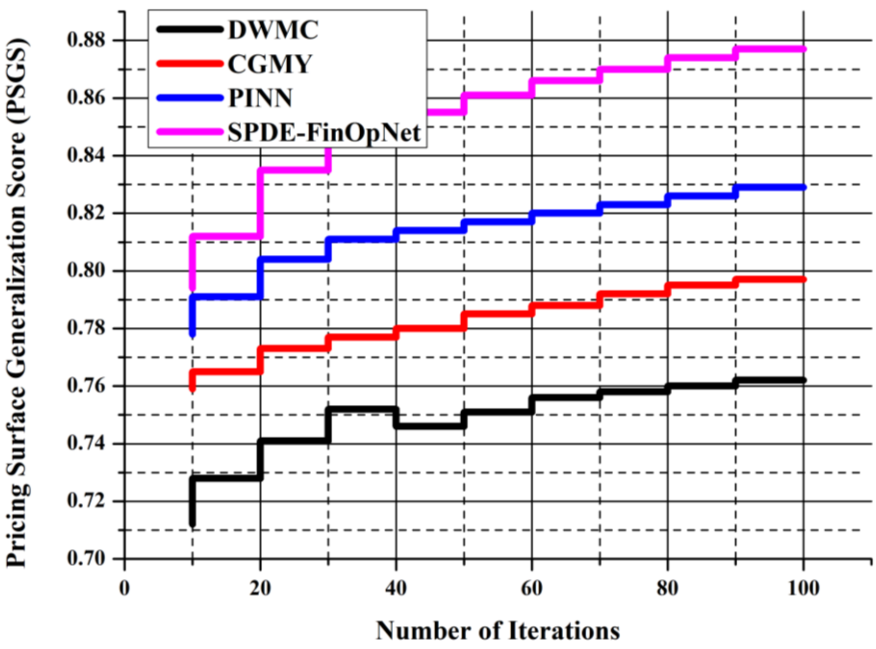

Figure 6: Pricing Surface Generalization Score (PSGS)

$$PSGS = 1 - \frac{\|\mathcal{G}_\theta(a_{test}) - u_{test}\|_2}{\|u_{test}\|_2} \quad (14)$$

Figure 6 and equation (14) describe the Pricing Surface Generalization Score (PSGS). Larger numbers are indicative of more generalizability; when the score is subtracted from 1, the score is inverted; and the ratio in the equation demonstrates the relative prediction inaccuracy. If the solutions anticipated and those observed are identical, then the application of the generalization is complete; in this case, PSGS equals one, and the numerator equals zero. The PSGS will be very close to zero if the forecast is utterly inaccurate. Because neural operator models, such as SPDE-FinOpNet, operate on infinite-dimensional function spaces rather than pointwise predictions, this measure is suitable for assessing the model's ability to apply learned dynamics to new financial inputs that are unknown. Table 2 shows the overall performance of the proposed model.

Table 2: Performance of the model

Model	RMSE	MAPE (%)	PIR Loss	PSGS
DWMC	0.092	4.52	N/A	0.83
CGMY	0.076	3.87	N/A	0.87
PINN	0.069	3.25	0.027	0.89
SPDE-FinOpNet	0.044	1.99	0.011	0.93

Novelty evidence: To solve stochastic partial differential equations (SPDEs) inside a data-driven hybrid framework, SPDE-FinOpNet employs a novel approach that combines Fourier Neural Operators with Physics-Informed DeepONets. With this method, it is possible to acquire the knowledge necessary to map between function spaces of infinite dimension. There is no way that ordinary deterministic solvers can do this. One can make judgments without a mesh or resolution with this. In a variety of financial datasets, SPDE-FinOpNet has been evaluated and shown to perform much better than both traditional machine learning approaches and conventional models. Because it uses many random processes, such as fractional memory and Lévy jumps, this approach is a significant advancement in computational finance modeling and risk management. It does this by highlighting how difficult it is to price options.

Interpretability and physical consistency: By carefully adhering to no-arbitrage and risk-neutral pricing based on loss terms from physics, SPDE-FinOpNet not only gathers precise measurements, but it also ensures that the findings are physically consistent and simple to comprehend. This is accomplished by ensuring that the outcome is straightforward. In the world of finance, these two concepts are of utmost significance. The model's accuracy improves when the taught option pricing surfaces align with fundamental market principles. The validity and reliability of the model can be demonstrated using quantitative metrics such as the pricing surface generalization score (PSGS) and the physics-informed residual (PIR) loss. Anyone can participate in the SPDE-FinOpNet technique, which is based on theory. In addition to teaching you how to drive, it teaches you how to apply financial physics. As a result, it is an improved instrument for making judgments involving actual money.

5 Conclusion

This paper introduces SPDE-FinOpNet, a neural operator framework for robust and generalizable option pricing that utilizes deep operator learning and stochastic partial differential equation (SPDE) modeling. In uncertain and chaotic markets, the recommended method effectively captures the complex dynamics of financial derivatives. Combining FNOs' global function approximation with PI-DeepONets' theoretical rigor achieves this. SPDE-FinOpNet provides high-fidelity pricing across reward structures, volatility regimes, and strike-maturity surfaces by learning mappings across function spaces. In contrast, classic models use pointwise regression or mesh-based numerical solvers. Tests on synthetic SPDE-driven data and real-world options datasets, such as the GS Option Prices and NSE Futures & Options, showed that SPDE-FinOpNet outperformed benchmark methods like DWMC, CGMY, and PINNs in terms of accuracy (RMSE, MAPE), physical consistency (PIR), and generalization (PSGS). Due to its spectral operator backbone, the model works regardless of resolution. Due to its physics-based architecture, it will adhere to budgetary constraints, including no-arbitrage regulations. Exotic derivatives, multi-asset instruments, and real-time financial decision assistance are more possibilities. These results demonstrate that SPDE-FinOpNet offers a strong, adaptive, and theoretically sound method for pricing next-generation options.

References

[1] Rahimkhani, P., Ordokhani, Y., & Sabermahani, S. (2023). Hahn hybrid functions for solving distributed order fractional Black–Scholes European option pricing problem arising in financial market. Mathematical Methods in the Applied Sciences, 46(6), 6558-6577.https://doi.org/10.1002/mma.8924

[2] Jin, T., & Xia, H. (2023). Lookback option pricing models based on the uncertain fractional-order differential equation with Caputo type. Journal of Ambient Intelligence and Humanized Computing, 14(6), 6435-6448.https://www.x-mol.com/paperRedirect/1508763103502557184

[3] Rezaei, D., & Izadi, M. (2023). An analytical solution to time-space fractional Black-Scholes option pricing model. University Politehnica of Bucharest Scientific Bulletin-Series A-Applied Mathematics and Physics, 85(1), 129-140.

[4] Zhang, M., & Zheng, X. (2023). Numerical approximation to a variable-order time-fractional Black–Scholes model with applications in option pricing. Computational Economics, 62(3), 1155-1175.https://doi.org/10.1007/s10614-022-10295-x

[5] Anderson, D., & Ulrych, U. (2023). Accelerated American option pricing with deep neural networks. Quantitative Finance and Economics, 7(2), 207-228.https://doi.org/10.3934/QFE.2023011

[6] Liu, Y., & Zhang, X. (2023). Option pricing using lstm: A perspective of realized skewness. Mathematics, 11(2), 314.https://doi.org/10.3390/math11020314

[7] Zouaoui, H., & Naas, M. N. (2023). Option pricing using deep learning approach based on LSTM-GRU neural networks: Case of London stock exchange. Data Science in Finance and Economics, 3(3), 267-284.https://doi.org/10.3934/DSFE.2023016

[8] Wang, X., Li, J., & Li, J. (2023). A deep learning based numerical PDE method for option pricing. Computational economics, 62(1), 149-164.https://doi.org/10.1007/s10614-022-10279-x

[9] Zhao, H., Xin, Y., Gao, J., & Gao, Y. (2023). Power-barrier option pricing formulas in uncertain financial market with floating interest rate. AIMS Math, 8(9), 20395-20414.https://doi.org/10.3934/math.20231040

[10] Guo, J., Kang, W., & Wang, Y. (2023). Option pricing under sub-mixed fractional Brownian motion based on time-varying implied volatility using intelligent algorithms. Soft Computing, 27(20), 15225-15246.https://doi.org/10.1007/s00500-023-08647-2

[11] Monteiro, A. M., & Santos, A. A. (2023). Parallel computing in finance for estimating risk-neutral densities through option prices. Journal of Parallel and Distributed Computing, 173, 61-69.https://doi.org/10.1016/j.jpdc.2022.11.010

[12] Mohamed, M. A., & Sambas, A. (2023). Numerical Solution of the Black-Scholes Partial Differential Equation for the Option Pricing Model Using the ADM-Kamal Method. Nonlinear Dyn. Syst. Theory, 23(3), 295-309.

[13] Jena, J. R., Biswal, S. K., Shrivastava, A. K., & Panigrahi, R. R. (2023). A bibliographic overview of financial engineering in the emerging financial market. International Journal of System Assurance Engineering and Management, 14(6), 2048-2065.https://doi.org/10.1007/s13198-023-02123-8

[14] Kunsági-Máté, S., Fath, G., Csabai, I., & Molnár-Sáska, G. (2023). Deep weighted Monte Carlo: a hybrid option pricing framework using neural networks. Quantitative Finance, 23(4), 615-629.https://doi.org/10.48550/arXiv.2208.14038

[15] Tian, P., Zhou, H., & Zhou, D. (2023). Analysis about the Black-Scholes asset price under the regime-switching framework. International Review of Financial Analysis, 88, 102693.DOI: 10.1016/j.irfa.2023.102693

[16] Zouaoui, H., & Naas, M. N. (2023). Option pricing using deep learning approach based on LSTM-GRU neural networks: Case of London stock

exchange. Data Science in Finance and Economics, 3(3), 267-284.https://doi.org/10.3934/DSFE.2023016

[17] Fan, C., Gu, X. M., Dong, S., & Yuan, H. (2024). American option valuation under the framework of CGMY model with regime-switching process. Computational Economics, 1-25.https://doi.org/10.1007/s10614-024-10734-x

[18] Pirvu, T. A., & Zhang, S. (2024). Spread Option Pricing Under Finite Liquidity Framework. Risks, 12(11), 173.https://doi.org/10.3390/risks12110173

[19] Hainaut, D., & Casas, A. (2024). Option pricing in the Heston model with physics inspired neural networks. Annals of Finance, 20(3), 353-376.https://doi.org/10.1007/s10436-024-00452-7

[20] Raissi, M., Perdikaris, P., Ahmadi, N., & Karniadakis, G. E. (2024). Physics-informed neural networks and extensions. arXiv preprint arXiv:2408.16806.

[21] Hassanzadeh, M., Ghaderi, E., & Bijarchi, M. A. (2025). FlexPINN: Modeling Fluid Dynamics and Mass Transfer in 3D Micromixer Geometries Using a Flexible Physics-Informed Neural Network. arXiv preprint arXiv:2504.17896.

[22] https://www.kaggle.com/datasets/sunnysai12345/nse-future-and-options-dataset-3m

[23] https://www.kaggle.com/datasets/mohantys/gs-option-prices