PHYSICS-INFORMED NEURAL NETWORK-BASED OPTION PRICING

SERHAT ALTINDAĞ

ÖZYEĞİN UNIVERSITY

DECEMBER, 2025

PHYSICS-INFORMED NEURAL NETWORK-BASED OPTION PRICING

by
Serhat Altındağ

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Data Science

Advisor: Assoc. Prof. Emre Sefer

Graduate School of Science and Engineering
Özyeğin University
İstanbul

December, 2025

PHYSICS-INFORMED NEURAL NETWORK-BASED OPTION PRICING

Approved by:

Assoc. Prof. Emre Sefer, Advisor
Department of Artificial Intelligence and Data Engineering
Özyeğin University

Prof. Taylan Akdoğan
Department of Natural and Mathematical Sciences
Özyeğin University

Prof. Burak Saltoğlu
Department of Economics
Boğaziçi University

Approval Date: December 18, 2025

To my wife and family

DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Serhat Altındağ

ABSTRACT

Option pricing is an important problem in finance. The most common option pricing methodologies are Black-Scholes, CEV, Heston, and SABR, where all of them can be described by Partial Differential Equations. On the other hand, as a deep learning technique, Physics-Informed Neural Networks (PINNs) have shown great potential in providing approximate solutions to Partial Differential Equations (PDEs). In this study, we come up with Physics-Informed Neural Network-based option pricing methods under these 4 pricing models. We evaluate our method using both simulated and actual market data, and assess its performance against analytical and numerical reference models. The empirical results for index options such as RUT and SPX, as well as index constituent companies options, show that the proposed deep learning models have lower absolute errors than the classical Black–Scholes model solution. Our proposed solution also performs better than the recently proposed PINNsFormer for the option pricing task. Our results on both European and American options show the potential importance of Physics-Informed Neural Networks in solving option pricing, which can also be extended to other financial asset pricing problems. Our methods, analysis code, and datasets are available at: `https://github.com/seferlab/PINN_for_Options`

Keywords: Option pricing, Physics-Informed Neural Networks, Deep Learning, Black-Scholes, SABR

ÖZET

Option fiyatlama, finans alanında önemli bir problemdir. En yaygın kullanılan fiyatlama yöntemleri Black–Scholes, CEV, Heston ve SABR modelleri olup, bu modellerin tamamı Kısmi Diferansiyel Denklemler ile ifade edilebilmektedir. Diğer yandan, bir derin öğrenme tekniği olarak Fizik-Temelli Sinir Ağları (Physics-Informed Neural Networks, PINNs), Kısmi Diferansiyel Denklemlere yaklaşık çözümler üretme konusunda büyük bir potansiyel sergilemektedir. Bu çalışmada, söz konusu dört fiyatlama modeli altında PINN tabanlı opsiyon fiyatlama yöntemleri geliştirilmiştir. Yöntemimiz hem simülasyon verileri hem de gerçek piyasa verileri üzerinde değerlendirilmiş; analitik ve sayısal referans modellerle karşılaştırmalı bir performans analizi gerçekleştirilmiştir. RUT ve SPX gibi endeks opsiyonları ile endeks bileşeni şirket opsiyonlarına ilişkin ampirik sonuçlarımız, önerilen derin öğrenme modellerinin mutlak hata bakımından klasik Black–Scholes modeline kıyasla daha başarılı olduğunu göstermektedir. Önerdiğimiz yöntem ayrıca, opsiyon fiyatlama görevi için yakın dönemde sunulan PINNsFormer modelinden de daha iyi performans sergilemektedir. Hem Avrupa tipi hem de Amerikan tipi opsiyonlar üzerinde elde ettiğimiz bulgular, PINN yöntemlerinin opsiyon fiyatlama probleminin çözümündeki potansiyel önemine işaret etmekte olup, bu yaklaşım diğer finansal varlık fiyatlama problemlerine de genişletilebilir. Yöntemlerimiz, kodlarımız ve veri setlerimiz şu adreste sunulmaktadır: `https://github.com/seferlab/PINN_for_Options`

Anahtar Kelimeler: Opsiyon fiyatlaması, Fizik Bilgili Sinir Ağları, Derin Öğrenme, Black–Scholes, SABR

ACKNOWLEDGEMENTS

I would like to sincerely thank my advisor, Assoc.Prof. Emre Sefer, for his continuous support, patience, and understanding throughout my thesis. This thesis was completed thanks to his support and belief in me.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY iv

ABSTRACT v

ÖZET vi

ACKNOWLEDGEMENTS vii

LIST OF TABLES x

LIST OF FIGURES xi

LIST OF ACRONYMS AND ABBREVIATIONS xiii

1 INTRODUCTION 1

1.1 Related Work 4

2 OPTION PRICING MODELS IN FINANCE 7

2.1 Black-Scholes Model 7

2.2 CEV Model 9

2.3 Heston Model 10

2.4 SABR Model 12

3 PHYSICS-INFORMED NEURAL NETWORKS 14

4 EXPERIMENTAL SETUP 19

4.1 Implementation 21

4.2 Performance Evaluation 22

5 RESULTS AND DISCUSSION 23

5.1 Results 23

5.1.1 Analysis of Activation Functions Using Random Datasets 23

5.1.2 Analysis of SABR-PINN Using Synthetic Dataset for Different β and ν 28

5.1.3 Comparison of All Option Pricing Models 29

5.1.4 Real Market Dataset - European Style Options 31

5.1.5 Real Market Dataset - American Style Options 35
5.1.6 Comparison of PINNs and PINNsFormer 40
5.2 Discussion 43
6 CONCLUSION 46
REFERENCES 48

LIST OF TABLES

Table 4.1: Random dataset and synthetic dataset ranges obtained using SABR parameters 21

Table 4.2: Random dataset ranges and synthetic dataset ranges obtained using SABR parameters 21

Table 5.1: The MSE results of the basic activation functions which are softplus, sin and tanh and new activation functions based on the basic activation functions. 24

Table 5.2: The MSE results for neural networks of different depths. 25

Table 5.3: MSE values between PINN solutions and benchmark models (Monte Carlo and Analytical Black Scholes). 29

Table 5.4: The selected European-style call option contracts for various underlying assets. The data includes the quote date, expiration date, strike price for each contract and close price of each index on quote date. 32

Table 5.5: MSE and MAPE values for PINN-based solution corresponding to each financial model across various underlying assets. 34

Table 5.6: The selected American-style call option contracts for various underlying assets. The data includes the quote date, expiration date, strike price for each contract, and the close price of each index on the quote date. 37

Table 5.7: MSE and MAPE values for PINN-based solution corresponding to each financial model across various underlying assets. 38

Table 5.8: Comparison of MAPE values between Conventional-PINN (C-PINN) and PINNsFormer (F-PINN) models. 41

LIST OF FIGURES

Figure 3.1: The figure depicts the general architecture of a PINN, including hidden layers and neurons, along with the PINN loss employed for solving the Black-Scholes PDE. The network takes input coordinates (X) and processes them through multiple hidden layers to predict the solution (u), assumed as an approximated solution of PDE. The training process incorporates three key physics-informed loss components: the PDE residual (Physics Loss) to enforce governing equations, initial condition loss to match temporal constraints, and boundary condition loss to satisfy spatial constraints. In this figure, the hidden layers and neurons show a general NN structure, and PINN loss for solving Black-Scholes PDE is shown. 15

Figure 4.1: Two Brownian motions with $\beta = 0.5$, $\nu = 0.3$, $\rho = -0.2$, and $\alpha_0 = 0.2$ 20

Figure 5.1: Loss plots of the SABR-PINN model for the case $\beta = 0.5$ and $\nu = 0.3$. 25

Figure 5.2: Results of the SABR-PINN model with $\beta = 0.5$ and $\nu = 0.3$ obtained using the random dataset. 26

Figure 5.3: 3D plots of random results of SABR-PINN with $\beta = 0.5$ and $\nu = 0.3$.. 27

Figure 5.4: Comparison with ($\beta = 0.5$, $\nu = 0.3$) and ($\beta = 1$, $\nu = 0$). In this context, the x-axis corresponds to the remaining time until the option's expiration, $t = T - t$. Specifically, $t = 0$ signifies the expiration time, while $t = 1$ indicates the initial time at which the option is acquired. 28

Figure 5.5: Comparison of option prices estimated by BS-PINN, CEV-PINN, Heston-PINN, SABR-PINN, the analytical Black–Scholes model, and Monte Carlo simulations using the random dataset. The left panel shows mean option prices over time, while the right panel illustrates option prices as a function of the underlying stock price. 30

Figure 5.6: Option prices generated by BS-PINN, CEV-PINN, Heston-PINN, SABR-PINN, the analytical Black–Scholes model, and Monte Carlo simulations using the synthetic dataset. 31

Figure 5.7: A comparison between the PINN-based option prices and the real option prices for Dow Jones Index (DJX), Oil Service Sector Index (OSX), Australian Dollar Currency Index (XDA) and Mini SPX Options Index (XSP), which yielded the lowest MAPE values, reveals a strong consistency between the predicted and real option prices. 35

Figure 5.8: A comparison between the PINN-based option prices and the real option prices for Agilent Technologies (A), Berkshire Hathaway (BRKB), Chevron (CVX), JPMorgan Chase (JPM), Philip Morris (PM) and Exxon Mobil (XOM). 40

LIST OF ACRONYMS AND ABBREVIATIONS

A	Agilent Technologies
AD	Automatic Differentiation
ATM	At The Money
BC	Boundary Condition
BRKB	Berkshire Hathaway
BS	Black-Scholes
CDF	Cumulative Distribution Function
CEV	Constant Elasticity of Variance
CVX	Chevron
C-PINN	Conventional Physics-Informed Neural Network
DJX	Dow Jones Index
FDM	Finite Difference Method
FEM	Finite Element Method
F-PINN	PINNsFormer (Transformer-based PINN)
IC	Initial Condition
ITM	In The Money
JPM	JP-Morgan Chase

MAPE	Mean Absolute Percentage Error
MC	Monte Carlo
MSE	Mean Square Error
NN	Neural Network
OSX	Oil Service Sector Index
OTM	Out of The Money
PDE	Partial Differential Equation
PINN	Physics-Informed Neural Network
PM	Philip Morris
RUT	Russell 2000 Index
SABR	Stochastic Alpha, Beta, Rho
SDE	Stochastic Differential Equation
SPX	S&P 500 Index
XDA	Australian Dollar Currency Index
XEO	S&P 100 Index
XOM	Exxon Mobil
XSP	Mini-S&P 500 Index

1. INTRODUCTION

Derivative pricing is an important problem in finance [1]. Among the derivative classes, options are frequently traded in markets. In finance, an option is a type of derivative contract that gives the holder the right, but not the obligation, to buy or sell an underlying asset at a specified price (called the strike price) on or before a specified expiration date [2]. There are a number of uses of options. Options can be used for hedging, where they are used to protect against losses (e.g., by buying a put option) [3]. They are also used for speculation purposes, betting on price movements with limited upfront cost [4]. Lastly, they can be used for income generation, selling options to earn premiums [5].

Accurately determining option prices plays a vital role for traders, investors, and financial institutions, enabling them to make well-informed choices regarding asset purchases, sales, or risk hedging strategies. Reliable pricing of options contributes to the stability of financial markets, as it influences how portfolios and investment plans are structured in response to price fluctuations [6]. In today's climate of heightened market volatility and unexpected shifts in Treasury yield curves, the challenge of achieving robust option pricing has become even more significant.

The most common option pricing methodology is Black–Scholes (BS) model [7]. In option pricing, the Black-Scholes model is expressed via partial differential equations (PDEs). Apart from the constant volatility, as in traditional Black-Scholes, volatility is generally not constant and tends to vary over time, unlike the Black-Scholes model. There are multiple cases for handling the underlying asset's non-constant volatility, where stochastic and local volatility models are the two most commonly used ones. In this case, the volatility of the underlying asset is assumed to be dependent on time t and asset price, where such a relationship is deterministic. Two common examples of local volatility models are Volatility Fitting Methods [8, 9] and the Constant Elastic Variance (CEV) model [10]. Both models are generalizations of the Black-Scholes model

where the volatility is elastic with respect to the asset price. On the other hand, stochastic volatility models come up with volatility processes such as the Heston model [11] and the Hull-White model [12]. Moreover, better models are obtained by combining stochastic and local volatility models. One common local stochastic volatility model is the Stochastic Alpha, Beta, Rho (SABR) model [13], which takes advantage of both stochastic and local volatility models at the same time.

Volatility models of this kind often yield pricing formulas that are either fully explicit or partially so. Nonetheless, due to inherent market uncertainties, these models commonly exhibit some level of model bias and must be regularly adjusted to align with current market data. For instance, [14] highlights that such models tend to underestimate the risk associated with changes in volatility. For more in-depth developments in the areas of option pricing and volatility modeling, readers may consult works such as [15, 16, 17, 18, 19, 20].

Physics-Informed Neural Networks (PINNs) are a class of deep learning models that incorporate known physical laws—typically expressed as partial differential equations (PDEs)—directly into the training process of a neural network. The key idea is to guide the learning process using both data and physics-based constraints, which helps the model learn solutions that are consistent with the underlying physical system. Instead of training only on labeled data, as in conventional neural networks, PINNs minimize a loss function that combines: 1- Data loss: The difference between the model's predictions and known data points (e.g., initial or boundary conditions), 2- Physics loss: The error in satisfying the governing PDEs (e.g., Navier-Stokes, heat equation, Schrödinger equation, etc.), calculated using automatic differentiation. PINNs have a number of advantages that make them suitable for large sets of problems. These advantages are: 1- They can solve forward and inverse problems involving PDEs, even with sparse or noisy

data, 2- PINNs are mesh-free, meaning they don't require a predefined grid or discretization (unlike traditional numerical solvers like finite element or finite difference methods), 3- They generalize well in high-dimensional problems, where traditional methods may suffer from the curse of dimensionality.

In this paper, we come up with PINN-based methods to price options under Black-Scholes, Constant Elastic Variance, Heston, and SABR models. We chose the PDE versions of these models for option pricing and applied them to predict call option prices. We also compare our results with the recently proposed transformer-based PINNsFormer [21]. Moreover, as these pricing models inherently involve second-order derivative terms, we will also explore the suitability of alternative activation functions beyond ReLU and Tanh. We evaluate our method using both simulated and actual market data and assess its performance against analytical and numerical reference models. The empirical results for Russell 2000 Index (RUT) and Dow Jones Index (DJX) options, as well as index constituent companies options, show that the proposed deep learning models have lower absolute errors than the classical Black–Scholes model solution. Distinct from most of the previous studies, our study places a primary emphasis on the use of real market data. Our main contributions can be summarized as follows:

1. We come up with PINN-based option pricing methods under 4 main option pricing models: Black Scholes, Constant Elastic Variance, Heston, and SABR volatility-based option pricing.

2. We integrate new neural network activation functions that are used in option pricing via PINNs, which enhance the pricing prediction performance.

3. We compare the proposed PINN-based solutions with more general transformer-based PINN (PINNformer) while pricing European options. We show that our specialized PINNs perform better than the transformer-based approach.

The remainder of this study is organized as follows. Section 1.1 discusses related studies on option pricing and PINNs. In chapter 2, we discuss four option pricing models which will be solved by PINNs. The following section 3 discusses how PINNs can be used in option pricing. Chapter 4 discusses the datasets and experimental setup, whereas chapter 5 presents results of our PINN-based solutions for option pricing. Finally, chapter 6 concludes the thesis.

1.1 Related Work

A number of deep learning based methods have been developed to price assets, real estates, as well as derivatives such as options [22, 23, 24, 25, 26, 27]. Traditionally, partial differential equations (PDEs) and option pricing problems have been addressed using classical numerical techniques such as finite difference methods (FDM), finite element methods (FEM), and Monte Carlo [28, 29, 30, 31]. While these methods have proven to be effective in many practical scenarios, they are not without limitations. For example, they often require finite discretization of the domain to achieve high accuracy, which can lead to a significant increase in computational cost and memory usage. Additionally, these mesh-based methods may struggle when applied to high-dimensional problems or domains with complex geometries, which are common in real-world financial and physical systems [32]. In response to these challenges, there has been growing interest in exploring alternative mathematical and computational frameworks for solving PDEs. Among these are analytical and semi-analytical methods such as Lie symmetry analysis, Lax pairs, and Bäcklund transformations, which aim to reduce PDEs to simpler forms or to generate new solutions from the known ones [33, 34, 35]. However, these approaches are typically limited to specific classes of equations and may not generalize well to complex, nonlinear scenarios.

In recent years, machine learning-based methods, particularly those employing

neural networks, have emerged as powerful tools for solving differential equations [36, 37, 38, 39]. Among these, PINNs, which are specialized subclasses of artificial neural networks, have shown considerable promise in solving PDEs. PINNs incorporate the underlying physical laws, expressed in the form of PDEs, directly into the loss function of the neural network during training. This allows the model not only to fit observational data but also to respect the governing physical principles of the system. As a result, PINNs have been successfully applied to a wide range of complex problems across various domains of physics and engineering [40, 41, 42, 43, 44].

In the field of quantitative finance, neural networks have recently garnered attention for their applicability to various option pricing problems [45, 46, 47, 48]. In addition to them, several studies have also successfully employed PINNs to solve option pricing problems under classical financial models, such as the Black-Scholes model, CEV model, and the Heston stochastic volatility model [32, 49, 50, 51, 52, 53, 54]. In these cases, the governing PDEs describe the evolution of option prices over time and are integrated into the neural network training process, allowing the model to learn solutions that are consistent both with data and with the underlying physics/financial theory. However, the majority of studies employing the PINN approach for option pricing have predominantly concentrated on the Black-Scholes model. Among these, only a limited number have utilized real market data in their analyses. Furthermore, the application of PINNs to alternative models such as the CEV and Heston models remains relatively scarce, and, to the best of our knowledge, none of the existing studies investigates the use of PINNs for solving the SABR model in the context of option pricing.

Another critical aspect in PINN-based studies is the choice of the activation function. ReLU, Sigmoid, and Tanh functions are the most commonly utilized activation functions in the existing literature of neural networks [55]. In this case, the second derivative of ReLU is zero, and the derivatives of the Tanh function diminish rapidly. Therefore,

the design or selection of activation functions that are more suitable for PDEs involving higher-order derivative terms is expected to yield improved performance [56].

2. OPTION PRICING MODELS IN FINANCE

Option pricing, a specialized area within the field of derivative finance, focuses on financial instruments whose value depends on the performance of underlying assets, such as stocks [57]. This subfield develops and analyzes methods to estimate the fair value of financial options under specific assumptions and prevailing market conditions, including asset prices and interest rates. Financial options are contracts that confer the right, but not the obligation, on the holder to buy or sell an underlying asset at a predetermined price within a specified period. The most common types of options are European and American options. European options can only be exercised at maturity, whereas American options can be exercised at any time before maturity. Furthermore, call options grant the holder the right to purchase the underlying asset, while put options provide the right to sell it. The study of option pricing is vital not only for accurately determining fair valuations but also for optimizing trading strategies, managing financial risks, and improving the overall efficiency of financial markets [58]. In this paper, we will focus on 4 types of option pricing models as discussed next.

2.1 Black-Scholes Model

The Black-Scholes model, developed in 1973 by the economists Fischer Black and Myron Scholes, with significant contributions from Robert Merton, is a mathematical model for pricing European-style options and derivatives [57, 59]. The model, which is a cornerstone in financial mathematics, provides a parabolic partial differential equation (PDE) for pricing an option. It assumes that the underlying asset price follows a stochastic process, specifically modeled through Brownian motion. Brownian motion represents the random nature of price movements over time, and the price of the underlying asset $S(t)$ is described by a geometric Brownian motion as follows:

$$dS(t) = \mu S(t)\,dt + \sigma S(t)\,dW(t) \tag{2.1}$$

where μ denotes the expected rate of return of the asset, σ represents the asset's volatility, also known as implied volatility, and $dW(t)$ is the stochastic component of Brownian motion. A major limitation of the Black-Scholes model is the assumption that both μ and σ remain constant over time.

The stochastic differential equation (SDE) governing the asset price is translated into the Black-Scholes partial differential equation (PDE) for the call option price $C(S, t)$. Under the assumptions of no arbitrage and a risk-neutral measure, the expected return of the asset is replaced by the risk-free interest rate r. The resulting PDE is given by:

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC = 0 \tag{2.2}$$

Here, $\frac{\partial C}{\partial t}$ represents the change in the option value over time, $\frac{\partial C}{\partial S}$ denotes the sensitivity of the option value to changes in the underlying asset price (also known as delta), and $\frac{\partial^2 C}{\partial S^2}$, known as gamma, measures the convexity of the option value with respect to the asset price [60].

As for the initial condition (IC) and boundary conditions (BCs) associated with the PDE, the following conditions can be applied for European call options:

- At $t = T$:

$$C(S, T) = \max(S - K, 0)$$

- For the lower boundary $S = S_{\min}$:

$$C(S_{\min}, t) = 0$$

- For the upper boundary $S = S_{\max}$:

$$C(S_{\max}, t) = S_{\max} - Ke^{-r(T-t)}$$

Here, K represents the strike price. The initial condition where $t = T$ corresponds to the option's expiration time. At maturity, the value of the option is defined by its

intrinsic value, which is the greater of the difference between the underlying asset price S and the strike price K, or zero. For the lower boundary condition at $S = S_{\min}$, the value of the call option is zero, since both the option and the underlying asset are considered useless at this point. For the upper boundary condition as $S = S_{\max}$, the option is almost certain to be exercised. Therefore, the value of the European call option asymptotically approaches the present value of the payoff $(S - K)$, discounted at the risk-free interest rate r over the remaining time to maturity $(T - t)$.

Finally, the Black-Scholes model admits a closed-form analytical solution for the European call option price, which is given by the following equation:

$$C(S,t) = S \cdot N(d_1) - K \cdot e^{-r(T-t)} \cdot N(d_2) \tag{2.3}$$

where $N(\cdot)$ is the cumulative distribution function (CDF) of the standard normal distribution. The terms d_1 and d_2 are defined as:

$$d_1 = \frac{\ln(S/K) + (r + \sigma^2/2)(T-t)}{\sigma\sqrt{T-t}} \tag{2.4}$$

$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{2.5}$$

2.2 CEV Model

In Black-Scholes model, the volatility of the underlying asset is assumed to be constant over time, which may contradict empirical market data. To overcome these limitations, the Constant Elasticity of Variance (CEV) model, proposed by Cox and Ross [61], extends the Black-Scholes framework by allowing the volatility of the asset price to vary as a function of the asset level. The CEV model assumes the following SDE for the asset price $S(t)$:

$$dS(t) = \mu S(t)\,dt + \alpha S^{\gamma}(t)\,dW(t) \tag{2.6}$$

Here, $\alpha > 0$ is the volatility scale constant and $\gamma \in [0, 1]$ is the elasticity parameter. For $\gamma < 1$, the volatility decreases with increasing asset price, capturing the leverage

effect observed in equity markets, and when $\gamma = 0.5$, the model simplifies to the square-root model proposed by Cox and Ross [62]. The instantaneous volatility function $\sigma(S,t)$ is given by:

$$\sigma(S,t) = \alpha S^{\gamma-1}(t) \tag{2.7}$$

Again, under the risk-neutral measure, μ in the SDE is replaced by r. The value $C(S,t)$ of a European call option under the CEV process satisfies the following PDE:

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 C}{\partial S^2} - rC = 0 \tag{2.8}$$

where the initial and boundary conditions for the PDE of CEV are the same as the conditions of the Black-Scholes model. Overall, CEV model explains the volatility smile or skew better than Black-Scholes. It introduces asymmetry in implied volatilities, and CEV model is especially useful for modeling equity options, where downward moves increase volatility.

2.3 Heston Model

Black-Scholes assumes constant volatility, which fails to capture several empirical features observed in financial markets, such as volatility smiles and skews. To better capture these market characteristics, several alternative models have been developed that incorporate stochastic volatility. Among the most influential of these models is the Heston stochastic volatility model, introduced by Steven Heston in 1993 [11].

The Heston model is a two-dimensional diffusion model, where both the asset price and its variance follow stochastic processes. Let $S(t)$ be the asset price at time t. The dynamics under the risk-neutral measure are given by the following SDEs:

$$dS(t) = rS(t)\,dt + \sqrt{v(t)}S(t)\,dW_1(t) \tag{2.9}$$

$$dv(t) = \kappa(\theta - v(t))\,dt + \sigma\sqrt{v(t)}\,dW_2(t) \tag{2.10}$$

where the two Brownian motions $W_1(t)$ and $W_2(t)$ are correlated by ρ, $dW_1(t)\,dW_2(t) = \rho\,dt$. In this case, $v(t)$ is the instantaneous variance, κ is the speed of mean reversion, θ is the long-term mean variance, and σ is the volatility of volatility. To ensure the positivity of the variance process, the Feller condition [63] must be satisfied, $2\kappa\theta \geq \sigma^2$.

To price an option under the Heston model, a partial differential equation that the option price must satisfy can be derived from the above SDEs. Let $C(S, v, t)$ be the price of a European call option, which depends on the underlying asset price S, its variance v, and time t. Then the associated PDE is:

$$\frac{\partial C}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 C}{\partial S^2} + \rho\sigma vS\frac{\partial^2 C}{\partial S \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 C}{\partial v^2} + rS\frac{\partial C}{\partial S} + \kappa(\theta - v)\frac{\partial C}{\partial v} - rC = 0 \quad (2.11)$$

The initial condition at $t = T$ is the same as the previous two models and is equal to the option's intrinsic value. Additionally, the boundary condition for $S_{\min}$ remains unchanged from the previous models, whereas the boundary condition for $S_{\max}$ is modified. Due to the time-dependent variation of the variance, two additional boundary conditions have been defined for this variable. The modified and newly added boundary conditions are as follows:

- Modified boundary condition at the upper asset price boundary $S = S_{\max}$:

$$\frac{\partial C}{\partial S}(S_{\max}, v, t) = 1$$

- New boundary condition at the upper variance boundary $v = v_{\max}$:

$$C(S, v_{\max}, t) = S$$

- New boundary condition at the lower variance boundary $v = 0$:

$$C(S, v = 0, t) = \max(S - Ke^{-r(T-t)}, 0)$$

As the value of the underlying asset increases, the option will undoubtedly be exercised and purchased. Consequently, the option's value converges to its intrinsic payoff, and its delta (the derivative with respect to the underlying asset price) approaches 1. As for the $v = v_{\max}$ condition, there is a positive relationship between variance and extreme events so high variance indicates that extreme events are more frequent, and option prices generally increase with variance. However, it would not make sense for an option to be more expensive than the asset itself, so the option's price is bounded above by the asset price. Lastly, when the variance tends to zero (expressing monotonicity), the option price is limited by the discounted intrinsic value. The key features of Heston model are: 1- Stochastic Volatility: Volatility follows a mean-reverting square-root process (Cox-Ingersoll-Ross), 2- Mean Reversion: Variance tends toward the long-run level θ at rate κ, 3- Leverage Effect: Negative correlation $\rho < 0$ captures the empirical observation that volatility tends to rise when prices fall.

2.4 SABR Model

The Stochastic Alpha Beta Rho (SABR) model [13] is a popular stochastic volatility model used for capturing financial features, such as volatility smiles and skews, better than Black-Scholes and CEV models. To achieve this, the SABR model is also built on two stochastic processes. Mathematically, the SABR model is defined by the following stochastic differential equations (SDEs):

$$dF(t) = \alpha(t)F^{\beta}(t)dW_1(t), \quad F(0) = f \tag{2.12}$$

$$d\alpha(t) = \nu\alpha(t)dW_2(t), \quad \alpha(0) = \alpha_0 \tag{2.13}$$

where the two Brownian motions $W_1(t)$ and $W_2(t)$ are correlated by ρ, $dW_1(t)\, dW_2(t) = \rho\, dt$. $F(t)$ represents the forward price of the underlying asset, f as the initial stock price, and t as time. The parameters ν, β, and ρ are free model parameters where $\nu > 0$ represents the volatility of volatility, $0 \leq \beta \leq 1$ controls the elasticity of the forward

price. PDE version of the SABR model can be written as follows according to [64];

$$\frac{\partial C}{\partial t} + \frac{1}{2}\alpha^2 F^{2\beta}\frac{\partial^2 C}{\partial F^2} + \rho\nu\alpha^2 F^{\beta}\frac{\partial^2 C}{\partial F \partial \alpha} + \frac{1}{2}\nu^2\alpha^2\frac{\partial^2 C}{\partial \alpha^2} = 0 \tag{2.14}$$

In 2009, [65] presented an extended version of the SABR model by converting the equations from forward price to stock price and added the terms risk free rate, continuous dividend rate, and mean reverting limit for volatility to the above stochastic equations. Then, he derived the PDE of his extended SABR model for equity options. The extended PDE is as follows:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\alpha^2 S^{2\beta}\frac{\partial^2 C}{\partial S^2} + \rho\nu\alpha^2 S^{\beta}\frac{\partial^2 C}{\partial S \partial \alpha} + \frac{1}{2}\nu^2\alpha^2\frac{\partial^2 C}{\partial \alpha^2} + rS\frac{\partial C}{\partial S} + \kappa(\theta(t) - \alpha)\frac{\partial C}{\partial \alpha} - rC = 0 \tag{2.15}$$

where κ ($\theta(t)$ - α) is volatility mean reverting term, with θ is the mean-reverting limit, and κ is the mean-reverting rate. However, for simplicity, the terms involving κ and θ have been omitted from the equation, and only the interest rate term has been added to the general SABR PDE expression. In this case, the SABR PDE for European call options can be written as:

$$\frac{\partial C}{\partial t} + \frac{1}{2}\alpha^2 S^{2\beta}\frac{\partial^2 C}{\partial S^2} + \rho\nu\alpha^2 S^{\beta}\frac{\partial^2 C}{\partial S \partial \alpha} + \frac{1}{2}\nu^2\alpha^2\frac{\partial^2 C}{\partial \alpha^2} + rS\frac{\partial C}{\partial S} - rC = 0 \tag{2.16}$$

The initial and boundary conditions of the above equation are taken to be the same as those of the Heston model, except for the boundary condition for maximum volatility. The boundary condition for $\alpha_{\max}$ is given by:

$$\frac{\partial C}{\partial t}(S, \alpha_{\max,t}) + \frac{1}{2}\alpha_{\max}^2 S^{2\beta}\frac{\partial^2 C}{\partial S^2}(S, \alpha_{\max,t}) + rS\frac{\partial C}{\partial S}(S, \alpha_{\max,t}) - rC(S, \alpha_{\max,t}) = 0 \tag{2.17}$$

3. PHYSICS-INFORMED NEURAL NETWORKS

The use of neural networks to solve differential equations was first proposed by Lagaris in 1998 [36]. Subsequently, the Physics-Informed Neural Networks (PINNs), in their current form, were developed through the work of Raissi et al. [40, 41].

PINNs is a novel class of deep learning models that integrates the fundamental principles of physics into the training of neural networks. This unique approach allows PINNs to approximate solutions to complex problems involving Partial Differential Equations (PDEs).

PINNs utilize the theoretical capabilities of neural networks to approximate any continuous function, as established by the Universal Approximation Theorem [66, 67, 68]. According to this theorem, a sufficiently wide and well-trained neural network can accurately approximate any continuous function within a bounded domain. This principle provides a solid foundation for employing neural networks to model solutions for complex physical systems described by PDEs.

One of the most functional aspects of this method is that it can be applied to both forward and inverse problems. Forward PDE problems typically involve predicting future states of a system based on known initial and boundary conditions, whereas inverse problems require deducing unknown parameters or initial conditions from observed data. Since this study focuses on the forward PDE solution, the PINN approach for the forward problem is illustrated in Figure 3.1.

At its core, a PINN aims to approximate the solution $u(\mathbf{X})$, where $\mathbf{X}$ is a vector of spatial and temporal coordinates, of a differential equation by employing a neural network $\text{NN}(\mathbf{X};\theta)$, where θ represents the trainable parameters. The neural network directly models the solution, i.e., $\text{NN}(\mathbf{X};\theta) \approx u(\mathbf{X})$.

To integrate physical laws into the training process, PINNs use automatic differentiation (AD), which is also known as algorithmic differentiation [69]. This method allows

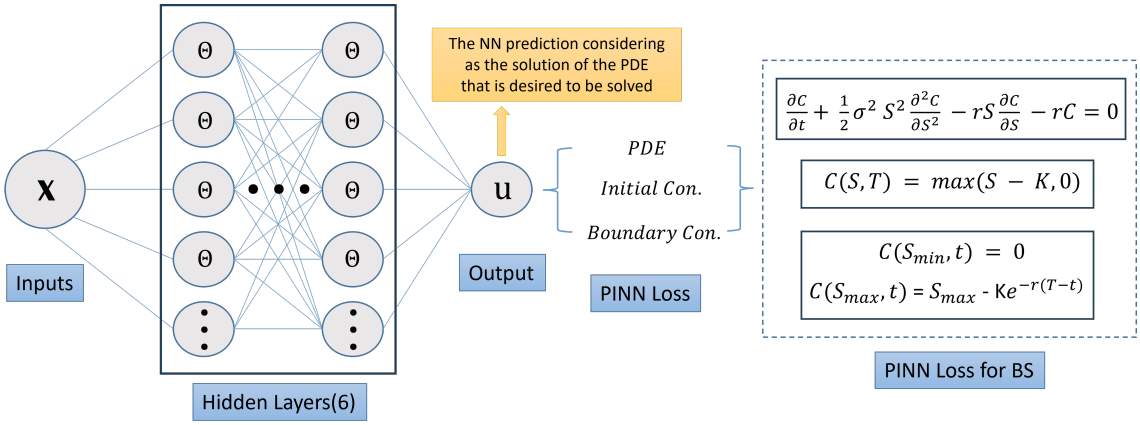


Figure 3.1: *The figure depicts the general architecture of a PINN, including hidden layers and neurons, along with the PINN loss employed for solving the Black-Scholes PDE. The network takes input coordinates (X) and processes them through multiple hidden layers to predict the solution (u), assumed as an approximated solution of PDE. The training process incorporates three key physics-informed loss components: the PDE residual (Physics Loss) to enforce governing equations, initial condition loss to match temporal constraints, and boundary condition loss to satisfy spatial constraints. In this figure, the hidden layers and neurons show a general NN structure, and PINN loss for solving Black-Scholes PDE is shown.*

derivatives of the outputs of the network with respect to its inputs to be calculated precisely and efficiently. Through AD, PINNs can directly derive the spatial and temporal derivatives needed by the PDEs from the neural network structure.

The main distinguishing feature of a PINN is its loss function, which enforces physical consistency by penalizing deviations from the governing equations. Specifically, the loss function comprises multiple critical components:

- **PDE Loss** ($\mathcal{L}_P$): Penalizes the residual of the governing PDE to ensure the predicted solution satisfies the differential equation.
- **Initial Condition Loss** ($\mathcal{L}_{IC}$): Guarantees the solution adheres to the prescribed initial conditions.
- **Boundary Condition Loss** ($\mathcal{L}_{BC}$): Constrains the solution to comply with the boundary conditions.

Traditionally, these losses are calculated with the mean square error (MSE) metric. The total loss is expressed as the linear sum of these three types of losses:

$$\mathcal{L}_{\text{Total}} = \omega_P \mathcal{L}_P + \omega_{IC} \mathcal{L}_{IC} + \omega_{BC} \mathcal{L}_{BC} \tag{3.1}$$

where ω_P, ω_{IC}, and ω_{BC} are hyperparameters that control the weighting of each loss component. By minimizing the total loss, the PINN learns a solution that not only fits the data but also respects the fundamental physical laws and constraints of the problem. Figure 3.1 illustrates the PINN losses corresponding to the Black-Scholes model. By way of example, utilizing the Black-Scholes equation and the conditions detailed in Chapter 2, PINN losses are computed as follows.

$$L_p = \frac{1}{N_P} \sum_{i=1}^{N_P} \left(\frac{\partial NN(S_i, t_i)}{\partial t} + \frac{1}{2}\sigma^2 S_i^2 \frac{\partial^2 NN(S_i, t_i)}{\partial S^2} + rS_i \frac{\partial NN(S_i, t_i)}{\partial S} - rNN(S_i, t_i) \right)^2 \tag{3.2}$$

$$L_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} \left(NN(S_i, T) - \max(S_i - K, 0) \right)^2 \tag{3.3}$$

$$L_{BC_{S_{\min}}} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left(NN(S_{min}, t_i) \right)^2 \tag{3.4}$$

$$L_{BC_{S_{\max}}} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left(NN(S_{max}, t_i) - \left(S_{max} - e^{-r(T-t_i)} \right) \right)^2 \tag{3.5}$$

where NN$(\mathbf{X}; \theta)$ is abbreviated as NN$(\mathbf{X})$ and $\mathbf{X}$ is (S,t) for the Black-Scholes equation. Lastly, N_i represents the number of data points used for the corresponding condition, and the expression N_{BC} denotes the number of data points corresponding to boundary conditions, including S_{min} and S_{max} conditions. As shown in Equation 3.1, the weighted sum of all these individual losses constitutes the total PINN loss for the Black-Scholes model. For CEV model, physics loss will be like this;

$$L_p = \left(\frac{1}{N_P} \sum_{i=1}^{N_P} (\frac{\partial NN(S_i,t_i)}{\partial t} + \frac{1}{2}\sigma^2(S_i,t_i)S^2 \frac{\partial^2 NN(S_i,t_i)}{\partial S^2} + rS\frac{\partial NN(S_i,t_i)}{\partial S} - rNN(S_i,t_i) \right)^2 \tag{3.6}$$

The main differences from Black-scholes is the instantaneous volatility which is depend on S and t. On the other hand, the IC and BCs expressions of CEV are the same as those in Equations 3.8 - 3.10 written for Black-Scholes. In the Heston and SABR models, since the PDE includes derivatives based on variance and volatility, it is necessary to write a longer and more complex physical loss than in the BS and CEV models. For the first of these, the physical loss is as follows:

$$L_p = \frac{1}{N_P} \sum_{i=1}^{N_P} \left(\frac{\partial NN(S_i,v_i,t_i)}{\partial t} + \frac{1}{2}v_i S_i^2 \frac{\partial^2 NN(S_i,v_i,t_i)}{\partial S^2} + \rho\sigma v_i S_i \frac{\partial^2 NN(S_i,v_i,t_i)}{\partial S \partial v} + \frac{1}{2}\sigma^2 v_i \frac{\partial^2 NN(S_i,v_i,t_i)}{\partial v^2} + rS_i \frac{\partial NN(S_i,v_i,t_i)}{\partial S} + \kappa(\theta - v_i)\frac{\partial NN(S_i,v_i,t_i)}{\partial v} - r\, NN(S_i,v_i,t_i) \right)^2 \tag{3.7}$$

IC and BC for S_{min} will be same as mentioned in Chapter 2 for both Heston and SABR but loss for other BCs will be as follows:

$$L_{BC_{S_{\max}}} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left(\frac{\partial NN}{\partial S}(S_{\max}, v_i, t_i) - 1) \right)^2 \tag{3.8}$$

$$L_{BC_{v_{\max}}} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left(NN(S_i, v_{\max}, t_i) - S_i \right)^2 \tag{3.9}$$

$$L_{BC_{v=0}} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left(NN(S_i, v_{v=0}, t_i) - \left(\max(S_i - Ke^{-r(T-t_i)}, 0) \right) \right)^2 \tag{3.10}$$

For Heston, the total loss function is the sum of the physics loss from the PDE and the condition losses, including the IC and BCs losses. Lastly, the physics loss equation for SABR, similar to the other three, is written as follows:

$$L_p = \frac{1}{N_P}\sum_{i=1}^{N_P}\left(\frac{\partial NN(S_i,\alpha_i,t_i)}{\partial t} + \frac{1}{2}\alpha_i^2 S_i^{2\beta}\frac{\partial^2 NN(S_i,\alpha_i,t_i)}{\partial S^2} + \rho\nu\alpha_i^2 S_i^{\beta}\frac{\partial^2 NN(S_i,\alpha_i,t_i)}{\partial S\partial\alpha} + \frac{1}{2}\nu^2\alpha_i^2\frac{\partial^2 NN(S_i,\alpha_i,t_i)}{\partial\alpha^2} + rS\frac{\partial NN(S_i,\alpha_i,t_i)}{\partial S} - rNN(S_i,\alpha_i,t_i)\right)^2 \tag{3.11}$$

As noted in the SABR discussion in Chapter 2, the Heston and SABR models share the same conditions, with the exception of the maximum volatility boundary. For the SABR model, the loss function corresponding to this boundary condition is modified as follows:

$$L_{BC_{\alpha_{\max}}} = \frac{1}{N_{BC}}\sum_{i=1}^{N_{BC}}\left(\frac{\partial NN}{\partial t}(S_i,\alpha_{\max,t_i}) + \frac{1}{2}\alpha_{\max}^2 S_i^{2\beta}\frac{\partial^2 NN}{\partial S^2}(S_i,\alpha_{\max,t_i}) + rS_i\frac{\partial NN}{\partial S}(S_i,\alpha_{\max,t_i}) - rNN(S_i,\alpha_{\max,t_i})\right)^2 \tag{3.12}$$

In conclusion, this approach combines the flexibility of neural networks with rigorous mathematical foundations, making it particularly effective for solving complex, nonlinear PDEs where traditional methods may struggle.

4. EXPERIMENTAL SETUP

In our study, we experimented with random, synthetic, and real market data. First, the model was trained using a randomly generated dataset. At this stage, the results obtained from the random dataset were compared with both Monte Carlo simulations and the analytical Black–Scholes outcomes. We implemented the Monte-Carlo method as described by Khomasuridze [65]. In addition, the selection of the most suitable activation function was performed based on the model's performance on this dataset. The synthetic dataset was generated using the stochastic dynamics of the SABR model. Subsequently, the model was evaluated on real-world market data.

To generate synthetic data compatible with the SABR model, it is necessary to simulate two correlated Brownian motions. For this purpose, the Wiener process, commonly used in financial models for stochastic processes [70], was employed. Then, a dataset was constructed with an initial time $t = 0$ and a terminal time T, consisting of 365 time steps with $\Delta t = \frac{1}{365}$ equal to one year. Using the Wiener process method, the graphs of W_1 and W_2 Brownian motions with a correlation of -0.2 for 365 steps can be seen in Figure 4.1. Additionally, this synthetic dataset was used to determine the boundaries of the random dataset.

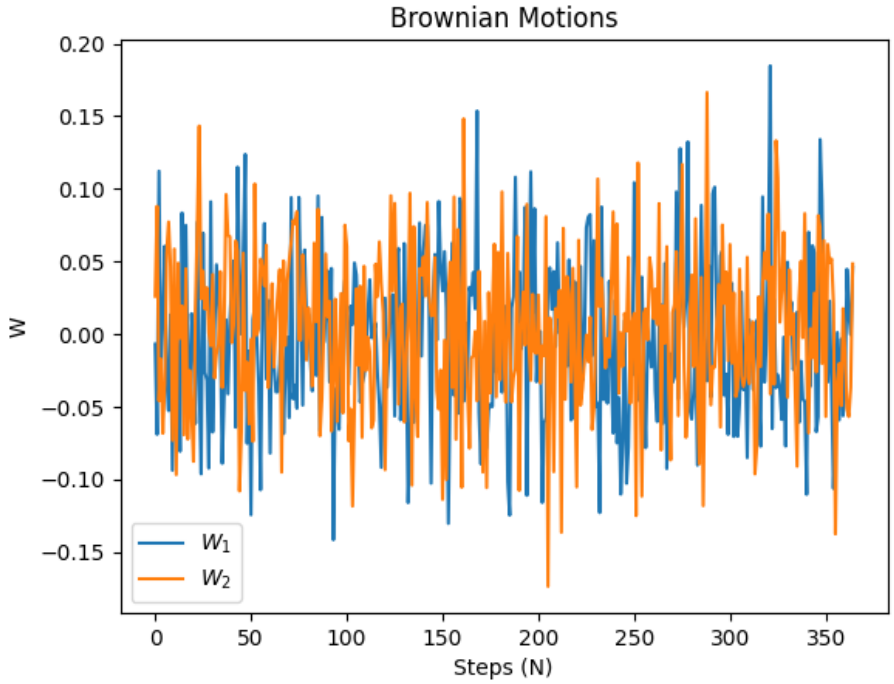


Figure 4.1: *Two Brownian motions with* $\beta = 0.5$, $\nu = 0.3$, $\rho = -0.2$, *and* $\alpha_0 = 0.2$

In Table 4.1, the parameters of this simulation and the ranges of the dataset obtained as a result of the simulation are shared. The dataset presented in Table 4.1 will be used to train the SABR-PINN model. Within this framework, an appropriate activation function will first be constructed for the model. Furthermore, in the case where $\beta = 0.5$, the SABR-PINN model will be compared with both the Monte Carlo approach and the analytical Black–Scholes model.

Table 4.2 presents the parameters of the dataset that will be used to evaluate the performance of all PINN models under the Black–Scholes setting (i.e., a single Brownian motion).

While using the real datasets, we used the publicly available options dataset in [71]. In this dataset, the contract types are categorized as European and American options. The dataset, constructed using data freely provided by the referenced website, comprises daily records for the first six months of 2013. It includes asset prices, bid and ask prices for

Table 4.1: *Random dataset and synthetic dataset ranges obtained using SABR parameters*

Parameter	Random Dataset	Synthetic Dataset
β	0.5	0.5
ν	0.3	0.3
ρ	-0.2	-0.2
F_0	100	100
α_0	0.2	0.2
Strike Price, K	50	50
F_{range}	$[0.0, 101.15]$	$[97.36, 101.15]$
α_{range}	$[0.0, 0.331]$	$[0.205, 0.331]$
t_{range}	$[0, 1]$	$[0, 1]$

Table 4.2: *Random dataset ranges and synthetic dataset ranges obtained using SABR parameters*

Parameter	Random Dataset	Synthetic Dataset
β	1	1
ν	0	0
ρ	-0.2	-0.2
F_0	100	100
α_0	0.2	0.2
Strike Price, K	50	50
F_{range}	$[0.627, 100.94]$	$[77.53, 111.24]$
α_{range}	$[0.0, 0.331]$	$[0.205]$
t_{range}	$[0, 1]$	$[0, 1]$

each contract, strike prices, and the corresponding implied volatilities for each business day during this period. Among the European-style contracts presented in the dataset, the longest-term contracts were selected as possible as, provided that their start date was 2013-01-02.

4.1 Implementation

For the implementation of PINNs, we employed a fully connected neural network structure. The network consists of six fully connected layers, each with 200 neurons, with batch normalization applied between consecutive layers. The model was trained for $1,000$ epochs using Adam optimizer [72] with $2e^{-4}$ learning rate. The best model in 1000

epochs was recorded, and the predictions were made with this model. While training the model, 1000 random points were generated for each condition in each epoch. All models are implemented in PyTorch [73]. All computations were carried out on Google Colab utilizing an NVIDIA T4 GPU with CUDA support.

4.2 Performance Evaluation

While evaluating the performance, we mainly used Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE). Let Y_i be the true price at time i, whereas $\hat{Y}_i$ be the predicted price at the same time step. Then, these evaluation metrics are defined as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{4.1}$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \tag{4.2}$$

where lower scores mean better performance for both metrics.

5. RESULTS AND DISCUSSION

5.1 Results

5.1.1 Analysis of Activation Functions Using Random Datasets

First of all, only the performance of SABR-PINN with parameters β = 0.5 and ν = 0.3 was examined in this section. Here, it was demonstrated that the SABR-PINN model produces reasonable results not only in the Black–Scholes case ($\beta = 1$, $\nu = 0$) but also under different parameter settings. Furthermore, suitable activation functions were tested within this single model. At this stage, the PDE for the general SABR model and its initial and boundary conditions were used, as shown in Equation 5.4.

Equation 5.4 was solved using different activation functions, and their performances were compared. Since the all financial model's (BS, CEV, Heston and SABR) equations contain parabolic expressions, activation functions such as ReLU (which is linear) or Tanh (which produces very small second derivative values) were found to be inappropriate. To determine the most appropriate activation function, they were treated as hyperparameters and their performance was evaluated. To compute numeric metrics such as mean square error (MSE) and mean absolute percentage error (MAPE) that help to evaluate the performance of the model, the option values were calculated by Monte Carlo method and analytical solution of the Black-Scholes given in Equation 2.3. These calculated values by Monte Carlo method were taken as the real option values and the MSE results were calculated accordingly in this section. The results of basic activation functions which are softplus, sin, and tanh and new activation functions based on the basic activation functions are presented in Table 5.1.

As can be seen in Table 5.1, the model trained with the activation function combined by Softplus and 0.005 x Sin yielded the best MSE score. Also, SABR-PINN results are more consistent with the Monte Carlo method in general. In addition, the epoch versus

Table 5.1: *The MSE results of the basic activation functions which are softplus, sin and tanh and new activation functions based on the basic activation functions.*

Activation Function	MC vs SABR-PINN	MC vs Analytical BS
Softplus	0.1964	0.217
Sin	12.17	0.4370
Tanh	7.26	0.3567
0.5 x Softplus + 0.5 x Sin	0.0812	0.5092
0.5 x Softplus + 0.5 x Tanh	0.2199	0.7196
Softplus + 0.005 x Sin	0.0762	0.4072
Softplus + 0.005 x Tanh	0.1090	0.5020
Softplus + 0.005 x Sin + 0.005 x Tanh	0.0881	0.3758

loss results of the model trained using the activation function can be seen in Figure 5.1. As can be observed from these graphs, the minimum volatility ($\alpha_{\min}$) boundary condition provides the largest contribution to the loss. Under this condition, the loss begins to level off around the 200th epoch.

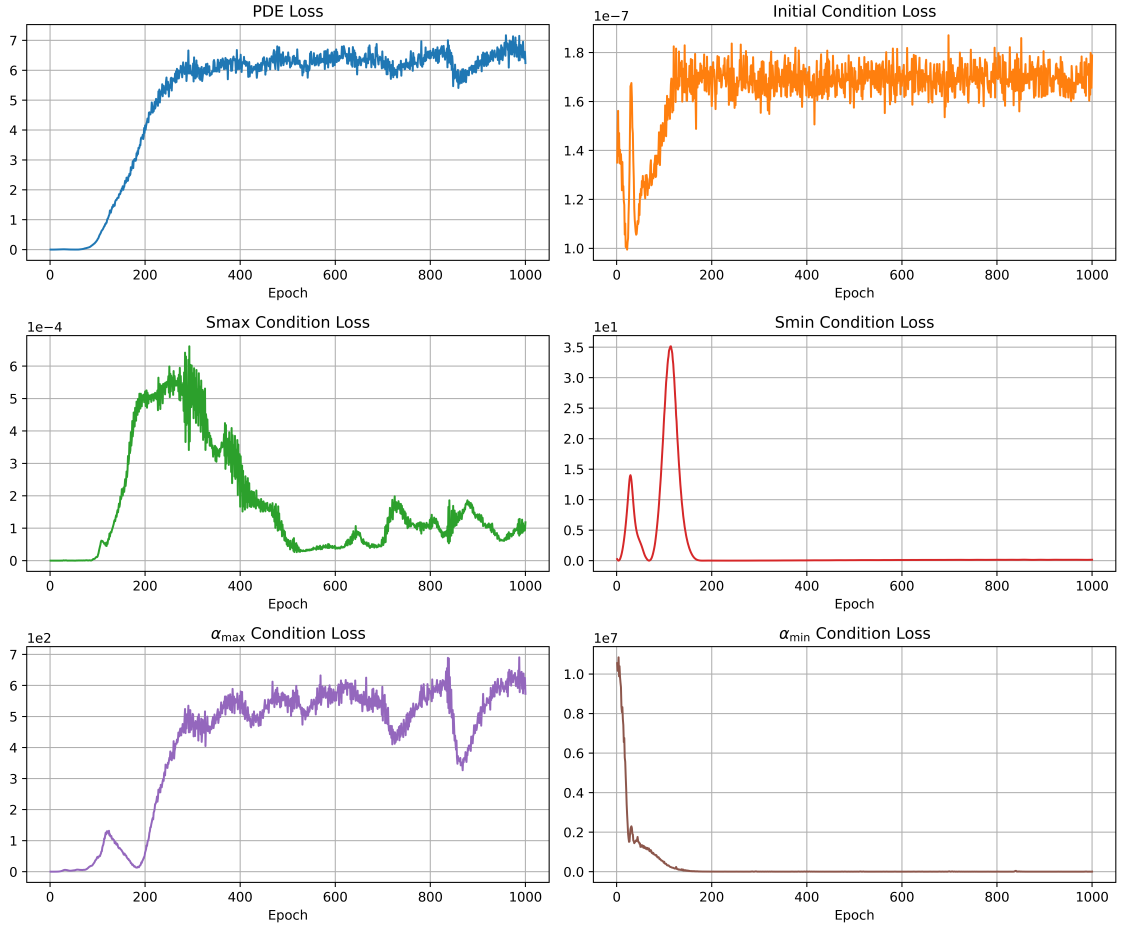


Figure 5.1: *Loss plots of the SABR-PINN model for the case* $\beta = 0.5$ *and* $\nu = 0.3$.

After determining the appropriate activation function, the optimal number of layers for the model was also examined. To this end, the model was additionally trained with 4 and 8 layers using the same activation function, and the MSE scores were compared, as shown in Table 5.2. According to these results, the 6-layer architecture was identified as the most optimal configuration, and the model was finalized with six layers.

Table 5.2: *The MSE results for neural networks of different depths.*

Number of Layers	MC vs SABR-PINN	MC vs Analytical BS
4-Layers	0.3918	0.6527
6-Layers	0.0762	0.4072
8-Layers	0.1370	0.4797

Graphical representations of the results obtained using the Softplus and 0.005 x Sin activation functions with the random dataset, shown in Figure 5.2. As illustrated in the Option Price and Stock Price plots, around the ATM (At The Money) region, where the stock price equals the strike price, SABR-PINN shows greater consistency with the Monte Carlo method, whereas the analytical Black–Scholes model produces higher option prices near the ATM region. Additionally, as the stock price moves away from the strike price, the three models converge more and more.

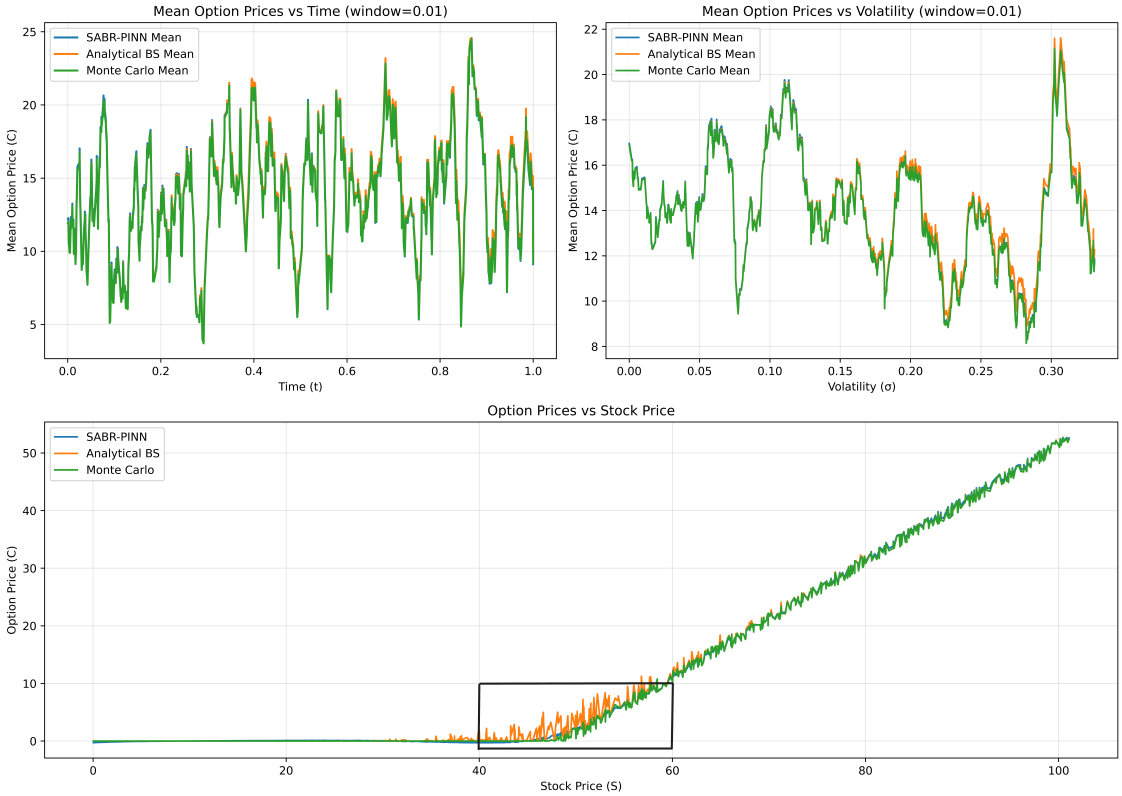


Figure 5.2: *Results of the SABR-PINN model with* $\beta = 0.5$ *and* $\nu = 0.3$ *obtained using the random dataset.*

Figure 5.2 also illustrates the relationship of option prices with time and volatility. Since a random dataset was used at this stage, the graphs were plotted based on average option prices within a 0.01 window interval. According to these graphs, although the

SABR-PINN, Monte Carlo, and analytical Black–Scholes models generally show consistency, the analytical Black–Scholes model diverges from the other two as volatility increases. However, the problem with SABR-PINN is that it can produce negative values close to zero, even though it should give the option price as zero in the Out of Money (OTM) region.

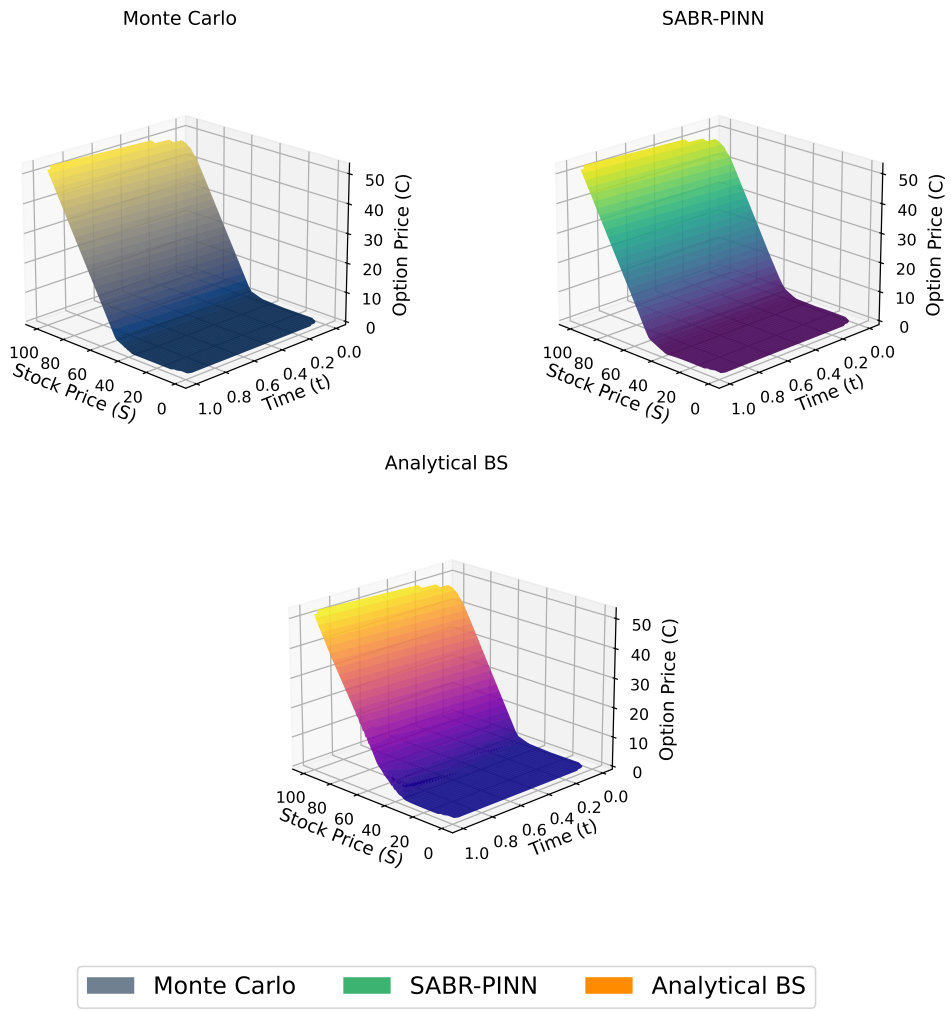


Figure 5.3: *3D plots of random results of SABR-PINN with* $\beta = 0.5$ *and* $\nu = 0.3$

Figure 5.3 presents a 3D visualization of the values produced by the SABR-PINN, Monte Carlo, and analytical Black–Scholes models for the random dataset.

5.1.2 *Analysis of SABR-PINN Using Synthetic Dataset for Different β and ν*

In addition to the previous study with random dataset, we present the studies conducted using the synthetic dataset. First, the performance of the SABR-PINN was examined for different values of β and ν. The parameters and ranges of the dataset constructed for this part were provided in Table 4.1 and Table 4.2. Figure 5.4 shows the results obtained using the synthetic dataset generated from the stochastic equations of the SABR model.

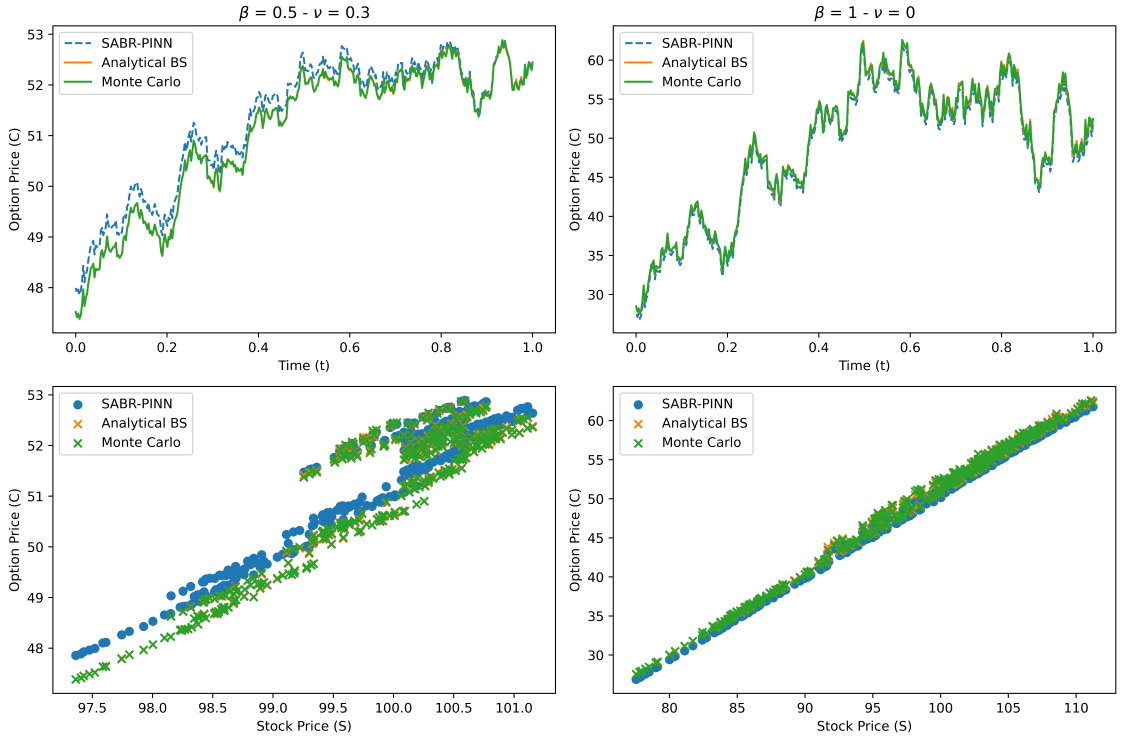


Figure 5.4: *Comparison with ($\beta = 0.5$, $\nu = 0.3$) and ($\beta = 1$, $\nu = 0$). In this context, the x-axis corresponds to the remaining time until the option's expiration, $t = T - t$. Specifically, $t = 0$ signifies the expiration time, while $t = 1$ indicates the initial time at which the option is acquired.*

As illustrated in the corresponding figure, the option prices obtained from the MC

method and the analytical Black–Scholes model are almost indistinguishable in the in-the-money (ITM) region for both parameter settings, $(\beta = 1, \nu = 0)$ and $(\beta = 0.5, \nu = 0.3)$. For the case $(\beta = 0.5, \nu = 0.3)$, the SABR-PINN model produces noticeably higher option values as $t \to 0$, where $t = 0$ denotes the expiration time. In contrast, when $(\beta = 1, \nu = 0)$, the SABR equations collapse to the Black-Scholes framework, and consequently, all three pricing methods yield overlapping curves, as clearly visible in Figure 5.4.

5.1.3 Comparison of All Option Pricing Models

In this section, the PINN results of other option pricing models, in addition to the SABR-PINN model, were also evaluated. To enable a common comparison, all models were reduced to the Black–Scholes case by adjusting with their parameters. Furthermore, all PINN models were compared with both the Monte Carlo method and the analytical Black–Scholes results. The performance differences among the models can also be seen from the MSE values presented in Table 5.3.

Table 5.3: *MSE values between PINN solutions and benchmark models (Monte Carlo and Analytical Black Scholes).*

	BS-PINN	**CEV-PINN**	**Heston-PINN**	**SABR-PINN**
MC vs PINN	416.25	283.73	231.43	0.0733
Analytical BS vs PINN	406.70	275.82	221.76	0.4660

Figure 5.5 presents the results of all models on the random dataset.

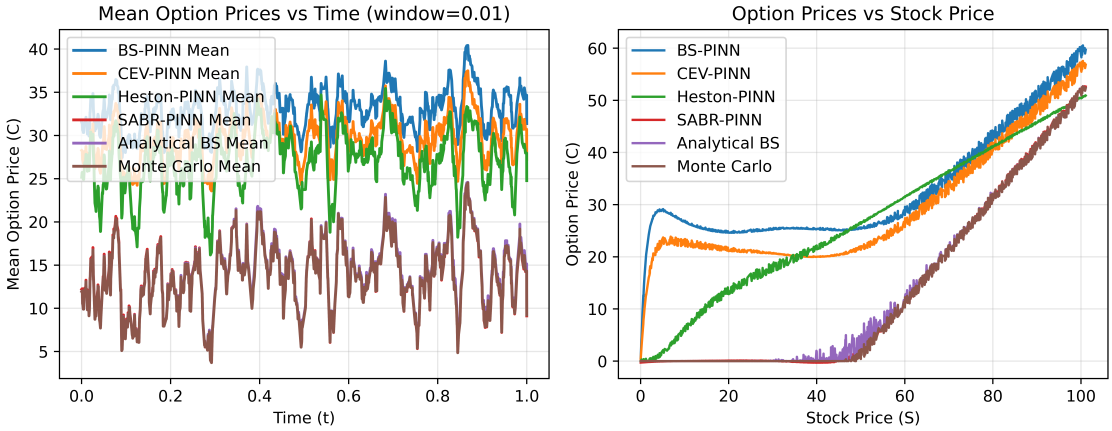


Figure 5.5: *Comparison of option prices estimated by BS-PINN, CEV-PINN, Heston-PINN, SABR-PINN, the analytical Black–Scholes model, and Monte Carlo simulations using the random dataset. The left panel shows mean option prices over time, while the right panel illustrates option prices as a function of the underlying stock price.*

As observed, the best results are obtained with SABR-PINN. The other three models, however, exhibit significant errors, particularly in the OTM (Out-of-the-money) region, where the strike price is lower than the stock price.

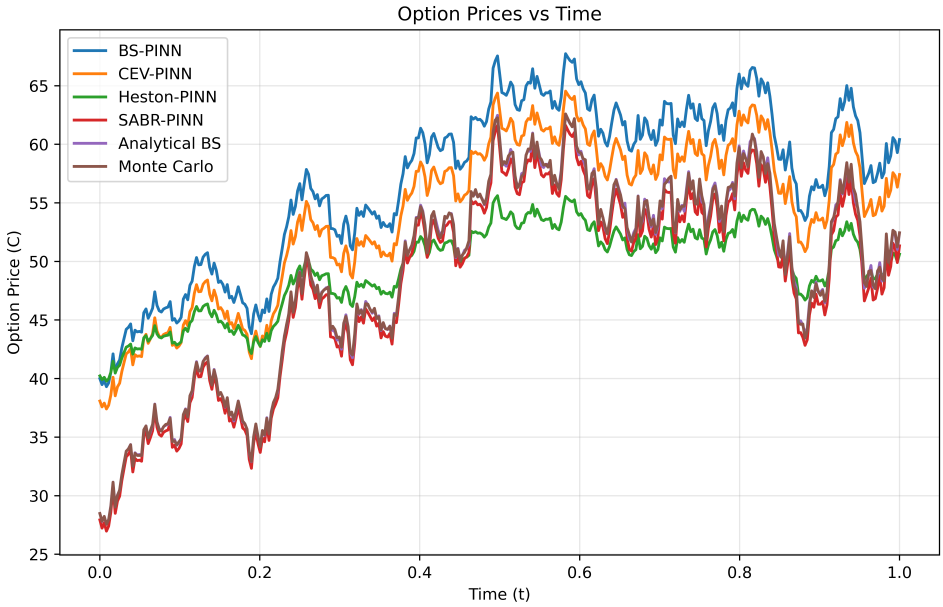


Figure 5.6: *Option prices generated by BS-PINN, CEV-PINN, Heston-PINN, SABR-PINN, the analytical Black–Scholes model, and Monte Carlo simulations using the synthetic dataset.*

As can be seen from these results, the Heston-PINN and SABR-PINN models performed better than the BS-PINN and CEV-PINN models. This outcome is not surprising, considering the additional variance and volatility terms present in the Heston and SABR equations. The reason for the difference between Heston-PINN and SABR-PINN models may be due to the maximum volatility boundary condition.

5.1.4 Real Market Dataset - European Style Options

By using the new activation function combining softplus and sin functions, we predict European style real market option prices. As discussed in Section 4, the dataset was obtained from the Optiondata website [71]. All previously presented option pricing PDEs were solved with PINN and then PINNformer, separately for all assets. To better align with real market conditions for SABR model, PDE of the extended-SABR model

in Equation 2.16 was employed, incorporating the risk-free interest rate r. Morever, the value of risk-free interest rate was obtained from website of U.S. Department of the Treasury [74] and the interest rate was set to $r = 0.0007$. The details of the selected option contracts are presented in Table 5.4.

Table 5.4: *The selected European-style call option contracts for various underlying assets. The data includes the quote date, expiration date, strike price for each contract and close price of each index on quote date.*

Underlying	Contract	Quote Date	Expiration	Strike	Close
BKX	BKX130316C00035000	2.01.2013	16.03.2013	200	52.93
DJX	DJX130316C00075000	2.01.2013	16.03.2013	75	134.13
HGX	HGX130316C00126000	2.01.2013	16.03.2013	126	177.11
NDX	NDX130316C02200000	2.01.2013	16.03.2013	2200	2746.47
OSX	OSXPM130328C00165000	2.01.2013	28.03.2013	165	225.91
RUT	RUT130316C00200000	2.01.2013	16.03.2013	200	873.42
SPX	SPX130316C00500000	2.01.2013	16.03.2013	500	1462.42
XDA	XDA130316C00078000	2.01.2013	16.03.2013	78	104.93
XEO	XEO130420C00560000	2.01.2013	20.04.2013	560	663.87
XSP	XSPPM130328C00090000	2.01.2013	28.03.2013	90	146.24

Unfortunately, Optiondata does not provide the actual market prices of the options directly, but it gives bid and ask prices for the contract instead of the real values. In similar studies, it is also a popular approach to use the average of bid and ask prices instead of the real option price [49]. However, in the dataset used in this European-Style study, bid and ask prices are not available for every time point, t, corresponding to the selected European-style option contracts. Moreover, all of the option contracts selected for real market study are in 'in-the-money' (ITM) region, where the strike price is lower than the close price of the underlying. Therefore, in the first part of the analysis, where a contract is selected from several different indices, the values obtained from the analytical Black-Scholes Equation 2.3 are taken as the real option prices for calculating MSE and MAPE metrics. This choice is justified because Section 5.1.2 shows that the analytical Black-Scholes and Monte Carlo prices are nearly identical. Given that these contracts

are ITM and the implied volatility is available for each time point t, the analytical Black-Scholes model is expected to provide sufficiently accurate values to be considered as the real option prices for this part of the study.

In addition to this, each financial model's own parameters must be adjusted in accordance with this real dataset. For this purpose, these parameters were tried to be adjusted in the most appropriate way for each contract. For the Black-Scholes model, the implied volatility values in the dataset were directly used. For the CEV model, since most of the selected indices exhibit the condition $\gamma < 1$, which implies that volatility decreases as the underlying asset price increases, the elasticity parameter γ was set to 0.5. Similarly, the scaling factor α was also set at 0.5, and the corresponding volatility σ values were calculated using the price data in the dataset according to Equation 2.7. Parameter calibration for the Heston model was conducted using the QuantLib library [75] in Python. For this calibration, average of the bid-ask prices available in the dataset and option prices obtained with the vanilla option function in the QuantLib library were used. In the SABR model, parameter calibration was performed using the implied volatility formula proposed by Hagan [13], along with the implied volatility data extracted from the dataset. For both the Heston and SABR models, the parameters were optimized using the minimize function from Python's SciPy library [76], aiming to minimize the squared differences between the model-generated values and the observed market values.

Additionally, the implied volatility values from the dataset were utilized as the instantaneous volatility in the SABR model. Since both the implied volatility values and the closing prices in the dataset are recorded on a daily basis, this temporal consistency is assumed to ensure the validity of the adopted approach. Table 5.5 summarizes the final MSE and MAPE values for each underlying.

In this study, which is based on European-type market data, the best results are obtained using the Heston-PINN and SABR-PINN models, consistent with the results

Table 5.5: *MSE and MAPE values for PINN-based solution corresponding to each financial model across various underlying assets.*

Underlying	BS-PINN		CEV-PINN		Heston-PINN		SABR-PINN	
	MSE	**MAPE**	**MSE**	**MAPE**	**MSE**	**MAPE**	**MSE**	**MAPE**
BKX	5.22	0.1037	55.24	0.3803	0.9306	0.0438	0.1189	0.0136
DJX	7.99	0.0367	235.24	0.2401	1.43	0.0180	0.1169	0.0051
HGX	125.31	0.1868	616.66	0.4225	4.22	0.0337	28.32	0.0852
NDX	53002.96	0.4164	163708.37	0.7323	17823.03	0.2356	14437.79	0.2111
OSX	12.94	0.0394	1240.80	0.4642	14.11	0.0474	0.4025	0.0077
RUT	1276.06	0.0495	1724.87	0.0587	81.31	0.0126	71.22	0.0119
SPX	978.92	0.0291	9780.76	0.0983	105.06	0.0099	4550.48	0.0669
XDA	0.0275	0.0054	205.21	0.5536	0.3732	0.0205	0.0347	0.0032
XEO	4476.49	0.5312	12614.77	0.8920	113.54	0.0746	322.76	0.1225
XSP	12.88	0.0485	306.01	0.2861	0.2353	0.0070	0.0625	0.0021
Average	**5989.88**	**0.1447**	**19048.79**	**0.4128**	**1814.42**	**0.0503**	**1941.13**	**0.0529**

presented in the other sections of the thesis. In contrast, the CEV-PINN model exhibits weaker performance compared to the other approaches. This situation may stem from the fact that the CEV parameters within the CEV-PINN framework have not been calibrated using market data, which underscores the importance of proper calibration.

Additionally, Figure 5.7 presents the graphs of the four underlying assets with the lowest MAPE values. As observed from these graphs, the model exhibited highly accurate predictions at t=1, corresponding to the contract's quote date, especially SABR model.

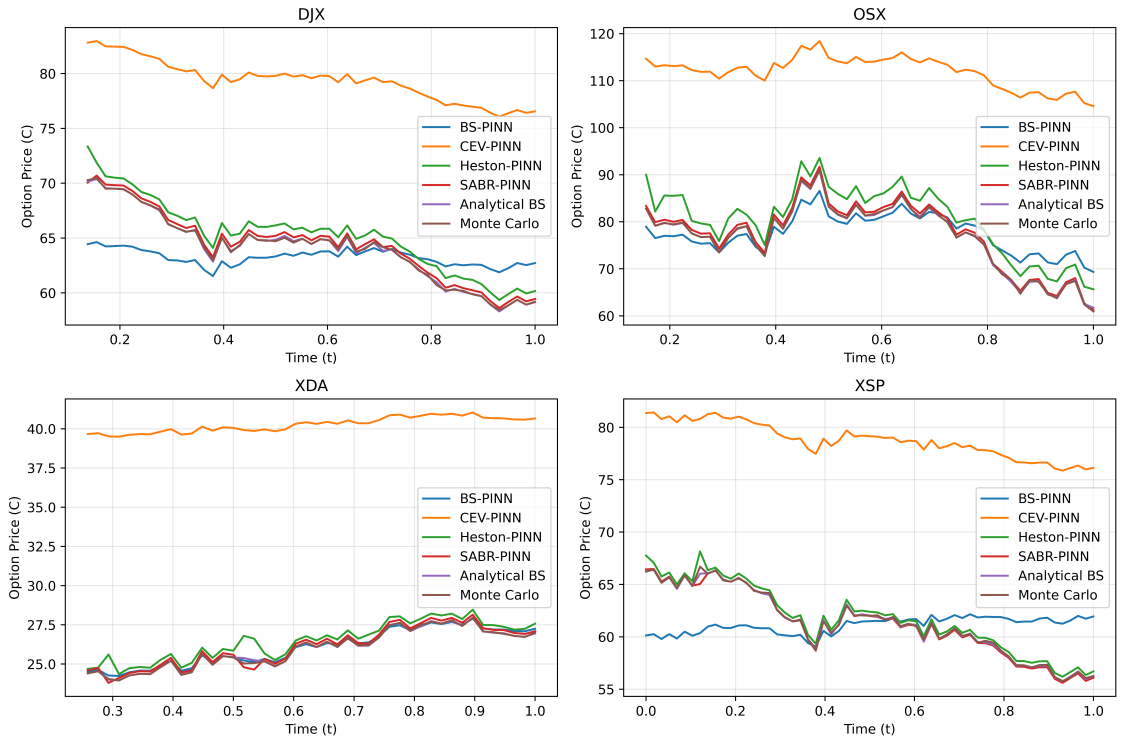


Figure 5.7: *A comparison between the PINN-based option prices and the real option prices for Dow Jones Index (DJX), Oil Service Sector Index (OSX), Australian Dollar Currency Index (XDA) and Mini SPX Options Index (XSP), which yielded the lowest MAPE values, reveals a strong consistency between the predicted and real option prices.*

5.1.5 Real Market Dataset - American Style Options

We also analyze the predictability of American-style call options. For this purpose, the models described in Chapter 2, along with their corresponding initial and boundary conditions, are employed. However, since American-style options may be exercised prior to the expiration date, an additional boundary condition, as defined in the Equations 5.1-5.7, has been incorporated into the problem formulation. First of all, all PDEs must now be less than or equal to zero.

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 C}{\partial S^2} - rC \leq 0 \quad \text{for BS} \tag{5.1}$$

$$\frac{\partial C}{\partial t} + rS\frac{\partial C}{\partial S} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 C}{\partial S^2} - rC \leq 0 \quad \text{for CEV} \tag{5.2}$$

$$\frac{\partial C}{\partial t} + \frac{1}{2}vS^2\frac{\partial^2 C}{\partial S^2} + \rho\sigma vS\frac{\partial^2 C}{\partial S \partial v} + \frac{1}{2}\sigma^2 v\frac{\partial^2 C}{\partial v^2} + rS\frac{\partial C}{\partial S} + \kappa(\theta - v)\frac{\partial C}{\partial v} - rC \leq 0 \quad \text{for Heston} \tag{5.3}$$

$$\frac{\partial C}{\partial t} + \frac{1}{2}\alpha^2 S^{2\beta}\frac{\partial^2 C}{\partial S^2} + \rho\nu\alpha^2 S^{\beta}\frac{\partial^2 C}{\partial S \partial \alpha} + \frac{1}{2}\nu^2\alpha^2\frac{\partial^2 C}{\partial \alpha^2} \leq 0 \quad \text{for SABR} \tag{5.4}$$

In addition, at any given time, it is also required for expiration that the stock price be higher than the strike price.

$$C(S,t) \geq \max(S - K, 0) \quad \text{for BS and CEV} \tag{5.5}$$

$$C(S,v,t) \geq \max(S - K, 0) \quad \text{for Heston} \tag{5.6}$$

$$C(S,\alpha,t) \geq \max(S - K, 0) \quad \text{for SABR} \tag{5.7}$$

This condition ensures that the option value does not fall below its intrinsic value at any time, which is essential to accurately model the possibility of early exercise allowed by American options.

This new conditions turn American style options into a free-boundary problem. Since the analytical-Black Scholes equation is basically for European style options and as there are ask and bid prices for all contracts in the dataset, the values obtained with PINN will be compared with the average of ask and bid prices, Equation 5.8.

$$P_{\text{real}} = \frac{P_{\text{ask}} + P_{\text{bid}}}{2} \tag{5.8}$$

Table 5.6: *The selected American-style call option contracts for various underlying assets. The data includes the quote date, expiration date, strike price for each contract, and the close price of each index on the quote date.*

Underlying	Contract	Quote Date	Expiration	Strike	Close
A	A130518C00018000	2.01.2013	18.05.2013	18	41.88
AAPL	AAPL130420C00260000	2.01.2013	20.04.2013	260	549.03
BAC	BAC130420C00003000	2.01.2013	20.04.2013	3	12.03
BRK.B	BRKB130316C00045000	2.01.2013	16.03.2013	45	93.20
CVX	CVX130316C00055000	2.01.2013	16.03.2013	55	110.39
FB	FB130316C00009000	2.01.2013	16.03.2013	9	28.00
GE	GE130316C00005000	2.01.2013	16.03.2013	5	21.34
GOOGL	GOOG130316C00310000	2.01.2013	16.03.2013	310	723.25
IBM	IBM130420C00100000	2.01.2013	20.04.2013	100	196.35
JNJ	JNJ130420C00042500	2.01.2013	20.04.2013	42.5	70.84
JPM	JPM130420C00022000	2.01.2013	20.04.2013	22	44.66
MSFT	MSFT130420C00015000	2.01.2013	20.04.2013	15	27.62
ORCL	ORCL130316C00015000	2.01.2013	16.03.2013	15	34.69
PFE	PFE130420C00016000	2.01.2013	20.04.2013	16	25.91
PG	PG130420C00035000	2.01.2013	20.04.2013	35	69.39
PM	PM130316C00045000	2.01.2013	16.03.2013	45	86.74
T	T130420C00017000	2.01.2013	20.04.2013	17	35.00
VZ	VZ130420C00021000	2.01.2013	20.04.2013	21	44.27
WFC	WFC130420C00016000	2.01.2013	20.04.2013	16	35.05
WMT	WMT130316C00037500	2.01.2013	16.03.2013	37.5	69.24
XOM	XOM130420C00045000	2.01.2013	20.04.2013	45	88.71

Selected stocks and contracts for American options can be seen in Table 5.6.

Careful consideration was given to ensuring that the selected stocks represented the most prominent and widely traded equities of 2013. Table 5.7 summarizes the final MSE and MAPE values for each underlying, which provides compelling evidence of the effectiveness of the proposed PINN framework in pricing American-style call options based on real market data. As in the previous sections, the results indicate that PINN-based models ,particularly PINN-Heston and PINN-SABR, show strong potential for application to American-style call option pricing. The network successfully captured both the nonlinear payoff characteristics and the early exercise feature inherent to American options. The incorporation of the free-boundary constraint into the loss function proved

particularly crucial in enforcing the correct optimal exercise boundary, enabling the PINN to align well with empirical price observed in market data. SABR-PINN performed best among all models, achieving an average MAPE of 0.0147 across all assets, surpassing Heston-PINN, which achieved an average MAPE of 0.0165. The SABR-PINN's ability to represent both skew and smile effects allowed it to adapt to volatility clustering and asymmetric market behaviors often observed in technology and index options. In contrast, although the simpler structure of BS-PINN and CEV-PINN is computationally efficient, they fall behind the other two models because they do not consider volatility as a variable of the option and therefore do not have volatility-dependent boundary conditions.

Table 5.7: *MSE and MAPE values for PINN-based solution corresponding to each financial model across various underlying assets.*

Underlying	**BS-PINN**		**CEV-PINN**		**Heston-PINN**		**SABR-PINN**	
	MSE	**MAPE**	**MSE**	**MAPE**	**MSE**	**MAPE**	**MSE**	**MAPE**
A	36.70	0.2331	14.50	0.1531	0.2667	0.0196	0.0774	0.0098
AAPL	1735.16	0.2137	6499.92	0.4255	76.26	0.0459	21.41	0.0239
BAC	5.01	0.2289	0.0004	0.0018	0.0387	0.0210	0.0450	0.0223
BRKB	2.03	0.0239	93.13	0.1815	1.81	0.0064	0.1375	0.0048
CVX	404.64	0.3324	122.56	0.1841	0.0966	0.0035	0.0543	0.0036
FB	18.56	0.2164	3.95	0.1004	0.0426	0.0100	0.0977	0.0151
GE	11.23	0.1848	0.37	0.0349	0.0251	0.0071	0.1037	0.0145
GOOG	314.31	0.0329	5298.41	0.1583	494.54	0.0483	1566.91	0.0816
IBM	427.96	0.1952	452.05	0.2050	0.0964	0.0024	4.03	0.0192
JNJ	68.79	0.2238	104.13	0.2992	2.4379	0.0288	0.1850	0.0103
JPM	31.07	0.1984	21.68	0.1796	0.1883	0.0145	0.0173	0.0034
MSFT	22.60	0.3586	9.36	0.2370	0.0956	0.0203	0.1519	0.0275
ORCL	26.16	0.2530	7.84	0.1396	0.1052	0.0137	0.0731	0.0111
PFE	14.18	0.3087	12.92	0.3077	0.0807	0.0180	0.0542	0.0165
PG	143.54	0.2843	65.99	0.2027	0.8028	0.0144	0.0481	0.0045
PM	147.96	0.2682	81.26	0.2019	0.3098	0.0116	0.0095	0.0016
T	26.31	0.2614	12.56	0.1887	0.2209	0.0132	0.1751	0.0127
VZ	27.96	0.1773	29.32	0.2171	0.1147	0.0089	0.0455	0.0057
WFC	31.02	0.2661	10.45	0.1627	0.1032	0.0142	0.0273	0.0061
WMT	70.66	0.2491	58.70	0.2322	0.8928	0.0112	0.2226	0.0102
XOM	132.55	0.2548	73.32	0.1937	0.5938	0.0137	0.0669	0.0036
Average	**176.11**	**0.2269**	**617.73**	**0.1908**	**27.58**	**0.0165**	**75.90**	**0.0147**

Additionally, Figure 5.8 presents the graphs of the six underlying assets, comparing PINN-based option prices and the real option prices. A key takeaway from these empirical findings is that the physics-informed approach effectively integrates theoretical consistency with data-driven flexibility. The model does not merely interpolate observed option prices but instead enforces the governing partial differential equation (PDE) and early exercise constraint as soft penalties during training. This yields solutions that are arbitrage-free, smooth, and consistent with financial theory, while simultaneously adapting to empirical market features. Moreover, when benchmarked against numerical solvers such as the finite-difference method, the PINN models achieved comparable accuracy with substantially reduced computation time once trained, an important advantage for real-time pricing and hedging applications.

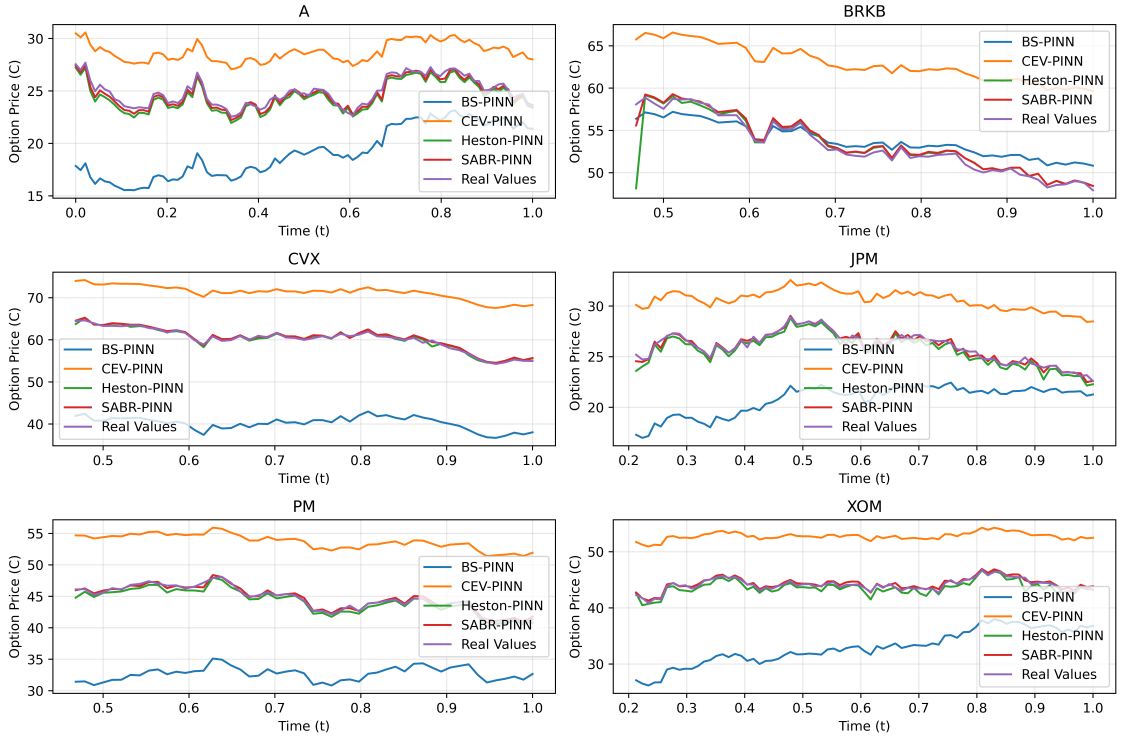


Figure 5.8: *A comparison between the PINN-based option prices and the real option prices for Agilent Technologies (A), Berkshire Hathaway (BRKB), Chevron (CVX), JP-Morgan Chase (JPM), Philip Morris (PM) and Exxon Mobil (XOM).*

5.1.6 Comparison of PINNs and PINNsFormer

The Physics-Informed Neural Network (PINN) framework implemented in Sections 5.1.1-5.1.5 represents the conventional PINN architecture originally proposed by Raissi, as comprehensively detailed in Section 3. In addition to the conventional PINN, the PINNsFormer model, which integrates the transformer architecture with the PINN framework, was also tested. This model was proposed by Zhao et al. [21], who demonstrated its performance on synthetic data through physics problems such as convection and Navier-Stokes in their study. According to their findings, although the PINNsFormer

introduces approximately a threefold increase in computational time compared to the conventional PINN, it yields better results for Convection and 1D-Reaction problems. Therefore, we also compared PINNsFormer with our specialized PINNs for solving option pricing PDEs. The results are presented in Table 5.8. PINNsformer incorporates the same activation function ($Softplus + 0.005xSin$) as our conventional PINNs. The model consists of an encoder-decoder layer with multi-head attention with two heads, followed by a feedforward network consisting of 6 layers, similar to our PINN discussed in previous sections, with the same activation. After the encoder-decoder layer, there are 3 linear layers. The first of these outputs the 4-dimensional input as 512 dimensions. The next hidden layer is of 512 dimensions. The output of the last layer is 1 dimension. Training also uses the Adam optimizer with learning rate $2e^{-5}$, and learning rate scheduling ReduceLROnPlateau.

Table 5.8: *Comparison of MAPE values between Conventional-PINN (C-PINN) and PINNsFormer (F-PINN) models.*

Underlying	BS		CEV		Heston		SABR	
	C-PINN	**F-PINN**	**C-PINN**	**F-PINN**	**C-PINN**	**F-PINN**	**C-PINN**	**F-PINN**
BKX	0.1037	0.5098	0.3803	0.7197	0.0438	0.7118	0.0136	0.9294
DJX	0.0367	0.7452	0.2401	0.2208	0.0180	1.03	0.0051	0.2409
HGX	0.1868	0.7061	0.4225	0.4481	0.0337	0.1125	0.0852	1.01
NDX	0.4164	0.5190	0.7323	0.3842	0.2356	0.8104	0.2111	0.5941
OSX	0.0394	0.2306	0.4642	0.4729	0.0474	0.3506	0.0077	0.0745
RUT	0.0495	0.4893	0.0587	0.0557	0.0126	0.8258	0.0119	0.9638
SPX	0.0291	0.3355	0.0983	0.0947	0.0099	0.6092	0.0669	0.3690
XDA	0.0054	0.1845	0.5536	0.5459	0.0205	0.5815	0.0032	1.00
XEO	0.5312	0.0476	0.8920	0.7982	0.0746	0.5988	0.1225	0.6487
XSP	0.0485	0.2708	0.2861	0.2800	0.0070	0.2241	0.0021	0.4612
Average	**0.1447**	**0.4313**	**0.4128**	**0.4020**	**0.0503**	**0.5854**	**0.0529**	**0.6298**

The results presented in Table 5.8 reveal a consistent performance advantage of the conventional Physics-Informed Neural Network (C-PINN) over the transformer-enhanced PINNsFormer (F-PINN) across nearly all volatility models and underlying assets. On average, C-PINN achieved substantially lower MAPE scores for the Black-Scholes, Heston,

and SABR models, by approximately $2.98\times$, $11.6\times$, and $11.9\times$, respectively, demonstrating that the simpler feed-forward PINN architecture is better suited for low-dimensional financial partial differential equations. Only marginal exceptions, such as the XEO index under the Black-Scholes case, exhibited slightly improved accuracy for the transformer-based variant; however, these were statistically insignificant relative to the overall trend.

A deeper examination of the individual results shows that the PINNsFormer often produced instability in the stochastic and mixed local-stochastic volatility models. In particular, for the Heston and SABR formulations, F-PINN errors exceeded 0.6 on average, while C-PINN maintained errors around 0.05. This gap highlights a key limitation of transformer-based PINNs when applied to financial PDEs, attention mechanisms, though powerful for sequential dependencies, tend to over-parameterize relatively smooth two-dimensional solution spaces such as (S, t) or (S, v, t). Consequently, the self-attention blocks may introduce unnecessary variance during optimization, amplifying numerical noise in the PDE residual term. By contrast, the fully connected structure of the conventional PINN enforces smoother and more stable convergence under physics-informed loss weighting.

From a computational standpoint, the F-PINN model also required roughly three times longer training time due to the multi-head attention and encoder-decoder modules, without a proportional gain in predictive accuracy. This trade-off underscores that transformer-based extensions, while advantageous for spatio-temporal turbulence or chaotic systems, do not necessarily confer benefits for option-pricing PDEs, which are governed by low-order parabolic dynamics. The attention mechanism's representational power appears redundant in this domain, where the governing equations and boundary conditions already provide strong structural priors. Nonetheless, PINNsFormer offers valuable architectural flexibility for potential high-dimensional financial applications. In scenarios

such as multi-asset basket options, correlated stochastic factors, or rough volatility models requiring long-range temporal memory, the transformer framework could mitigate the limitations of conventional PINNs by capturing non-local dependencies more effectively. However, the present findings indicate that for single-asset, well-behaved PDEs such as those in the Black-Scholes, CEV, Heston, and SABR formulations, the conventional PINN achieves superior balance between accuracy, interpretability, and computational efficiency.

5.2 Discussion

The results obtained in this study provide substantial evidence that Physics-Informed Neural Networks (PINNs) can serve as an efficient and accurate framework for solving option pricing problems governed by partial differential equations. Unlike traditional numerical solvers, such as finite difference or Monte Carlo methods, PINNs embed the structure of the governing PDE directly into the neural network loss function, enabling solutions that are both data-driven and physically consistent. This intrinsic coupling between data and physics allows the models to generalize better across varying boundary conditions and volatility regimes, a crucial property for financial applications where market conditions fluctuate rapidly.

A major outcome of our analysis is that the PINN-based formulations of stochastic and mixed local–stochastic volatility models (i.e., Heston-PINN and SABR-PINN) significantly outperform both the classical analytical models and their deterministic PINN counterparts (Black–Scholes-PINN and CEV-PINN). This superiority is particularly evident under real market conditions, where volatility dynamics deviate from the constant-volatility assumption of the Black–Scholes model. Moreover, the improved activation function combining Softplus and sinusoidal components yielded notable performance

gains, suggesting that activation smoothness and higher-order differentiability are critical for PDEs involving second-order derivatives.

From a computational perspective, the proposed PINN models achieved high numerical accuracy with relatively modest computational requirements. While transformer-based PINNsFormer demonstrated competitive flexibility, its increased complexity and computational cost did not translate into proportional improvements in predictive performance for financial PDEs. This observation suggests that, for low- to medium-dimensional option pricing problems, conventional PINN architectures remain more efficient without sacrificing accuracy. Nevertheless, transformer-based extensions could become more valuable for multi-asset or portfolio-level pricing tasks involving high-dimensional stochastic processes.

An important practical insight concerns the models robustness across different types of option contracts. Both European and American-style options were priced accurately, with the latter requiring additional boundary constraints to capture early exercise behavior. The incorporation of real market data further confirmed that the PINN framework is not merely a theoretical construct but can be effectively calibrated to real-world observations using implied volatility surfaces and market-consistent parameters. In particular, the SABR-PINN exhibited the lowest mean absolute percentage error across multiple underlyings, showing its adaptability to diverse asset classes.

Despite these promising findings, some limitations persist. The convergence of PINNs can be sensitive to hyperparameter tuning, particularly the relative weights of PDE, initial, and boundary losses. Moreover, while PINNs handle smooth stochastic processes well, modeling abrupt regime shifts, jumps, or heavy-tailed behaviors remains an open challenge. Addressing these issues may require integrating PINNs with stochastic differential equation solvers, variational inference, or rough volatility formulations. Another promising direction involves adaptive collocation sampling to focus training on regions

where the PDE residual is large, thus improving convergence and stability.

In summary, this study demonstrates that embedding financial PDEs into neural architectures via physics-informed learning provides a powerful paradigm for option pricing. By harmonizing theoretical consistency with deep learning flexibility, PINNs can bridge the gap between classical financial modeling and modern data-driven techniques, paving the way for more realistic, explainable, and computationally efficient pricing systems in quantitative finance.

6. CONCLUSION

In this study, we investigated the effectiveness of Physics-Informed Neural Networks (PINNs) for solving option pricing problems governed by major financial models, including Black–Scholes, Constant Elastic Variance (CEV), Heston, and SABR. By embedding the governing partial differential equations into the neural network training process, our framework enables robust approximation of option prices while preserving theoretical financial constraints. We further introduced improved activation function formulations that better capture the second-order derivative behavior inherent in pricing PDEs, addressing well-known issues of conventional activation functions such as ReLU and Tanh.

Through extensive experiments on synthetic, randomly sampled, and real market datasets, we demonstrated that the proposed PINN-based pricing approach achieves lower pricing errors than the classical Black–Scholes analytical solution, particularly for models with stochastic or mixed local-stochastic volatility structures. Our results show that while all models benefit from the PINN formulation, the Heston-PINN and SABR-PINN approaches consistently outperform Black–Scholes-PINN and CEV-PINN, highlighting the value of modeling volatility dynamics more realistically. Moreover, for both European and American option contracts, the PINN-based solutions provided accurate estimates across different underlying assets and market conditions, supporting the practical applicability of this approach.

Overall, our findings demonstrate that PINNs offer a promising and flexible alternative for option pricing tasks, with the potential to extend toward more complex derivatives, higher-dimensional financial instruments, and market calibration workflows. Future research directions include improved handling of American-style free-boundary conditions, adaptive weighting strategies to balance PDE and boundary losses, the incorporation of jumps or rough volatility, and real-time deployment in high-frequency trading

environments. As markets continue to demand accurate and computationally scalable forecasting tools, PINN-based models provide an important step toward next-generation financial engineering methodologies.

REFERENCES

[1] M. Baxter and A. Rennie, *Financial calculus: an introduction to derivative pricing*. Cambridge university press, 1996.

[2] A. Hirsa, *Computational methods in finance*. Chapman and Hall/CRC, 2024.

[3] D. Wu and S. Jaimungal, "Robust risk-aware option hedging," *Applied Mathematical Finance*, vol. 30, no. 3, pp. 153–174, 2023.

[4] A. Simsek, "The macroeconomics of financial speculation," *Annual Review of Economics*, vol. 13, no. 1, pp. 335–369, 2021.

[5] B. Eraker, "The volatility premium," *The Quarterly Journal of Finance*, vol. 11, no. 03, p. 2150014, 2021.

[6] G. Bakshi, C. Cao, and Z. Chen, "Empirical performance of alternative option pricing models," *The Journal of finance*, vol. 52, no. 5, pp. 2003–2049, 1997.

[7] J. C. Hull and S. Basu, *Options, futures, and other derivatives*. Pearson Education India, 2016.

[8] B. Dupire *et al.*, "Pricing with a smile," *Risk*, vol. 7, no. 1, pp. 18–20, 1994.

[9] E. Derman and I. Kani, "Riding on a smile," *Risk*, vol. 7, no. 2, pp. 32–39, 1994.

[10] J. C. Cox, "The constant elasticity of variance option pricing model," *Journal of Portfolio Management*, p. 15, 1996.

[11] S. L. Heston, "A closed-form solution for options with stochastic volatility with applications to bond and currency options," *The review of financial studies*, vol. 6, no. 2, pp. 327–343, 1993.

[12] J. Hull and A. White, "The pricing of options on assets with stochastic volatilities," *The journal of finance*, vol. 42, no. 2, pp. 281–300, 1987.

[13] P. S. Hagan, D. Kumar, A. S. Lesniewski, and D. E. Woodward, "Managing smile risk," *The Best of Wilmott*, vol. 1, pp. 249–296, 2002.

[14] B. Eraker, M. Johannes, and N. Polson, "The impact of jumps in volatility and returns," *The Journal of Finance*, vol. 58, no. 3, pp. 1269–1300, 2003.

[15] D. B. Madan, P. P. Carr, and E. C. Chang, "The variance gamma process and option pricing," *Review of Finance*, vol. 2, no. 1, pp. 79–105, 1998.

[16] R. Cont, J. d. Fonseca, and V. Durrleman, "Stochastic models of implied volatility surfaces," *Economic Notes*, vol. 31, no. 2, pp. 361–377, 2002.

[17] J. Gatheral, "A parsimonious arbitrage-free implied volatility parameterization with application to the valuation of volatility derivatives," *Presentation at Global Derivatives & Risk Management, Madrid*, p. 0, 2004.

[18] J. Gatheral and A. Jacquier, "Arbitrage-free svi volatility surfaces," *Quantitative Finance*, vol. 14, no. 1, pp. 59–71, 2014.

[19] S. N. Cohen, C. Reisinger, and S. Wang, "Arbitrage-free neural-sde market models," *Applied Mathematical Finance*, vol. 30, no. 1, pp. 1–46, 2023.

[20] J. Guyon and J. Lekeufack, "Volatility is (mostly) path-dependent," *Quantitative Finance*, vol. 23, no. 9, pp. 1221–1258, 2023.

[21] Z. Zhao, X. Ding, and B. A. Prakash, "Pinnsformer: A transformer-based framework for physics-informed neural networks," 2024.

[22] Y. Uygun and E. Sefer, "Financial asset price prediction with graph neural network-based temporal deep learning models," *Neural Computing and Applications*, vol. 37, no. 30, pp. 25445–25471, 2025.

[23] B. Arat, H. F. Ateş, and E. Sefer, "Multi-context real estate market prediction," in *2025 33rd Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2025.

[24] P. Celik and E. Sefer, "Analyzing transaction graphs via motif-based graph representation learning for cryptocurrency price prediction," *Computational Economics*, pp. 1–22, 2025.

[25] M. Pala and E. Sefer, "Nft price and sales characteristics prediction by transfer learning of visual attributes," *The Journal of Finance and Data Science*, vol. 10, p. 100148, 2024.

[26] B. Seyhan and E. Sefer, "Nft primary sale price and secondary sale prediction via deep learning," in *Proceedings of the Fourth ACM International Conference on AI in Finance*, pp. 116–123, 2023.

[27] T. Tuncer, U. Kaya, E. Sefer, O. Alacam, and T. Hoser, "Asset price and direction prediction via deep 2d transformer and convolutional neural networks," in *Proceedings of the third ACM international conference on AI in Finance*, pp. 79–86, 2022.

[28] H. Han and X. Wu, "A fast numerical method for the black-scholes equation of american options," *SIAM Journal on Numerical Analysis*, vol. 41, no. 6, pp. 2081–2095, 2004.

[29] M. N. Anwar and L. Andallah, "A study on numerical solution of black-scholes model," *Journal of Mathematical Finance*, vol. 08, pp. 372–381, 01 2018.

[30] D. Jeong, J. Kim, and I.-S. Wee, "An accurate and efficient numerical method for black-scholes equations," *Commun. Korean Math. Soc*, vol. 24, pp. 617–628, 10 2009.

[31] D. Lamberton, "On the binomial approximation of the american put," *Applied Mathematics & Optimization*, vol. 82, no. 2, pp. 687–720, 2020.

[32] F. Gatta, V. S. Di Cola, F. Giampaolo, F. Piccialli, and S. Cuomo, "Meshless methods for american option pricing through physics-informed neural networks," *Engineering Analysis with Boundary Elements*, vol. 151, pp. 68–82, 2023.

[33] B. C. Kaibe, "Application of lie symmetries to solving partial differential equations associated with the mathematics of finance." May 2021.

[34] C. J. Papachristou, "Symmetry and Integrability of Classical Field Equations," 3 2008.

[35] C. Papachristou, "The maxwell equations as a bäcklund transformation," *Advanced Electromagnetics*, vol. 4, p. 52, 07 2015.

[36] I. Lagaris, A. Likas, and D. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, pp. 987–1000, 09 1998.

[37] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

[38] J. Sirignano and K. Spiliopoulos, "Dgm: A deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, 08 2017.

[39] T. Dockhorn, "A discussion on solving partial differential equations using neural networks," *CoRR*, vol. abs/1904.07200, 2019.

[40] M. Raissi and G. E. Karniadakis, "Hidden physics models: Machine learning of nonlinear partial differential equations," *Journal of Computational Physics*, vol. 357, pp. 125–141, 2018.

[41] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[42] B. Moseley, A. Markham, and T. Nissen-Meyer, "Solving the wave equation with physics-informed deep learning," 2020.

[43] J. Lim and D. Psaltis, "Maxwellnet: Physics-driven deep neural network training based on maxwell's equations," *APL Photonics*, vol. 7, no. 1, p. 011301, 2022.

[44] A. Aygün, "Physics informed neural networks for computational fluid dynamics," Master's thesis, Middle East Technical University, 2023.

[45] T. Hahn, *Option Pricing Using Artificial Neural Networks : an Australian Perspective*. PhD thesis, Bond University, 2013.

[46] H. A. Trønnes, "Pricing options with an artificial neural network: A reinforcement learning approach," Master's thesis, Institutt for samfunnsøkonomi, 2018.

[47] J. Ruf and W. Wang, "Neural networks for option pricing and hedging: A literature review," *Journal of Computational Finance*, Nov. 2019. Available at SSRN: `https://ssrn.com/abstract=3486363`.

[48] H. F. and, "Artificial neural network for option pricing with and without asymptotic correction," *Quantitative Finance*, vol. 21, no. 4, pp. 575–592, 2021.

[49] A. Dhiman and Y. Hu, "Physics informed neural network for option pricing," 2023.

[50] X. Wang, J. Li, and J. Li, "A deep learning based numerical pde method for option pricing," *Computational Economics*, vol. 62, no. 1, pp. 149–164, 2023.

[51] C. Z. Liu, Y. Zhang, L. Qin, and Y. Liu, "Kolmogorov–arnold finance-informed neural network in option pricing," *Applied Sciences*, vol. 14, no. 24, 2024.

[52] D. Hainaut and A. Casas, "Option pricing in the heston model with physics inspired neural networks," *Annals of Finance*, vol. 20, no. 3, pp. 353–376, 2024.

[53] N. Zamxaka, "Option pricing with physics-informed neural networks (pinns)," Master's thesis, University of Cape Town, 2024. OpenUCT. Field: Finance and Tax.

[54] T. Socaciu and P. Pașcu, "Physics-informed neural networks in pricing financial options," *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, vol. 16, no. 2, pp. 474–488, 2025.

[55] S. Cuomo, V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics–informed neural networks: Where we are and what's next," *Journal of Scientific Computing*, vol. 92, no. 3, p. 88, 2022.

[56] H. Wang, L. Lu, S. Song, and G. Huang, "Learning specialized activation functions for physics-informed neural networks," *Communications in Computational Physics*, vol. 34, pp. 869–906, 11 2023.

[57] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of political economy*, vol. 81, no. 3, pp. 637–654, 1973.

[58] L. Hodder, W. J. Mayew, M. L. McAnally, and C. D. Weaver, "Employee stock option fair-value estimates: do managerial discretion and incentives explain accuracy?," *Contemporary Accounting Research*, vol. 23, no. 4, pp. 933–975, 2006.

[59] R. C. Merton, "Theory of rational option pricing," *The Bell Journal of Economics and Management Science*, vol. 4, no. 1, pp. 141–183, 1973.

[60] E. Kandel and N. D. Pearson, "Option value, uncertainty, and the investment decision," *Journal of Financial and Quantitative Analysis*, vol. 37, no. 3, pp. 341–374, 2002.

[61] J. C. Cox and S. A. Ross, "The valuation of options for alternative stochastic processes," *Journal of Financial Economics*, vol. 3, no. 1-2, pp. 145–166, 1976.

[62] V. Linetsky and R. Mendoza, "Constant elasticity of variance (cev) diffusion model," in *Encyclopedia of Quantitative Finance*, Wiley, 2010.

[63] H. Albrecher, P. Mayer, W. Schoutens, and J. Tistaert, "The little heston trap," *Wilmott*, no. 1, pp. 83–92, 2007.

[64] J. Kienitz, T. McWalter, and R. Sheppard, *PDE Methods for SABR*, pp. 265–291. 09 2017.

[65] I. Khomasuridze, "Extensions of the sabr model for equity options," 2009.

[66] H. W. Lin, M. Tegmark, and D. Rolnick, "Why does deep and cheap learning work so well?," *Journal of Statistical Physics*, vol. 168, no. 6, pp. 1223–1247, 2017.

[67] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[68] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, 1989.

[69] A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind, "Automatic differentiation in machine learning: A survey," *Journal of Machine Learning Research*, vol. 18, pp. 1–43, 04 2018.

[70] S. Zou, X. Dong, and B. Yan, "Application of stochastic processes in financial market models," *Advances in Economics Management and Political Sciences*, 2024.

[71] OptionData.org, "Optiondata.org - option market data." Accessed: 2025-06-04.

[72] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[73] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[74] U.S. Department of the Treasury, "U.s. department of the treasury," 2025. Accessed: 2025-06-03.

[75] QuantLib Team, "QuantLib: a free/open-source library for quantitative finance," Oct. 2021.

[76] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.