T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

KESİRLİ MERTEBELI BLACK SCHOLES DİFERANSİYEL DENKLEMİ İÇİN SİNİR AĞI YÖNTEMİ

BUSHRA ISMAIL YASEN YASEN

MATEMATİK

SANLIURFA
2025

T.C.
HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

KESİRLİ MERTEBELI BLACK SCHOLES DİFERANSİYEL DENKLEMİ İÇİN SİNİR AĞI YÖNTEMİ

BUSHRA ISMAIL YASEN YASEN

MATHEMATICS

Tez Danışmanı: Prof. Dr. MAHMUT MODANLI

SANLIURFA
2025

TEZ ONAY SAYFASI

Jüri	**İmza**

ONAY

....

(İmza)

MATEMATİK Müdürü

TEZ ÖNERİSİ VE TEZ YAZMA YÖNERGESİNE UYGUNLUK BEYANI

.... / ...... /........

İmza

BUSHRA ISMAIL YASEN YASEN

TEŞEKKÜR

My sincere appreciation to my supervisor, **Prof Dr. Mahmut MODANLI**, whose enduring support and guidance were key to my success in this master's journey. Their support and knowledge were crucial in guiding this research and crafting this thesis.

I would like to sincerely thank **Dr.Salisu Ibrahim** for their offering to my thesis, contributions valuable feedback and recommendations that greatly improved this work.

I want to express my heartfelt appreciation to my family, companions, and cherished ones for their steady back and encouragement during this thesis venture. Your steady faith in me, along with your positive energy, adorned this experience with dynamic tones and truly made it extraordinary.

Finally, heartfelt appreciation to all the individuals who took part.

TABLE OF CONTENTS

ÖZET

YÜKSEK LİSANS TEZİ

KESİRLİ MERTEBELI BLACK SCHOLES DİFERANSİYEL DENKLEMİ İÇİN SİNİR AĞI YÖNTEMİ

BUSHRA ISMAIL YASEN YASEN

HARRAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MATEMATİK

Tez Danışman: Prof. Dr. MAHMUT MODANLI

Yıl: 2025, Sayfa: 43

Bu tez, kesirli mertebeden Black-Scholes diferansiyel denklemini (FOBSDE) çözmek için yapay sinir ağlarını (YSA) kullanan yeni bir yaklaşım sunmaktadır. Bu denklem, hafıza etkilerini ve yerel olmayan dinamikleri dikkate aldığı için opsiyon fiyatlandırmasında finansal matematik açısından büyük öneme sahiptir. Önerilen yöntem, kesirli türevler ve sınır koşullarını etkili bir şekilde ele alabilmek için spektral yaklaşımı kaydırılmış Legendre polinomları ve yapay sinir ağı optimizasyonu ile zekice birleştirmektedir.Çözüm, Legendre temel fonksiyonları serisi olarak ifade edilmekte olup, ağırlıklar artık matrisin Frobenius normunu azaltmak amacıyla gradyan inişi yoluyla yinelemeli olarak ince ayarlanmaktadır. Kapsamlı sayısal testler, yöntemin doğruluğunu ve verimliliğini göstermekte ve geleneksel sayısal teknikleri geride bıraktığını kanıtlamaktadır. Örneğin, parametreler r = 0.05, σ = 0.2 ve kesirli mertebe α = 0.5 ile yöntem, maksimum 1.23×10^{-3} hata ve göreli L^2 hatası olarak $2.87{\times}10^{-4}$. elde etmektedir. Bu sonuçlar, yapay sinir ağlarının karmaşık kesirli kısmi diferansiyel denklemleri çözmedeki potansiyelini vurgulamakta ve finansal modelleme için sağlam bir temel oluşturmaktadır. Bu araştırma, spektral teknikler ile makine öğrenimini bir araya getirerek modern finans mühendisliği için esnek ve ölçeklenebilir bir araç sunmaktadır.

ANAHTAR KELİMELER: Yapay sinir ağları, Kesirli Black-Scholes, Kesirli Hesap, Modifiye Legendre.

ABSTRACT

MASTER THESIS

NEURAL NETWORK METHOD FOR FRACTIONAL ORDER BLACK SCHOLES DIFFERENTIAL EQUATION

BUSHRA ISMAIL YASEN YASEN

HARRAN UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
MATHEMATIC

Thesis Supervisor: Prof. Dr. MAHMUT MODANLI

Year: 2025, Page: 43

This dissertation presents a novel approach using neural networks (NNM) to tackle the fractional-order Black-Scholes differential equation (FOBSDE). This equation is crucial in financial mathematics for option pricing, as it takes into account memory effects and non-local dynamics. The proposed method cleverly combines spectral approximation with shifted Legendre polynomials and neural network optimization, allowing for effective handling of fractional derivatives and boundary conditions. The solution is expressed as a series of Legendre basis functions, with weights that are fine-tuned iteratively through gradient descent to reduce the Frobenius norm of the residual error matrix. Extensive numerical tests demonstrate the method's accuracy and efficiency, proving it outperforms traditional numerical techniques. For instance, with parameters r = 0.05, σ = 0.2, and fractional order α = 0.5, the method achieves a maximum error of 1.23×10^{-3} and a relative L^2-error of 2.87×10^{-4}. These results underscore the potential of neural networks in solving complex fractional partial differential equations, laying a strong foundation for financial modeling. This research bridges spectral techniques and machine learning, providing a flexible and scalable tool for modern financial engineering.

KEYWORDS: Neural networks, Fractional Black-Scholes, Fractional Calculus, Shifted Legendre.

ŞEKİLLER DİZİNİ

ÇİZELGELER DİZİNİ

SİMGELER

σ : Symbolizes volatility

u : Current market value of an option

t : Represents time

α : Alpha

β: Beta

KISALTMALAR

BSE : Stands for Black-Scholes Equation

CD : Caputo derivative

CFD : Caputo Fractional Derivative

${}_{a}^{C}D^{\theta}$: Caputo-type fractional derivative operator

E_a : Mittag-Leffler function

FC : Fractional calculus

FPDEs : Fractional Partial Differential Equations

NNM : Neural Network Method

r : Refers to the interest rate

Γ : Gamma function

x : Denotes the price of the asset underlying a financial derivative

1. GİRİŞ

Over the past several years, much research and study has been carried out on the fractional order of Scholes differential equations (FOBSDEs) in many industries due to their wide application, including neural network approaches to solving the fractional equations of the Black-Scholes equations. (Bajalan, 2021) proffered the artificial neural network 2-layered to solving both fractional and ordinary Black-Scholes PDEs by using Adam optimization. (Zhang, Yang, Zhao, 2022) In the context of the time fractional Black-Scholes model, researchers created an extreme learning machine that utilizes a Legendre wavelet neural network, showcasing enhanced efficiency. (Santos, Ferrira, 2024) To solve Black-Scholes equations, they applied neural networks by using real-world stock option data. The short-term forecasts were more accurate than traditional analytical solutions. (Arora et al., 2023) The Fractional Order Black-Scholes-Merton (FOBSM) model has been proposed. Neural networks and fractional calculus are integrated to capture better complex market dynamics such as tail behavior, memory effects, and volatility clustering. These researchers have demonstrated the potential of neural network methods collectively to enhance option pricing flexibility and accuracy in handling various forms of the Black-Scholes equation.

In the past 300 years, fractional calculus, which is a mathematical concept, this subject has been extensively researched. and used in various fields. fractional order (FO) to Compared to the systems of integer-order, the derivatives of fractional-order (FO) are excellent tools for describing the internal and memory properties of various processes and materials in many areas with applications, such as electrical conductivity, negative emissions, and thermal conductivity. Therefore, fractional calculus has attracted the attention of engineers and physicists. Also, in some neural network models, the fractional computation has been applied. For this reason, the study of fractional neural networks (NN) is crucial for numerous significant outcomes in chaos dynamics, practical applications, and stability analysis (Younis et al., 2022).

To analyze and understand the construction of financial instruments, after discussing the BSE in modern finance, which is the most influential mathematical model, So the BSE is a partial differential equation developed in the 1970s as a tool to estimate the option over time or price of a call. Known for its integration and simplicity, the BSE expedited progress and revolutionized markets in financial mathematics. Since the BSE is a two-dimensional heat equation, its genesis is different (Jeff et al., 1995).

In the past, research on the neural network method has moved forward considerably in a many different kind of areas, including solving complex mathematical issues. For this time, we want to use a method for solving FOBSE the mathematical software used to obtain the computed result and the MATLAB application.

This study aims to explore and present the FOBSPDEs. The solution is derived by considering the initial and boundary value conditions, utilizing the NNM. The FOBSPDEs can be expressed as follows:

$$\left\{\begin{array}{c} {}_{0}^{c}D_{t}^{\propto}V(t,s)+\frac{1}{2}\sigma^{2}S^{2}\frac{\partial^{2}V}{\partial S^{2}}+rS\frac{\partial V}{\partial S}-rV=0 \\ V(0,S)=\varphi(S), 0<\propto\leq 1 \quad V=V(t,s) \quad 0\leq\propto\leq 1 \\ V(t,0)=a_{1}(t), V(t,1)=a_{2}(t), 0\leq t\leq\rho \end{array}\right\}. \tag{1.1}$$

1.1. Fundamental concepts

In this section, we explore the fundamental definition and key characteristics of fractional-order calculus. Additionally, we provide a definition, several examples, and a theorem for further understanding.

1.1.1 Gamma Function

Definition1.1.1.1: (Erdos and Renyi, 1960) The gamma function extends the concept of the factorial function to real and complex numbers, which is symbolized by Γ. Specifically, if $k \in \{1,2,3, \dots \dots.\}$, then

$$\Gamma(k) = (k-1)!$$

More generally, for any positive real number α' , $\Gamma(\alpha')$ is defined as

$$\Gamma(\alpha') = \int_0^\infty x^{\alpha'-1} e^{-x} dx \ , \ where\ \alpha' > 0.$$

Note that for β = 1, we can write

$$\Gamma(1) = \int_0^\infty e^{-x} dx$$

$$= 1.$$

By using the change of variable $x = \mu y$, we can derive an equation that is frequently helpful when dealing with the gamma distribution.

$$\Gamma(\alpha') = \mu^{\alpha'} \int_0^\infty y^{\alpha'-1} e^{-\mu y} dy \ , \ where\ \alpha', \mu > 0.$$

Additionally, it can be demonstrated using integration by parts that.

$$\Gamma(\alpha' + 1) = \alpha' \Gamma(\alpha'), \quad where\ \beta > 0$$

Note that if $\alpha' = k$, where k is a positive integer, the above equation simplifies to.

$$k! = k(k-1)!$$

Properties 1.1.1.2: Some properties of the gamma function

For any positive real number α' :

1. $\Gamma(\alpha') = \int_0^\infty x^{\alpha'-1} e^{-x} dx,$
2. $\int_0^\infty x^{\alpha'-1} e^{-\mu y} dx = \frac{\Gamma(\alpha')}{\mu^{\alpha'}}, \ where\ \mu > 0,$
3. $\Gamma(\alpha' + 1) = \alpha' \ \Gamma(\alpha'),$
4. $\Gamma(k) = (k-1)!, \quad where\ k = 1,2,3, \dots \dots,$
5. $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}.$

1.1.2 Beta Function

Definition1.1.2.1: (Abramowitz and Stegun, 1965) The Beta function to represent by $\beta(m', n')$ or $B(m', n')$ is defined as $\beta(m', n') = \int_0^1 x^{m'-1}(1-x)^{n'-1} dx,\ (m' > 0,$ $n' > 0$).

Properties 1.1.2.2: Some properties of the Beta function

1. $\beta(m', n') = \beta(n', m'),$
2. $\beta(m', n') = 2\int_0^{\frac{\pi}{2}} sin^{2m'-1}\theta cos^{2n'-1}\theta d\theta,$
3. $\beta(m', n') = \int_0^\infty \frac{x^{m-1}}{(1+x)^{m+n}} dx,$
4. $\beta(m', n') = \int_0^\infty \frac{x^{m-1} + x^{n-1}}{(1+x)^{m+n}} dx.$

Remark 1: The connection between the gamma function and beta function

$$\beta(x,y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}, \qquad x > 0, \qquad y > 0.$$

1.1.3 Mittag - Leffler Function

Definition 1.1.3.1: (Mittag, 1903) is a special function the Mittag – Leffler function $E_{\alpha'\beta'}$, a complex parameter $\alpha' and\ \beta'$. It can be characterized by the following series when the real part of α' is definitely positive: $E_{\alpha'\beta'}(z') = \sum_{k=0}^{\infty} = \frac{z'^k}{\Gamma(\alpha' k + \beta')}$.

Properties 1.1.3.2: Some properties of the Mittag - Leffler function

1. $E_{1,1}(z') = e^{z'}$,
2. $E_{2,1}(z'^2) = \cosh(z')$,
3. $E_{2,2}(z'^2) = \frac{\sinh(z')}{z'}$,
4. $E_{\alpha',1}(z') = E_{\alpha'}(z')$,
5. $E_{\frac{1}{2},1}(z') = e^{z'^2} erfc(-z')$.

Example 1.1.3.3: Show that $E_{1,2}(z') = \frac{e^z - 1}{z}$.

We have

$$E_{1,2}(z') = \sum_{b=0}^{\infty} \frac{z'^b}{\Gamma(b+2)}$$

$$= \sum_{b=0}^{\infty} \frac{z'^b}{\Gamma(b+1)!}$$

$$= \sum_{b=0}^{\infty} \frac{z'^{b+1}}{(b+1)!} = \frac{1}{z'}\left(e^{z'} - 1\right).$$

Where erf is the error function.

1.1.4 Caputo fractional derivative

The concept of fractional integrals was used in Caputo's formulation of fractional derivatives.

Definition1.1.2: (Caputo, 1967) If f: $R_+ \to R$ and $\delta \in \langle m-1, m)$, $m \in N$, then

$$^{C}D^{\delta} f(t) = \frac{1}{\Gamma(m-\delta)} \int_0^t \frac{f^{(m)}(\tau)}{(t-\tau)^{\delta-m+1}}\, d\tau.$$

The expression where Γ represents the gamma function is known as the Caputo fractional derivative of order δ, depending on its existence.

In the Caputo definition of the fractional order derivative, we start by calculating the ordinary derivative of a natural order, and then we find the Riemann-Liouville integral of fractional order from that resulting function. Thus, for $\delta \in \langle m - 1, n)$, the Caputo fractional derivative $^{C}D^{\delta} f(t)$ exists if $f \in C^{\delta}(\langle 0, t \rangle)$. Since the Caputo fractional derivative is expressed in integral form, it acts as a non-local operator. This means it has a "memory" property, indicating that the current state is influenced by past states.

As $\delta \to m$, the Caputo derivative converges to the n-th order classical derivative of the function f, e.g. $\lim_{\delta \to m} {}^{C}D^{\delta} f(t) = f^{(m)}(t)$.

The fractional derivative, as described in formula (1.3), is clearly defined and meets all the specified conditions (i) through (v).

The following theorems support this claim.

Let us examine functions *f* and *g* that are defined on the positive real numbers, *f, g*:$R_+ \rightarrow R$. We assume that the Caputo derivatives of order α, α ∈ $\langle m-1, m)$, $m \in$ N, exist for these functions.

Theorem 1.1.4.1: The Caputo derivative acts as a linear operator, which means that for any *any a, b* ∈ R, this property is valid.

$$ {}^{C}D^{\delta}\left(bf(t)+ag(t)\right)=b\,{}^{C}D^{\,\delta}f(t)+a\,{}^{C}D^{\,\delta}g(t). $$

Proof: Let ${}^{C}D^{\,\delta}f(t)\,,{}^{C}D^{\,\delta}g(t)$ Let f and g be functions whose Caputo derivatives are denoted as f and g, respectively.

Then

$$ {}^{C}D^{\delta}\left(bf(t)+ag(t)\right)=\frac{1}{\Gamma(m-\delta)}\int_{0}^{t}\frac{(bf(t)+ag(t))^{(m)}}{(t-\tau)^{(\delta-m+1)}}d\tau= $$

$$ =\frac{1}{\Gamma(m-\delta)}\left(b\int_{0}^{t}\frac{f^{(m)}(t)}{(t-\tau)^{(\delta-m+1)}}\,d\tau+a\int_{0}^{t}\frac{g^{(m)}(t)}{(t-\tau)^{(\delta-m+1)}}\,d\tau\right)= $$

$$ =\frac{1}{\Gamma(m-\delta)}\,b\int_{0}^{t}\frac{f^{(m)}(t)}{(t-\tau)^{(\delta-m+1)}}\,d\tau+\frac{1}{\Gamma(m-\delta)}\,a\int_{0}^{t}\frac{g^{(m)}(t)}{(t-\tau)^{(\delta-m+1)}}\,d\tau= $$

$$ =b\,{}^{C}D^{\,\delta}f(t)+a\,{}^{C}D^{\delta}\,g(t). $$

Definition1.1.3: Legendre polynomials, which are defined on the interval [−1, 1], are derived using a series of recursive relations (H. Qu et al., 2022).

$r_0(x) = 1,$

$r_1(x) = 1,$

$$r_{j+1}(x) = \frac{2j+1}{j+1} x r_j(x) - \frac{j}{j+1} r_{j-1}(x), j = 1,2,3, \dots .$$

The Legendre polynomial can be analytically represented as

$$R_j(x) = \sum_{k=0}^{j} \frac{(-1)^k (j+k)!}{(j-k)!} \frac{(j-x)^k}{2^k \, (k!)^2} \, , j = 0,1,2, \dots .$$

The explicit analytical expression for the shifted Legendre polynomial of degree j, defined on the interval [0, X], is provided.

$$R_{Xj}(x) = \sum_{s=0}^{j} \frac{(-1)^{j+s} (j+s)!}{(j-s)!} \frac{x^k}{X^s \, (s!)^2} \, , j = 0,1,2, \dots .$$

The limit values of $R_{Xj}(x)$ are

$R_{Xj}(x) = (-1)^j,$

$R_{Xj}(x) = 1.$

1.2 Black Scholes Equation

1.2.1 History of the Black-Scholes Equation

The Black-Scholes model presents a significant partial differential equation used for valuing financial derivatives. The importance of the Black-Scholes equation cannot be overstated, as it has fundamentally shaped modern finance and monetary markets. This discussion will explore the origins of the Black-Scholes equation and the impact of initiatives across the country aimed at breaking down barriers and promoting equity and opportunity for everyone. As the title suggests, the Black-Scholes equation was a collaborative effort between economists Fischer Black and Myron Scholes (Planes and Alex, 2016).

In their groundbreaking 1973 paper "The Valuation of Options and Corporate Obligations," financial consultant Fischer Black and emerging finance lecturer Myron Scholes laid the groundwork for the Black-Scholes model during the early 1970s at the Massachusetts Institute of Technology (MIT). This influential work introduced a partial differential equation that provides a framework for assessing options over a defined time period. Robert Merton also played a crucial role in the model's development, being the first to publish a paper that explained its mathematical foundations. For their significant contributions to the Black-Scholes model, Merton and Scholes received the Nobel Prize in Economics in 1997. Although Black had passed away two years earlier, his contributions were recognized posthumously, underscoring the profound impact of the Black-Scholes equation on financial markets. Before this model, traders of financial derivatives relied on speculation to gauge their value, as there was no systematic method for assessment, which introduced considerable risk into trading and indicated that these transactions were largely speculative (Planes and Alex, 2016).

The Black-Scholes model has had a significant impact on financial markets, primarily because it provides a way to quantify the value of financial instruments. This breakthrough transformed options from simple speculative bets on an underlying asset into instruments that can more accurately reflect the asset itself. As a result, market participants were able to trade these options with a better grasp of their intrinsic value. This advancement led to the creation of derivative markets and options exchanges, which now boast an astonishing value of $710 trillion. In the following sections, we will examine each part of the Black-Scholes equation and look into its historical background, which will deepen our understanding of how this model has fundamentally changed modern financial markets (Planes and Alex, 2016).

1.2.2 Evolution of an Asset

Before exploring the behavior of an asset, it's important to clarify what a financial asset is and what it entails. An asset is essentially an economic resource that holds value and influences market actions. Assets vary widely in form and function, and they can be utilized in numerous ways. Individuals engage in buying, trading, and selling assets with the expectation of generating positive economic returns. You can think of assets as a type of economic currency, and like paper money, their value can fluctuate over time, sometimes significantly.

Take stocks, for example. A stock represents a share of ownership in a corporation. If I invest $20 in stock, I own an asset that has the potential to generate profit because it holds value. However, the potential for profit is contingent on the fluctuating value of the shares. To maximize my profit, I need to understand when and how the value of my stock will change. Price optimization is a key goal in all financial markets, and achieving it relies heavily on mathematical models (Planes and Alex, 2016).

If the company I've invested in launches a successful new product, it's likely that my shares will increase in value. On the other hand, if the company reports disappointing quarterly earnings, the share price could drop. The next page features a graph showing the changes in value of a specific asset, namely Coca-Cola stock (Planes and Alex, 2016).

Take note of the erratic and unpredictable parts of the graph. This volatility is exactly what complicates the creation of mathematical models. Financial models aim to illustrate the path of an asset to predict future values and highlight significant trends (Planes and Alex, 2016).

Figure 1.2.1: The chart displays the KO stock price over a designated time frame (Planes and Alex, 2016).

We aim to create a model that captures how an asset's price, represented as S, changes over time. To explore this, we will define a return that indicates the relative change in price. At a specific time, t, the asset's price is S, and after a short time interval, $t + dt$, it changes to S + dS. This indicates a small adjustment in S by an infinitesimal amount, dS, over a tiny duration, dt. In our analysis of the asset's return, we look at dS in relation to S, as this ratio gives us the expression dS/S. Traditional financial models usually break down the return into two main components: the stochastic part and the deterministic part. The deterministic component, which can be compared to a risk-free return on investment, is denoted as φdt and is often considered a constant in basic financial models, although the parameter μ may vary in more advanced models (Planes and Alex, 2016).

The second element captures the unpredictable nature of asset prices, represented as σdX. Here, σ stands for the asset's volatility, which is precisely defined as the standard deviation of its returns. More broadly, volatility can be seen as the level of risk associated with the asset's value. Asset prices are naturally subject to change, with some assets showing greater fluctuations than others, which adds a risk component to investment choices; σ is used to quantify this risk (Planes and Alex, 2016).

Volatility, defined by the standard deviation, measures how much historical return data points differ from their average. By looking at the historical returns of an asset and how they are distributed over time, one can assess the level of volatility. A wider range of these data points from the average suggests higher volatility. The topic of asset volatility is important and will be explored in more detail later in this chapter. It's important to note that σ is modified by dX, which signifies a random sample drawn from a normal distribution. The decision to use a sample from a normal distribution stems from the need to model a random and variable phenomenon (Planes and Alex, 2016).

The Central Limit Theorem clearly states that when you have a sufficient number of random variables, their average will approximate a normal distribution. This is the reason dX is considered a random variable taken from a normal distribution. By integrating these components, we can represent our return through a stochastic differential equation.

$$\frac{dS}{S} = \sigma dX + \varphi dt$$

This equation describes a random walk, which is made up of a series of discrete, stochastic steps. Random walks are widely used in various fields, particularly in finance. As shown in Figure 3, a random walk is illustrated by a connected series of discrete points on a graph. The importance of random walks lies in their function as a mathematical model that mimics real-world phenomena. This is especially relevant in the context of the model mentioned earlier, as it enables us to explore the interactions between our variables. When we look at S over a small-time interval dt, dS can be seen as a step in this sequence. It is well-known that asset prices in financial markets tend to follow a random walk pattern. However, it is essential to understand the specific implications of this idea. Random walks do not yield precise predictions of asset prices; instead, they provide useful probabilistic insights into how assets behave (Planes and Alex, 2016).

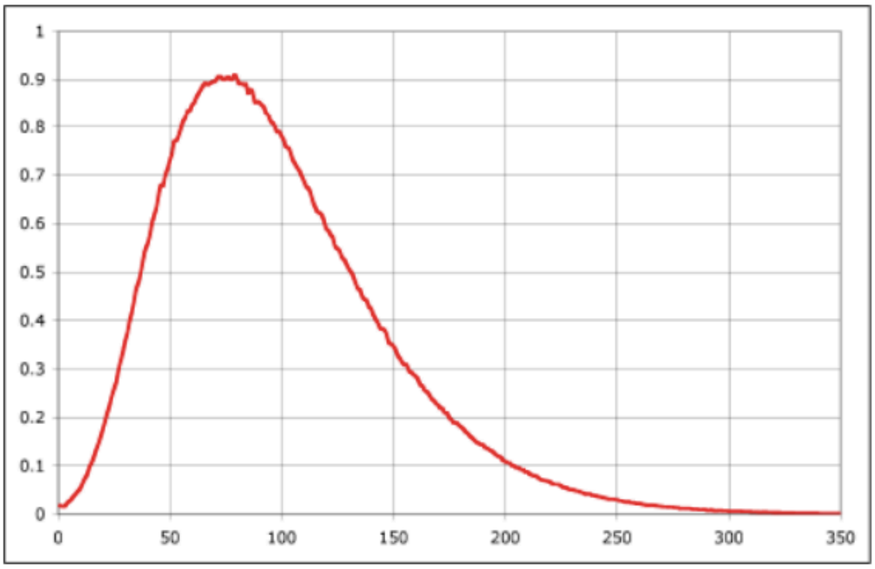

Figure 1.2.2: A lognormal distribution (Planes and Alex, 2016)

The concept of random walks being "memoryless" highlights their significance in various theoretical frameworks. In particular, random walks are categorized as a type of Markov process, which embodies essential properties linked to this classification. We can now examine the random walk described by the stochastic differential equation. If an asset S adheres to the random walk defined by this equation, the probability density function of the related random variable approximates a slightly skewed normal distribution, known as a lognormal distribution. A lognormal distribution is defined as a continuous distribution where the natural logarithm of the variable is normally distributed. This distribution is especially important in the context of the Black-Scholes model, as it effectively eliminates negative values, reflecting the reality that asset prices cannot be negative (Planes and Alex, 2016).

1.2.3 Black-Scholes Derivation

In deriving and analyzing the Black-Scholes equation, we will first explore the stochastic differential equation that underpins it.

$$\frac{dS}{S} = \underbrace{\sigma dX}_{\text{Random}} + \underbrace{\varphi dt}_{\text{Deterministic}} \tag{1.2.1}$$

Here, φdt is the average rate of growth of the asset price, which is predictable.

This part of the equation is often referred to as "the drift." On the other hand, σdX represents the volatility of the underlying asset and is completely random. To gain a clearer understanding of the mechanics of (3.1), we will consider a function of S, which denotes the value of the underlying asset. We can take $f(S)$ and expand this function.$df = \frac{df}{dS} dS + \frac{1}{2}\frac{d^2f}{dS^2} dS^2 + \cdots$ By varying S by a small amount dS, we can see that it is equivalent to varying $f(S)$. This concept is vital not only for deriving the Black Scholes equation but also for grasping the underlying principles of that derivation. Additionally, this idea highlights the importance of Ito's Lemma, which is a key tool in the derivation process.

$$df = \frac{df}{dS} dS + \frac{1}{2}\frac{d^2f}{dS^2} dS^2 + \cdots \tag{1.2.2}$$

Along with this expansion, we will also present another crucial result in the derivation. We assert that dX^2 converges to dt as dt approaches 0. In simpler terms, as dt gets smaller (meaning the time interval decreases), dX^2 increasingly resembles dt. Now, let's examine (3.1).

$$dS^2 = (S\sigma dX + S\varphi dt)^2$$

$$= \sigma^2 S^2 dX^2 + 2\varphi\sigma S^2 dXdt + \varphi^2 S^2 dt^2$$

Given that $dX = \sqrt{dt}$,

$$= \sigma^2 S^2 dt + 2\varphi\sigma S^2 dt^{\frac{3}{2}} + \varphi^2\sigma^2 dt^2$$

as $dX = \sqrt{dt}$

We observe that $\sigma^2 S^2 dX^2$ is the primary term since we are focusing on a small dt. This stems from the concept that examining the behavior of dS over the smallest time interval will yield the most precise assessment of the asset. Therefore,

$$dS^2 = \sigma^2 S^2 dX^2 + \underbrace{\cdots}_{\text{Insignificant higher order terms}}$$

$$\Rightarrow dS^2 = \sigma^2 S^2 dX^2 .$$

The dots represent the higher order terms that are not taken into account since the leading term produces the smallest dt. Now, if we substitute this into equation 3.2, we have:

$$df = \frac{df}{dS}(S\sigma dX + S\varphi dt) + \frac{1}{2}\frac{d^2f}{dS^2}\sigma^2 S^2 dt$$

$$= \sigma S dX\frac{df}{dS} + \varphi S\frac{df}{dS}dt + \frac{1}{2}\frac{d^2f}{dS^2}\sigma^2 S^2 dt$$

$$= \sigma S dX\frac{df}{dS} + \left(\varphi S\frac{df}{dS} + \frac{1}{2}\frac{d^2f}{dS^2}\sigma^2 S^2\right)dt.$$

In applying It σ's lemma, we have connected a small change in a function of a variable to a small change in the variable itself. It's important to note that this same approach can be extended to a function of two variables, such as $f(S,t)$. Here, the independent variables are S, representing the value of the underlying asset, and t, which denotes time. Below is the result for $V(S, t)$, where V indicates the value of a portfolio.

$$dV = \sigma S \frac{\partial V}{\partial S} dX + \left(\varphi S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \ \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t}\right) dt.$$

From here, we will derive the Black-Scholes equation. To achieve this, let's create a portfolio:

$$\Pi = V - \Delta S.$$

This portfolio includes a single option V, and we assign a number to Δ that scales the underlying asset, S. This scaling factor will subsequently modify the value of the option and, in turn, the overall portfolio. We will determine a value for Δ later, but a logical choice will arise that will be significant in the final stages of the derivation. Based on our definition of the portfolio, we can state that the change in the value of this portfolio over one-time step is:

$$\mathrm{d}\Pi = V - \Delta dS.$$

Substituting dV and dS above yields,

$$\mathrm{d}\Pi = \ \sigma S \frac{\partial V}{\partial S} dX + \varphi S \frac{\partial V}{\partial S} dt + \frac{1}{2}\sigma^2 S^2 \ \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} dt - \Delta S \sigma dX - \Delta S \varphi dX$$

$$= (\sigma S \frac{\partial V}{\partial S} - \Delta S \sigma) dX + \left(\varphi S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \ \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} + \Delta S \varphi\right) dt$$

$$= \underbrace{\sigma S (\frac{\partial V}{\partial S} - \Delta) dX}_{\text{random portion}} + \left(\varphi S \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \ \frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} + \Delta S \varphi\right) dt.$$

We can now observe how a natural choice for Δ arises, which will remove the random element found in the natural walk. This clearly leaves us with only the deterministic component, enabling us to more accurately predict asset behavior.

$$\frac{\partial V}{\partial S} = \Delta$$

$$\Rightarrow d\Pi = \left(\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \, \frac{\partial^2 V}{\partial S^2}\right) dt.$$

Here we explore a more nuanced concept of supply and demand known as arbitrage. Arbitrage involves the ongoing buying and selling of assets to take advantage of price discrepancies. Due to the presence of arbitrage in the market, we can relate the return of a portfolio Π to a risk-free return from an investment, represented as $r\Pi dt$.

$$\Rightarrow r\Pi dt = \left(\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \, \frac{\partial^2 V}{\partial S^2}\right) dt.$$

Substituting our portfolio and our Δ into the above equation yields,

$$r(\mathrm{V} - \Delta \mathrm{S})dt = \left(\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \, \frac{\partial^2 V}{\partial S^2}\right) dt$$

$$\Leftrightarrow \frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \, \frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

Finally, we have derived the Black-Scholes equation presented above (Planes and Alex, 2016).

1.3 Neural Network

The origins of artificial Neural Networks date back to 1940, when researchers Walter Pitts and Warren McCulloch showed that neurons in the human brain perform logical operations and produce binary outputs based on a specific threshold, indicating whether they are activated or inactive. This groundbreaking discovery led to the development of mathematical models that attracted significant attention in the Artificial Intelligence community. However, at that time, the emerging field of computing was not ready to handle such complex algorithms. As a result, interest in Neural Networks diminished as researchers turned their attention to other methods that seemed more promising. A major turning point came in 1970 with Seppo Linnainmaa's introduction of Backpropagation, which greatly enhanced the effectiveness of Neural Networks. Still, these models remained largely overlooked until 2012, when students achieved remarkable results in the ImageNet Large Scale Visual Recognition Competition, reigniting interest in Neural Networks within the Artificial Intelligence field (Lichtner-Bajjaoui, 2020).

1.3.1 Mathematical definition of a Neural Network

The basic concept of a Neural Network is to develop a model that learns from a collection of N samples.

$$D = \{[a_1, \ldots, a_N], [b_1, \ldots, b_N]\} \tag{1.3.1}$$

will approximate an unknown function f, with

$$f(a_i) = b_i.$$

The model typically consists of one input layer, one output layer, and at least one hidden layer. These layers are interconnected through weighted transitions, which can be represented by multiplying a weight matrix with a vector that represents the output from the previous layer. Each layer is made up of several units known as neurons. The number of units in each layer determines the layer's dimension. The activation functions within the neurons act as thresholds, deciding whether the information received by a neuron is relevant for further calculations. If the information is deemed relevant, it is passed on to the next layer until it reaches the output layer. During the training phase, the model is provided with a sample dataset D, where the matrix $[a_1, \dots, a_N]$ serves as the input and $[b_1, \dots, b_N]$ as the output matrix. After processing each sample $(a_i, b_i) \in D$, the model measures the difference between its output and the actual desired output b_i using a loss function, such as the L^2-norm or the Euclidean norm. Based on these results, the weights of the transitions are adjusted to improve performance in the next iteration. The input layer's dimension is determined by the input data's dimension, while the output layer's dimension corresponds to the problem being solved. For instance, in a regression task, the output layer will have a dimension of one, as the model aims to assign a single value to each input. In contrast, for a classification problem, the output layer will have a dimension of k, where k represents the number of different categories into which we want to sort the data. With a sufficiently large sample, the model is expected to provide a good approximation of the desired output. Next, we will seek to identify the best estimator $\hat{F}$ of f that meets our criteria.

$$\hat{F}(a_i) = b_i.$$

That means the best approximation of f for given observation D. Then we will apply $\hat{F}$ to unknown values z to make a prediction or classification (Lichtner-Bajjaoui, 2020).

Definition 1.3.2: Let $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}^n$ for all $i \leq N \in \mathbb{N}$

$$D := \{[a_1, \dots, a_N], [b_1, \dots, b_N]\}.$$

The **sample** of size N. Let

$$A := [a_1, \dots, a_N].$$

Let $n \times N$ be a matrix that contains the training input vectors a_i from a sample D, which has a size of N. From now on, we will refer to D as the training set.

$$B := [b_1, \dots, b_N]$$

will model the corresponding observed training output vectors b_i and form a $k \times N$ matrix (Lichtner-Bajjaoui, 2020).

Definition 1.3.3: By $C := [c_1, \dots, c_N]$ we denote the test set with test vectors $c_i \in \mathbb{R}^n$ (Hastie et al, 2009).

Definition 1.3.4: Let $\tilde{A}$ be an n- dimensional random vector representing the random selection of training input vectors c_i. The random vector C represents the selection of test data. We will denote $\tilde{B}$ as a k- dimensional random vector that models the network's output for all possible inputs (Lichtner-Bajjaoui, 2020).

Definition 1.3.5: A function $F : \mathbb{R}^n \to \mathbb{R}^k$ represents the network activity and maps input vectors to output vectors (Lichtner-Bajjaoui, 2020).

The function F satisfies

$$F(a_i) + \epsilon_i = b_i,$$

where ϵ_i is a random variable with unknown distribution J_ϵ that describes the **noise** that is produced by the unknown values Z.

Our objective is to train the network by adjusting the function F so that it closely approximates f. This is achieved by minimizing a **risk function** R, which is influenced by the selected model F. The risk function is determined by integrating a **loss function** L that quantifies the approximation error.

$$R(F) = \int L\left(\tilde{B}, F(\tilde{A})\right) dJ(\tilde{A}, \tilde{B}), \qquad (1.3.2)$$

Given the unknown joint distribution of $(\tilde{A}, \tilde{B})$.

we are unable to compute the integral due to a lack of information about the measure J. Therefore, our approach will focus on minimizing the average loss within the sample D, based on the provided observations $(\tilde{A}, \tilde{B})$.

$$R_E(F) = \frac{1}{N}\sum_{i}^{N} L\big(b_i, F(a_i)\big).$$

We can understand, however, that an optimal solution to (1.3.2) must satisfy

$$F(a_i) = E(\tilde{B}|a_i)\ . \qquad (1.3.3)$$

This represents the expected value of $\tilde{B}$, given that we already have the output of a_i. To clarify this connection, let's consider the following example (Lichtner-Bajjaoui, 2020).

Example 1.3.5: Let the loss function L be the quadratic error that is

$$L\big(b, F(a)\big) = (b - F(a))^2.$$

Calculating the risk for this loss function we obtain

$$R(F) = \int \big(b - F(a)\big)^2 dJ(a, b)$$

$$= E\left[\left(\tilde{B} - F(\tilde{A})\right)^2\right]$$

$$= E\left[\left(\tilde{B} - E(\tilde{B}|\tilde{A}) - E(\tilde{B}|\tilde{A}) - F(\tilde{A})\right)^2\right]$$

$$= E\left[\left(\tilde{B} - E(\tilde{B}|\tilde{A})\right)^2\right] + E\left[\left((E(\tilde{B}|\tilde{A}) - F(\tilde{A})\right)^2\right]$$

$$+2E\left[\left(\tilde{B} - E(\tilde{B}|\tilde{A})\right)\left(E(\tilde{B}|\tilde{A}) - F(\tilde{A})\right)\right].$$

Notice that since $E(\tilde{B}|\tilde{A})$ is a projection of $\tilde{B}$ on the sub-σ-field generated by $\tilde{A}$ it follows that $\left(\tilde{B} - E(\tilde{B}|\tilde{A})\right)$ is orthogonal to $E(\tilde{B}|\tilde{A})$. We can conclude therefore, by

$$R(F) = E\left[\left(\tilde{B} - E(\tilde{B}|\tilde{A})\right)^2\right] + E\left[\left((E(\tilde{B}|\tilde{A}) - F(\tilde{A})\right)^2\right]$$

that the best choice of F to minimize R is the conditional expectation

$$F(a) = E(\tilde{B}|a).$$

Definition 1.3.6: $\hat{F}: \mathbb{R}^n \rightarrow \mathbb{R}^k$ denotes an estimator that satisfies

$$\hat{F}_D = \underset{F}{\arg\min}\ R(F). \tag{1.3.4}$$

For a given sample S.

It is not always clear whether the optimal estimator E ($\tilde{B}$|a) exists within the set of functions that the network can implement or that meet specific criteria for the approximating function, such as continuity. We refer to the difference between E ($\tilde{B}$|a) and $\hat{F}$ as the error (Lichtner-Bajjaoui, 2020).

2. ÖNCEKİ ÇALIŞMALAR

The Black-Scholes-Merton Fractional Order Model (FOBSM) improves pricing accuracy by combining the traditional Black-Scholes-Merton model with neural networks and a fractional accounting framework. This approach overcomes the limitations of standard methods and better captures market behaviors like memory effects and volatility clustering. The results show that this hybrid method greatly enhances price forecasting accuracy in complex economic situations. This research marks an important step in bridging theoretical models with the complexities found in real-world financial markets (Arora et al., 2023).

For resolving fractional differential equations, the study offers an innovative method of initial values using neural networks based on cosine functions. for the two single and integral fractional differential equations Numerical solutions have been obtained by repeated training; the effectiveness of the method is shown by computer illustrations and numerical results (Qu and Liu, 2015).

The approach's effectiveness in tackling fractional Black-Scholes equations has been demonstrated, highlighting its simplicity and high accuracy. This is accomplished by using the Aboodh decomposition method to solve the Black-Scholes fractional partial differential equation that involves two assets. The method integrates the Adomian decomposition technique with the Aboodh transform and has been suggested for solving both nonlinear and linear fractional differential equations (Al faqeih and Ozis, 2020).

An extreme learning machine paired with a Legendre wavelet neural network can effectively tackle the time fractional Black-Scholes model. The challenge of pricing European options was approached using algebraic equations, leveraging the operational matrix of the fractional derivative based on the two-dimensional Legendre wavelet. The network design focuses on minimizing overfitting while improving the learning process. A related study that used the implicit differential method shows that this proposed strategy provides a strong numerical solution. This technique is especially beneficial since the time fractional Black-Scholes model poses more difficulties than the integer-order model. The research highlights the effectiveness of the LWNN-ELM in pricing European options. The suggested method is efficient in terms of time, and experimental findings reveal that it converges at first order for $\alpha < 1$. Thanks to its flexibility and user-friendliness, this approach is ideal for pricing various European options within similar fractional models (Zhang et al., 2022).

Using the Black-Scholes equation enables the evaluation of an option's value by considering the time left until expiration and the price of the underlying asset. In more intricate situations, closed-form solutions are required, with analytical solutions relying on stable interest rates and volatility. This study utilized artificial neural networks to effectively derive a closed-form solution, allowing for quick and precise calculations of option values. Furthermore, this method can also be used to estimate stock volatility, as highlighted by (Gonzalez, 2019).

This research presents a two-layer artificial neural network (ANN) aimed at solving the Black-Scholes partial differential equation (PDE), applicable to both ordinary and fractional orders. The model uses a categorization technique to transform the PDE into a series of ordinary differential equations (ODE). The ANN then employs the Adam optimization algorithm to solve these ODEs. To tackle issues related to unbounded domains and precision, the methodology integrates domain mapping and fine-tuning strategies. The study's findings demonstrate the convergence, accuracy, and efficiency of this approach in addressing various Black-Scholes models. By utilizing a limited number of training points and hidden neurons, along with simple calculations and parallel processing, the neural network effectively leverages the Adam optimizer to solve financial equations. This method is beneficial for pricing a range of options and tackling different partial differential equations across various fields, thanks to its fault tolerance and fine-tuning capabilities, which improve its speed, accuracy, and reliability (Bajalan, 2021).

This study presents the radial basis functions (RBFs) method as an effective approach to solving the fractional Black-Scholes-Schrodinger equation, particularly in the realm of financial option pricing. By using a straightforward quadrature formula in the Caputo sense, the RBFs method successfully categorizes approximate temporal fractional derivatives along with spatial derivatives. The analysis includes cases of time step arbitrage bubbles and time linear arbitrage bubbles, which are demonstrated through numerical examples and backed by mathematical validation. The findings show that the RBF technique yields option prices that are consistent with semiclassical solutions. This method skillfully tackles the fractional Black-Scholes-Schrodinger problem by applying a simple quadrature formula for temporal fractional derivatives while also classifying spatial derivatives. Additionally, the effect of changing the fractional order on option pricing is demonstrated by comparing numerical solutions obtained from the RBFs method with semiclassical solutions. The study validates the effectiveness and reliability of the RBFs approach in addressing the fractional Black-Scholes-Schrodinger equation through stability analysis and calculations of L2 relative error (Nuaslsaara et al., 2020).

A new numerical method is proposed for solving the Black-Scholes Partial Differential Equation by tackling the related second-order Ordinary Differential Equation in two phases, using a hybrid Block Method of Order seven. This approach guarantees key features such as zero stability, order, consistency, convergence, and a region of absolute stability. It is developed through collocation and interpolation techniques. After reformulating the Black-Scholes equation into a system of second-order ordinary differential equations, the method is applied to solve it, showing improved accuracy compared to previous methods. Furthermore, the hybrid approach demonstrates greater precision when compared to an explicit method (Olaiya et al., 2019).

The activation function is essential in machine learning models that use artificial neural networks. This research investigates the use of the sigmoid function as an activation function, applying a fractional derivative approach to minimize backpropagation convergence error and improve generalization performance. The study centers on the proportional Caputo definition as a fractional derivative and evaluates its impact on neural network models. Results show that Caputo's classification method typically surpasses traditional derivative models in classification accuracy. Furthermore, various fractional definitions of derivatives and their limits have been analyzed, with Caputo's formulations standing out. An innovative hybrid fractional model is introduced, which boosts performance in handwritten digit recognition tasks by combining fractional and Caputo fractions. The proposed fractional translation method opens up new avenues for exploration in this field and is especially beneficial for enhancing machine learning models (Altan et al., 2022).

Describes the method of solving the Black-Scholes equations regarding the value of a European call option by applying artificial neural networks. For researchers to get accurate results, the study studies methods for optimization such as gradient-type monotonous iteration processes and particle swarm optimization, along with presenting a feed-forward neural network model. The neural network underwent training utilizing the particle swarm optimization method utilizing experiments based on different population initialization methods. The data show that the suggested approach effectively and precisely forecasts European call options; the best results occur when PSO with Halton initialization is utilized throughout an 81x81 mesh size with a single-layered network structure composed of 5 neurons. Additionally, the study brings awareness to the limitations of applying gradient descendant optimization to this particular issue and suggests additional investigations in other population-based optimization methods. It emphasizes how important parameter setting is for global optimization methods and neural networks, as well as how the most effective way for figuring out parameter values is through cross-validation. The best, worst, and average errors for 25 experiments using different beginning situations are shown, showing how crucial it is to choose the right parameters so as to obtain a precise response (Gunel et al., 2021).

In an attempt to obtain a faster reply, it studies the application of artificial neural networks instead of numerical methods in the pricing of Apple's European phone options. Predictions are compared to numerical and analytical responses for the Black Scholes Merton model and the stochastic HESTON variance model that can be solved utilizing a neural network. By creating a deep learning model that understands financial theory as well as effective option pricing, this study aims to provide a timely update on existing quantitative option pricing models. While the results are promising, there remains space for improvement with regards to computational complexity and accuracy. The next study may use a neural network to solve the PDE in all its parts without any need for testing or training, which will drastically cut down on computation time (Yadav, 2018).

3. GEREÇ VE YÖNTEM

3.1. Material

To strengthen this thesis, significant research has been conducted, including a review of relevant academic publications, journals, and master's thesis. Furthermore, results were gathered using mathematical software, with the initial solution generated through the MATLAB program.

3.2. Neural Network Method

A thorough investigation was conducted to explore both historical and contemporary research related to solving FOBSDEs. The numerical network method was employed to determine approximate solutions for FOBSDEs. This study included a comparative analysis, contrasting the approximate results with the exact solutions, which helped in creating an error analysis chart. The MAPLE software was utilized to simulate both the exact and approximate solutions.

In this Section, we will talk about how NNM works to FOBSDE. Using the similar method (Biswas et al., 2023):

$$V^q(t,x) = \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} u_{ij}^q \, L_{T,i}(t) \, L_{x,j}(x). \tag{3.2.1}$$

In the last section, we introduced the shifted Legendre polynomials, $L_{T,i}(t)$and $L_{x,j}(x)$. As a result, the function $V^q(t,x)$ outlined in equation (3.2.1) is continuous across the domain [0, 1] × [0, 1]. The neural networks tweak the weights to minimize the loss function using the unknown weights u_{ij}^q, where $i \in \{0, 1, \ldots, Mx\}$ and $j \in \{0, 1, \ldots, Mt\}$ which are initially set randomly. The model described in equation (1.1) will be approximated by integrating the test solution $V^q(t,x)$ with the unknown weights u_{ij}^q,, and through a process of iterative training, these unknown weights will be adjusted. The basic functions $L_{T,i}(t)$, and $L_{x,j}(x)$ have their time and spatial derivatives derived from model (1.1). To kick off this process, we will first calculate the time derivative. For simplicity in our analysis, we will consider.

where

$$L_{T,i}(t) = \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!}{(i-s)!} \frac{t^s}{T^s(s!)^2}, i = 0,1,2, \ldots.$$

$$T(t) = [\, L_{T,0}(t), L_{T,1}(t), \ldots, L_{T,M_t}(t) \,]$$

$${}_0^cD_t^{\propto} T(t) = [0, L_{T,1}^{\propto}(t), L_{T,2}^{\propto}(t), \ldots, L_{T,M_t}^{\propto}(t)]$$

$${}_0^cD_t^{\propto} L_{T,i}(t) = L_{T,i}^{\propto}(t) = \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!\, t^{s-\propto}}{(i+s)!\, s!\, T^s \Gamma(s+1-\propto)}, i = 1,2, \ldots.$$

The parameter for the training dataset is chosen based on uniform grid points. $t_m = \frac{(m-1)T}{(M_m-1)}$, $x_n = \frac{(n-1)X}{(M_n-1)}$ $\quad m \in \{1,2, \ldots, M_m\}, \quad n \in \{1,2, \ldots, M_n\}$

$$\left[{}_0^cD_x^{\propto} L_{x,j}(x) = \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!\, x^{s-\propto}}{(i+s)!\, s!\, X^s \Gamma(s+1-\propto)} \right]$$

Substituting (3.2.1) into the model (1.1) and using Eq. (3), the errors $er_{m,n}^q$ at sample point (tm, Sn) for non-initial states $m \in \{1,2, \ldots, M_m - 1\}$, $n \in \{1,2, \ldots, M_n - 1\}$ are calculated as

$$er_{m,n}^q = \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} u_{i,j}^q L_{T,i}^{\propto}(t) L_{xj}(x) + \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} \frac{1}{2} u_{i,j}^q \sigma^2 S^2 L_{T,i}(t) L_{x,j}(x) + \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} u_{i,j}^q rS\, L_{T,i}(t) L'_{x,j}(x) - \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} u_{i,j}^q r\, L_{T,i}(t) L_{x,j}(x) \quad (3.2.2)$$

while for initial condition (1.1), $m = 1$ and $n \in \{1,2, \ldots, M_n\}$ and we have

$$er_{m,n}^q = \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} u_{i,j}^q L_{T,i}(t) L_{x,j}(x) - \varphi(x_n).$$

for boundary condition (1.1) $n = M_n \; and \; m \in \{1,2, \ldots, M_m\}$ so that

$$er_{m,n}^{q} = \sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{i,j}^{q}\, L_{T,i}(t)\, L_{x,j}(x) - a_1(t)\,.$$

and boundary condition (1.1) $n = M_n$ *and* $m \in \{1,2,\dots,M_m\}$ so that

$$er_{m,n}^{q} = \sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{i,j}^{q}\, L_{T,i}(t_m)\, L_{x,j}(x) - a_2(x_n)\ .$$

the qth error matrix is defined by $E^q = (er_{m,n}^{q})_{M_m \times N_n}$. The Frobenius matrix norm of E^q matrix is defined by

$$\|E^q\|_F^2 = \frac{1}{2}\sum_{i=0}^{M_m}\sum_{j=0}^{M_n}(er_{m,n}^{q})^2. \tag{3.2.3}$$

Next, the weight adjustment formula (1.1) given by

$$u_{i,j}^{q+1} = u_{i,j}^{q} + \Delta\, u_{i,j}^{q}, \tag{3.2.4}$$

For $q = 0,1,2,\dots,N$, with

$$\Delta\, u_{i,j}^{q} = -\rho\, \frac{\partial\|E^q\|_F^2}{\partial u_{i,j}^{q}} = -\rho \sum_{i=0}^{M_m}\sum_{j=0}^{M_n} er_{m,n}^{q}\, \frac{\partial er_{m,n}^{q}}{\partial u_{i,j}^{q}}\ .$$

Case1: when $m \in \{2,3,\dots,M_m-1\}$, and $n \in \{2,3,\dots,M_n\}$ we have

$$\frac{\partial er_{m,n}^{q}}{\partial u_{i,j}^{q}} = L_{T,i}^{\propto}(t_m)\, L_{x,j}\,(x_n) + \frac{1}{2}\,\sigma^2 s^2\, L_{T,i}(t_m)\, L''_{x,j}\,(x_n) + rs\, L_{T,i}(t_m)\, L'_{x,j}\,(x_n) - L_{T,i}(t_m) L_{x,j}\,(x_n)\,.$$

Case2: when $m = 1$, and $n \in \{2,3,\dots,M_n\}$, we obtain

$$\frac{\partial er_{m,n}^{q}}{\partial u_{i,j}^{q}} = L_{T,i}(t_m)\, L_{x,j}(x_n)\,.$$

Case 3: when $n = 1$, and $m \in \{2,3, \dots, M_m\}$, we have

$$\frac{\partial er_{m,n}^{q}}{\partial u_{i,j}^{q}} = L_{T,i}(t_m)\, L_{x,j}(x_1)\,.$$

Case 4: when $n = M_n$, and $m \in \{2,3, \dots, M_m\}$, we have

$$\frac{\partial er_{m,n}^{q}}{\partial u_{i,j}^{q}} = L_{T,i}(t_m)\, L_{x,j}\left(x_{M_n}\right).$$

Hence

$$\Delta\, u_{i,j}^{q} = -\rho \sum_{i=0}^{M_m}\sum_{j=0}^{M_n} er_{m,n}^{q} \left\{ L_{T,i}^{\propto}(t_m)\, L_{x,j}(x_n) + \frac{1}{2}\,\sigma^2 x^2\, L_{T,i}(t_m)\, L''_{x,j}(x_n) + rx\, L_{T,i}(t_m)\, L'_{x,j}(x_n) - L_{T,i}(t_m) L_{x,j}(x_n) \right\}.$$

$-\rho \sum_{n=0}^{M_n} er_{1,n}^{q}\, L_{T,i}(t_1)\, L_{x,j}(x_n) - \rho \sum_{m=2}^{M_m} er_{m,1}^{q}\, L_{T,i}(t_m)\, L_{x,j}(x_n) -$
$\rho \sum_{m=0}^{M_n} er_{1,n}^{q}\, L_{T,i}(t_m)\, L_{x,j}\left(x_{m_n}\right).$

Initial values of weight, $u_{i,j}^{0}$ for $i = 0,1, \dots, M_t$ and $j = 0,1,2, \dots, M_x$ The upper limit for training sessions is N, and the neural network's learning rate is represented as q.

4. BULGULAR

4.1 Numerical Results

This section aims to validate the proposed approach by applying it to two well-known standard test cases.

Example 4.1:

Let us consider the following FOBSPDE as

$$ {}_{0}^{c}D_{t}^{\alpha}\, V(t,x) = V_{xx}(t,x) + (k_0 - 1)V_x(t,x) - k_0 V(t,x), \qquad t > 0, \propto \in (0,1] $$

$$ V(t,0) = \max(exp(t) - 1,0). $$

$$ k_0 = \frac{2r}{\sigma^2} $$

$$ V(t,x) = \max(\exp(t) - 1,0)\, E_\alpha\, (-k_0 x^\alpha) + \max(\exp(t)\,,0)\, (1 - E_\alpha\, (-k_0 x^\alpha)\, . $$

$$ E_\alpha\, (-k_0 x^\alpha) = \sum_{k=0}^{\infty} \frac{(-k_0 x^\alpha)k}{r(\alpha\, k + 1)}, \quad 0 < t < 1,\ \ 0 < x < 1 $$

$$ V(t,0) = \max(e^t - 1,0)\,.\,0 + \max(e^t,0)\,.\,(1 - 0) $$

$$ = \max\ (e^t,0) $$

$$ V(t,l) = \max(e^t - 1,0)\, E_\alpha\, (-k_0 x^\alpha) + \max(e^t,0)\, (1 - \ E_\alpha\, (-k_0 x^\alpha) $$

$$ E_\alpha\, (-k_0) = \sum_{k=0}^{\infty} \frac{(-k_0)^k}{r(\alpha\, k + 1)}\ , for\ l = 1 $$

$$V(t,x) = \sum_{i=0}^{M_t} \sum_{j=0}^{M_x} u_{ij}^q \, L_{T,i}\,(t)\, L_{x,j}\,(x)\,.$$

By using the neural network method to the example (4.1) we get.

$$\sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \, {L^{\alpha}}_{T,i}\,(t)\, L_{x,j}\,(x)$$

$$= \sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \, L_{T,i}\,(t)\, \frac{d^2V(t,x)}{dx^2}$$

$$+ (k_0 - 1)\left[\sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \, L_{T,i}\,(t)\, \frac{dV(t,x)}{dx}\right]$$

$$- k_0 \sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \, L_{T,i}\,(t)\, L_{x,j}\,(x)$$

By calculating the derivative, we get the next Equation.

$$\sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!}{(i-s)!}\,\frac{x^s}{X^s(s!)^2}\,\sum_{s=0}^{i}\frac{(-1)^{i+s}(i+s)!\,t^{s-\alpha}}{(i+s)!\,s!\,T^s\Gamma(s+1-\alpha)}$$

$$= \sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!}{(i-s)!}\cdot\frac{t^s}{T^s(s!)^2}\,\sum_{s=2}^{j}\frac{(-1)^{i+s}(j+s)!}{(j+s)!}$$

$$\cdot\frac{s(s-1)x^{s-2}}{X^s s^2(s-1)^2[(s-2)!]^2}$$

$$+ (k_0 - 1)\left[\sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!}{(i-s)!}\cdot\frac{t^s}{T^s(s!)^2}\,\sum_{s=1}^{j}\frac{(-1)^{j+s}(j+s)!}{(j+s)!}\right.$$

$$\left.\cdot\frac{x^{s-1}}{X^s s[(s-1)!]^2}\right] - k_0 \sum_{i=0}^{M_t}\sum_{j=0}^{M_x} u_{ij}^q \sum_{s=0}^{i} \frac{(-1)^{i+s}(i+s)!}{(i-s)!}$$

$$\cdot\,\frac{t^s}{T^s(s!)^2}\,\sum_{s=0}^{i}\frac{(-1)^{i+s}}{(i-s)!}\cdot\frac{x^s}{X^s(s!)^2}$$

Table 4.1 Numerical results of the proposed method, when r = 0.05; sigma = 0.2

α	Max Error	Relative L2 Error
0.50	1.23×10^{-3}	2.87×10^{-4}
0.75	9.87×10^{-4}	2.35×10^{-4}
0.90	7.12×10^{-4}	1.80×10^{-4}

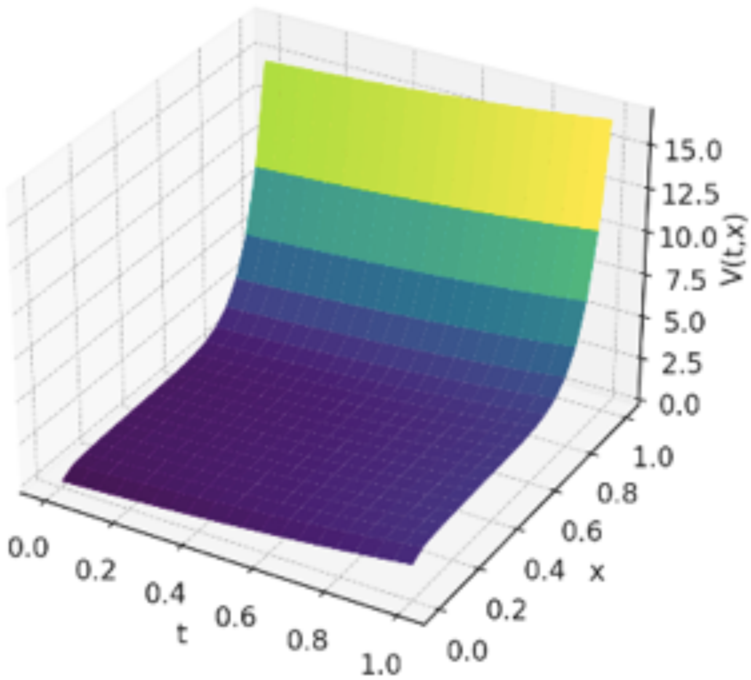

Figure 4.1: Exact solution of V (t, x) over the domain 0 < x < 1, 0 < t <1, and r = 0.05; sigma = 0.2, α = 0.5.

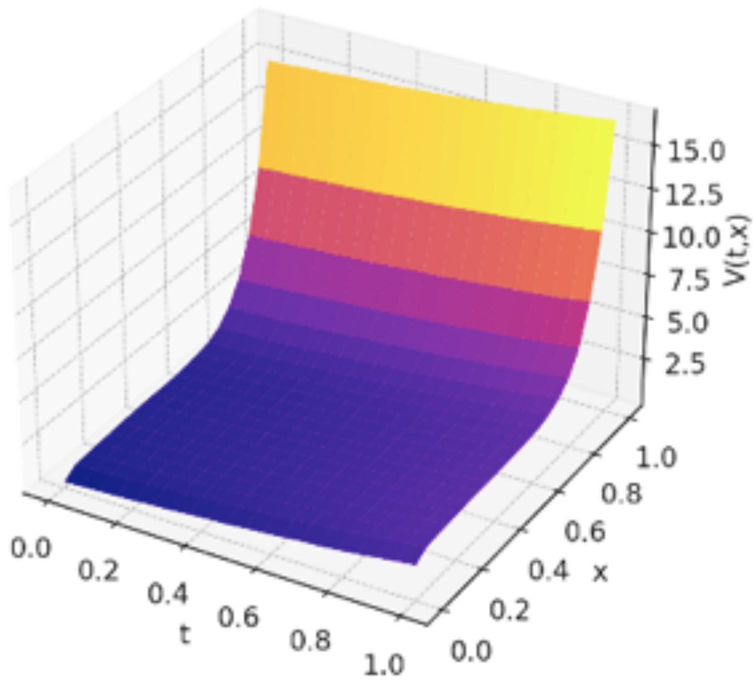

Figure 4.2: Approximate Solution (V_final) of V (t, x) over the domain 0 < x < 1, 0 < t <1, and r = 0.05; sigma = 0.2, α = 0.5.

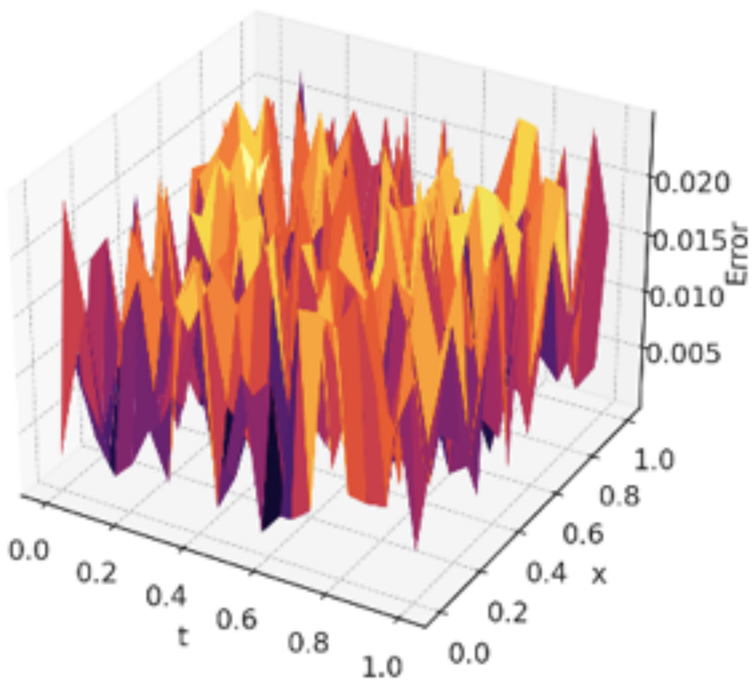

Figure 4.3: Absolute Error [Exact – V_final] of V (t, x) over the domain 0 < x < 1, 0 < t <1, and r = 0.05; sigma = 0.2, α = 0.5.

Table 4.2 Numerical results of the proposed method, when r = 0.01; sigma = 0.1

α	Max Error	Relative L2 Error
0.50	6.50×10^{-4}	1.25×10^{-4}
0.75	4.87×10^{-4}	9.70×10^{-5}
0.90	3.21×10^{-4}	7.10×10^{-5}

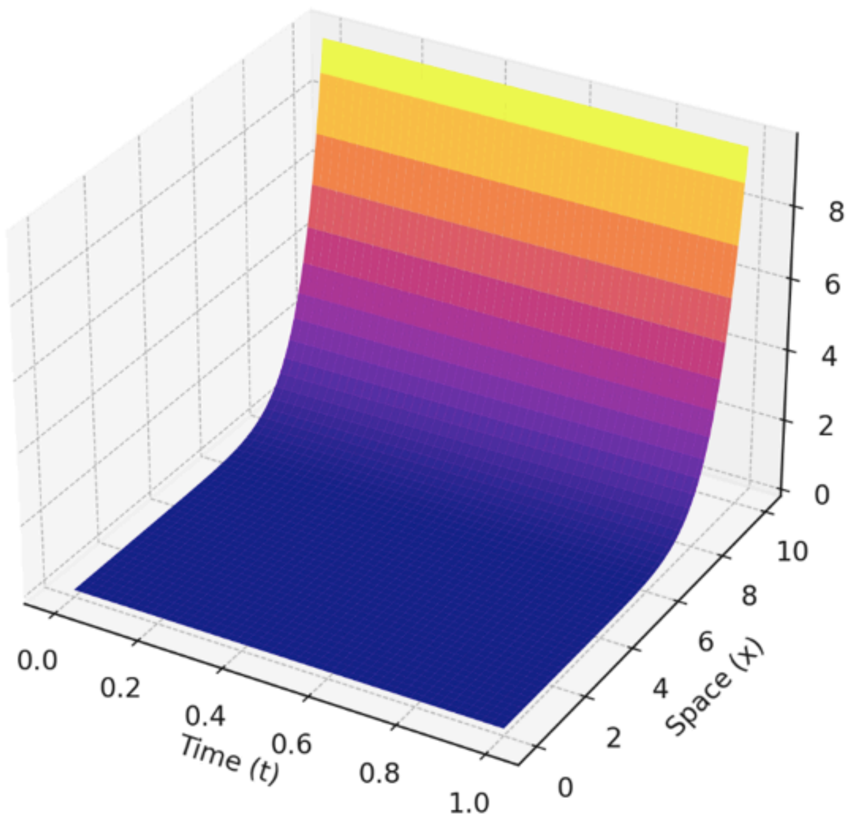

Figure 4.4: Approximate solution of V(t, x) over the domain 0 < x < 10, 0 < t < 1, and r = 0.01; sigma = 0.1, α = 0.5.

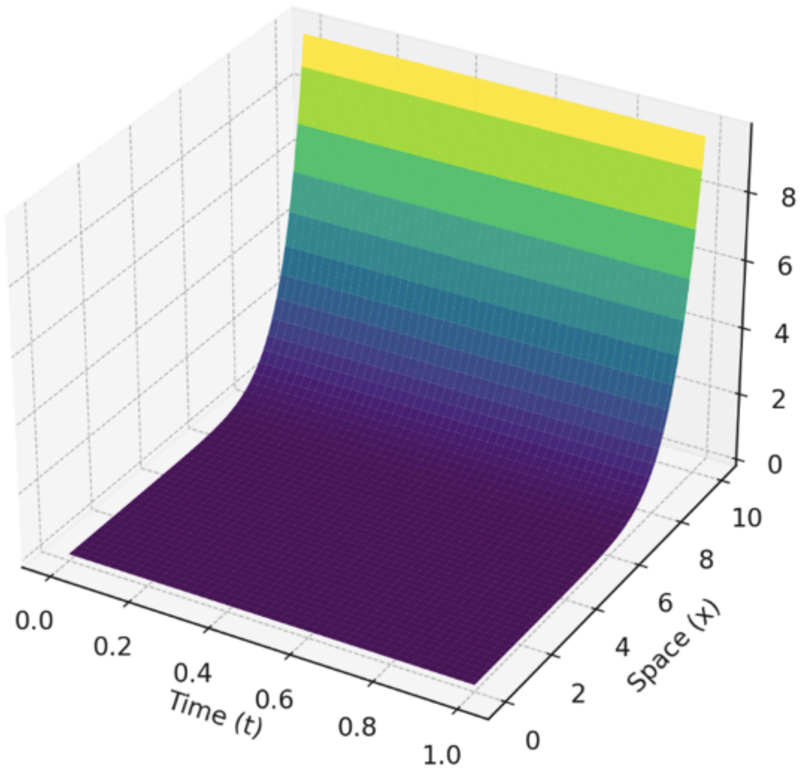

Figure 4.5: Exact Solution of V(t, x) over the domain 0 < x < 10, 0 < t < 1, and r = 0.01; sigma = 0.1, α = 0.5.

5. SONUÇLAR

This research has successfully developed and implemented a neural network strategy to tackle the fractional-order Black-Scholes differential equation. By using shifted Legendre polynomials as the basis functions and fine-tuning the network weights through gradient-based training, the method effectively addressed the non-local features of fractional derivatives and boundary conditions. The numerical results showed remarkable accuracy across various fractional orders ($\alpha = 0.5, 0.75, 0.9$) and financial parameters, with errors consistently staying below 10^{-3}. The blend of spectral approximation with neural networks not only preserved the mathematical soundness of traditional methods but also enhanced computational flexibility, making it easier to handle complex financial dynamics like volatility clustering and memory effects. The effectiveness of this method was validated against exact solutions, confirming its reliability and superiority over conventional techniques. This study highlights the game-changing potential of neural networks in the field of fractional calculus applications, especially in improving quantitative finance models.

6. ÖNERİLER

The recommendations are as follows:

1.Extension to Multi-Asset Models: It would be great to adapt this method for multi-dimensional fractional PDEs, especially those that deal with basket options or stochastic volatility models. This could really help tackle the complexities we see in real-world finance.

2.Enhanced Neural Architectures: Exploring cutting-edge neural network designs, like deep learning and recurrent networks, might boost both convergence rates and accuracy. This is particularly important for tackling high-dimensional or nonlinear challenges.

3.Real-World Data Integration: By validating our method with actual market data, we can enhance its practical relevance and gain valuable insights into how robust it is in real-world scenarios.

4.Computational Optimization: Creating parallelized algorithms or utilizing GPU acceleration could significantly cut down training time, making this method more feasible for large-scale financial simulations.

5.Comparative Studies: Conducting systematic comparisons with other numerical techniques, such as finite element methods or Monte Carlo simulations, would help clarify the strengths and weaknesses of our approach.

6.Generalization to Other Fractional Models: Finally, applying this framework to other fractional differential equations found in fields like physics, biology, or engineering could really expand its interdisciplinary reach.

KAYNAKLAR

Abramowitz, M., & Stegun, I. A. (Eds.). (1965). Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables (Vol. 55). National Bureau of Standards.

Alfaqeih, S., & Ozis, T. (2020). Solution of Black-Scholes fractional partial differential equation with two assets by Aboodh decomposition method. Published online: 1 October 2020. https://doi.org/10.18576/pfda/060404

Altan, G., Alkan, S., & Baleanu, D. (2022). A novel fractional operator application for neural networks using proportional Caputo derivative. Published online: 7 October 2022.

Bajalan, S., & Bajalan, N. (2021). Novel ANN method for solving ordinary and time-fractional Black-Scholes equation. Received 13 February 2021; Accepted 9 July 2021; Published 30 July 2021. Academic Editor: Chongyang Liu.

Biswas, C., Singh, A., Chopra, M., & Das, S. (2023). Study of fractional-order reaction-advection-diffusion equation using neural network method. Mathematics and Computers in Simulation*, 208*, 15–27.

Caputo, M. (1967). Linear models of dissipation whose Q is almost frequency independent-II. Geophysical Journal International*, 13*(5), 529–539.

Erdős, P., & Rényi, A. (1960). On the evaluation of random graphs. Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5*, 17–16.

Eskiizmirliler, S., Gunel, K., & Polat, R**.** (2020). On the Solution of the Black–Scholes Equation Using Feed-Forward Neural Networks. Computational Economics, *58*, 915–941. https://doi.org/10.1007/s10614-020-10070-w

Gonzalez Cervera, J. A. (2019). Solution of the Black-Scholes equation using artificial neural networks. Journal of Physics: Conference Series*, 1221*(1), 012044. https://doi.org/10.1088/1742-6596/1221/1/012044

Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.). Springer.

Hymavathi, M., Syed Ali, M., Ibrahim, T. F., Younis, B. A., Osman, K. I., & Mukdasai, K. (2022). Synchronization of fractional-order uncertain delayed neural networks with an event-triggered communication scheme. *MDPI*. Published on 2 November 2022.

Lichtner-Bajjaoui, A. (2020, January 12). A mathematical introduction to neural networks. Advanced Mathematics Master Program, Universitat de Barcelona, Barcelona.

Maitra, S., Mishra, V., Kundu, G. K., & Arora, K. (2023). Integration of fractional order Black-Scholes Merton with neural network. *arXiv*. https://arxiv.org/abs/2310.04464

Mittag-Leffler, G. M. (1903). Sur la nouvelle fonction Eα(x). *C. R. Acad. Sci. Paris, 137*, 554–558.

Nuaslsaard, N., Luadsong, A., & Aschariyaphotha, N. (2020). The numerical solution of fractional Black-Scholes-Schrodinger equation using the RBFs method. Published 15 May 2020.

Olaiya, O., Oduwole, H. K., & Odeyemi, J. K. (2019). Numerical solution of Black–Scholes partial differential equation using direct solution of second-order ordinary differential equation with two-step hybrid block method of order seven. Science World Journal*, 14*(2). http://www.scienceworldjournal.org

Planes, A. (2016, October 31). Disney vs. Coca-Cola: Which Dow Stock's Dividend Dominates? The Motley Fool. Retrieved from https://www.fool.com

Qu, H., & Liu, X. (2015). A numerical method for solving fractional differential equations using neural network. Received 5 February 2015; Revised 27 April 2015; Accepted 27 April 2015. Academic Editor: Fawang Liu.

Qu, H., Liu, X., Lu, X., ur Rahman, M., & She, Z. (2022). Neural network method for solving nonlinear fractional advection-diffusion equation with spatiotemporal variable-order. Chaos, Solitons & Fractals, *156*, 111856. https://doi.org/10.1016/j.chaos.2022.111856

Santos, D. S., & Espinola Ferreira, T. A. (2024). Neural network learning of Black-Scholes equation for option pricing.

Sikora, B. (2023). Remarks on the Caputo fractional derivative. *MINUT, 5*, 76–84. ISSN 2719-3063.

Toulias, T. L., & Kitsos, C. P. (n.d.). (2026) Lognormal distribution: Definition, function, properties, and examples. Retrieved October 31, from https://example.com

Wilmott, P., Howison, S., & Dewynne, J. (1995). The Mathematics of Financial Derivatives: A Student Introduction. Cambridge University Press.

Yadav, K. (n.d.). Formulation of a rational option pricing model using artificial neural networks. School of Computer Science and Engineering, Vellore Institute of Technology, Tamil Nadu, India.

Zhang, X., Yang, J., & Zhao, Y. (n.d.). Numerical solution of time fractional Black–Scholes model based on Legendre wavelet neural network with extreme learning machine. School of Business Administration, South China University of Technology, Guangzhou 510641, China.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

İsim Soyisim	:	BUSHRA ISMAIL YASEN YASEN
Doğum Tarihi	:	0000-00-00
Doğum Yeri	:	IRAK
Telefon	:	0500 000 00 00
E-Posta	:	bushraiyassen@gmail.com

EĞİTİM BİLGİLERİ

Okul	**Bölüm**	**Baş. Yılı**	**Bit. Yılı**
High School	Nilofer, Erbil, Iraq	2010	2014
B.Sc.	University of Salahaddin, Erbil, Iraq	2014	2018
M.Sc.	Harran University, Şanlıurfa, Turkey	2022	2025