Adv. Math. Fin. App., 2025, **10**(3), P. 307-323

Advances in Mathematical Finance & Applications
www.amfa.iau-arak.ac.ir
Print ISSN: 2538-5569
Online ISSN: 2645-4610
Doi:10.71716/amfa.2025.11118687

Original Research

The Using Neural Network and Finite Difference Method for Option Pricing under Black-Scholes-Vasicek Model

Mahdiyeh Mohammadi[a], Elham Dastranj[b], Abdolmajid Abdolbaghi Ataabadi[c,*], Hossein Sahebi Fard[d]

[a,b,d] *Department of Mathematics, Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Semnan, Iran*
[c]*Department of Management, Faculty of Industrial Engineering and Management, Shahrood University of Technology, Shahrood, Semnan, Iran*

ARTICLE INFO
Article history:
Received 2024-05-01
Accepted 2024-12-11

Keywords:
Option Pricing
Neural Networks
Finite Difference Method
Black-Scholes-Vasicek Model

ABSTRACT
In this paper, the European option pricing is done using neural networks in the Black-Scholes-Vasicek market. The general purpose of this research is to compare the accuracy of neural network and Black-Scholes-Vasicek models for the pricing of call options. In the sequel, the finite difference method is applied to find approximate solutions of partial differential equation related to option pricing in the considered market. In the design of the artificial neural network required for this research, the parameters of the Black-Scholes-Vasicek model have been used as network inputs, as well as 720 data from the daily price of stock options available in the Tehran Stock Exchange market (in 1400) as the network output. The approximate solutions obtained in this article, which were carried out by two methods of neural networks and finite differences on the Tehran stock exchange based on the daily price of stock options, are shown that neural networks are more accurate method comparing with finite difference. The comparison of pricing results using neural networks with real prices in the assumed market is presented and shown via diagram, as well.

1 Introduction

Option pricing is considered one of the important issues in financial markets, which has significant growth in recent years and solve this issue has absorbed traders and investors to these markets. An option is a type of contract that gives the holder the option to buy or sale of an asset for a certain period of time [7]. Nowadays derivative instruments are used in the stock market for trading, and one of the most important derivative instruments is the option, which is considered as a privilege for its holder.
A lot of models have been presented for option pricing and are available in literature. The Black-Scholes model is known as the first model in European option pricing, which was presented by Fisher Black and Mayron Scholes. In this model, the price of a stock was simulated with a geometric Brownian

* Corresponding author. Tel.: 09133215022
E-mail address: abdolbaghi@shahroodut.ac.ir

motion. In the Black-Scholes model, it is assumed that the probability distribution of the future price of underlying asset is log-normal, but the information obtained from the real market dedicates that the mentioned probability distribution has a heavier tail [3]. This model had maintained its efficiency for a long time despite its defects such as the stability of stock price volatility and the absence of transaction costs. But after five years and with the collapse of the financial markets, it lost its efficiency, and it was used as a basis for models [8]. Vasicek model is one of the stochastic interest rate models that enables mean reversion. This model was introduced by Uldrich Vasicek, which was a type of single factor short term rate and described interest rate changes using a kind of risk in the market. The separation of the stochastic interest rate from other underlying asset prices is one of the important features of Vasicek model. In this model if the financial prices increase indefinitely, the interest rate will not increase indefinitely [12].

In numerical methods for solving boundary value problems and equations with partial derivatives, finite differences are one of the most widely used methods. The basis of the finite difference method is local approximations of partial derivatives, which are obtained by expanding the Taylor series from lower orders. This method is simple in terms of definition and implementation and works on simple and uniform areas like rectangles. Since 1928, the finite difference method has been used to find approximate solutions of partial differential equations (PDE) and ordinary differential equations (ODE). Bull, Thomson and Jordan are among the researchers who worked in this field. Finite difference methods with discretization of time interval and spatial domain are considered a numerical technique for solving differential equations. At any point of the grid, we are faced with an unknown value or a function value, so it is possible that the approximation of the differential equation will eventually become a device that can be solved by a suitable algorithm [15].

Finite difference method is widely used in financial researches. In [11] numerical solution for multi-assets option pricing (rainbow option) under two dimensional Black-Scholes model are presented via finite difference method. Results available in [9] dedicates the pricing of Bitcoin options sheets are systematically overpriced by classical methods, while there is a significant improvement in price prediction using neural network models. In [4], European options with transaction cost under some Black-Scholes markets using the finite difference method, are presented. In [12] the finite difference method is applied to find numerical solutions for zero-coupon bond option pricing.

The results in [14] show using the neural network, an accurate prediction can be obtained to evaluate the result, which makes sure that it will be favorable in different sectors and earn money. One of the most important concepts in artificial intelligence are neural networks, which are widely used in various sciences, including financial sciences and markets. Parallel learning and processing is one of the most important features of artificial neural networks. The main part of neural networks is machine learning.

This paper is organized in 7 sections. In section 2 the Theoretical Fundamentals and Research Background are stated. In section 3 the considered market equipped with combination of Black-Scholes and Vasicek model is presented. Section 4 is devoted to a numerical solution of partial differential equation related to European option pricing in Black-Scholes-Vasicek market using finite difference method. In section 5, the neural networks used in the article are described. Neural network prediction and finite difference method results for desired option pricing under the Black-Scholes-Vasicek model are compared in section 6 by assuming 720 stock option prices in the Tehran stock market. The paper is concluded in section 7.

2 Theoretical Fundamentals and Research Background

Neural network science researchers, McCulloch and Pitts, conducted studies on the internal communication ability of a neuron model. The result of their work was to present a computational model based on a simple pseudo-neuronal element. At that time, other scientists such as Donald Hebb were also working on the rules of adaptation in neuronal systems. Donald Hebb proposed a learning rule for adapting connections between artificial neurons. Verbus published a backpropagation algorithm in his doctoral dissertation, and finally Rosenblatt rediscovered this technique [9]. In [1], the Black-Scholes model and neural networks were compared and the results show that the neural network is more accurate for option pricing than the Black-Scholes model. In this article, finite difference method and neural network are used in the market of Black-Scholes-Vasicek for European option pricing. The purpose was a comparison of efficiency of these two models in option pricing issue. So 120 options were priced by these two methods. In this paper, the partial differential equation related to European option pricing in the Black-Scholes-Vasicek market is approximated by the finite difference method. It has been done on a number of options in the Tehran Stock Exchange market in 2021-2022. In the sequel, a neural network was designed using Black-Scholes-Vasicek model parameters, and using it, pricing on options was presented. The results obtained from the pricing by the two mentioned methods indicates that neural networks have higher accuracy in comparison with the finite difference method. Combining artificial neural networks and two Black-Scholes and Heston models in [13], it is concluded that artificial neural networks can be considered as an efficient alternative to existing quantitative models for option pricing. The researchers' findings show that risk premium was a determining factor in explaining changes in investors' expected rate of return, and that there was a conditional relationship between the Downside Beta and expected return. Therefore, to explain the relationship between risk and return, one must pay attention to the market direction [21].Other findings research, through a regular and logical process based on the judgment method in a survey of 14 experts in the field of capital market investment and a quantitative and multivariate model of fuzzy network analysis, to assess the level of importance, ranking and refining the effective factors. Portfolio optimization was undertaken. Based on the analysis, the variables of profit volatility, return on capital, company value, market risk, stock profitability, financial structure, liquidity and survival index can be introduced as the most important factors affecting the optimization of the stock portfolio [22].

3 The Model

In the 1980s and 1990s, the Black-Scholes model played an essential and important role in the success of financial engineering because option pricing is one of the most important topics in financial economics, and the Black-Scholes model has revolutionized this pricing method. For this reason, in this section, this model is investigated, then option pricing under the Vasicek model is presented.
The Black-Scholes model is a model that includes two assets; riskless asset (B_t) and risky asset (X_t) which respectively have the following dynamics [2].

$$dB_t = rB_t dt, \qquad B(0) = 1 \tag{1}$$

$$dX_t = \mu X_t dt + \sigma_1 X_t dW_t^{(1)}, \tag{2}$$

where r is interest rate, σ_1 is volatility of risky underlying assets, $W_t^{(1)}$ is standard Wiener process and μ is drift.

Vasicek model is a financial mathematical model to describe the interest rate improvement, which is a single factor short term rate and describes the interest rate volatility using a kind of risk in the market. Instantaneous interest rate in Vasicek model is calculated as follows.

$$dr_t = a(b - r_t) + \sigma_2 dW_t^{(2)}. \tag{3}$$

The parameters of eq. (3) define as follow.

- b (Long-term mean of return process); All curves of r are close to the mean value.
- a (Speed of return to the mean return process); Assuming it is non-negative, the parameter a determines the speed at which the curves converge to b.
- σ_2 (Instantaneous volatilities); It measures the stochastic numbers are entered into the system in real time. The relationship of σ_2 with stochastic numbers is direct, as σ_2 increases with the increase of stochastic numbers.
- $W_t^{(2)}$(Wiener process); It is stochastic and independent of modeling in the market risk factor. suppose there exists a constant ρ that implies the correlation between two processes $W_t^{(1)}$ and $W_t^{(2)}$,

$$W_t^{(1)} W_t^{(2)} = \rho dt, \tag{4}$$

Using Ito's lemma, the pde of Black-Scholes-Vasicek model obtained as follow.

$$\begin{gathered}\frac{\partial V}{\partial t} + \frac{1}{2}\sigma_1^2 X^2 \frac{\partial^2 V}{\partial X^2} + \sigma_1\sigma_2\rho X \frac{\partial^2 V}{\partial X \partial r} + \frac{1}{2}\sigma_2^2 \frac{\partial^2 V}{\partial r^2} + rX\frac{\partial V}{\partial X} + a(b - r)\frac{\partial V}{\partial r} - rV \ = 0 \\ (r \in \mathbb{R}, X \in \mathbb{R}^+, t \in [0, T])\end{gathered} \tag{5}$$

In eq. (4), V is the option price equation [17].

3.1 Parameter estimation

At first, the following points have been considered in our computation; the initial value of r_t, called r_0, is always positive. a which denotes the rate of return to the mean of the underlying asset price process, σ the underlying asset price volatility process, and b the long-term mean of the underlying asset price process get positive values. Feller's condition, which is expressed as $2ab = \sigma_2^2$ [10].

Theorem 1: The MLE estimation for calculating parameters a and b in each time step is

$$\hat{a} = \frac{2}{\delta}\left(1 + \frac{\hat{P}\delta}{2}\frac{1}{n}\sum_{k=1}^{n}\frac{1}{r_{k-1}} - \frac{1}{n}\sum_{k=1}^{n}\sqrt{\frac{r_k}{r_{k-1}}}\right), \tag{6}$$

$$\hat{b} = \frac{\hat{P} + \frac{1}{4}\sigma_2^2}{\hat{a}}, \tag{7}$$

$$\hat{P} = \frac{\frac{1}{n}\sum_{k=1}^{n}\sqrt{r_{k-1}r_k} - \frac{1}{n^2}\sum_{k=1}^{n}\sqrt{\frac{r_k}{r_{k-1}}}\sum_{k=1}^{n} r_{k-1}}{\frac{\delta}{2} - \frac{\delta}{2}\frac{1}{n^2}\sum_{k=1}^{n}\frac{1}{r_{k-1}}\sum_{k=1}^{n} r_{k-1}}, \tag{8}$$

where $n = \frac{T}{\delta}$ and δ is an arbitrary positive number [10].

Proof: Conditioned on $\sqrt{r_{k-1}}$, $\sqrt{r_k}$ is subject to the following Gaussian distribution [18]:

$$\sqrt{r_k} \sim \mathrm{N}\left(m_{k-1} , \frac{\sigma^2}{4}\delta \right), \tag{9}$$

$$m_{k-1} = \sqrt{r_{k-1}} + \frac{1}{2\sqrt{r_{k-1}}}\left[ab - ar_k - ar_{k-1} - \frac{1}{4}\sigma^2 \right]\delta. \tag{10}$$

Denote the likelihood function as

$$\ell(\ominus) = \prod_{k=1}^{n} \frac{1}{\sqrt{2\pi}\sqrt{\frac{\sigma^2}{4}\delta}} \exp\left(- \frac{\left(\sqrt{r_k} - m_{k-1}\right)^2}{\frac{\sigma^2}{2}\delta} \right), \tag{11}$$

Where $\ominus = (a , ab , \sigma)$. Taking the logarithm of the above likelihood function (11), omitting the constant variables, the logarithm likelihood function can be written as

$$\ell(\ominus) = -\sum_{k=1}^{n} \frac{1}{2}\log\sigma^2 - \frac{\left(\sqrt{r_k} - m_{k-1}\right)^2}{\frac{\sigma^2}{2}\delta}. \tag{12}$$

Taking the derivative of the function $\ell(\ominus)$ with respect to the parameters, ab, a, σ^2, respectively, we can obtain

$$\begin{cases} \dfrac{\partial \ell(\ominus)}{\partial (ab)} = -\delta \displaystyle\sum_{k=1}^{n} \left(\sqrt{r_k} - m_{k-1}\right) \frac{1}{\sqrt{r_{k-1}}}, \\ \dfrac{\partial \ell(\ominus)}{\partial a} = \delta \displaystyle\sum_{k=1}^{n} \left(\sqrt{r_k} - m_{k-1}\right)\sqrt{r_{k-1}}, \\ \dfrac{\partial \ell(\ominus)}{\partial \sigma^2} = -\dfrac{n}{2\sigma^2} + \dfrac{1}{\delta}\displaystyle\sum_{k=1}^{n} \frac{2\left(\sqrt{r_k} - m_{k-1}\right)^2}{\sigma^4}. \end{cases} \tag{13}$$

To obtain the MLE of the parameter $\ominus$, the derivatives of the parameters are set as follows:

$$\begin{cases} \dfrac{\partial \ell(\ominus)}{\partial (ab)} = 0, \\ \dfrac{\partial \ell(\ominus)}{\partial a} = 0, \\ \dfrac{\partial \ell(\ominus)}{\partial \sigma^2} = 0. \end{cases} \tag{14}$$

Then the equations in (14) implies that

$$\sum_{k=1}^{n} \left(\sqrt{r_k} - m_{k-1} \right) \frac{1}{\sqrt{r_{k-1}}} = 0, \tag{15}$$

$$\sum_{k=1}^{n} \left(\sqrt{r_k} - m_{k-1} \right)\sqrt{r_{k-1}} = 0, \tag{16}$$

$$\frac{n}{2\sigma^2} - \frac{1}{\delta}\sum_{k=1}^{n} \frac{2\left(\sqrt{r_k} - m_{k-1} \right)^2}{\sigma^4} = 0. \tag{17}$$

With an ingenious technique, setting $\mathrm{P} = ab - \frac{1}{4}\sigma^2$, sorting (15) and (16), we can get

$$\frac{na\delta}{2} = n + \frac{\mathrm{P}\delta}{2}\sum_{k=1}^{n}\frac{1}{r_{k-1}} - \sum_{k=1}^{n}\sqrt{\frac{r_k}{r_{k-1}}}, \tag{18}$$

$$\frac{n\mathrm{P}\delta}{2} = \sum_{k=1}^{n}\sqrt{r_{k-1}r_k} - \sum_{k=1}^{n} r_{k-1} + \frac{a\delta}{2}\sum_{k=1}^{n} r_{k-1}. \tag{19}$$

The above equations can be solved as

$$\hat{\mathrm{P}} = \frac{\frac{1}{n}\,\sigma\,\sum_{k=1}^{n}\sqrt{r_{k-1}r_k} - \frac{1}{n^2}\sum_{k=1}^{n}\sqrt{\frac{r_k}{r_{k-1}}}\,\sum_{k=1}^{n} r_{k-1}}{\frac{\delta}{2} - \frac{\delta}{2}\frac{1}{n^2}\sum_{k=1}^{n}\frac{1}{r_{k-1}}\,\sum_{k=1}^{n} r_{k-1}}, \tag{20}$$

$$\hat{a} = \frac{2}{\delta}\left(1 + \frac{\hat{\mathrm{P}}\delta}{2}\frac{1}{n}\sum_{k=1}^{n}\frac{1}{r_{k-1}} - \frac{1}{n}\sum_{k=1}^{n}\sqrt{\frac{r_k}{r_{k-1}}}\right), \tag{21}$$

Then according to (17),

$$\hat{\sigma}^2 = \frac{4}{\delta}\frac{1}{n}\sum_{k=1}^{n}\left[\sqrt{r_k} - \sqrt{r_{k-1}} - \frac{\delta}{2\sqrt{r_{k-1}}}\left(\hat{\mathrm{P}} - a\, r_{k-1}\right)\right]^2, \tag{22}$$

thus

$$\hat{\sigma} = \sqrt{\frac{4}{\delta}\frac{1}{n}\sum_{k=1}^{n}\left[\sqrt{r_k} - \sqrt{r_{k-1}} - \frac{\delta}{2\sqrt{r_{k-1}}}\left(\hat{\mathrm{P}} - a\, r_{k-1}\right)\right]^2}. \tag{23}$$

With $\hat{\mathrm{P}}$, $\hat{a}$, $\hat{\sigma}^2$, we can get that

$$\hat{b} = \frac{\hat{\mathrm{P}} + \frac{1}{4}\hat{\sigma}^2}{\hat{a}}. \tag{24}$$

Then Eqs. $(21) - (24)$ are the estimations of a, σ^2, σ, b by MLE, respectively. Since the MLE of the parameters are derived through the Gaussian distribution of $\sqrt{r_k}$ rather than the approximation of chisquare distribution, the estimation method is called MLE [18].

To calculate the parameters in Vasicek model, first $\hat{P}$ is selected using equation (8). Then substituting $\hat{P}$ into equation (6), a is obtained. To calculate b, σ_2 is needed as well, which is obtained from the relationship of the deviation of the daily returns of the underlying asset in the entire period. Parameters a and b are used as input layer nodes of the neural network.

4 Finite Difference Method

In this section, finite difference method is used to construct a numerical simulation for pde (5). In relation (5), t denotes maturity time which decreases from T to zero. To simplify finite difference method calculations, changing the variable t as $\tau(t) = T - t$, the backward equation has been transformed to a forward equation [5]. Note that τ increases from zero to T.

$$\frac{\partial V(X,r,t)}{\partial t} = \frac{\partial V\big(X,r,\tau(t)\big)}{\partial \tau(t)}\tau'(t) = -\frac{\partial V(X,r,\tau)}{\partial \tau} \tag{25}$$

Replacing $\tau(t)$ with t, eq. (5) is rewritten as

$$\frac{\partial V}{\partial \tau} - \frac{1}{2}\sigma_1^2 X^2 \frac{\partial^2 V}{\partial X^2} - \sigma_1\sigma_2\rho X \frac{\partial^2 V}{\partial X \partial r} - \frac{1}{2}\sigma_2^2 \frac{\partial^2 V}{\partial r^2} - rX\frac{\partial V}{\partial X} - a(b-r)\frac{\partial V}{\partial r} + rV = 0\,,$$
$$(r \in \mathbb{R}, X \in \mathbb{R}^+, \tau(t) \in [0,T])\,. \tag{26}$$

The interval $[0,T]$ divided into K subintervals of length Δt. We also choose an upper bound r_{max} for r and (X_{max}) for X. Dividing $[0, r_{max}]$ to N subintervals of length Δr and $[0, X_{max}]$ to M intervals of length ΔX, a grid point is created, denote by $(n\Delta r, m\Delta x, k\Delta t)$ where $n = 0,1,\dots,N, m = 0,1,\dots,M$, and $k = 0,1,\dots,K$.

We use a forward difference approximation of the time derivative, a central difference approximation of the first order X and r derivative and a symmetric central difference approximation of the second order X and r derivative, i.e.

$$\frac{\partial V}{\partial t}(m\Delta X, n\Delta r, k\Delta t) = \frac{\upsilon_{m,n}^{k+1} - \upsilon_{m,n}^{k}}{\Delta t} + O(\Delta t), \tag{27}$$

$$\frac{\partial V}{\partial X}(m\Delta X, n\Delta r, k\Delta t) = \frac{\upsilon_{m+1,n}^{k} - \upsilon_{m-1,n}^{k}}{2\Delta X} + O((\Delta X)^2) \tag{28}$$

$$\frac{\partial V}{\partial r}(m\Delta X, n\Delta r, k\Delta t) = \frac{\upsilon_{m,n+1}^{k} - \upsilon_{m,n-1}^{k}}{2\Delta r} + O((\Delta r)^2) \tag{29}$$

$$\frac{\partial^2 V}{\partial X \partial r}(m\Delta X, n\Delta r, k\Delta t) = \frac{\upsilon_{m+1,n+1}^{k} - \upsilon_{m+1,n-1}^{k} - \upsilon_{m-1,n+1}^{k} + \upsilon_{m-1,n-1}^{k}}{4\Delta X \Delta r} + O((\Delta X)^2, (\Delta r)^2), \tag{30}$$

$$\frac{\partial^2 V}{\partial X^2}(m\Delta X, n\Delta r, k\Delta t) = \frac{\upsilon_{m+1,n}^{k} - 2\upsilon_{m,n}^{k} + \upsilon_{m-1,n}^{k}}{(\Delta X)^2} + O((\Delta X)^2), \tag{31}$$

$$\frac{\partial^2 V}{\partial r^2}(m\Delta X, n\Delta r, k\Delta t) = \frac{\upsilon_{m,n+1}^{k} - 2\upsilon_{m,n}^{k} + \upsilon_{m,n-1}^{k}}{(\Delta r)^2} + O((\Delta r)^2, \tag{32}$$

Using these approximations, equation (27) transforms to

$$\frac{\upsilon_{m,n}^{k+1} - \upsilon_{m,n}^{k}}{\Delta t} - \frac{1}{2}\sigma_1^2 m^2 (\Delta X)^2 \frac{\upsilon_{m+1,n}^{k} - 2\upsilon_{m,n}^{k} + \upsilon_{m-1,n}^{k}}{(\Delta X)^2} - \sigma_1\sigma_2\rho m\Delta X$$
$$\frac{\upsilon_{m+1,n+1}^{k} - \upsilon_{m+1,n-1}^{k} - \upsilon_{m-1,n+1}^{k} + \upsilon_{m-1,n-1}^{k}}{4\Delta X \Delta r} - \frac{1}{2}\sigma_2^2 \frac{\upsilon_{m,n+1}^{k} - 2\upsilon_{m,n}^{k} + \upsilon_{m,n-1}^{k}}{(\Delta r)^2} \tag{33}$$
$$-n\Delta r m\Delta X \frac{\upsilon_{m+1,n}^{k} - \upsilon_{m-1,n}^{k}}{2\Delta X} - a(b - n\Delta r)\frac{\upsilon_{m,n+1}^{k} - \upsilon_{m,n-1}^{k}}{2\Delta r} + n\Delta r \upsilon_{m,n}^{k} = 0\,.$$

For equation (33), the error is of the order of $O(\Delta t+(\Delta X)^2+(\Delta r)^2)$ and also for $n=1,\dots,N, m=1,\dots,M$, and $k=1,\dots,K$. From (33),

$$\begin{aligned}\upsilon_{m,n}^{k+1} &= \upsilon_{m,n}^{k}+\frac{\sigma_1^2 m^2\Delta t}{2}\left(\upsilon_{m+1,n}^{k}-2\upsilon_{m,n}^{k}+\upsilon_{m-1,n}^{k}\right)+\frac{\sigma_1\sigma_2\rho m\Delta t}{4\Delta r}\\ &\left(\upsilon_{m+1,n+1}^{k}-\upsilon_{m+1,n-1}^{k}-\upsilon_{m-1,n+1}^{k}+\upsilon_{m-1,n-1}^{k}\right)+\frac{\sigma_2^2\Delta t}{2(\Delta r)^2}\left(\upsilon_{m,n+1}^{k}-2\upsilon_{m,n}^{k}+\upsilon_{m,n-1}^{k}\right)+\\ &\frac{nm\Delta r\Delta t}{2}\left(\upsilon_{m+1,n}^{k}-\upsilon_{m-1,n}^{k}\right)+\frac{a\Delta t(b-n\Delta r)}{2\Delta r}\left(\upsilon_{m,n+1}^{k}-\upsilon_{m,n-1}^{k}\right)-n\Delta r\Delta t\,\upsilon_{m,n}^{k}\,.\end{aligned} \tag{34}$$

Let

$$\begin{aligned}&A(m)=\frac{\sigma_1^2 m^2\Delta t}{2}, \quad B(m)=\frac{\sigma_1\sigma_2\rho m\Delta t}{4\Delta r}, \quad C(m)=\frac{\sigma_2^2\Delta t}{2(\Delta r)^2},\\ &E(m,n)=\frac{nm\Delta r\Delta t}{2}, \quad D(n)=\frac{a\Delta t(b-n\Delta r)}{2\Delta r}, \quad F(n)=n\Delta r\Delta t\,.\end{aligned} \tag{35}$$

So equation (35) can be written as

$$\begin{aligned}\upsilon_{m,n}^{k+1} &= \upsilon_{m,n}^{k}+A(m)\left(\upsilon_{m+1,n}^{k}-2\upsilon_{m,n}^{k}+\upsilon_{m-1,n}^{k}\right)+B(m)(\upsilon_{m+1,n+1}^{k}-\upsilon_{m+1,n-1}^{k}-\\ &\upsilon_{m-1,n+1}^{k}+\upsilon_{m-1,n-1}^{k})+C(m)\left(\upsilon_{m,n+1}^{k}-2\upsilon_{m,n}^{k}+\upsilon_{m,n-1}^{k}\right)+E(m,n)\left(\upsilon_{m+1,n}^{k}-\upsilon_{m-1,n}^{k}\right)\\ &+D(n)\left(\upsilon_{m.n+1}^{k}-\upsilon_{m.n-1}^{k}\right)-F(n)\,\upsilon_{m.n}^{k}\,.\end{aligned} \tag{36}$$

where the value of $\upsilon_{M,N}^{k}$ ،$\upsilon_{m,N}^{k}$ ،$\upsilon_{M,n}^{k}$ ،$\upsilon_{m,0}^{k}$ ،$\upsilon_{0,n}^{k}$ ،$\upsilon_{0,0}^{k}$ ،$\upsilon_{m,n}^{0}$ for $k=0,1,2,\dots,K$, $n=0,1,2,\dots,N$ and $m=0,1,2,\dots,M$ are known from initial and boundary conditions.

5 Neural Networks

Machine learning is used as a computational technique using experience to improve the result or make accurate predictions. Because machine learning algorithms achieve success using word data. So machine learning depends on data analysis and statistics.

A neural network is a method for simulating neurons or a network of neurons in the brain, which is divided into the following three significant parts;

- Cell body
- Input section or dendrite
- Output segment or axon

Neurons are computing units that take electrical pulses as input with dendrites and direct them to the output part, named axon [9]. So, neurons communicate with each other by electrical pulses. This neural network, artificial neurons or nodes are the processing elements. The neural network consists of three types of neural layers; Input layer, hidden layer and output layer. In this model, there is a one-way flow of information from input to output.

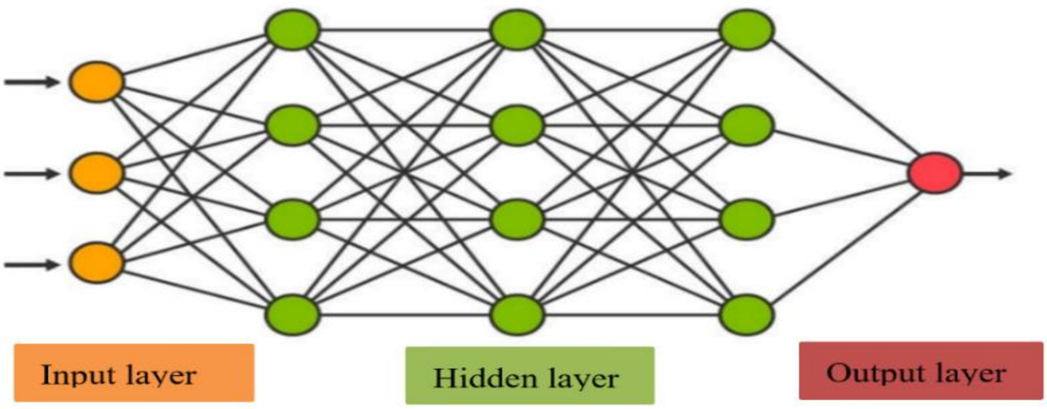

Fig. 1: Layers of Neural Network

The components of the neural network are: Input layer, hidden layer, output layer, weights, propensity and activation function [20]. These components have been shown in Figure 2 and equation (37).

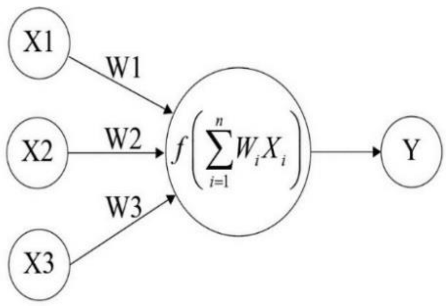

Fig. 2: Mathematical scheme of a neuron

$$y = f(x) = f(\sum x_i w_i + B), \qquad i = 1,2, \dots, n \,. \tag{37}$$

5.1 Bias

bias is an additional parameter used to bias the neural networks. The function of this parameter is that by adding to the product of variables and weights, it directs the activator function towards positive or negative numbers [6].

5.2 Back Propagation

In Back propagation, the direction of data movement is forward. In this way, the data is entered into the network from the input layer and after performing the calculations, it is placed in the next layer by means of the activator function in the hidden layer. Neural network using back propagation tries to reduce the cost function by adjusting the weights of the network and the propensity value of the function. One of the important parts of neural network design is the cost function, the purpose of which is to calculate the amount of error in the system with the real value, and how to calculate it for the neural network that has one output follows the following relationship.

$$cost(t) = y^{(t)} \log\left(h_\theta\left(x^{(t)}\right)\right) + \left(1 - y^{(t)}\right) \log(1 - h_\theta\,(x^{(t)}))\,. \tag{38}$$

In relation (38), $y^{(t)}$ is the actual value of the t-th node in the output layer and $h_\theta(x^{(t)})$ the prediction value of the neural network in the t-th node of the output layer [6].

5.3 The Gradient Descent

The gradient in the neural network determines the direction and amount of learning during network training. So, the gradient is used to update the weights of the network. In complex networks, it is possible for the calculated errors to accumulate and the gradient to grow to a great extent, which causes the weights to increase. This issue causes network instability.

The gradient descent algorithm, which is an optimization algorithm, tries to calculate the minimum value of the cost function. So, it first checks the local gradient of the cost function with respect to w and moves towards reducing the gradient. This progression continues until the value of the gradient tends to zero.In machine learning, if the cost function is drawn based on the parameter w, the resulting graph will be a convex curve with only one minimum. The slope of this minimum point becomes zero and the cost function must converge to this point. To find the point, the parameters of the cost function must be calculated for all w, which is very time-consuming and is not an optimal method. This is where the necessity of using a decreasing gradient becomes clear. In this method, instead of using all the samples, a random batch of samples is used, which is called batch gradient algorithm. If the number of categories is assumed equal to one, it is called stochastic gradient algorithm.

Let $L(w)$ be the considered cost function which depends on the weights. The update of the weights in each step is obtained using the gradient reduction algorithm from the following equation;

$$W_{n+1} = W_n - \alpha \nabla_w L(W_n) \tag{39}$$

In each step of the learning, the algorithm goes against the direction of the gradient towards the optimization of the weights. In relationship (39), α shows how much the algorithm moves against the direction of the gradient in each step, which is called the learning rate. In each iteration, the value of the cost function decreases so that $L(W_0) \geq L(W_1) \geq L(W_2) \geq \cdots$. This algorithm continues until there is no cost to reduce and the algorithm converges.

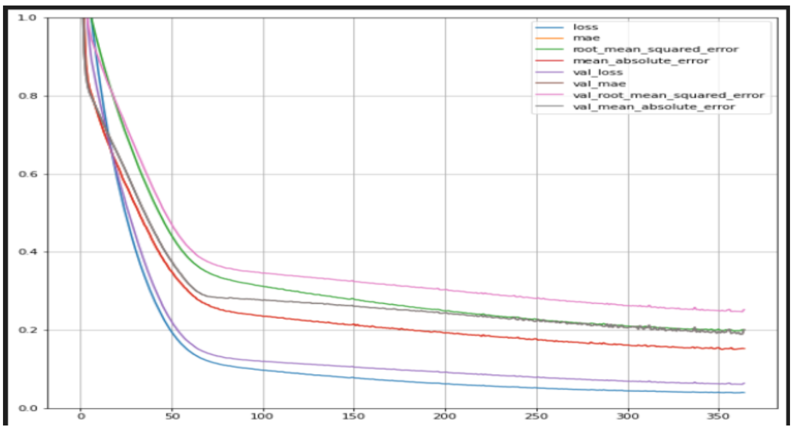

Fig. 3: Learning Rate

The diagram drawn in Figure 3 is related to neural network learning in this article. As can be seen in figure 3, the cost function converges to the minimum value almost from the 70th stage onwards, and the learning almost stops from this stage onwards.

That is, out of the 600 data that were considered for training, after 70 training cases, the learning rate will decrease drastically and the main training base will take place. From this stage onwards, small changes are made in neural network system for each case of training. so, the learning rate is very fast at the beginning of the training, and its speed decreases and becomes almost equal to zero.

5.4 Activation Function

The main part of neural networks are the activation functions that produce the expected output by being placed on the neurons. Since the neural network is used for non-linear and complex functions, its work without the activation functions becomes the same as the linear function and loses its efficiency. So, the presence of the activation function is necessary for the neural network. Many activation functions can be used in the design of neural networks, such as sigmoid, tanh, SELU (Scaled Exponential Linear Unit), ReLU (Rectified Linear Unit).

Considering that in terms of calculations, the function ReLU performs better than other functions and also does not have the problem of gradient vanishing in sigmoid and tanh functions, in this research the ReLU function is used for calculations. One of the important features of the ReLU function that is used most of the time is deep learning. This function works in such a way that it considers numbers less than zero as zero and numbers greater than zero and equal to zero as the value they are [19]. The ReLU function define as follow

$$f(x) = \max(0, x)\,. \quad (40)$$

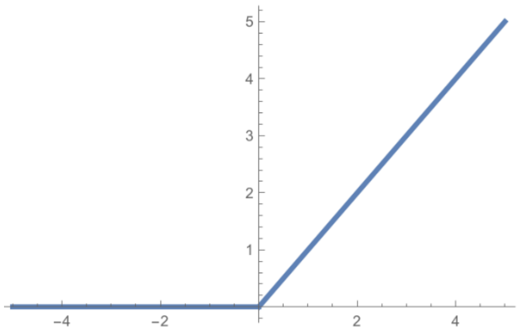

Fig. 4: ReLU Function

Although this function is similar to the linear function, it is derivable and is used in the backward propagation stage. In the ReLU function, all nodes are not activated simultaneously. In other words, when the input value of the function is greater than zero, the output is equal to the input, but if the input of the function is smaller than zero, the output is equal to zero.

In the neural network, to achieve more optimal weights and more accurate prediction, the number of nodes and hidden layers should be increased. But if their number is too high, the generalization power of the network will decrease. To find the exact number of hidden nodes and layers, there is no specific formula and it is obtained by trial and error. But to calculate the number of nodes in the hidden layer, the following relationship is often used.

$$N_h = \frac{N_s}{\alpha(N_i + N_o)}, \tag{41}$$

where N_h is the number of hidden layer neurons, N_s is the number of samples in the training data, N_i is the number of input layer neurons, N_o is the number of output layer neurons and α is scaling factor [19].

6 Approximate Solutions in Real Market

In this article, in order to compare the Black-Scholes-Vasicek model and neural networks and also to check which model has more accurate predictions, 720 stock option prices in the Tehran stock market have been studied. So 600 data were used to learn the neural network and the remaining 120 data were used to compare the predictions of these two models. All the contracts are for 6 months and the daily return of the base share has been used to calculate the expected return. The parameter values of the Black-Scholes-Vasicek model are shown in Table 1.

Table. 1: Parameter estimation

Parameter	σ_1	σ_2	ρ	a	b
Value	178.177	0.03535	0.16456	141.5135	0.0054

To calculate the volatility of the underlying asset price σ_1, first the closing price of the stock at the end of each day was considered as the price of the underlying asset, and then the standard deviation of the base share prices was calculated in the entire period.
The interest rate volatility σ_2 using the standard deviation of the daily returns of the underlying stock as a whole the course is obtained.
The correlation coefficient (ρ) between the price process of the underlying asset and the rate of return process. The long-term average of the interest rate process b and speed of return to the average of the interest rate process a are calculated from equations (8) and (6), respectively. The rate of return in this research is calculated daily i.e.

$$r = \ln\left(\frac{The\ daily\ price\ of\ the\ underlying\ asset}{The\ previous\ day's\ price}\right)$$

The assumed Variables in two considered methods are the parameters of the Black-Scholes-Vasicek model. As in the finite difference method, 9 variables are needed to find the numerical solution of the partial differential equation related to European option pricing in Black-Scholes-Vasicek model, in the neural network, the same 9 variables are used for the input layer. The neural network model built for this research is shown in Figure 5.

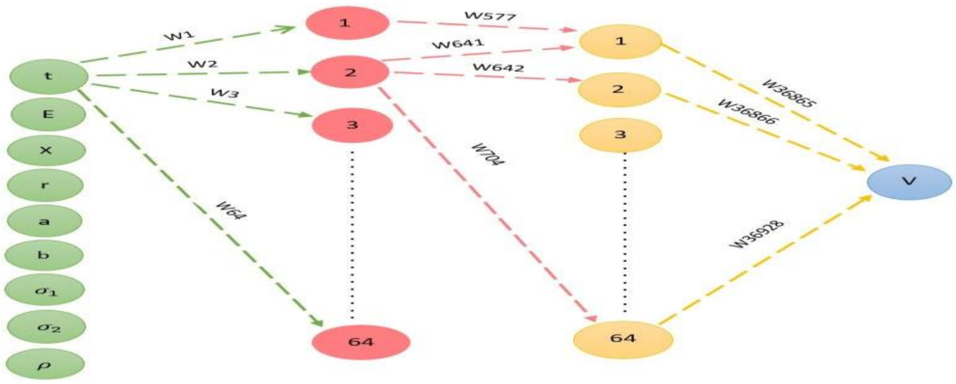

Fig. 5: The Neural Network Model of Research

The neural network designed in this research consists of an input layer, two hidden layers and an output layer. The input layer has 9 nodes, the hidden layers have 64 nodes each, and the output layer has one node. There is no exact formula to calculate the number of neurons used in the hidden layers of the neural network, but equation (26) is often used. According to the model designed in Figure (5), the sum of input and output layer neurons is equal to 10 (9+1=10) and the total number of data used in neural network training is equal to 600 data. Using equation (26), the number of neurons in hidden layers is calculated, which is equal to 60 neurons in each layer (α=1). But with a little trial and error, it is clear that the presence of 60 neurons in each hidden layer is not the most optimal state. The most optimal state is achieved with 64 neurons in each hidden layer, and numbers less and more than that have lower prediction accuracy. In this model, the ReLU function is used for the activation function and equation (23) is used for the cost function. E is the strike price of the contract and X is the final price of the underlying share (asset). The designed neural networks is initially learned using 85% of the data to predict the future. The result of this learning can be seen in the following diagram.In Figures 6(a) and 6(b), the effect of the underlying stock price and also the time elapsed from the option contract on the option price can be seen well. In these diagrams, other parameters are assumed to be constant. These two forms are different due to the difference in the constant value of E. For Figure 6(a), E=1500 and for Figure 6(b), E=2500 is considered. In both charts, as the price of the underlying stock increases, the price of the option also increases.

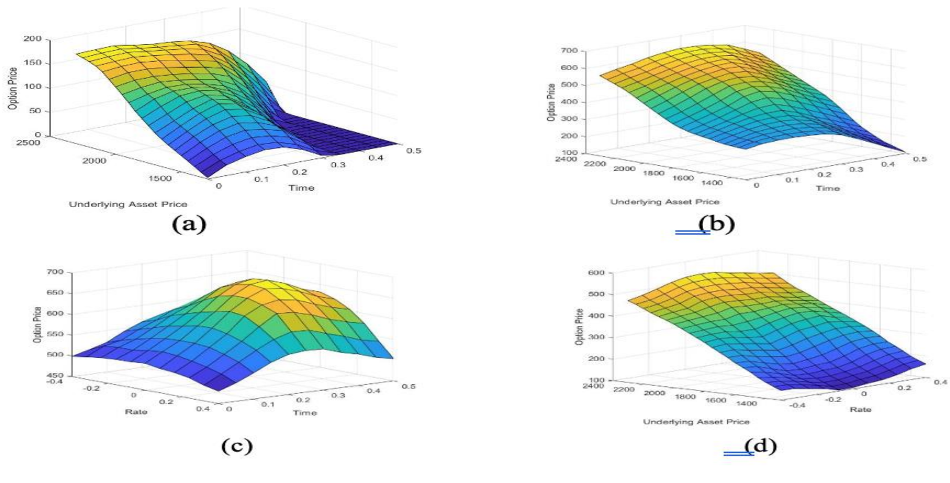

Fig. 6: The Result of this Learning by Neural Networks

Figure 6(c) shows the effects of return and underlying stock (share) price on the option price. The other parameters are assumed to be constant. In Figure 6(d), the effects of the time elapsed from the contract and the return of the underlying share on the option price can be seen. Now, using the constructed neural network model and the ReLU activator function, the price prediction of 120 existing option contracts is discussed. The result of this prediction is shown in Figure 7.

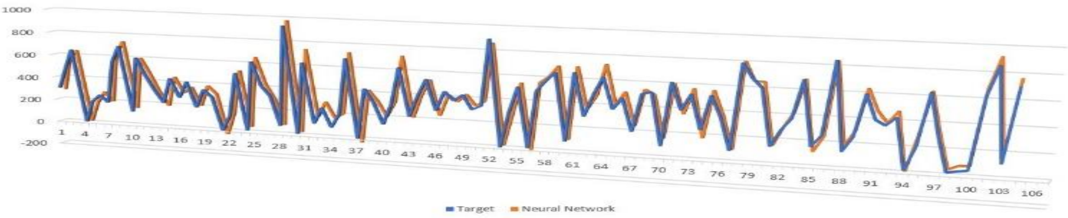

Fig. 7: The ccomparison of neural network prediction with actual market price

Again, in the Black-Scholes-Vasicek model and using the finite difference method, European option pricing is done on the same 120 option contracts, the results of which can be seen in figures 8 and 9.

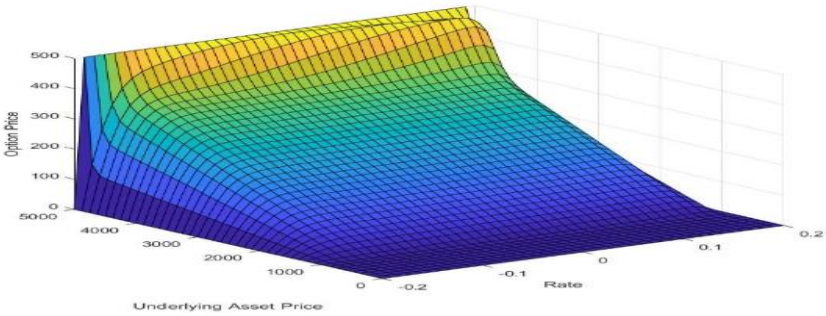

Fig. 8: Finite difference method for Black-Scholes-Vasicek model

Figure 8 shows the effects of return and underlying stock price on the option price. The rest of the parameters are considered constant. As it is known, with the increase in the underlying stock price, the option price is fixed and equal to zero at first and then increases. The stability of the option price is initially due to the fact that the price of the base share is lower than the strike price (E=1200), which makes the option worthless. But as the price of the basic share increases compared to the strike price, the value of the option also increases. The return does not have much effect on the option price as long as the difference between the basic share price and the strike price is small, but when the difference between the basic share price and the strike price increases, the value of the option increases as the return increases.

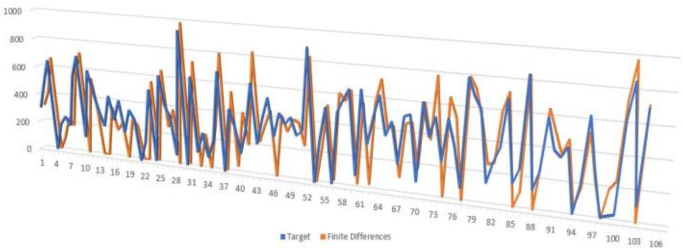

Fig. 9: The comparison of finite difference method with actual market price

Table. 2: Some comparison resolts

Real Price	Finite difference method	Absolute error	Neural networks prediction	Absolute error
304	312.3468	8.3468	280.16214	23.83786
483	409.5531	73.4469	557.47437	74.47437
649	661.233	12.233	634.96106	14.03894
10	0.3349	9.6651	4.083268	5.916732
195	85.0144	109.9856	174.13712	20.86288
255	232.8785	22.1215	272.70822	17.70822
195	184.4834	10.5166	192.9528	2.0472
557	571.2108	14.2108	623.65674	66.65674
699	714.7934	15.7934	731.64575	32.64575
124	0.0098	123.9902	145.16171	21.16171
605	539.2252	65.7748	593.52094	11.47906
455	336.732	118.268	454.798	0.202
324	3.3361	320.6639	299.83353	24.16647
220	*0*	220	181.86008	38.13992
438	348.0758	89.9242	439.57736	1.57736
274	190.8255	83.1745	305.41263	31.41263
414	246.8528	167.1472	358.87515	55.12485
196	*0*	196	198.14636	2.14636
354	282.2846	71.7154	377.56622	23.56622
292	234.5367	57.4633	309.6387	17.6387
4	0.8368	3.1632	-40.36049	44.36049
140	*0*	140	152.51457	12.51457
514	560.5295	46.5295	526.1534	12.1534
17	6.3745	10.6255	36.831345	19.831345
620	646.9938	26.9938	648.5445	28.5445
415	255.185	159.815	421.5689	6.5689
332	377.9276	45.9276	294.21518	37.78482
76	0.0015	75.9985	72.62544	3.37456
939	983.4203	44.4203	974.00464	35.00464
15	11.9038	3.0962	18.632647	3.632647
632	727.5793	95.5793	736.674	104.674
115	*0*	115	160.39772	45.39772
253	234.5367	18.4633	293.2677	40.2677
Total Mean Absolute Error		78.062		26.633

The error of each model can be calculated using the MAE (Mean Absolute Error) method

$$e_n = \text{real price} - \text{predicted price}\,, \tag{42}$$

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N} |e_n|\,. \tag{43}$$

Table 2 shows the result of the prediction of these two models.
As can be seen in Table (2), in the first column, the real price of the call option sheets available in the market is given. In the next columns, the value of the price prediction of the same call option sheets made by finite differences and neural network is placed. In front of the prediction of each method, its error rate (the difference between the prediction and the actual value of the first column) is also written. In the sequel, using the equation (42) and (43), the total error value of each model was calculated.

7 Conclusions

In this article, finite difference method and neural network are used in the market of Black-Scholes-Vasicek for European option pricing. The purpose was compare the efficiency of these two models for option pricing. So 120 options were priced by these two methods. So at the beginning, in the Black-Scholes-Vasicek market, partial differential equation related to pricing was solved using the finite difference method, and the difference between the numerical value obtained and the actual value available in the market was investigated. Then the parameters of the Black-Scholes-Vasicek model were considered as the input layer of the neural network and after learning the neural network by the activation function, the price prediction was made for those 120 data and compared with the real price in the market. As can be observed in Table 2, Figures 7 and 9, the prediction of the neural network by the function is more accurate than the finite difference method in the Black-Scholes-Vasicek market and it can be used as an important tool for option pricing in the future. The results of this research are in agreement with the research results of Julia Bennell and Charles Sutcliffe who found that option pricing with the neural network method is more accurate than the Black-Scholes model [1] and also with the results of the research of Swan Mitra who found that the combination of neural networks with the Black-Scholes model improves the accuracy of option price forecasting [16]. In another research in 2021, Kaustubh Yadav concluded that the combination of artificial neural network with Black-Scholes model can be considered as a suitable alternative for quantitative models [13]. According to the results of this research as well as related researches, it can be seen that the combination of neural network with Black-Scholes-Vasicek model solves the shortcomings in each of these models and provides more accurate prediction.

References

[1] Bennell, J., Sutcliffe, C., Black-Scholes versus neural networks in pricing FTSE 100 options, *Intelligent Systems in Accounting, Finance & Management: International Journal*, 2000; 12(4), 243-260. Doi: 10.1002/isaf.254.

[2] Björk, T., *Arbitrage theory in continuous time*. Oxford university press, 2009.

[3] Dastranj, E., Sahebi Fard, H., Abdolbaghi, A., Hejazi, S, R., Power option pricing under the unstable conditions (Evidence of power option pricing under fractional Heston model in the Iran gold market), *Physica A: Statistical Mechanics and its Applications*, 2020; 537, 122690. Doi: 10.1016/j.physa.2019.122690

[4] Dastranj, E., Sahebi Fard, H., Exact solutions and numerical simulation for Bakstein-Howison model, *Computational Methods for Differential Equations*, 2022; 10(2), 461-474. Doi:10.22034/cmde.2021.42640.1834.

[5] Dura, G., Mosneagu, A. M., Numerical approximation of Black-Scholes equation, *Annals of the Alexandru Ioan Cuza University-Mathematics*, 2010; 56(1), 39-64. Doi:10.2478/v10157-010-0004-x.
[6] François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., Pineau, J., An introduction to deep reinforcement learning, *Foundations and Trends® in Machine Learning*, 2018;11(3-4): 219-354. Doi:10.1561/2200000071.
[7] Hull, J.c, Risk management and financial institutions, *John Wiley & Sons*, 2012.
[8] MacBeth, J. D., Merville, L, J., An empirical examination of the Black-Scholes call option pricing model, *The journal of finance*, 1979; 34(5), 1173-1186. Doi:10.2307/2327242
[9] Pagnottoni, P., Neural network models for Bitcoin option pricing, *Frontiers in Artificial Intelligence*, 2019; 2: 5. Doi:10.3389/frai.2019.00005
[10] Ren, P., Parametric estimation of the Heston model under the indirect observability framework, *Doctoral dissertation*, University of Houston. 2014.
[11] Sahebi Fard, H., Easapour, H., Dastranj, E., Stochastic analysis to find exact solutions of rainbow option under 2-D Black-Scholes model, *Journal of Interdisciplinary Mathematics*, accepted.
[12] Sahebi Fard, H., Dastranj, E., Abdolbaghi Ataabadi, A., Analytical and numerical solutions for the pricing of a combination of two financial derivatives in a market under Hull-White model, *Advances in Mathematical Finance and Applications*, 2022; *7*(4):1013-1023. Doi: 10.22034/AMFA.2021.1902303.1447
[13] Kaustubh, Y., Formulation of a rational option pricing model using artificial neural networks, SoutheastCon, *IEEE*, 2021, Doi:10.20944\preprints202012.0426.v1
[14] Shahvaroughi Farahani, M., Prediction of interest rate using artificial neural network and novel meta-heuristic algorithms, *Iranian Journal of Accounting, Auditing and Finance*, 2021; *5*(1): 1-30. Doi:10.22067/ijaaf.2021.67957.1017
[15] Roul, P., Goura, V, P., A compact finite difference scheme for fractional Black-Scholes option pricing model, *Applied Numerical Mathematics*, 2021; *166*: 40-60. Doi:10.1016/j.apnum.2021.03.017
[16] Mitra, S.K., An option pricing model that combines neural network approach and Black-Scholes formula, *Global Journal of Computer Science and Technology*, 2012; **12**(4): 7-15.
[17] Chalasani, P., and Jha, S., Steven Shreve: Stochastic Calculus and Financ, *Lecture notes*, 1997;1-343.
[18] Wang, X., He, X., Bao, Y., Zhao, Y., Parameter estimates of Heston stochastic volatility model with MLE and consistent EKF algorithm, *Science China Information Sciences*, 2018; *6:*1-17.
[19] Araghi, M., Dastranj, E., Abdolbaghi Ataabadi, A., Sahebi Fard, H., Option pricing with artificial neural network in a time dependent market, *Advances in Mathematical Finance and Applications*, 2024; *2*(2): 723.
[20] Zohrabi, A., Tehrani, R., Souri, A., Sadeghi Sharif, S.J., Provide an improved factor pricing model using neural networks and the gray wolf optimization algorithm, *Advances in Mathematical Finance and Applications*, 2023; 8(1):23-45. Doi: 10.22034/AMFA.2022.1942547.1641.
[21] Nikumaram, H., Rahnamay Roodposhti, F., Zanjirdar, M., The explanation of risk and expected rate of return by using of Conditional Downside Capital Assets Pricing Model, *Financial knowledge of securities analysis*,2008;3(1):55-77

[22] Zamanpour,A., Zanjirdar, M., Davodi Nasr,M, Identify and rank the factors affecting stock portfolio optimization with fuzzy network analysis approach, *Financial Engineering And Portfolio Management*,2021;12(47): 210-236