LUT University

Pricing and Hedging options with Neural Networks: A case of FTSE 100 index options

Lappeenranta–Lahti University of Technology LUT

Master's Programme in Strategic Finance & Analytics, Master's Thesis

2022

Tuukka Pohjonen

Examiner(s): Professor Mikael Collan D.Sc (Econ.)

Post-doctoral researcher Jyrki Savolainen D.Sc (Econ.)

TIIVISTELMÄ

Tekijä	Tuukka Pohjonen
Tutkielman nimi	Pricing and Hedging options with Neural Networks: A case of FTSE 100 index options
Akateeminen yksikkö	LUT-kauppakorkeakoulu
Koulutusohjelma	Kauppatieteet, Strateginen Rahoitus
Ohjaajat	Mikael Collan, Jyrki Savolainen
Hakusanat	Neural Networks, Options, Derivatives, Deep-Learning

Teknologinen vallankumous ja datan määrän lisääntyminen maailmassa on saanut tutkijat kiinnostumaan datalähtöisten koneoppimismallien adaptoimisesta akateemisen rahoituksen piiriin, hyödyntääkseen perinteisten rahoitusmallien epätehokkuuksia. Tutkimuksen tarkoituksena on testata, voiko eteenpäin kytkettyä monikerroksista perseptroniverkkoa käyttää FTSE 100 osto-optioiden hinnoitteluun ja suojaamistarkoitukseen. Tuloksia vertaillaan Black & Scholes option hinnoittelumalliin, joka antaa optioille teoreettisen hinnan perustuen kohde-etuuden hintaan, strike tasoon, riskittömään korkotasoon, volatiliteetti-ennusteeseen sekä jäljellä olevaan maturiteettiin.

Tutkimuksen data koostuu yli 3000:sta optiosta aikavälillä 2019-2021. Lopullinen aineisto rajausten jälkeen koostuu 271 689 itsenäisestä havainnosta. Tutkimuksessa rakennettiin yhteensä kaksi erilaista monikerroksista perseptroniverkkoa: yksi koostuen Black & Scholesin perusmuuttujista sekä toinen, jossa on lisäksi GARCH-perusteinen volatiliteettiparametri. Molemmat neuroverkkosovitukset menestyivät Black & Scholes mallia paremmin optioiden hinnoittelussa myös silloin, kun aineisto jaettiin option maturiteetin ja preemion mukaan. Neuroverkkosovitus, jossa oli mukana ylimääräinen GARCH volatiliteettiparametri, ei tuonut lisää suorituskykyä optioiden hinnoitteluun. Suojaamisen testaus indikoi ylivoimaista suorituskykyä neuroverkkosovitukselle verrattaen Black & Scholes malliin varmistaen aikaisemman kirjallisuuden tutkimustulokset.

ABSTRACT

Author	Tuukka Pohjonen
Title	Pricing and Hedging options with Neural Networks: A case of FTSE 100 index options
School	LUT School of Business and Management
Degree programme	Strategic Finance & Analytics
Supervisors	Mikael Collan, Jyrki Savolainen
Keywords	Neural Networks, Options, Derivatives, Deep-Learning

The technological revolution and the rapidly increasing amount of data have spurred researchers to adapt data-based deep-learning models to academic finance for the purpose of leveraging inefficiencies in traditional analytical models. The objective of this research is to study whether Multilayer Perceptron Artificial Neural Networks (ANN) can be used to price and delta hedge FTSE 100 index call options. The acquired findings are compared to the Black & Scholes option pricing model, which estimates the theoretical price of an option based on underlying asset price and volatility, risk-free interest rate, strike price and time to maturity.

The research data contains roughly 3000 unique FTSE 100 call option quotes from 2019 to 2021, containing 271 689 datapoints after filtering. Two distinct MLP architectures were implemented: one with Black & Scholes input variables and the other with the inclusion of the GARCH volatility input parameter. Both of the tested ANN models outperformed Black & Scholes in terms of pricing performance, even when the data was partitioned based on options time-to-maturity and moneyness rate. The artificial neural network model with an additional GARCH volatility parameter added no new explanatory power to pricing. Hedging results also indicated promising results as Artificial Neural Networks were found superior compared to Black & Scholes model supporting the findings of previous research.

Contents

LIST OF ABBREVIATIONS

BSM	Black&Scholes Model
ANN	Artificial Neural Network
MLP	Multilayer Perceptron
MND	Mixture Density Network
RBF	Radial Basis Function
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
EKF	Extended Kalman Filter
IEKF	Iterative Extended Kalman Filter
VG	Variance Gamma model
GNN	Gated Neural Network
GARCH	Generalized Autoregressive Conditional Heteroskedasticity

1 Introduction

This thesis evaluates how the multilayer perceptron (MLP) type of Artificial Neural Network (ANN) can be applied to price and hedge index options. Traditionally index option's are priced with the most famous option's pricing model, namely Black & Scholes model (BSM). The BSM derives a theoretical option price from the market inputs based on complex differential equations.(Black & Scholes 1973) Artificial Neural Networks represent one type of computational machine learning technique (ML) that mimics the behavior of human brains, specifically how neuron signals interact. The form of Neural Networks used in this thesis is called feed-forward Multilayer perceptron Neural Networks, consisting of multiple computational layers in a network.

1.1 Background and Motivation

The development of neural networks can be traced back to the 1940s when McCulloch & Pitts (1943) discovered how the functioning of neurons can be expressed mathematically and how binary-valued neurons could execute computations.(Detienne et al. 2003) The universal approximation theory suggests that Neural Networks can estimate any continuous function in a closed interval based on input variables, which makes it possible to model and capture complex non-linear relationships (Ruf & Wang 2020).

In recent decades, Neural Networks have gained momentum among practitioners and academics. The computational power of modern computers and the exponentially growing amount of data have made it possible to solve real-life problems with Neural Networks and other Deep-learning methods. These problems include aerospace, computer vision, self-driving cars, and health care. In finance, deep-learning applications primarily stem from securities pricing, fraud detection, and loan risk assessments, but due to rapidly increasing data flows, more applications are being constantly developed.

Although the history of derivatives dates back to a far-reaching history, their theoretical pricing only began to gain attention among academics and practitioners in the 1970s, when Fischer Black and Myron Scholes developed the famous option pricing model (Black & Scholes 1973). BSM relies on several assumptions and simplifications, which will be introduced in subsequent chapters, including time stochastic diffusion processes for the underlying asset. The motivation behind the thesis is to research from the theoretical and practical point of view whether the neural network can capture the edge-cases of BSM and produce more accurate pricing and hedging results for FTSE 100 index options.

Malliaris & Salchenberger (1993) and Hutchinson et al. (1994) were the first researchers to successfully point out the possibility of estimating option prices via a non-parametric data-driven approach. After these papers were published, the pricing of financial derivatives was given a new direction in this research domain. In addition to the abovementioned researchers, numerous other papers have reported outstanding performance of Neural Networks in option pricing when benchmarked against traditional option pricing models Bennell & Sutcliffe (2004);Barunıkova & Barunık (2011);Gradojevic et al. (2009). However, Gradojevic et al. (2009) reported that Neural Network models do not produce constant superiority compared to BSM, and Yao et al. (2000) reported that the BSM model only constitutes better performance based on error metrics in at-the-money contracts compared to ANN's. The earliest papers in the field of study have been done with simple neural network architectures as well as scant data, which affects the performance of neural networks compared to traditional pricing models significantly. However, the extended computational power has enabled more complex neural network architectures and larger datasets to be used programmatically, leaving more gaps to be filled in this research area.

1.2 Objectives

The main objective is to study how Multilayer perceptron Neural Networks (MLP) can be used to price and hedge FTSE 100 European-styled index options during the years 2019-2021. This thesis will review different methodologies and ANN architectures used for option pricing based on existing literature and how those have fared against traditional option pricing models. The second objective of this thesis is to examine whether the best-in-practice methodology can be improved when the constant volatility assumption of BSM is relaxed by adding additional Generalized Autoregressive Conditional Heteroscedasticity (GARCH) volatility component to represent stochastic underlying asset price dynamics. Lastly, this thesis will re-evaluate the hedging performance of ANN compared to BSM. The applied ANN models are further partitioned by the option's charasteristics,time-to-maturity and moneyness rate to determine whether any generalizations of the findings can be inferred.

The research questions in this particular study are introduced as:

> *What type of Neural Network models are the most suitable for option pricing and hedging and what is their relative performance compared to Black & Scholes model according to the literature?*
>
> *Can the pricing performance of the currently applied Neural Network models for option pricing be improved with additional GARCH volatility parameter?*
>
> *What is the relative hedging and pricing performance of MLP neural network compared to Black & Scholes option pricing model?*

1.3 Limitations

This thesis is limited to covering only European-styled call options. Call options represent a contract granting the right to purchase the underlying instrument on specific maturity date, whereas a put option is an opposite alternative to call option. European-styled options can be exercised only on the settlement date, whereas American-styled options can be exercised at any point prior to expiration in the option's time grid. The restriction to use only European-styled call options is done to simplify the empirical structure and the amount of data and to have the most suitable financial instrument to incorporate the properties of the BSM model. The underlying instrument is European-styled FTSE 100 index options, one of the European continent's most actively traded options contracts. The observation period is restricted to cover the time grid from January 2019 to December 2021.

1.4 Thesis structure

Firstly, a comprehensive literature review will be covered during the first three chapters introducing the theoretical and mathematical concepts of artificial neural networks alongside option theory and the Black & Scholes model. Chapter six covers the previous academic literature on option pricing with ANNs. The standard methodology and data structure are explained in the seventh chapter of the thesis. The chaper eight presents the empirical results of the thesis, and after that, conclusions are drawn.

2 Artificial Neural Networks

The concept of Artificial Neural Networks stems from the behavior of human neurons. Scientists are still precarious about how every element in the human brain interconnects, although the ANNs, which resemble certain properties in biological neurons, have already shown the capability to perform complex non-linear computations. (Yegnanarayana 2009, 15) Indeed, the capability of neural networks cannot be glorified too much since it has been shown by (Hornik et al. 1989) and (Hornik et al. 1990) that neural networks are capable of learning approximation of any function and its derivatives (Can & Fadda 2014).

Although the development cycle of neural networks can be traced back to the 1940s when the first mathematical model of biological neurons by McCulloch & Pitts (1943) was published, the more relevant findings with regards to the boundaries of this thesis was found in 1958, when Rosenblatt (1958) invented a pillar for learning networks, specifically referred as the perceptron. Perceptron is a single-layer network model which is suitable for making predictions for binary classes. The perceptron's input signal is combined with a sum of weight and the input value, which is later processed through a threshold to obtain the final binary prediction. (Yegnanarayana 2009, 15); (Nielsen 2015, 3-4). According to Krøse (1996) the output y of the perceptron can be modeled mathematically as follows:

$$y = F(\sum_{i=1}^{2} w_i x_i + \theta) \tag{1}$$

Where $F(\cdot)$ is the activation function or threshold, and x is the input vector and w the weights or bias term.

The perceptron's capability of solving only linear classification tasks led to a breakthrough in the development of feed-forward ANNs during the 1980s when inventions such as the backpropagation algorithm were built by Rumelhart et al. (1986). During the same time the first Multilayer-Perceptron networks (MLP) were tested. MLP represents a subset of Feed-Forward ANN, among other Neural Networks such as Convolutional Neural Networks (CNN) and Radial Basis Functions (RBF). The information in MLPs is flooding into one direction without any loops through the architecture of multiple different layers. The significant difference between regular Perceptron and MLP is the number of used layers and the fact that MLP uses non-linear activation functions, which increases its capability to characterize features from non-linear data. The "learning" of the network is happening during the aforementioned procedure called backpropagation which is covered later in this chapter.

In the simplest form, the MLP contains an input layer, single or multiple hidden layers associated with many neurons, and an output layer (Abraham 2005). The input layer is the first obstacle of the signal, and it handles the network's external communication and passes the inbound data to other layers. The network's output layer is the last obstacle and produces ANN's prediction. Hidden layers can also be called "transitional" layers between input- and output layer's as they operate in the middle of the network. Each layer contains artificial neurons, inbound data from previous layers, and an activation function that manipulates the outbound flowing data to the next layer.

In MLP, neurons' transformed signals, which are modified by bias and non-linear activation function, are transferred to neurons in the following layers and repeated sequentially until the final result is produced at the end of the output layer. The output is afterward compared against a cost function that models the errors. The error is minimized during the backpropagation phase by tuning the weights in each epoch until the desired target accuracy is met. (Amilon 2003)

2.1 Backpropagation Algorithm

The backpropagation algorithm, developed by Rumelhart et al. (1986) is the fundamental part of training ANNs to learn from data. The objective of the process is to update the weights in MLP's layers so that it minimizes the error between the network's output and desired actual output. The backpropagation algorithm is divided into two phases called a Forward Pass and a Backward pass.

In the forward pass, the network's weights are first initialized by random numbers before the output vector for each layer h^i is calculated. Firstly the input values x are multiplied with initial weights W and then the output of the multiplication is calculated through activation function $f(\cdot)$. The next layer is receiving the output of $h(N)$ as an input, and this process is prolonged until final layer. This procedure can be mathematically defined as follows:

$$h^{(1)} = f(1)(W^{(1)T}x + b^{(1)}) \tag{2}$$

$$h^{(2)} = f(2)(W^{(2)T}h^{(1)} + b^{(2)}) \tag{3}$$

$$h^{(i)} = f(i)(W^{(i)T}h^{(i-1)} + b^{(i)} \tag{4}$$

$$\hat{y}x = W^{(N)T}h^{(N-1)} + b^{(B)} \tag{5}$$

where W^i is the weight matrix in i:th layer, x input vector, $f(\cdot)$ the activation function and b is the bias matrix. (Kubat & Kubat 2017, 104) ; (Murphy 2012, 565-570)

Gradient Descent is a commonly used technique in backward pass phase to improve the initial random weights in a network to achieve a state where the error between the model's output and actual output is minimized. Let's estimate the squared difference between the model's output (f) and the actual output (d):

$$\epsilon = \sum (d_i - f_i)^2 \tag{6}$$

The next step in backpropagation is to go backwards from the output layer to next layers and modifying weights by the negative magnitude of the partial derivative with respect to weight. Beginning with the output layer, the approach works backwards, updating initial weights by the amount that ϵ varies in comparison to weight (W). The gradient with respect to weight ϕ is derived by applying the chain rule for the formula below:

$$\nabla w\phi = \left(\frac{\partial \phi}{\partial w_{1i}^{j}}, \ldots, \frac{\partial \phi}{\partial w_{li}^{j}}, \ldots, \frac{\partial \phi}{\partial w_{mj-1}^{j} + 1, i} \right) \tag{7}$$

The general equation for changing the weights in a network can be expressed as:

$$W_i^j \leftarrow W_i^j + c_i^j \delta_i^j X^{j-1} \tag{8}$$

where c_i^j is the learning rate, δ_i^j sensitivity parameter calculated from the Equation 7. (Nilsson 1996, 53-60)

2.2 Activation functions

Activation functions are the parts between layers that produce the output, which is afterward used in the next layer in a network. Activation functions are crucial in terms of learning complex dimensions of data and the possible non-linearity which is presented (Sharma et al. 2017). Activation functions that are used in the empirical section of this thesis are presented next.

Rectified Linear Unit (ReLu) introduced by Hahnloser et al. (2000) is one of the most used activation function in ANN's (Ramachandran et al. 2017). The benefit of ReLu stems from its higher training performance compared to other activation functions, such as the hyperbolic tangent function (Lu et al. 2019). The basic rule of ReLu states that the estimation is always linear if the output value is positive and zero if the output value is negative. Despite the superior training performance, ReLu activation might incorporate a condition called "Dying ReLu" where ReLu activation produces zero output regardless of given input (Arnekvist et al. 2020). The equation of ReLu can be mathematically denoted as follows:

$$g(x) = max(x, 0) \tag{9}$$

A superior form of ReLu, specifically called the LeaKy ReLu (LReLu) activation function, prevents the occurrence of dying relu condition by containing a small negative slope to keep updating weights during the whole backpropagation phase (Nwankpa et al. 2018). This slight adjustment makes LReLu-based gradient optimization more robust during the backpropagation phase without making it too costly in terms of training time (Maas et al. 2013). We can observe from the Equation 10 that gradient x will be multiplied with a small α constant parameter, which is in this case 0.3 [1] if the gradient turns out to be less than zero. Otherwise, the backpropagation procedure is identical to ReLu.

$$g(x) = max(\alpha x, 0) \tag{10}$$

[1]Keras's default α parameters are used without any further adjustments in the empirical section. More information may be found on Keras' website: https:keras.io/api/layers/

Exponential Linear Unit (ELU) is a similar activation function in contrast to LReLu, except the negative values in a layer are transformed with the exponential function. ELU activation also solves the "dying ReLu" condition like LReLu, but it is also capable of producing negative output values allowing near-zero mean activation without the cost of increased training speed (Nwankpa et al. 2018). The exponential activation can be expressed as follows:

$$g(x) = \begin{cases} x & \text{if } x >= 0 \\ \alpha * (e^x - 1) & \text{if } x < 0 \end{cases} \tag{11}$$

The last activation function used in the empirical section is called the exponential function. Generally, the exponential function is referred to as any particular linear function for positive real numbers. This activation is used in the final layer to produce positive only values for option prices between the range of $[0, \infty]$ (Culkin & Das 2017). More formally the exponential can be denotes as:

$$g(x) = e^x \tag{12}$$

3 Financial Options

A non-linear derivatives contract that grants the right to buy or sell the underlying asset at a predetermined strike price is known as an option. Options are typically employed for speculative reasons or to manage the tail risks of a portfolio. Options contracts can be traded over-the-counter (OTC) or on exchanges. Put options and Call options are the two types of options. The holder of a call option has the right to purchase the underlying asset at the expiration date based on the predetermined strike price. The put option is reversed to the call option, thus giving the right to sell the underlying constraint when the contract matures. Black & Scholes (1973);

(Hull 2003, 10-14)

Options are exercised at the maturity date or any time interval during the tenure, depending on the option's exercise style. The exercising style for options can be either European- or American-styled. American-styled options can be exercised at any point in the time grid before expiration, whereas European-styled contracts can be solely exercised upon maturity date. (Hull 2003, 10-14) There are always at least two counterparties between option transactions, one taking a long position and the other taking a short position. Option writers are typically banks, and especially the OTC contracts are typically collateralized since one counterparty always has liabilities due to a netting under the master agreements. The writer's benefit in option is always reversed to another counterparty (Taleb 1997, 18).

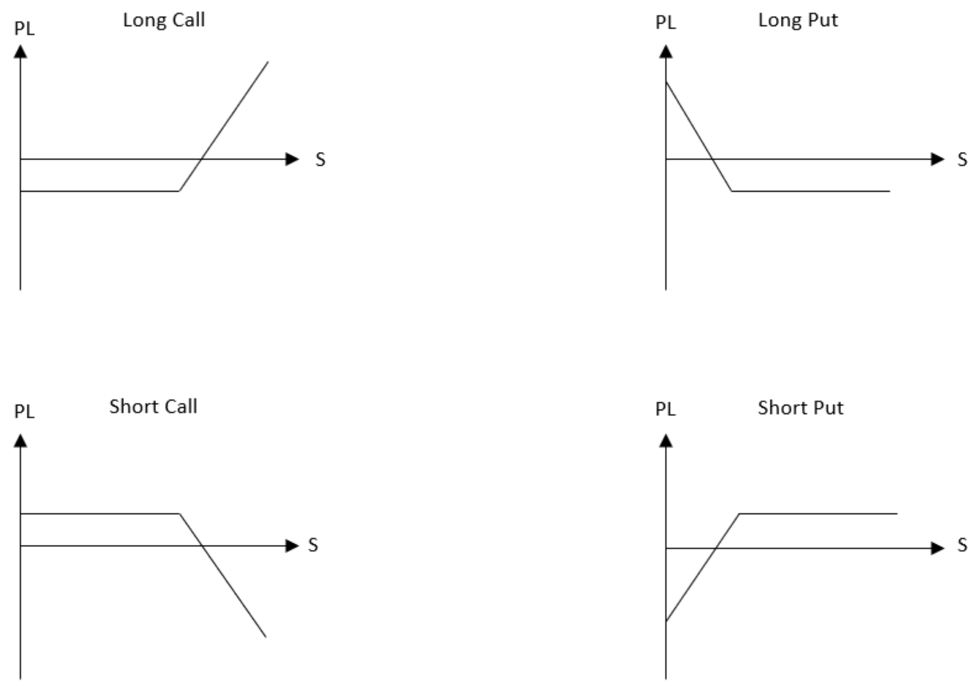

Figure 1: Option Payoff

As previously stated, the payout structure for options is non-linear and differs between call and put options. Figure 1 shows a graphical representation of a long call and a short put. The option's moneyness specifies its present position in relation to the defined strike price. (Cox & Rubinstein 1985, 9-22) When the strike price of an call option is more than the embedded underlying asset, the option is considered out-of-the-money (OTM). When the situation is reversed and the strike price is lower than the underlying asset, the call option is in-the-money (ITM). Due to the asset's time-varying fluctuations, options are seldom traded at-the-money (ATM), denoting the condition when strike equals underlying asset. The intrinsic value of options indicates how much ITM the option is currently. For a call option the intrinsic value can be formulated as:

$$max[S - X, 0] \tag{13}$$

And for put option:

$$max[X - S, 0] \tag{14}$$

Where S represents the current stock price and X represents the Strike price of an option (Hull 2003, 107-110).

3.1 Black & Scholes option pricing model

Fischer Black and Myron Scholes made an immense breakthrough in academia in the same year when Chicago Board Options Exchange started the sales of call options in 1973 as they developed one of the most famous option pricing model, namely called Black & Scholes model. The equation for Black & Scholes can be formulated as:

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 f}{\partial S^2} = rf \tag{15}$$

Where f is the function of stock price S , time-to-maturity t, risk-free rate r and volatility of the underlying asset σ. (Black & Scholes 1973)

Their novel idea was to construct a closed-form solution based on complex differential equations to model the theoretical price for European-styled options based on market inputs and several assumptions. BSM can be formulated for European-styled puts and calls as follows:

The Black & Scholes formula can be expressed as follows:

$$Call = S_0\Phi(d_1) - Ke^{-rT}\Phi(d_2) \tag{16}$$

$$Put = Ke^{-rT}\Phi(-d_2) - S_0\Phi(-d_1) \tag{17}$$

where:

$$d_1 = \frac{\log(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}} \tag{18}$$

$$d_2 = \frac{\log(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}} = d_1 - \sigma.\sqrt{T}. \tag{19}$$

Where K is the Strike Price, S stands for underlying asset price, K, r is the risk-free rate σ is the volatility input, T time-to-maturity, $d1$ and $d2$ are parameters to the phi ϕ in equations 5 and 6. ϕ and $N(x)$ stands for a cumulative probability distribution function for a standardized normal variable, which can be expressed as:

$$N(X) = (2\pi)^{-\frac{1}{2}} \int_{-\infty}^{x} exp(-\frac{z^2}{2})dz \tag{20}$$

According to Hull (2003, 92-93), the assumptions of Black & Scholes option pricing model are listed as:

1) The stock price follows Geometric Brownian motion. Volatility of the underlying asset is constant over the lifetime of an option.
2) Short selling is permitted
3) No transaction costs or taxes
4) No dividends or other distributions are paid during the lifetime of the security
5) No riskless arbitrage opportunities
6) Trading of security is continuous
7) The risk-free rate r is constant for all maturities

Despite being one of the most significant contributions in academic finance, the BSM model has received criticism among academics and practitioners due to its strict assumptions. Even the model's inventors expressed concern about the underlying assumptions being "idealizations" of market circumstances (MacKenzie 2006). Jackwerth & Rubinstein (1996) for example, revealed very substantial evidence of infrequently log-normally distributed stock prices indicating skewed stock price distributions. This conclusion contradicts BSM since the underlying asset increments are represented using a log-normal distribution. Another flaw with BSM is that it implies continuous trading and hedging at arbitrary time intervals. Haug & Taleb (2011) criticizes the dynamic hedging context since dynamic hedging is technically not achievable due largely to technical constraints. From a private person's perspective, the assumption of zero breakage commissions, structuring- or transaction fees, does not correspond to genuine market circumstances, which should be considered in terms of valuation.

3.2 Delta Hedging

Delta is one of the most used ”Greeks” of options. Greeks measure the dimension of risks involved in options. Delta presents a proportion of movement in underlying asset price with respect to option price. The mathematical representation of this partial derivative can be noted as:

$$\Delta = \frac{\partial V}{\partial S} \tag{21}$$

Where V is the option price and S the underlying asset price. Hull (2003, 285-286) In other words, option delta can be thought as a sensitivity metric that option traders are typically monitoring to cover their positions with respect to delta movements. If for example the delta of call option is measured to be 0.5, for every one dollar movement in the underlying asset changes value of call option 0.5 dollars. Hull (2003, 402-403)

Delta hedging refers to a option trading strategy where the portfolio is constructed by offsetting positive and negative deltas in a way they create the overall portfolio delta as close to zero as possible. (Hull & White 2017). The balancing is usually done with the combination of options and underlying assets. However delta hedging is quite cumbersome in practice as the portfolio keeps delta hedged for only a short time period, and it needs to be re-balanced at discrete time intervals to keep the portfolio delta neutral.

4 Performance Measurement

The pricing performance of the empirical section will be evaluated based on the error metrics used by Bennell & Sutcliffe (2004), Anders et al. (1996) and Culkin & Das (2017). Let's assume that C_{ANN} is the call price estimated with Neural Network, and C represents the call price observed by markets.

$$\text{Mean Square Error (MSE)} = \frac{1}{N}\sum_{i=1}^{N}(C_{ANN}^{(i)} - C^{(i)})^2 \tag{22}$$

$$\text{Mean Absolute Error (MAE)} = \frac{1}{N}\sum_{i=1}^{N}|C_{ANN}^{(i)} - C^{(i)}| \tag{23}$$

$$\text{Root Mean Square Error (RMSE)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(C_{ANN}^{(i)} - C^{(i)})^2} \tag{24}$$

The risk metrics in the empirical study are applied to BSM by replacing C_{ANN} with C_{BSM}.

5 Asset Volatility

Volatility, which is said to be the measure of uncertainty, represents the spread of all likely outcomes of an asset during a specific timeframe. (Poon & Granger 2003) Volatility is one of the key components in option's pricing since the BSM model inputs the volatility of an underlying asset. The so called implied volatility can be calculated backwards from the BSM formula with numerical approximation, which tells the estimate of the option's future volatility until the maturity date. Option traders are usually trading volatility based on current implied volatility levels, since it is the only unobservable BSM parameter in the market.

According to Hull (2003, 121), the realized volatility, which give the ex-post estimate can be calculated for an asset as follows:

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(u_i - \bar{u})}, \tag{25}$$

where i represents the time interval, u_i is the continuously compounded log returns and $\bar{u}$ is the mean of continuously compounded log returns.

5.1 GARCH

Another volatility estimate used in this study is the Generalized Autoregressive Autoregressive Conditional Heteroscedasticity model (GARCH). The history of GARCH stems back in the 1980's when Bollerslev (1986) invented the GARCH model. GARCH model forecasts the future volatility by combining the fitted value of the previous forecasting period with a longer term moving average component (Brooks 2002).

Let ϵ_t denote a discrete stochastic process and the information set by ψ_t. The general form of Generalized Autoregressive Conditional Heteroscedasticity model GARCH(p,q) invented by Bollerslev (1986) can model historical returns h_t in a combination of ARCH as follows:

$$\epsilon_t|\psi_t \sim N(0, h_t) \tag{26}$$

$$
\begin{aligned}
h_t &= \alpha_0 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-1}^2 + \sum_{i=1}^{p} \beta_i h_{t-1} && (27) \\
&= \alpha_0 + A(L)\epsilon_t^2 + B(L)h_t && (28)
\end{aligned}
$$

Where α is the short run persistence and β is the long run persistence.

And $p \geq 0$, $q \geq 0$, $\alpha_0 \geq 0$, $\alpha_i \geq 0, i = 1....q$, $\beta_i \geq 0, i = 1....p$

Finally, L is the lag operator:

$$
\begin{aligned}
\alpha(L) &= \alpha_1 L + \alpha_2 L^2 + ...\alpha_q L^q && (29) \\
\beta(L) &= \beta_1 L + \beta_2 L^2 + ...\beta_p L^p && (30)
\end{aligned}
$$

When $p = 0$ the process is simply ARCH(q), in other words, a linear function of past sample variances and ϵ_t is the white noice. GARCH(p,q) function adds up to the formula previous lagged conditional variances obtained by simple linear regression. (Bollerslev 1986)

The existing literature enclose large number of different methods built on top of GARCH model family, but this thesis uses GARCH(1,1) order model volatility input in one of the Neural Network models since multiple studies has shown that GARCH(1,1) is sufficient for capturing volatilities in most financial time series (Brooks 2002, 394). According to Bollerslev (1986), the mathematical definition of GARCH(1,1) can be expressed as follows:

$$
h_t = \alpha_0 \alpha_1 \epsilon_{t-1}^2 + \beta_1 h_{t-1}, \;\; \alpha_0 \geq 0, \alpha_1 \geq 0, \beta_1 \geq 0 \qquad (31)
$$

6 Literature Review

Artificial Neural Networks have been widely used as an alternative tool in derivatives pricing since the 1990s, as the sophisticated data mining techniques allow the pricing formulas to be more relaxed from theoretical assumptions that drive the existing theoretical models. In most studies, ANNs are trained to learn the price of an option as a function of BSM parameters, but some studies have also trained the ANNs to learn implied volatility surfaces and hedging ratios.

The following chapters describe the majority of the previous literature based on the research domain. The history of options pricing with ANNs will be briefly addressed, followed by a presentation of several ANN models and designs utilized for option pricing purposes. Lastly, a Table showing majority of the relevant previous studies is presented.

6.1 Early Studies

The very first study about neural networks in options pricing was published in the 1990s by Malliaris & Salchenberger (1993). The authors trained MLP neural network to estimate S&P 100 call option prices using Black & Scholes model input variables. As a result, authors reported that ANNs outperform Black & Scholes model roughly in half of the cases for in- and out-of-money options when benchmarked with MSE metrics.

Hutchinson et al. (1994) published the most referenced paper in this field of study right after Malliaris and Salchenberg. They suggested a nonparametric technique for predicting the pricing formula for options based on radial basis functions (RBFs) with four multiquadric centers and one output sigmoid, as well as MLP neural networks with a single hidden layer and four neurons. A neural network that employs radial basis functions as activation functions is known as a radial basis function network (RBF). Using synthetic data and S&P 500 futures options during the time-frame of 1987-1991, the authors investigated the pricing and hedging performance of feedforward neural networks in comparison to the Black & Scholes model.

Hutchinson et al. (1994) reduced the functional form of the ANN's pricing formula based on the BSM's homogeneity assumption, where the underlying asset's return is independent of the level of Stock price (S), such that the pricing formula required to be estimated was $f(S(t)/X, 1, T-t)$, relaxing the potential overfitting dilemma. With synthetic data, the authors obtained above 99 percent out-of-sample $R2$ values for both RBF and MLP networks. The aforementioned neural networks also out-performed the Black & Scholes model with real data, as the experiment applied to S&P 500 future options yielded above 90% $R2$ values. Arbitrage conditions were also tested with encouraging results: hedging errors for both out-of-sample datasets were less than that of the Black & Scholes model.

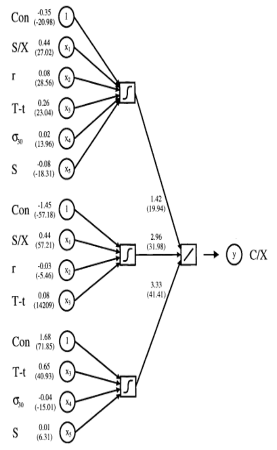

Figure 2: Anders et al. (1996) ANN architechture with four input parameters and three hidden layers

[2]

[2] S represents Stock Price, S/X is the moneyness assumed by homogenity degree one form, r is the risk-free rate, $T-t$ refers time-to-maturity and σ_{30} is the 30-day volatility input

Anders et al. (1996) priced DAX index options with MLP network using similar methodology as Hutchinson et al. (1994). Instead of using only regular BSM parameters, Anders et al. (1996) used multiple volatility estimates alongside hint form as input for pricing options. Their study showed that the MLP network with a single hidden layer can be adopted to a pricing formula with promising results.

Arguably the form of the pricing function to be estimated is one of the biggest question marks when neural networks are applied to derivatives pricing. Garcia & Gençay (2000) continued Hutchinson et al. (1994) and Anders et al. (1996) studies which incorporated the so-called homogeneity hint to pricing formula, a superior functional form, which considers ANN's first part controller by moneyness and the second one by time-to-maturity. Garcia & Gençay (2000) stated that the homogeneity hint always reduces the MSE statistics, and they found the usage of the homogeneity degree one form for ANNs is crucial for outperforming BSM in the S&P 500 index options.

Nonetheless, the BSM is the standard benchmark for pricing European-styled options, alternatives such as American-styled- and exotic-styled options necessitate slightly different valuation techniques. The likelihood of early exercise in American Styled options and contract-specific features for Exotic options must be considered when estimating the option's fair pricing. ANN's capability to estimate these types of options have also been tested in the existing literature several times. Kelly et al. (1994) taught to MLP network a price function of an American-styled option. Kelly et al. (1994) used equity put options listed for IBM, Chrysler, GM, and Merck from 1993 to 1994. The findings were benchmarked against the binomial option pricing model, which is an iterative method for pricing options. Neural Network was found superior over the binomial option pricing model. Other researchers, such as Zapart (2002); Pande & Sahu (2006); Teddy et al. (2006); Hirsa et al. (2019) have also concentrated on their research for exotic and American-styled options.

6.2 Extended ANN models for Option's pricing

Most models utilized in the study domain have been implemented with MLP and RBF type-of neural networks, but multiple extensions to the conventional models have been built according to the literature. Niranjan (1996) estimated FTSE 100 index option prices with radial basis function similar to that employed by Hutchinson et al. (1994). Unlike Hutchinson et al. (1994), Niranjan (1996) protracted the learning procedure with Extended Kalman Filter (EKF) algorithm. Kalman filter is a recursive algorithm that estimates the state of the process by reducing the mean of squared error. By training the model with Kalman filter, Niranjan (1996) reported that radial basis function with Extended Kalman filter produces fewer errors than that of basic BSM, but found the best performance in BSM associated with Extended Kalman Filter. Ormoneit (1999) used Iterative Extended Kalman Filter (IEKF) as a learning algorithm for pricing DAX index options. They also found MLP with two sigmoidal hidden units and IEKF outperformed several benchmark models and alternative learning procedures, such as online backpropagation and regular EKF based on MSE and $R2$ metrics. In the context of training neural networks with Kalman Filters, these researchers have also illustrated novel usage of Kalman Filters in deep-learning-based Option pricing: Freitas et al. (2000); Huang (2008)

Schittenkopf & Dorffner (2001) constructed a mixture density network model (MDN) to extract risk-neutral densities from FTSE 100 index option contracts. The primary concept behind MNDs is to approximate conditional density function as a mixture of Gaussians in a way where the distribution and the mixture coefficients are parameterized as a function of input variables. The authors estimated the parameters by using MLPs with a single input, T, which stands for time-to-maturity, and the model produced Gaussian parameters. As a consequence of comparing two alternative network specifications, the authors found that MNDs performed better than standard BSM and adjusted BSM. The conclusion hold for both pricing- and hedging performance.

Gradojevic et al. (2009) used a modular neural network to price S&P 500 index options from 1987 to 1993. A modular neural network is a type of artificial neural network which consists of a sequence of autonomous neural networks controlled by an intermediate. The authors decomposed the data into modules consisting of moneyness and time-to-maturity. Each of the nine modules of the network was trained independently, improving the parameter generalization and cutting out the recency effect incorporated in single ANN models. Gradojevic et al. (2009) reported superior performance over BSM with three modules selected by moneyness criterion, but the alternative model with the time-to-maturity criterion was found inferior compared to the BSM model indicating an inconsistency in pricing performance.

The final study of interest concerning different neural network designs for options pricing represents gated neural networks (GNN). GNN uses a very similar approach to modular neural networks but contains multiplicative gating connections. As Gradojevic et al. (2009) decomposed the data into different modules consisting of independent pricing models, Yang et al. (2017) proposed a slightly different modular approach, where the option grouping is automated and learned from data instead of heuristically labeled. Hence GNN dynamically adjusts the pricing functions and labeling of each group when the market changes gradually. Their model was designed so that they integrated economic constraints into their neural networks to produce more economically meaningful predictions. [3] Firstly, Yang et al. (2017) proposed a single model consisting of time-to-maturity and moneyness inputs. The performance of the single model benchmarked with MSE statistics was poor compared to BSM, Variance-Gamma (VG), and Kou Jump models, which represents different statistical option pricing models. However, they proposed also a multimodel, which trains a number of single pricing models as well as a weighting model to transition between the single models. Based on the MSE statistics, multimodel outperformed all three benchmark models.

[3]The specifications for economic axioms are not presented in this study.

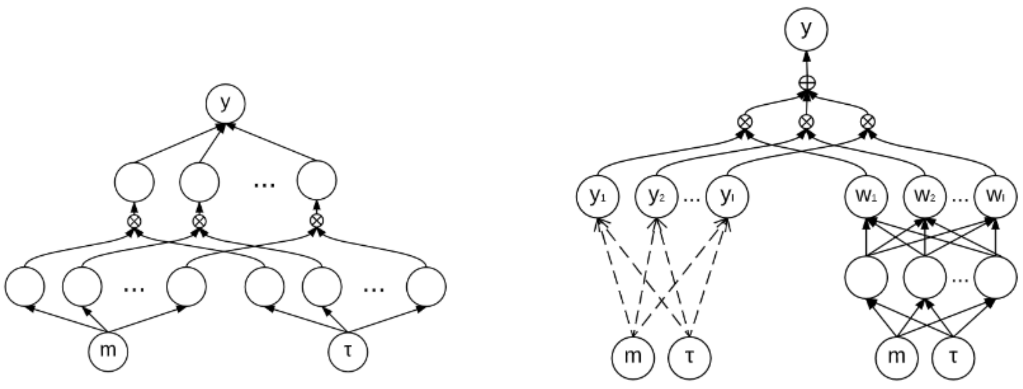

Figure 3: Yang et al. (2017) design for single model (left) and multimodel (right) [4]

6.3 Summary of the Literature Review

To shortly recap the history of options pricing with Artificial Neural Networks, the vast majority of studies have focused on the pricing performance instead of testing hedging performance or implied volatility forecasting performance. Most of the ANN models are based on either RBF or MLP networks, but more complex ANN architectures are also presented in the literature. Lately, the focus on the research area has been more on thinly traded markets and products and on more complicated neural network architectures. Ruf & Wang (2020) has comprehensively listed the majority of the studies in their literature review. Citing their literature review, Table 1 displays over 20 relevant studies regarding option pricing and hedging with ANNs compared to the BSM model.

[4]The single model (left) with input variables m and t is depicted in the picture below. The multi model (right) is composed of a series of M single modules as well as a weighted model that rotates between the single models. $\otimes$ is a multiplication gate that outputs the product of its inputs.

Table 1: Previous literature presented in table

Authors & Year	Input	Output	Data
Malliaris & Salchenberger (1993)	$S, K, t, \sigma I, r$	C	S&P 100
Hutchinson et al. (1994)	$S/K, t$	C/K	Simulated, S&P 500
Kelly et al. (1994)	$S, K, t, \sigma H$	C	Individual stocks
Niranjan (1996)	$S/K, t$	C/K	FTSE 100
Anders et al. (1996)	$S/K, S, t, \sigma H, \sigma V, r$	C/K	DAX
Ormoneit (1999)	S/K	C/K	DAX
Briegel & Tresp (2000)	S, t	C	FTSE 100
Ghaziri et al. (2000)	$S, K, t, \sigma H, r, OI$	C	S&P 500
Dugas et al. (2001)	$S/K, t$	C/K	S&P 500
Garcia & Gençay (2000)	$S/K, t$	C	S&P 500
Healy et al. (2002)	$S, K, t, \sigma I, SD, OI, V$	C	FTSE 100
Zapart (2002)	Wavelet coefficients with lags	Wavelet coefficients	Individual stocks
Bennell & Sutcliffe (2004)	$S/K, t, \sigma_I H, OI, V$	$C, C/K$	FTSE 100
Lin & Yeh (2005)	$S, K, t, \sigma H, r$	C	TAIEX
Jung et al. (2006)	$S, K, t, \sigma IH$	C	KOSPI 200
Mitra (2006)	$S, K, t, \sigma H, r$	C	NIFTY 50
Pande & Sahu (2006)	$S, K, t, \sigma PCA, r$	C	Individual stocks
Teddy et al. (2006)	$S - K, t, \sigma H$	C	GBP-USD
Gençay & Gibson (2007)	$S, K, t, \sigma G, r$	C	S&P 500
Thomaidis et al. (2007)	S, K, t	C	S&P 500

Andreou et al. (2008)	S/K,r,skewness, kurtosis, $\sigma H, \sigma CAL, \sigma V$	C	S&P 500
Gradojevic et al. (2009)	$S/K, t$	C/K	S&P 500
Barunıkova & Barunık (2011)	S, K, t	C	S&P 500
Can & Fadda (2014)	$S/K, S, t, r$	C/K	S&P 100
Montesdeoca & Niranjan (2016)	$S/K, t, \sigma_H, volume,$	C/K	FTSE 100, individual stocks
Culkin & Das (2017)	$S/K, t, \sigma, r$	C/K	Simulation

Terms: Option Price (C), Strike Price (K), Underlying Price (S), Volatility parameter (σ), risk-free rate(r), trading volume (volume), historical volatility (σ_h), GARCH-generated volatiltiy (σ_G), Calibrated volatility (σ_{CAL}), Open-Interest (OI)

7 Data and Methodology

The quantitative data used in the empirical section of this thesis is retrieved from the Thomson Refinitiv Eikon. FTSE 100 index options traded in LIFFE were chosen for the analysis as they are European-styled and do not pay dividends. The data is also restricted to call options as the appropriate version of the Black & Scholes model for put options requires separate training for ANN. The 3-month LIBOR rate is used as a proxy for the risk-free rate in BSM and ANN. The study domain has mostly focused on the S&P 500 index options, and for the last ten years, FTSE 100 index options haven't been shown in the research domain, filling the gaps for pricing UK options with neural networks and fresh data.

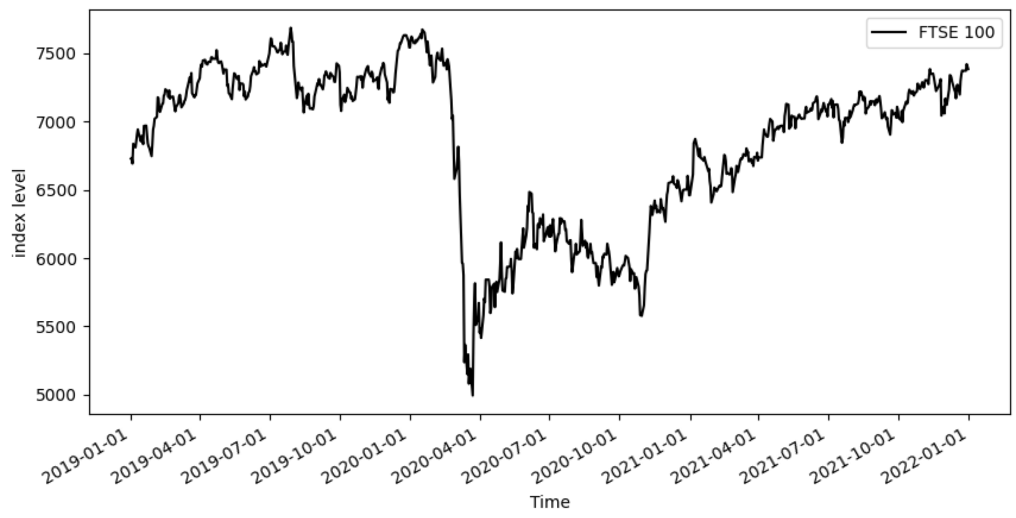

Figure 4: FTSE index during 2019-2021

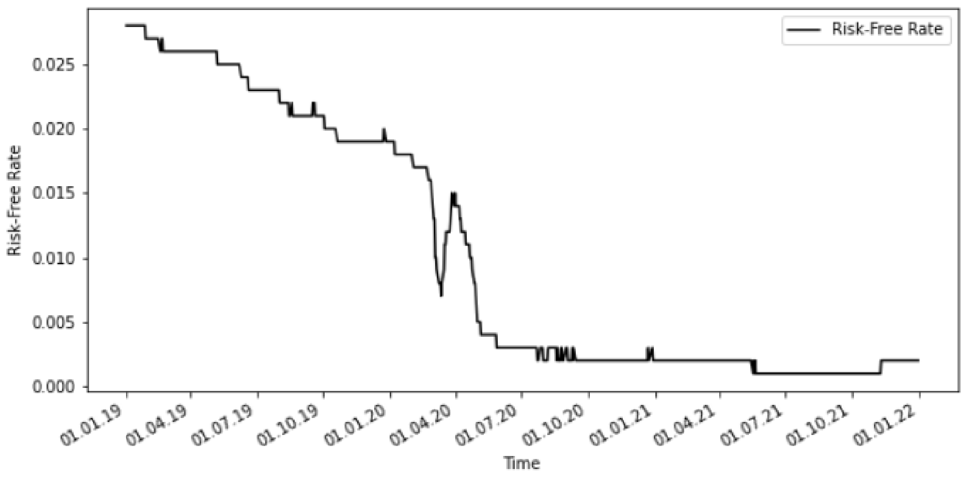

Figure 5: 3-month LIBOR rate during 2019-2021

The time frame used in this study is from January 2019 to December 2021, containing the high-volatility periods and v-shaped recovery caused by the COVID-19 pandemic. FTSE 100 index options traded in LIFFE are cash-settled index option contracts. The interval between the strike prices is determined by the expiry month's lifetime and proximity to the at-the-money strike. The contracts are listed between 25, 50, 100, and 200 index points, and the last trading day is the third Friday or the next available trading day in the delivery month. (LIFFE 2022). The whole dataset before filtration covers 3070 unique FTSE 100 index option contracts between 5000-8000 index points, in total consisting of 578 715 observations across all maturities and strike prices.

7.1 Research Environment

Python 3.9.5, an open-source general-purpose computer language, is used for the main part of the empirical study. The computations are carried out using the RTX 580 8Gb GPU and the AMD Ryzen 5 3600 (4Ghz) computer hardware. The MLP neural network is built with Keras 2.8.0, a deep learning API that uses Tensorflow as a computational engine. Keras allows to construct complicated neural network architechtures in a flexible manner.[5]

7.2 Data pre-processing

The data filtering for the final dataset follows the logic proposed by Anders et al. (1996), where the purpose of the filtering is to cut off extraneous and non-representative observations from the data in order to obtain more meaningful results. The first criterion excludes low-priced options, which may lead to high deviations between theoretical and observed option prices. The second criterion filters options with a small amount of time-value as they may also cause deviations between theoretical and observed prices. The third criterion filters options that are not consistent with no-arbitrage conditions. The last criterion filters the options that are deep-in- or deep-out-of-money, as they are rarely traded, containing almost zero informational content. (Anders et al. 1996)

[5]More information about Teras and Kensorflow can be found directly from their websites: https://www.tensorflow.org/ ; https://keras.io/

The filtering is formulated as follows:

1) The call option is traded at less than 10 points:

$$Ct < 10 \tag{32}$$

2) Time-to-Maturity $(T - t)$ is less than 15 days:

$$T - t < 15 \tag{33}$$

3) The lower boundary condition of option is violated:

$$C < S - Ke^{-r*r} \tag{34}$$

Where C represents call price, S Underlying price and r risk-free rate.

4) The option is deep-in- or deep-out-of-money:

$$\frac{S}{K} < 0.85 \quad \text{or} \quad \frac{S}{K} > 1.15 \tag{35}$$

After the applied filtering, the final dataset consist of 271 689 independent observations. Distribution of different strike prices and descriptive statistics are shown below.

Table 2: Descriptive statistics for options

Days till Expiration	<60	60-180	>180	All Options
Average Price £				
OTM (<0.95)	18.92	77.65	140.73	70.98
ATM (0.95-1.05)	154.69	257.96	340.97	249.90
ITM (>1.05)	580.36	654.21	710.68	647
Count				
OTM (<0.95)	195591	37237	11085	68502
ATM (0.95-1.05)	35250	76798	30208	143513
ITM (>1.05)	14038	34150	10973	59674

As shown in the Table 2, majority of the options in the final dataset are close to at-the-money (0.95-1.05 moneyness rate) and have less than 60 days until expiration. The lowest average price for options is for out-of-money options with less than 60 days until expiration, and the highest average price is for options that are close to at-the-money but have more than 180 days until expiration. Naturally, in-the-money options have the highest average prices, while out-of-the-money options have the lowest average prices.

Following Hutchinson et al. (1994) study, 60-day volatility is used as an input variable for BSM. The picture below shows the estimates of 1-month,2-month and 3-month historical volatilitities during 2018-2022, including the study period.

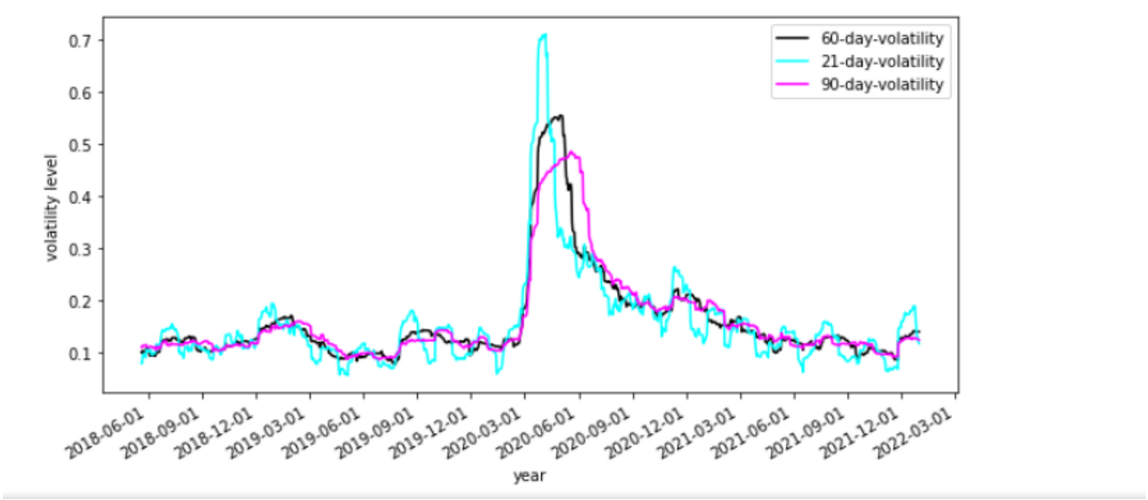

Figure 6: different volatility levels during 2018-2021

Volatility estimates are annualized for both BSM and ANN models using the formula below:

$$Vol(p.a) = S * \sqrt{252}, \tag{36}$$

Another volatility estimate used in this study is the GARCH (1,1) volatility estimate introduced in section 5.1. Bollerslev (1986) showed, that the order of the GARCH form can be evaluated graphically by using autocorrelation function (ACF) and partial autocorrelation function (PACF) plots. The magnitude of significant lags in the graph below suggest using higher order GARCH model, but multiple studies have shown that GARCH(1,1) is sufficient for capturing volatilities in most financial time series (Brooks 2002, 394).

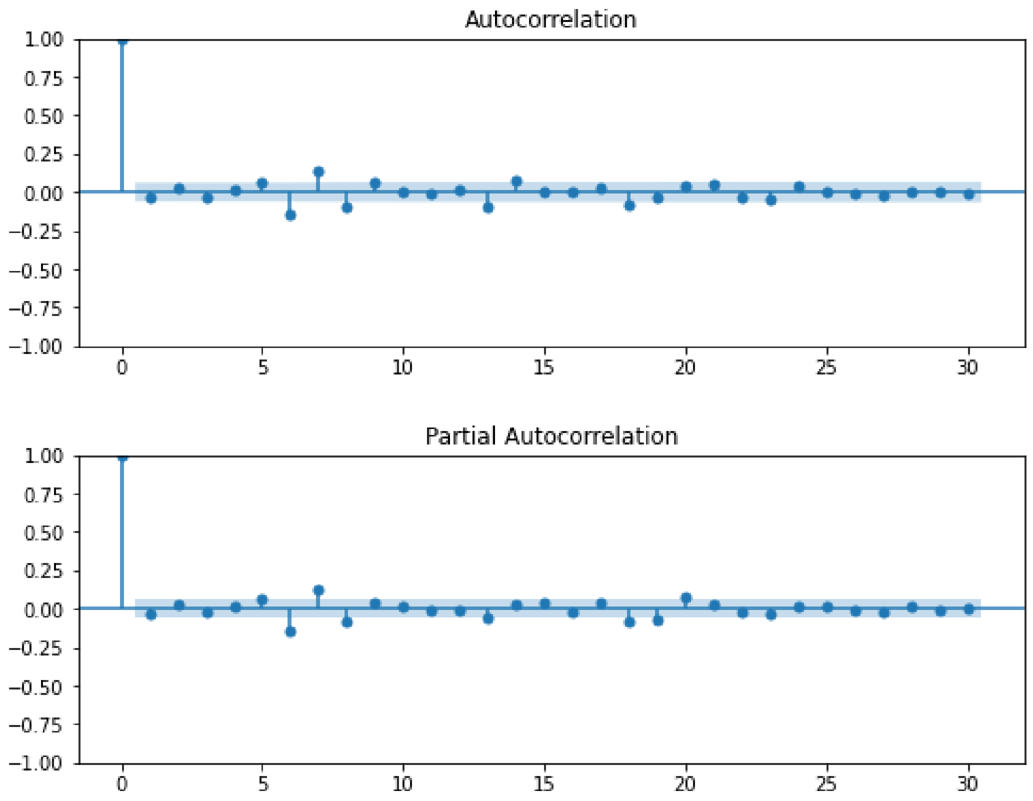

Figure 7: ACF & PACF

The fitted parameters of GARCH(1,1) model are shown below in Table 3. Samples are fitted from 1.6.2018 to 1.3.2021 but the timeframe is later on adjusted to cover only the observation period between 1.1.2019 to 31.12.2021.

Table 3: GARCH(1,1) estimate

Garch	coefficients	Standard Error	t
μ	0.0350	0.544	0.804
ω	0.0525	0.01408	2.455
α	0.1249	0.04668	2.829
β	0.8241	0.05195	15.864

Where μ refers to mean returns, ω to variance intercept, α represents the ARCH effect in equation whereas β refers to GARCH equation.

The mean returns of the estimation, specifically referred as μ are approximately 0.035 during the observation period. When observing the GARCH equation coefficients, both α and β terms appears to be statistically significant in 1% confidence interval indicating that the order of the model fits well to data. Also the constant ω is statistically significant in 1 % confidence interval. Graphical representation of GARCH(1,1) model is also compared to short term volatility in the Figure shown below. The data visualization reveals a close co-integration between the GARCH(1,1) and 5-day volatility series.

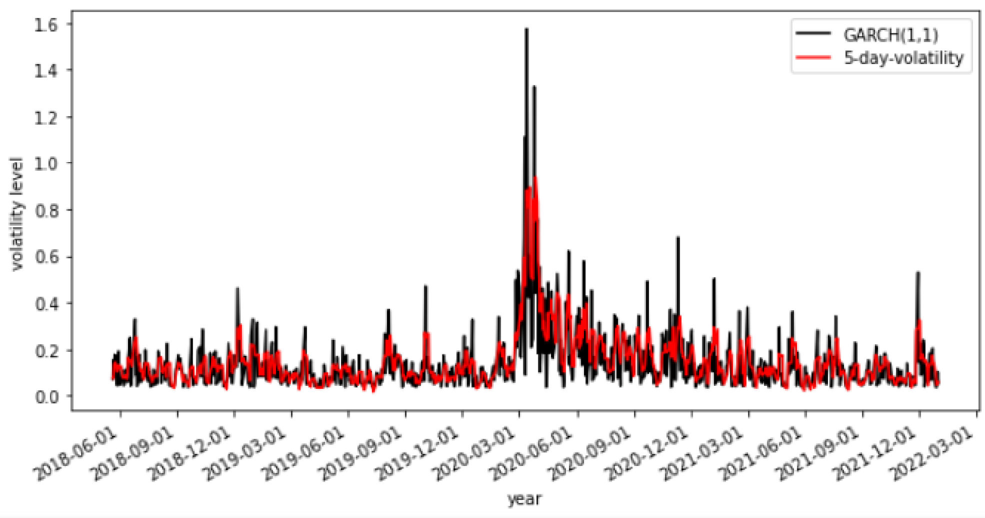

Figure 8: GARCH(1,1) vs 5-day volatility during 2018-2021

7.3 Neural Network Architecture

The architecture of the applied MLP neural networks are influenced by Culkin & Das (2017) study. The input parameters consists of: Underlying price S, Strike price K, risk-free interest rate r, time-to-maturity $T - t$ and volatility estimates $\sigma_{30}, \sigma_{60}, \sigma_{90}$ and σ_{GARCH}.

Two distinct architectures will be implemented. The first ANN contains all other aforementioned input parameters than σ_{GARCH}, and the second model includes an extra GARCH input parameter. These networks' purpose is to approximate the option's pricing function $f(\cdot)$. According to Merton (1973), the return of the underlying asset is independent of the level of the price of the underlying asset S such that it is also independent of the pricing function $f(\cdot)$ of an option price C. Therefore ANNs are taught to estimate the price C divided by strike K. Following Hutchinson et al. (1994) methodology, the input variables are normalized to point out to a homogeneity hint form. The functional form of the first model can be denoted as:

$$\frac{C}{K} = f(S/K, 1, T - t, r, \sigma_{30}, \sigma_{60}, \sigma_{90}) \tag{37}$$

And for the second model with additional GARCH parameter:

$$\frac{C}{K} = f(S/K, 1, T - t, r, \sigma_{30}, \sigma_{60}, \sigma_{90}, \sigma_{GARCH}) \tag{38}$$

The entire network architecture is composed of one input layer, four hidden layers with 120 neurons each, and one output layer that outputs the projected call option price. The first hidden layer utilizes LeakyRelu as activation function, the second and fourth hidden layers Elu and the third hidden layer uses Relu. Since the option prices can not be negative, the exponential function is used at output layer to yield only positive values (Culkin & Das 2017).

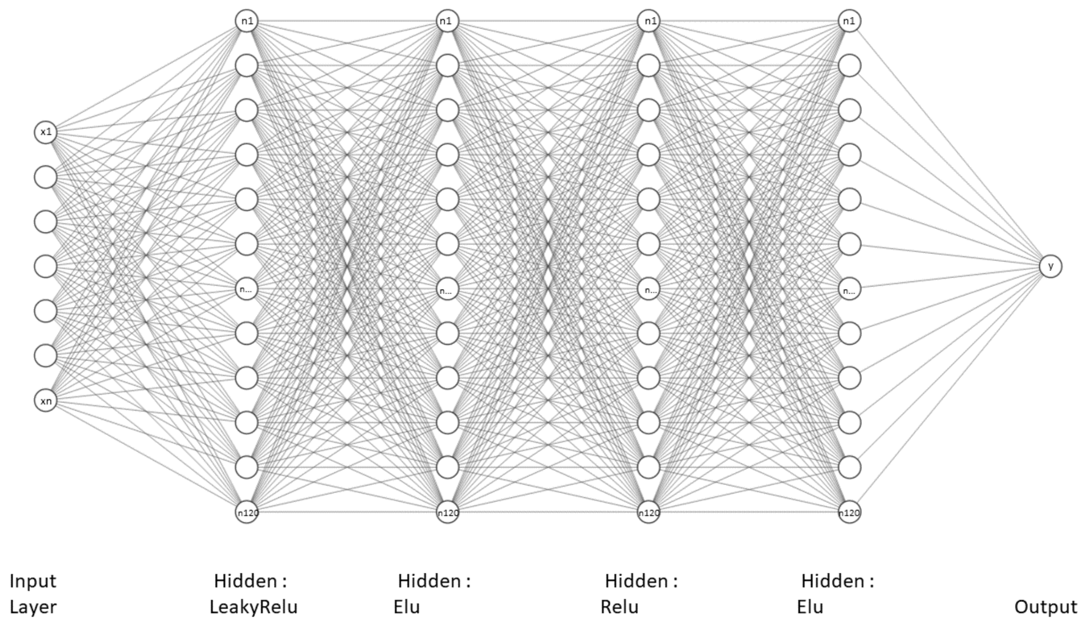

Figure 9: Applied Neural Network Model with 4 hidden layers

Furthermore, the data is divided according to the 80/20 principle, with the first 80% of the dataset serving as a training sample and the remaining 20% serving as a test sample. The model's performance is evaluated with out-of-sample testing set after training it with a training set. The hyperparameters are also tuned with a 25% dropout rate. The dropout rate is a regularization technique that prevents the neural network from overfitting (Srivastava et al. 2014). The number of epochs used for training is 1000 per learning network, and the batch size, which reflects the number of samples processed before updating the model, is set to 64. In addition to this, the loss function is optimized using MSE statistics formulated in Equation 22.

7.4 Delta Hedging performance measure

Delta hedging performance of ANN compared to BSM is investigated through *tracking error* proposed by Hutchinson et al. (1994). The idea behind the methodology is to replicate a delta hedged position by offsetting the risk from the option position. If the pricing formula is correctly identified and there is a possibility to trade without costs on a continuous basis, then the combined value of the stock and bond position should offset the value of the call position. The BSM model assumes continuous delta hedging, but in reality, this is impossible, and there will always be some level of tracking error due to discreteness. Hence the hedging performance will be evaluated by comparing if the tracking error of the neural network is lower than the one with BSM.

Lets denote the portfolio value at time t as $V(t)$.

$$V(t) = V_S(t) + V_B(t) + V_C(t) \tag{39}$$

Where $V_S(t)$ is the value of spot value of stocks, $V_B(t)$ is the value of bond position and $V_C(t)$ is the spot value of calls taken at date t. The composition of the portfolio at date 0 can be denoted as:

$$S(0) = S(0)\Delta_{ANN}(0), \quad \Delta ANN(0) = \frac{\partial F_{ANN}(0)}{\partial S} \tag{40}$$

$$V_C(0) = -F_{BSM}(0) \tag{41}$$

$$V_B(0) = -(V_S(0) + V_C(0)) \tag{42}$$

The hedging strategy goes as follows: one writes a call option according to Equation 41, takes simultaneously a long position for the spot shown in equation 40, and lastly shorts the bond to finance the hedge according to eq. 42. F_{BSM} represents the approximation of the call value calculated by BSM, F_{ANN} is the corresponding call approximation calculated by the neural network. Note that the Δ_{ANN} represents the partial derivative of call price computed via Neural Network with respect to spot price. The delta for the Neural Network can be computed analytically from the network approximation. The computer graph of Tensorflow allows us to calculate the delta measure easily.

Preceding the expiration, the portfolio will be rebalanced in a regular interval τ. The stock and bond positions are rebalanced as follows:

$$V_S(t) = S(t)\Delta_{ANN}(t), \quad \Delta ANN(t) = \frac{\partial F_{ANN}(t)}{\partial S} \tag{43}$$

$$V_B(t) = e^{rt}V_B(t-\tau) - S(t)(\Delta_{ANN}(t) - \Delta_{ANN}(t-\tau)) \tag{44}$$

τ represents one trading day, and r is the definition of risk-free rate.
Finally the tracking error of the portfolio can be formulated as:

$$\xi = e^{-rT}E[|V(T)|] \tag{45}$$

The experiment will be done simultaneously for Black & Scholes model, where Δ_{ANN}

is replaced by:

$$\Delta BSM(0) = \frac{\partial F_{BSM}(0)}{\partial S} = N(d1) \tag{46}$$

A paired t-test will be done to test the statistical properties of the absolute hedging errors. Hutchinson et al. (1994) argues that the statistical dependence in option-price paths might cause nonreliable results and suggests a less formal methodology to investigate the differences in tracking errors. The null hypothesis of the paired t-test is that the average difference in tracking errors are zero and the one-sized hypothesis states that the difference is positive:

$$D = |\mu_{ANN}| - |\mu_{BSM}| \tag{47}$$

- H_0: D $= 0$
- H_1: D $\neq 0$

8 Results

In this chapter, the findings of the empirical data analysis are presented based on the methodology described in the last chapter. At first, the pricing performance of ANN models and BSM for the whole out-of-sample is evaluated using performance metrics listed in Section 4. Pricing performance is hereafter analyzed by segmenting data based on the option's moneyness rate and time-to-maturity to obtain more insights. Lastly, the results of the hedging performance measured by paired t-test are discussed.

8.1 Pricing Results

Table 4: Results for test set without partition

Full test set	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.565	7.57	5.85
NN_Garch	0.764	8.17	7.16
Black&Scholes	5.3	23.2	15.7

The Table 4 shows MSE, RMSE, and MAE (Eqs. 22,24,23) measures for the whole out-of-sample without data partition. Both RMSE and MAE quantify price discrepancies in absolute units, while MSE assesses pricing performance more holistically (Anders et al. 1996). The results indicate superior pricing performance for both ANN models over BSM measured with MSE, RMSE, and MAE during the observation period. This finding is consistent with the previous studies of, e.g. Hutchinson et al. (1994), Anders et al. (1996) and Bennell & Sutcliffe (2004).

When the whole out-sample data set is considered, the MLP network with additional GARCH parameters is inferior compared to the regular MLP neural network in all performance metrics. This outcome might be explained by the fact that the GARCH-generated volatility estimate (eq.28) is insufficient to provide any explanatory power on the ANN's pricing function. Despite the fact that several studies have demonstrated the GARCH(1,1) process's volatility estimation capabilities, the use of GARCH-derived volatility estimates in option pricing formulas has faced criticism in the literature. For example Anders et al. (1996) contends that the GARCH model is incompatible with the BSM valuation formula (eq.15), and more advanced EGARCH models should be used instead.

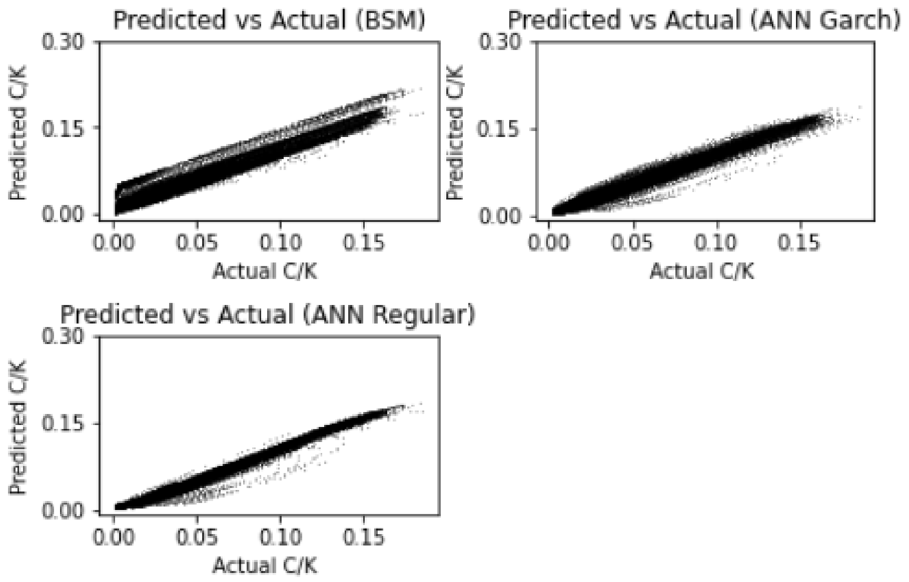

Figure 10: Predicted vs Actual plots

The performance for the overall out-of-sample dataset is also evaluated graphically. The predicted vs. Actual plots in Figure 10 illustrate the best-fitting line for the regular ANN model and the worst-fitting line for BSM. The equivalent plot also illustrates some deviation in higher and lower C/K levels for the ANN GARCH model and BSM model. One rationale for this interpretation is because deep-in- or deep-out-of-money options are traded far less frequently than near-the-money options.

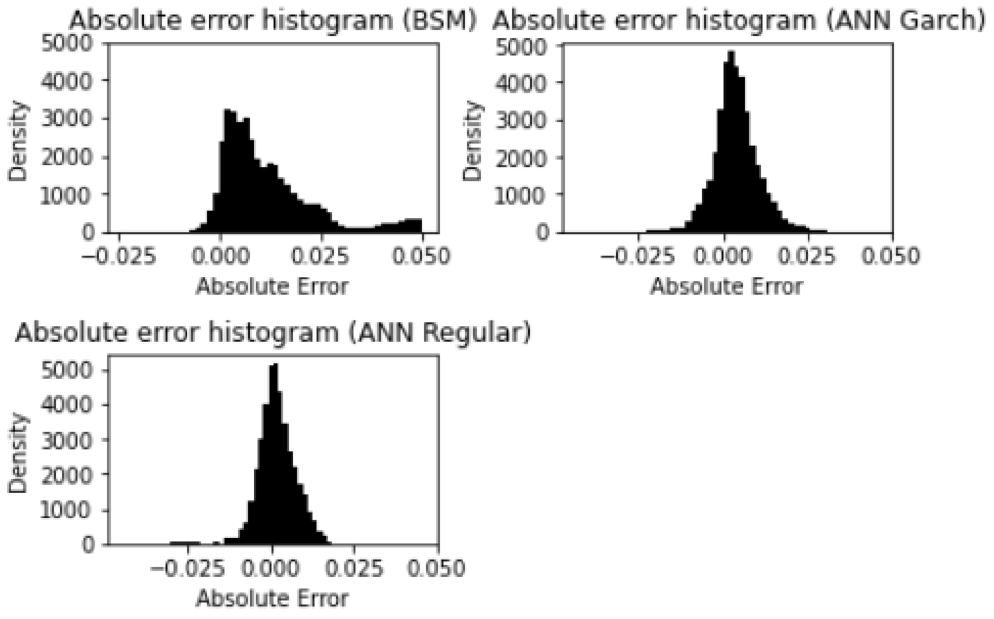

Figure 11: Absolute Error Histograms

Pricing error distribution plots are found in Figure 11. Pricing errors are calculated by subtracting the model price from the actual option price per each data point pair. Pricing error distribution is relatively normally distributed based on visual inspection in the case of regular- and GARCH ANN model, but positively skewed for BSM indicating overpricing in options. One possible explanation for this stems from the constant 60-day volatility assumption made for BSM. During the observation period, the 60-day volatility was higher compared to longer-term volatility, which causes some form of overpricing of options and leads to a right-tailed absolute error histogram.

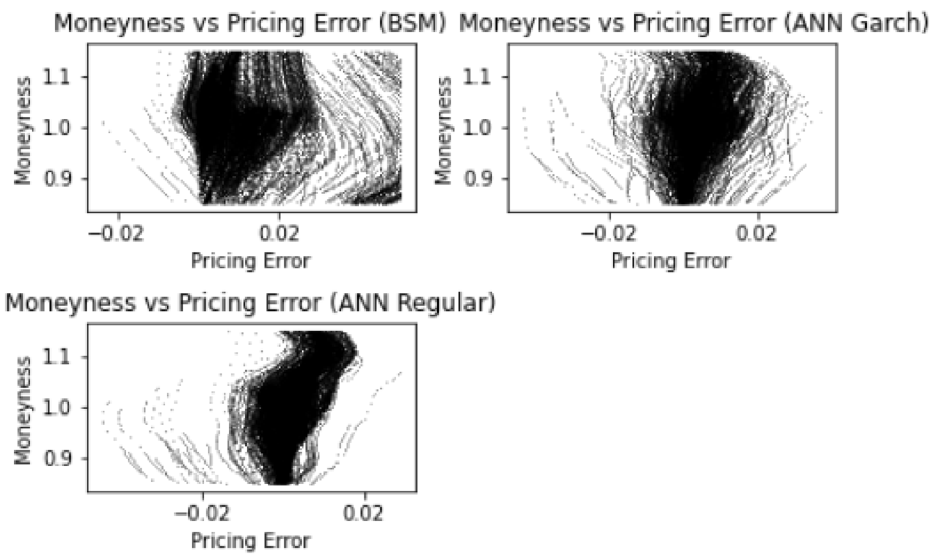

Figure 12: Moneyness vs Pricing Error

Moneyness vs. Pricing error plots are displayed in Figure 12 for all three models. The plots show the magnitude of pricing error compared to the moneyness level. According to the visualization, the regular ANN model seems to underprice deep-out-of-money options. Similar observations can also be drawn for ANN Garch and BSM models. Both BSM and MLP GARCH models also overprice deep-in-the-money options according to the plots. Figure 12 also show that BSM tends to underprice deep-out-of-money options similarly as mentioned for deep-ITM options. This tendency has also shown in the papers of Yao et al. (2000) and Gençay & Salih (2003).

Table 5: Results for Pricing performance, in-the-money (>1.05)

ITM	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.319	5.07	4.1
NN_Garch	2,9	17.22	16.53
Black&Scholes	3.5	19.12	13.75

Table 6: Results for Pricing performance, out-the-money (<0.95)

OTM	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.294	5.4	3.11
NN_Garch	0.676	8.2	5.26
Black&Scholes	9.1	30.12	22.17

Table 7: Results for Pricing performance, at-the-money, (0.95-1.05)

ATM	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.846	9.14	7.93
NN_Garch	0.848	9.22	8.24
Black&Scholes	4.6	21.55	14.41

Pricing results partitioned by their maturity are displayed in tables 5-7. Data is partitioned by grouping options based on their moneyness level following the filtering criteria: ITM, when moneyness > 1.05, ATM when moneyness between 0.95-1.05, and OTM when moneyness < 0.95. Both ANN models outperform BSM in all three performance metrics even though the data is partitioned by moneyness rate. Regular ANN model outperforms ANN GARCH model in ITM and OTM category, but contains almost identical MSE statistics compared to ANN GARCH model in ATM category. Rather small deviations in the ATM category are not sufficient enough to explain which one of the following ANN models is better to price ATM options.

Table 6 also illustrates that both ANN models compound the lowest MSE, RMSE, and MAE values in the OTM category indicating that they are better at pricing OTM options than ITM or ATM options. This finding is also in line with Malliaris & Salchenberger (1993) and Healy et al. (2002) and Bennell & Sutcliffe (2004). A potential explanation for why ANN models perform better than the BSM model in the OTM category can be found in the underlying assumptions of BSM. BSM assumes continuous and cost-free trading on options (Black & Scholes 1973). As mentioned previously, very thin trading in deep in- or out-of-money options may create potential mispricing effects on the markets, which ANN models have clearly captured during the training. Another interesting finding can be found in the ITM category. BSM model compounds the best performance in the ITM category compared to all other moneyness categories. This observation is in line with Bennell & Sutcliffe (2004). Their paper also found BSM superior compared to ANN, which is not in line with these results.

Table 8: Results for Pricing performance, long-term options (>180d)

Long term	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.304	5.45	4.2
NN_Garch	0.933	9.67	5.01
Black&Scholes	3.3	18.3	17.17

Table 9: Results for Pricing performance, medium-term options (60-180d)

Medium term	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.25	5.19	3.43
NN_Garch	0.491	7.28	5.30
Black&Scholes	5.8	7.31	5.62

Table 10: Results for Pricing performance, Short-term options (<60d)

Short term	MSEx10-4	RMSEx10-3	MAEx10-3
NN	0.1683	4.17	3.23
NN_Garch	0.395	6.26	5.13
Black&Scholes	0.541	7.32	6.50

Furthermore, pricing results are also partitioned by time-to-maturity following the criterion: Short-term options $<$ 60 days, medium-term options 60-180 days, and long-term options $>$ 180 days. Time-to-maturity partitioned results are seen in tables 7-10. Overall, we can say that the regular ANN model has the best pricing performance compared to other models in all three error metrics when different maturity levels are considered. The maturity split also shows that both ANN models outperform BSM, and therefore it can be concluded that the neural network model is more suitable for pricing FTSE 100 index options than the theoretical BSM model.

To open a little bit more of the partitioned results based on the time-to-maturity level, the regular ANN model has the best fit for medium-term options compared to the two other models. ANN GARCH model and BSM model have the best fit for short-term options. This stems from the fact that GARCH(1,1) volatility's explanatory power is superior for short-term options due to the fact that GARCH(1,1) is sufficient for capturing short-term fluctuations. This finding is also straightforward for BSM, since the calculation for BSM is done by following Hutchinson et al. (1994) methodology, and constant 60 days historical volatility has been used to calculate theoretical BSM prices. This procedure automatically leads to some degree of maturity mismatch dilemma compared to exact volatility matching, especially for options with longer time-to-expiration, thus potentially under- or overpricing options depending on the current volatility level. However, on a larger scale, this effect should not be relatively significant to actually explain why the pricing performance of BSM is inferior compared to ANN. The regular ANN model also outperforms BSM, and the ANN Garch model with long-term category than its peers, thus containing the lowest errors in every category when the time-to-matyrity and moneyness is considered.

8.2 Hedging Results

The hedging performance for ANN vs. BSM is introduced in this subsection. The hedging performance measurement is followed by the procedure introduced in chapter 7.4. The results are stored for the whole out-of-sample dataset and also partitioned by moneyness. Options with time-to-maturity less than 50 days are removed from the dataset following Hutchinson et al. (1994) procedure. Only the regular ANN model is chosen as a benchmark against BSM since the pricing performance was better compared to ANN GARCH, meaning that regular ANN's delta must be more realistic compared to the corresponding one of the ANN GARCH model.

Table 11: Hedging results partitioned by moneyness

Sample	t-statistic	p-value	Tracking error ANN £	Tracking error BSM £
Full	-3.11	<0.0019	557.14	613.99

Table 11 shows hedging results for the whole out-of-sample. Based on the obtained results, the ANN outperforms BSM in delta hedging based on paired t-test in 1% confidence interval. A similar finding can also be made by looking at average tracking errors: The average tracking error for ANN is smaller than for BSM. More comprehensive analysis can be done when options are partitioned by moneyness. The moneyness partition criteria are different compared to the pricing criteria to keep more observations in different categories. An option is considered to be ITM when moneyness is greater than 1, and OTM when moneyness is lower than 1.

Table 12: Hedging results partitioned by moneyness

Sample	t-statistic	p-value	Tracking error ANN £	Tracking error BSM £
ITM	-1.23	<0.21	1265.52	1314.28
OTM	-7.11	0.00001	423.09	525.45

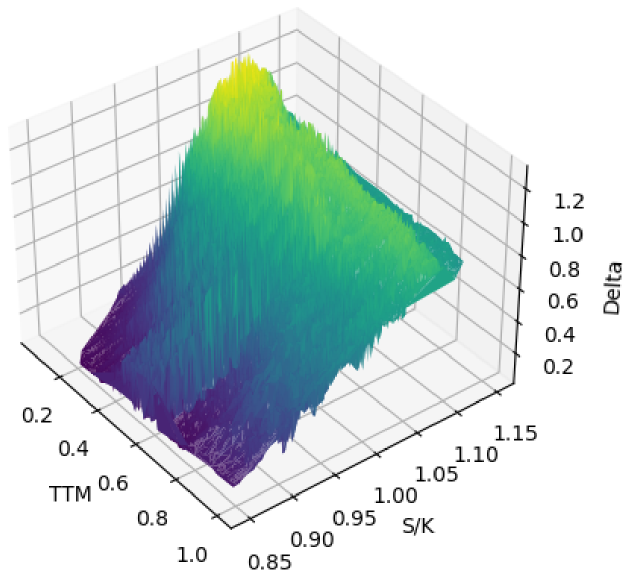

Figure 13: Δ_{ANN} surface

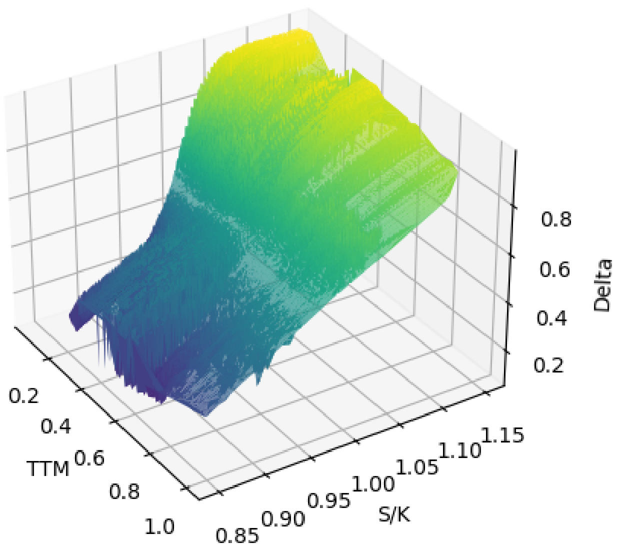

Figure 14: Δ_{BSM} surface

From Table 12, we can see that neither of the models is better for pricing ITM options. One reason for this is caused by the delta properties of ANN: for deep-in-the-money options, the delta level boundary is violated. According to theory, the delta for the options should be between 0 and 1 (Cox & Rubinstein 1985, 44-47). The Figure 13 shows a delta greater than 1 for short-term ITM options, showing the delta level boundary violation graphically. Similarly, Figure 14 shows delta distribution for the BSM model, which explicitly shows that the delta values are distributed between 0 to 1, indicating consistent delta estimation. This finding is aligned with (Anders et al. 1996), indicating the neural network hasn't learned the implicit properties of the pricing formula.

When OTM options are considered, the result is clear. ANN outperforms BSM with 1% confidence interval, and the tracking error is significantly smaller compared to BSM. The result is not surprising since the ANN's pricing performance was already found to be highest for OTM options. The obtained results are also in-line with Hutchinson et al. (1994), who also found superior hedging performance in MLP neural network over BSM based on the paired t-test.

9 Conclusion and Discussion

This study evaluated the option pricing & delta hedging performance of Artificial Neural Networks in comparison against the Black & Scholes option pricing model using FTSE 100 index options during 2019-2021. In total, two distinct MLP models were tested against BSM: One with regular BSM input variables and the other with additional GARCH(1,1) parameters.

The first research question: *What type of Neural Network models are the most suitable for option pricing and hedging and what is their relative performance compared to Black & Scholes model according to the literature?* was covered in the literature review section by looking latter research about option pricing and hedging with Neural Networks. To summarize the findings, the research domain compounds multiple different Artificial Neural Network models for option pricing. Much of the research has been made with MLP, and they have mostly provided better results compared to BSM and other option pricing models.

In the empirical part, both MLP models built for the purposes of this thesis were found superior in option pricing when benchmarked against the BSM model in the given data, giving answers to the second research question: *What is the relative hedging and pricing performance of MLP neural network compared to Black & Scholes option pricing model?* The key findings in pricing also cover superior performance for both MLP models when the data was split based on moneyness and time-to-maturity, confirming the previous studies in this research area. Answers were also found for the second part of the research question, as the delta hedging performance was tested against the BSM model. Delta hedging performance of ANNs suggests superior performance over BSM in the overall dataset, but when the data was split based on moneyness, only OTM options were found statistically significant to outperform the BSM model.

The last key finding of the empirical testing concludes that the additional GARCH(1,1) parameter did not improve pricing performance giving answers to the last research question: *Can the pricing performance of the currently applied Neural Network models for option pricing be improved with additional GARCH volatility parameter?* This conclusion might be explained by the data and time period employed, or by the fact that the regular GARCH volatility is an inadequate parameter in the option pricing formula. However, these findings gives a basis for the fact that ANNs are actually capable of learning the option pricing formula and improving the inefficiencies in the BSM formula to construct more accurate estimates for option's pricing and hedging. Traders may find this thesis useful by using neural network models for observing arbitrage opportunities in option markets or when creating more efficient hedging strategies to cover their positions.

The research area has been focusing much on the pricing performance, whereas the hedging performance hasn't been explored so widely. Potential research ideas in this field of study could be associated with hedging performance with more complex neural network architectures and other option types such as commodity options. Another interesting research idea would be to study whether the more advanced GARCH volatility estimates used as an input variable in Neural Networks, such as the E-GARCH volatility proposed by Anders et al. (1996), would increase the option pricing accuracy.

References

Abraham, A. (2005), ‘Artificial neural networks’, *Handbook of measuring system design* .

Amilon, H. (2003), ‘A neural network versus black–scholes: a comparison of pricing and hedging performances’, *Journal of Forecasting* **22**(4), 317–335.

Anders, U., Korn, O. & Schmitt, C. (1996), ‘Improving the pricing of options: A neural network approach’, *Journal of forecasting* **17**(5-6), 369–388.

Andreou, P. C., Charalambous, C. & Martzoukos, S. H. (2008), ‘Pricing and trading european options by combining artificial neural networks and parametric models with implied parameters’, *European Journal of Operational Research* **185**(3), 1415–1433.

Arnekvist, I., Carvalho, J. F., Kragic, D. & Stork, J. A. (2020), ‘The effect of target normalization and momentum on dying relu’, *arXiv preprint arXiv:2005.06195* .

Barunıkova, M. & Barunık, J. (2011), ‘Neural networks as a semiparametric option pricing tool’.

Bennell, J. & Sutcliffe, C. (2004), ‘Black–scholes versus artificial neural networks in pricing ftse 100 options’, *Intelligent Systems in Accounting, Finance & Management: International Journal* **12**(4), 243–260.

Black, F. & Scholes, M. (1973), ‘The pricing of options and corporate liabilities’, *Journal of political economy* **81**(3), 637–654.

Bollerslev, T. (1986), ‘Generalized autoregressive conditional heteroskedasticity’, *Journal of econometrics* **31**(3), 307–327.

Briegel, T. & Tresp, V. (2000), ‘Dynamic neural regression models’.

Brooks, C. (2002), *Introductory Econometrics for Finance*, Cambridge University Press.
URL: *https://books.google.fi/books?id=3lqHDwAAQBAJ*

Can, M. & Fadda, S. (2014), ‘A nonparametric approach to pricing options learning networks’, *Southeast Europe Journal of Soft Computing* **3**(1).

Cox, J. C. & Rubinstein, M. (1985), *Options markets*, Prentice Hall.

Culkin, R. & Das, S. R. (2017), ‘Machine learning in finance: the case of deep learning for option pricing’, *Journal of Investment Management* **15**(4), 92–100.

Detienne, K. B., Detienne, D. H. & Joshi, S. A. (2003), ‘Neural networks as statistical tools for business researchers’, *Organizational Research Methods* **6**(2), 236–265.

Dugas, C., Bengio, Y., Bélisle, F., Nadeau, C. & Garcia, R. (2001), ‘Incorporating second-order functional knowledge for better option pricing’, *Advances in neural information processing systems* pp. 472–478.

Freitas, J. d., Niranjan, M. & Gee, A. H. (2000), ‘Hierarchical bayesian models for regularization in sequential learning’, *Neural computation* **12**(4), 933–953.

Garcia, R. & Gençay, R. (2000), ‘Pricing and hedging derivative securities with neural networks and a homogeneity hint’, *Journal of Econometrics* **94**(1-2), 93–115.

Gençay, R. & Gibson, R. (2007), ‘Model risk for european-style stock index options’, *IEEE transactions on neural networks* **18**(1), 193–202.

Gençay, R. & Salih, A. (2003), ‘Degree of mispricing with the black-scholes model and nonparametric cures’, *Economics and Finance. Annals* **4**, 73–101.

Ghaziri, H., Elfakhani, S. & Assi, J. (2000), ‘Neural, netvvorks approach to pricinc, options’, *Neural Network World* **1**(2/00), 271–277.

Gradojevic, N., Gençay, R. & Kukolj, D. (2009), ‘Option pricing with modular neural networks’, *IEEE transactions on neural networks* **20**(4), 626–637.

Hahnloser, R. H., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J. & Seung, H. S. (2000), ‘Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit’, *nature* **405**(6789), 947–951.

Haug, E. G. & Taleb, N. N. (2011), ‘Option traders use (very) sophisticated heuristics, never the black–scholes–merton formula’, *Journal of Economic Behavior & Organization* **77**(2), 97–106.

Healy, J., Dixon, M., Read, B. & Cai, F. (2002), ‘A data-centric approach to understanding the pricing of financial options’, *The European Physical Journal B-Condensed Matter and Complex Systems* **27**(2), 219–227.

Hirsa, A., Karatas, T. & Oskoui, A. (2019), ‘Supervised deep neural networks (dnns) for pricing/calibration of vanilla/exotic options under various different processes’, *arXiv preprint arXiv:1902.05810* .

Hornik, K., Stinchcombe, M. & White, H. (1989), ‘Multilayer feedforward networks are universal approximators’, *Neural networks* **2**(5), 359–366.

Hornik, K., Stinchcombe, M. & White, H. (1990), ‘Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks’, *Neural networks* **3**(5), 551–560.

Huang, S.-C. (2008), ‘Online option price forecasting by using unscented kalman filters and support vector machines’, *Expert Systems with Applications* **34**(4), 2819–2825.

Hull, J. C. (2003), *Options,futures,and other derivatives*, Prentice-Hall International.

Hull, J. & White, A. (2017), ‘Optimal delta hedging for options’, *Journal of Banking & Finance* **82**, 180–190.

Hutchinson, J. M., Lo, A. W. & Poggio, T. (1994), ‘A nonparametric approach to pricing and hedging derivative securities via learning networks’, *The journal of Finance* **49**(3), 851–889.

Jackwerth, J. C. & Rubinstein, M. (1996), ‘Recovering probability distributions from option prices’, *The journal of Finance* **51**(5), 1611–1631.

Jung, K.-H., Kim, H.-C. & Lee, J. (2006), ‘A novel learning network for option pricing with confidence interval information’, pp. 491–497.

Kelly, D. L., Shorish, J. et al. (1994), ‘Valuing and hedging american put options using neural networks’, *Unpublished manuscript, Carnegie Mellon University* .

Krøse, B. (1996), ‘Patrick van der smagt’, *An introduction to neural networks* .

Kubat, M. & Kubat (2017), *An introduction to machine learning*, Vol. 2, Springer.

Lin, C.-T. & Yeh, H.-Y. (2005), ‘The valuation of taiwan stock index option price-comparison of performances between black-scholes and neural network model’, *Journal of Statistics and Management Systems* **8**(2), 355–367.

Lu, L., Shin, Y., Su, Y. & Karniadakis, G. E. (2019), ‘Dying relu and initialization: Theory and numerical examples’, *arXiv preprint arXiv:1903.06733* .

Maas, A. L., Hannun, A. Y., Ng, A. Y. et al. (2013), Rectifier nonlinearities improve neural network acoustic models, *in* ‘Proc. icml’, Vol. 30, Atlanta, Georgia, USA, p. 3.

MacKenzie, D. (2006), ‘Is economics performative? option theory and the construction of derivatives markets’, *Journal of the history of economic thought* **28**(1), 29–55.

Malliaris, M. & Salchenberger, L. (1993), ‘A neural network model for estimating option prices’, *Applied Intelligence* **3**(3), 193–206.

McCulloch, W. S. & Pitts, W. (1943), ‘A logical calculus of the ideas immanent in nervous activity’, *The bulletin of mathematical biophysics* **5**(4), 115–133.

Merton, R. C. (1973), ‘Theory of rational option pricing’, *The Bell Journal of economics and management science* pp. 141–183.

Mitra, S. K. (2006), ‘Improving accuracy of option price estimation using artificial neural networks’.

Montesdeoca, L. & Niranjan, M. (2016), Extending the feature set of a data-driven artificial neural network model of pricing financial options, *in* ‘2016 IEEE symposium series on computational intelligence (SSCI)’, IEEE, pp. 1–6.

Murphy, K. P. (2012), *Machine learning: a probabilistic perspective*, MIT press.

Nielsen, M. A. (2015), *Neural networks and deep learning*, Vol. 25, Determination press San Francisco, CA, USA.

Nilsson, N. J. (1996), 'Introduction to machine learning. an early draft of a proposed textbook'.

Niranjan, M. (1996), 'Sequential tracking in pricing financial options using model based and neural network approaches', *Advances in neural information processing systems* **9**, 960–966.

Nwankpa, C., Ijomah, W., Gachagan, A. & Marshall, S. (2018), 'Activation functions: Comparison of trends in practice and research for deep learning', *arXiv preprint arXiv:1811.03378* .

Ormoneit, D. (1999), 'A regularization approach to continuous learning with an application to financial derivatives pricing', *Neural Networks* **12**(10), 1405–1412.

Pande, A. & Sahu, R. (2006), A new approach to volatility estimation and option price prediction for dividend paying stocks, *in* 'WEHIA 2006–1st International Conference on Economic Sciences with Heterogeneous Interacting Agents; 15–17 June 2006, University of Bologna, Italy', Citeseer.

Poon, S.-H. & Granger, C. W. (2003), 'Forecasting volatility in financial markets: A review', *Journal of economic literature* **41**(2), 478–539.

Ramachandran, P., Zoph, B. & Le, Q. V. (2017), 'Searching for activation functions', *arXiv preprint arXiv:1710.05941* .

Rosenblatt, F. (1958), 'The perceptron: a probabilistic model for information storage and organization in the brain.', *Psychological review* **65**(6), 386.

Ruf, J. & Wang, W. (2020), 'Neural networks for option pricing and hedging: a literature review', *Journal of Computational Finance, Forthcoming* .

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986), 'Learning representations by back-propagating errors', *nature* **323**(6088), 533–536.

Schittenkopf, C. & Dorffner, G. (2001), 'Risk-neutral density extraction from option prices: improved pricing with mixture density networks', *IEEE Transactions on Neural Networks* **12**(4), 716–725.

Sharma, S., Sharma, S. & Athaiya, A. (2017), 'Activation functions in neural networks', *towards data science* **6**(12), 310–316.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), 'Dropout: a simple way to prevent neural networks from overfitting', *The journal of machine learning research* **15**(1), 1929–1958.

Taleb, N. N. (1997), *Dynamic hedging: managing vanilla and exotic options*, Vol. 64, John Wiley & Sons.

Teddy, S. D., Lai, E.-K. & Quek, C. (2006), A brain-inspired cerebellar associative memory approach to option pricing and arbitrage trading, *in* 'International Conference on Neural Information Processing', Springer, pp. 370–379.

Thomaidis, N. S., Tzastoudis, V. S. & Dounias, G. (2007), 'A comparison of neural network model selection strategies for the pricing of s&p 500 stock index options', *International Journal on Artificial Intelligence Tools* **16**(06), 1093–1113.

Yang, Y., Zheng, Y. & Hospedales, T. (2017), Gated neural networks for option pricing: Rationality by design, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', Vol. 31.

Yao, J., Li, Y. & Tan, C. L. (2000), 'Option price forecasting using neural networks', *Omega* **28**(4), 455–466.

Yegnanarayana, B. (2009), *Artificial neural networks*, PHI Learning Pvt. Ltd.

Zapart, C. (2002), 'Stochastic volatility options pricing with wavelets and artificial neural networks', *Quantitative Finance* **2**(6), 487.