QUANTITATIVE COMPARISON OF OPTION PRICING MODELS: NEURAL NETWORKS VS. STOCHASTIC MODELS

Jyväskylä University
School of Business and Economics

Master's Thesis

2024

Author: Sebastian Nyholm
Subject: Economics
Supervisor: Esa Mangeloja

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

ABSTRACT

Author Sebastian Nyholm	
Title Quantitative comparison of option pricing models: neural networks vs. stochastic models	
Subject Economics	Type of work Master's thesis
Date 03.05.2024	Number of pages 73

This study presents a quantitative comparison of three different option pricing models. The emphasis is on the quite recent artificial neural network model, which is compared to the Monte Carlo simulation and the Black-Scholes-Merton pricing model. The financial markets' complexity demands increasingly sophisticated models, and recent advances in computing power have facilitated the development of intricate option pricing models.

Especially the sub-model derived from artificial neural networks, the multilayer perceptron has been used in pricing European call options. Existing literature demonstrates the multilayer perceptron's superiority in pricing accuracy compared to the Black-Scholes-Merton model. However, successful implementation necessitates specific model inputs, defined network architecture, and a substantial amount of data.

The results of this study underscore the better predictive accuracy of the artificial neural networks when compared to the stochastic models, as it is more accurate in predicting the option prices when using the complete testing dataset. Notably, the artificial neural network exhibits exceptional performance when pricing out-of-the-money options, with diminishing discrepancies to the stochastic models observed with in-the-money options, to the point of the network's results being comparable to the results of the stochastic models. The two stochastic models used in this thesis expectedly perform extremely similarly.

The optimal network architecture identified diverges notably from those architectures used in prior literature, featuring significantly greater numbers of hidden layers and neurons per layer. However, despite the large network size this does not cause overfitting problems, and this is somewhat attributable to the large reliable dataset. The time period used, along with the chronological data partitioning method, caused problems, ultimately leading to the decision to drop the interest rate variable from the network model altogether.

Keywords
Option pricing, options, neural networks, machine learning, stochastic models, derivatives pricing, computational finance

Place of storage
Jyväskylä University Library

TIIVISTELMÄ

Tekijä
Sebastian Nyholm

Työn nimi
Optiohinnoittelumallien kvantitatiivinen vertailu: neuroverkot vs. stokastiset mallit

Oppiaine	Työn laji
Taloustiede	Pro gradu -tutkielma
Päivämäärä	**Sivumäärä**
03.05.2024	73

Tämä tutkimus vertailee kolmea erilaista optioiden hinnoittelumallia kvantitatiivisesti. Painopiste on melko uuden keinotekoisen neuroverkkomallin käytössä, jota vertaillaan Monte Carlo simulaatioon sekä Black-Scholes-Merton hinnoittelumalliin. Rahoitusmarkkinoiden monimutkaisuus sekä sen datamassat vaativat monimutkaisempien mallien käyttöä kuin koskaan aikaisemmin, ja viimeaikaiset edistysaskeleet laskentatehossa ovat mahdollistaneet hienovaraisten hinnoittelumallien kehityksen.

Erityisesti keinotekoisten neuroverkkojen alamallia monikerroksista perseptroniverkkoa on käytetty eurooppalaisten osto-optioiden hinnoittelussa. Kirjallisuus todistaa monikerroksisten perseptroniverkkojen olevan tarkempia Black-Scholes-Merton malliin verrattuna, vaikkakin tietynlaiset syötteet, verkkoarkkitehtuuri sekä suuri datasetti ovat välttämättömiä tämän tarkkuuden saavuttamiseksi.

Tämän tutkimuksen tulokset korostavat keinotekoisten neuroverkkojen tarkempaa hinnoittelua verrattuna stokastisiin malleihin, sillä ne ovat tarkempia optiohinnoittelussa, kun käytetään vertailussa koko datasettiä. Keinotekoinen neuroverkko osoittaa poikkeuksellista suorituskykyä miinusoptioilla (out-of-the-money), ja erot stokastisiin malleihin pienenevät siirryttäessä plusoptioihin (in-the-money), joka johtaa mallien samantasoiseen suorituskykyyn plusoptioilla. Tutkimuksessa käytettävät stokastiset mallit suoriutuivat odotetusti erittäin samankaltaisesti.

Tuloksissa tunnistettu optimaalinen neuroverkkorakenne poikkeaa huomattavasti aiemmassa kirjallisuudessa käytetyistä rakenteista, sillä siinä on merkittävästi enemmän piilokerroksia sekä neuroneita per kerros. Vastoin odotuksia suuri koko ei kuitenkaan aiheuta ylisovittamisongelmia, mikä johtuu osittain suuresta ja luotettavasta datasetistä. Käytetty aikaperiodi kronologisen jaottelun kanssa aiheuttaa ongelmia ja tämä johtaa lopulta korkotaso muuttujan poistamiseen neuroverkkomallista kokonaan.

Asiasanat
Optioiden hinnoittelu, optiot, neuroverkot, koneoppiminen, stokastiset mallit, johdannaisten hinnoittelu, laskennallinen rahoitus

Säilytyspaikka
Jyväskylä yliopiston kirjasto

FIGURES

TABLES

CONTENTS

ABBREVIATIONS

ANN = Artificial neural network
ATM = At the money
BSM = Black-Scholes-Merton
BSMDE = Black-Scholes-Merton differential equation
ITM = In-the-money
MAE = Mean Absolute Error
MAPE = Mean absolute percentage error
MCS = Monte Carlo simulation
MLP = Multilayer perceptron
MSE = Mean squared error
NTM = Near-the-money
OTC = Over-the-counter
OTM = Out-of-the-money
ReLU = Rectified linear unit
RMSE = Root mean squared error
TTM =Time-to-maturity

1 INTRODUCTION

Option pricing has long been a subject of extensive research and great interest. Because options, as well as other derivatives, are used for risk management as well as speculation and portfolio management the pricing of these financial instruments needs to be as accurate as possible. Over the last half a century different methods have been created in an attempt to make the pricing of options as accurate as possible, and these models have been ranging from statistical models to stochastic models derived from theory.

As computing power has become more affordable and more efficient it has enabled the use of more complex models, such as the artificial neural network (ANN) and a sub-method of that the multilayer perceptron (MLP). It used to be hard to run complex neural network models before, as they are heavy on computing power and thus, they must have been simple methods not containing too many layers, neurons, or iterations in the model.

This thesis aims to take advantage of the advancements in computing power and try to evaluate the performance of an ANN MLP model in option pricing and compare it to the Black-Scholes-Merton (BSM) model and the Monte Carlo simulation (MCS). The advancement in computing power is one of the driving motivators for this thesis. Other motivation lies in the improved risk management, that extends beyond financial institutions and mutual funds to traditional industries as well as individual economic agents. For instance, well-priced options can mitigate the price increase of a construction material or can fortify an investment portfolio against a big economic crisis. Many industries are already using derivatives in risk management, but an argument can be made that they would be utilized even more if the pricing was more accurate. The premise is that if the pricing was more accurate it would mitigate concerns related to overpaying for the hedging the options provide.

Some other motivation for this subject is also drawn from the ANN MLP model being a great practical finance tool for finance professionals, that does not only work in theory but also in practice. Continuous financial model improvements are crucial to keeping track of the constantly evolving finance world.

Artificial intelligence tools such as ChatGPT were used in making this master's thesis. These programs were used to aid in programming the ANN and managing the large dataset using Python. It seems that such programs are extremely useful in finding errors in the code and they were used especially for this. Artificial intelligence tools were not asked or commanded to write this thesis or construct ready to use Python code.

1.1 Background

Financial derivatives are financial instruments that derive their value from another asset. They often trade in the future, meaning that one can for example agree to buy a specific asset in six months at a specific price. The two main categories of financial derivatives are options and futures, where the options give the holder the option (not the obligation) to buy or sell the underlying asset at a specific time at a specific price. For the futures, this turns into an obligation. Derivatives like these are most often used for risk management, as well as speculating. The pricing of these derivatives is important so that the risks can be managed at a fair price.

Traditionally options have been priced with a Black-Scholes-Merton model. This model was made by Black and Scholes in 1973 and was continued in the same year by Merton. This model was the groundbreaking option pricing model at that time, as there weren't that reliable pricing models before that. However, options have been around for much longer than the BSM pricing model. The BSM model is a stochastic model that has a strong theoretical background. There have been several extensions to the BSM model, such as ones where the volatility used in the model is not considered constant or is derived in another way than in the original model. For example, John Cox and Stephen Ross have done a lot of work regarding the different BSM model extensions. These can already be considered as advancements in option pricing, but also a lot of other models have been composed.

Another stochastic model used in option pricing is the Monte Carlo simulation. The idea is to run many different price paths for the underlying asset and calculate the option price for each of these paths. An average of these option prices are taken which is supposed to then represent the option's real value. It is based on the law of large numbers, which means that if enough paths are run ($n \rightarrow \infty$) the average of the option payoffs should be the actual value.

The latest big advancement seems to be the machine learning models, mainly the ANN MLP models. These were first introduced by Malliaris and Salchenberger (1993a). The ANN models resemble standard regression models but in practice are built from brain-like neurons. Input variables are run through the neural network and its neurons minimizing an error function until it cannot be minimized anymore. The ANN models have been compared to different BSM models, and the results vary depending on the specific models used as well as depending on the dataset used. There seem to be a few connecting factors between the models that consistently outperform the BSM model, and those will be used in this thesis.

The research regarding the ANN models has mainly been observing its performance in European options. It seems like it should also be examined in American option pricing, as well as in exotic option pricing. Some literature on American option pricing with the ANN does exist but it is nowhere near as extensive as for the European options. There also doesn't seem to be that much literature regarding the different hidden layer architectures, that would prove a specific number of hidden neurons or hidden layers to be superior to others. It could also be argued that there exists a gap in the literature on comparing the ANN models to other option pricing models, and not only to the BSM model variations although some literature on this already exists (see for example Liu et al. (2019)).

1.2 Research questions

This thesis aims to find out how well the artificial neural network models compare to the Monte Carlo simulation method, and as a final benchmark compare these results to the Black-Scholes-Merton model. The main research question is:

- Is the artificial neural network better at pricing the DAX-index call options than the Monte Carlo simulation method, and the Black-Scholes-Merton option pricing model?

Additional research questions are as follows:

- What is the optimal number of hidden layers, hidden neurons, and epochs in the artificial neural network?
- Are the pricing performances of the artificial neural network, the Monte Carlo simulation, and the Black-Scholes-Merton model dependent on the moneyness of the option?

1.3 Structure

This thesis has been divided into eight different chapters, where the first one is the introduction to the topic. This chapter will contain the background, motivation, research questions, and research methods as well as the structure of the thesis.

The second chapter aims to explain financial options, with emphasis on the styles of options used in this thesis. This chapter won't be an in-depth analysis of options, but rather just an overview or a recap of them.

The third chapter discusses the stochastic models used in option pricing, especially the BSM model and the MCS. Before these, a mathematical foundation will be established which the methods will be built on top of. The derivation of the stochastic models requires a mathematical foundation.

Chapter four contains an introduction to machine learning, and especially to the model of ANN. Machine learning and the ANN are defined, and the ANN used in this thesis is specified.

The fifth chapter includes a literature review of the artificial neural networks in option pricing. This chapter will focus on the attributes of the models used and also on the results that have been found when using an ANN model in option pricing. This chapter will also contain a summary table of a sample of the literature done before.

In the sixth chapter an overview of the dataset it presented, and the time period used in this study is discussed. This chapter also presents the model used for the ANN and explains how the dataset is filtered.

The seventh chapter is the main focus of this study. It presents the results of the thesis and compares the different models to each other. Before this, an optimal structure for the artificial neural network is acquired. The models are compared to each other using the whole dataset and some subsets of it.

The eighth chapter concludes the thesis. This chapter discusses the findings and summarizes them as well as presents some ideas for upcoming research based on the same subject. The limitations of this study are also discussed.

2 FINANCIAL OPTIONS

In the following chapter, the reader is familiarized with different financial options. The chapter will not contain an in-depth analysis of option contracts or option types but rather a quick overview of them[1]. The primary reference for this chapter is Hull's (2013) book "Options, futures and other derivatives. It seems to be the most used book when it comes to either basic options and derivatives or more complex ones, and it is used as the course book for many universities' derivatives courses.

Black and Scholes (1973), and Hull (2013, p. 1) explain that financial options are financial instruments and derivatives that derive their value from another asset. They explain that they yield the holder the right (but not an obligation) to buy or sell the underlying asset at an earlier specified price, which is known as the strike price. They add that the underlying assets might be for example commodities, stocks, or indices. Options can be categorized in popular ways based on two key divisions: put vs. call options and European vs. American options, and on top of European and American options there also exists so-called exotic options(Black & Scholes, 1973; Hull, 2013, p. 574). Because of the existence of exotic options, the European and American options are occasionally referred to as vanilla options, Black and Scholes (1973), and Hull (2013, p. 574) add. Options can, according to them, be traded in exchanges or in over-the-counter markets (OTC) which refers to private parties agreeing on an option deal.

The main characteristics of an option are its type, the underlying asset, expiration date, and its strike price. The type is either a put option or a call option. A call (put) gives the holder the option to buy (sell) an asset by the expiration date for the strike price. Depending on if the option is European or American the option can be exercised only at the expiration date or on any day, and thus the European which can be exercised only on the expiration date is easier to analyze. (Hull, 2013, p. 7-8) Because the American option can also be

---

[1] If the reader wants to familiarize themself more with options Hull's (2013) book "Options futures and other derivatives", which is also cited in this chapter a lot, is a great place to start.

exercised on any other day, it gives the holder more possibilities, and thus the price of the option should be higher than its European counterpart. According to Hull (2013, p. 7) most of the options traded in exchanges are American[2].

Options are most often used for speculation purposes or to manage risks (hedge). Hull (2013, p. 11) explains hedging as trying to reduce a particular risk they have, which can be almost anything that causes fluctuations in the investment. It is important to note, that there should always be a price to pay for hedging.

Different risks of financial options can measured by so-called "Greeks" (Hull, 2013, p. 377). These refer to different Greek letters such as delta, theta, gamma vega, and rho. The only Greek relevant in this thesis is the delta in the context of delta hedging. Hull (2013, p. 264) explains the delta of an option to measure how sensitive the price of the option is to a change in the price of the underlying asset. If the delta of an option is 0.2 the option's price changes around 20 percent of the amount of the change in the underlying's price.

2.1 European option positions and payoff functions and profiles

Option contracts have two sides, where one side is the buyer(long), and one side is the seller(short). The seller (or writer) of the option has the obligation to sell (or buy) the underlying at the specified strike price if the holder of the option so wishes. There are four types of option positions, a long call, a long put, a short call, and a short put. This subchapter will review the attributes and payoff profiles of long and short calls and puts, and to make this overview simple only European options and their payoff profiles and payoff functions will be examined.

Let K be the strike price of the option, and S_T be the price of the underlying asset at time $t \in \{0, T\}$. Now the payoff function for a European long call option is

$$\max(S_T - K, 0) \tag{1}$$

The option has value if $S_T > K$ and no value if $S_T \leq K$. If it has no value, it should not be exercised as the market offers the asset at a lower or equal price when compared to the strike price K. Figure 1 demonstrates the payoff profile of a European long call option, which has a strike price of 100 and an initial option price of 7.

[2] American or European options as types do not refer to geographical locations, but rather just the type of the option. European options can be traded in American exchanges and vice versa.

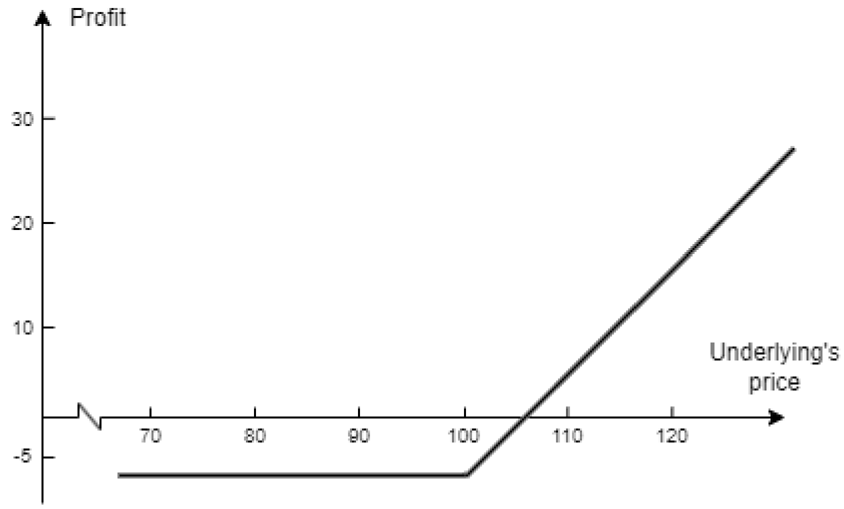

FIGURE 1 Payoff profile of a European long call option

The payoff function for a European long put option is

$$\max(K - S_T, 0) \tag{2}$$

And this time the option has value if $S_T < K$ and no value if $S_T \geq K$. The higher the strike is in this situation the higher the price the option allows the holder to sell the underlying. Figure 2 demonstrates the payoff profile of a European long put option with a strike price of 90 and an initial option price of 5.

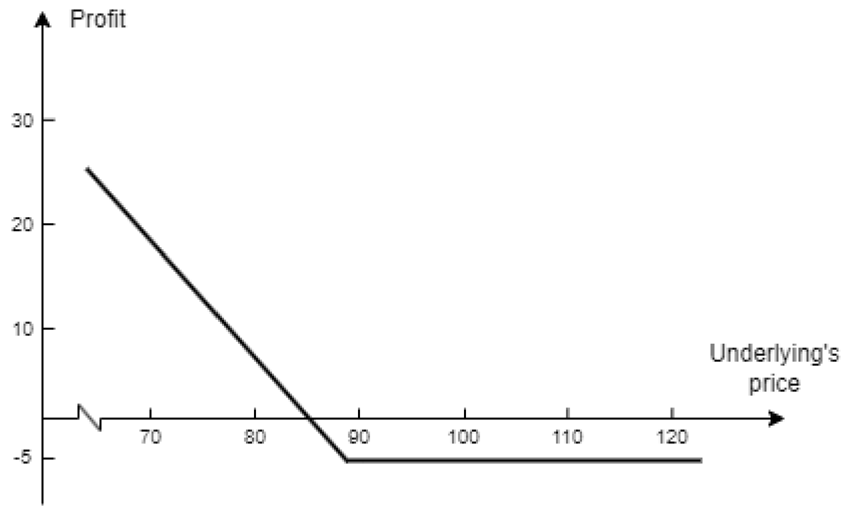

FIGURE 2 Payoff profile of a European long put option

Payoff for a European short call position is

$$\min(K - S_T, 0) \tag{3}$$

Which is demonstrated in Figure 3, with an initial option price of 5 and a strike price of 100.

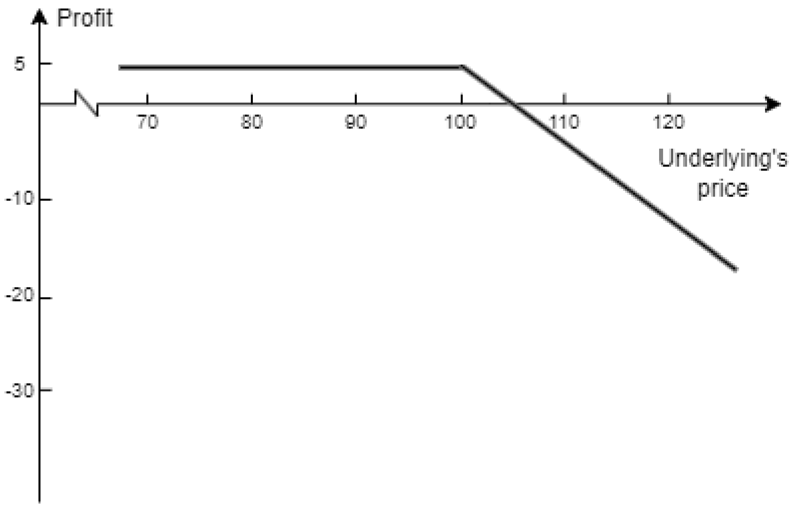

FIGURE 3 Payoff profile of a European short call option

And the payoff for a European short put is

$$\min(S_T - K, 0) \tag{4}$$

Which is demonstrated in Figure 4, with an initial option price of 5 and a strike price of 90.

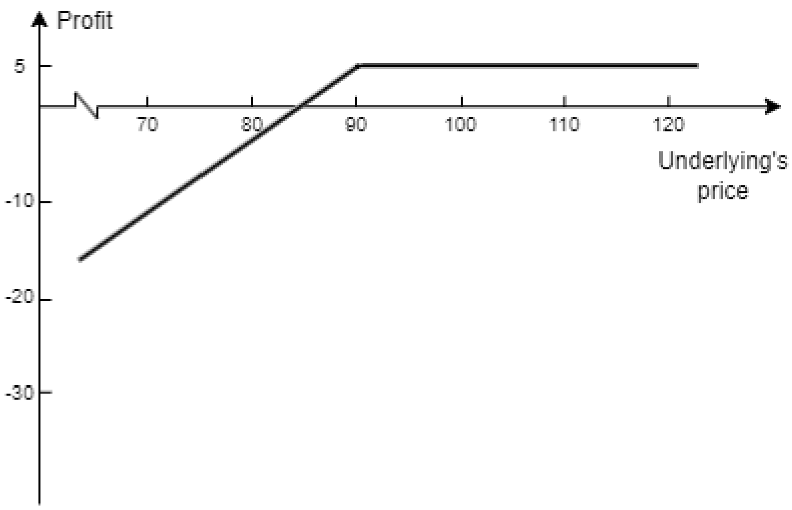

FIGURE 4 Payoff profile of a European short put option

2.2 Other options

On top of the vanilla options, there exists the so-called exotic options. They often do not contain easily definable properties and are not traded actively. Because of this, the exotics are usually OTC traded options. The market for these kinds of options is small, and they are often tailored to fit a very niche use case. That is why it is not often of interest to trade them publicly in the exchanges. Because the exotics are often even more complex than the vanilla options, their pricing is even more difficult. Below are a few examples of the exotics. These will be covered in short as they are not a part of this thesis. However, they are very interesting in the option pricing framework, as they are difficult to price with traditional methods. One of the more exciting future research areas could be exotic pricing with an ANN. The examples below are based on Hull (2013).

Packages are portfolios consisting of different assets. These are usually made to be similar to a hedging strategy, which is easily purchasable as a complete package. These strategies are for example butterfly spreads or bull and bear spreads.

Asian options' payoffs are dependent on the arithmetic average of the price of the underlying asset during the life of the option. Asian options might be more appropriate to some of the hedging needs than for example European options because they do indeed count for the average price of the whole period and are not only hedging against the price at the TTM.

The holder of a shout option can once during the options life shout at the writer. At maturity, the holder receives the payoff of the option as it was a usual European option, or the intrinsic value at the time of the shout, whichever of these is greater.

Compound options are options on options. A call on a call, a call on a put, a put on a call, or a put on a put are compound options. The two different options have their own exercise dates and two strike prices.

As the name exotic options might tell you, the contracts are often quite exotic and thus difficult to price or evaluate. Usually, these options do net the dealer a lot of money (Hull, 2013, p. 574). This is mainly because the dealer and the buyer both have a hard time pricing them and thus are not aware of the fair value of the option. The dealer must play it safe and ask for more risk premium because of the uncertainty.

2.3 Moneyness

An option can be classified with respect to the underlying's current price and the contract's strike price, and this classification is called moneyness. This classification helps in determining whether the option contract currently has intrinsic value or not. Hull (2013, p. 201) explains that option contracts are referred to as in-the-money (ITM), at-the-money (ATM) or out-of-the-money (OTM) depending on where the underlying's price is with respect to the strike price. ITM means that the option has at least some intrinsic value. He continues that with S being the underlying's price and with K being the strike price options are referred to as presented in Table 1. Figure 5 also demonstrates the different moneyness points in a payoff profile graph of a European call option.

TABLE 1 Moneyness of option contracts

Moneyness	**S<K**	**S=K**	**S>K**
Call option	OTM	ATM	ITM
Put option	ITM	ATM	OTM

Without transaction costs only an option that is ITM is exercised, and because of this moneyness and especially ITM is a signal that the option has value at that specific time. However, OTM and ATM options do have a price, because the underlying's price can always change before the expiration date. If there is at least a small chance that the underlying asset's price reaches the strike price before the expiration date the option should have some value and thus have a price. Options that are deep ITM or OTM are hard to price because the underlying's price changes would have to be so dramatic that a possibility like that is hard to calculate. Malliaris and Salchenberger (1993a) note that the BSM model's pricing bias is the largest for deep OTM options, and later in this thesis it is observed that the ANN also has trouble pricing deep ITM options. Moneyness is thus a crucial factor in option pricing. Deep ITM or deep OTM options are also not traded much (Anders et al., 1998)[3].

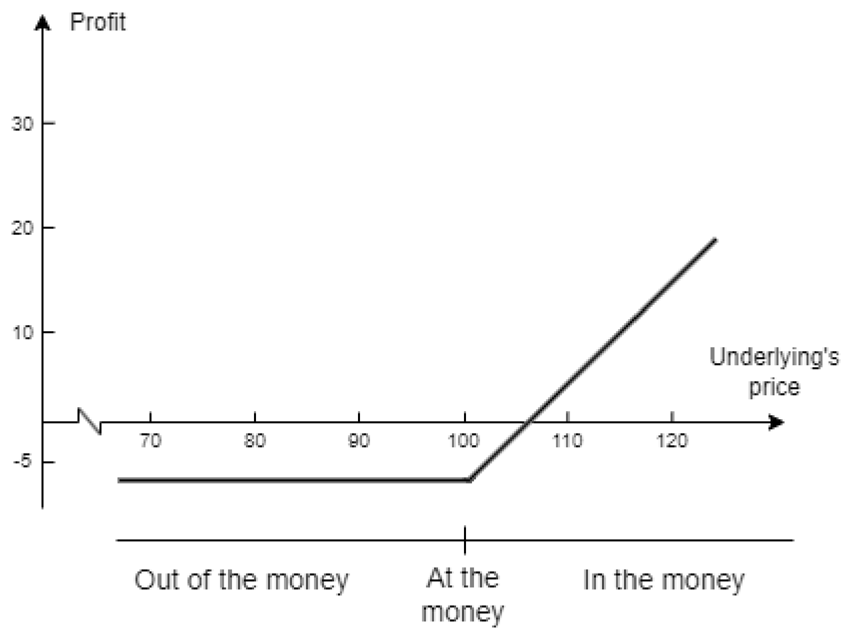

FIGURE 5 Moneyness of a European call option

[3] Deep ITM or deep OTM simply means that the underlying asset's current price is far from the strike price, so far from ATM.

3 STOCHASTIC MODELS IN OPTION PRICING

This chapter aims to provide a base-level understanding of stochastic models and their foundation, to then later on be able to connect this knowledge to the models in option pricing. First crucial parts of stochastic models will be discussed which are the Brownian motion which has been used in the modeling of security prices (Wang, 2012, p. 31), the Itô's lemma, and the law of large numbers. Later two stochastic option pricing models are examined: the traditional model by Black and Scholes (1973) and Merton (1973) and the Monte Carlo simulation which was first developed for physics by Stanislaw Ulam in the 1940s. The main idea of stochastic models such as the BSM and the MCS is to include the uncertainty factor into their structure so that it can be accounted for. Random variables are the uncertainty variables, that are incorporated into the models.

The assumption for this chapter is that the reader has at least a foundational understanding of probability concepts like random variables at an introductory level or higher. However, it is intended that the latter parts of this thesis can be read and understood without these, but this chapter might be hard to understand without.

3.1 The mathematical foundation

3.1.1 Brownian motion

The Brownian motion was first discovered by botanist Robert Brown in 1827 while studying pollen grains and their movements(Kac, 1947; Wang, 2012, p. 31). Later Norbert Wiener mathematically constructed the stochastic process of the Brownian motion[4] and Figure 6 presents two similar examples of the Brownian motion but with different time increments. Brownian motion has according to Hull (2013, p. 282) been used and developed for the use cases of

[4] Brownian motion and Wiener process as definitions are used interchangeably.

physics to model the motion of particles, and it functions in the same type of way as Markov chains do. Wang (2012, p. 31) explains that this mathematical process has also been used extensively in finance to model security price movements, and Bachelier (1900) was the pioneer in applying the Brownian motion to finance in his doctoral thesis. Security prices do seem to have uncertainty in them, so a probability model to demonstrate this random walk type of movement is essential.

Let's consider a sample space Ω, which is the collection of all possible outcomes. First, a stochastic process W = $\{W_t : t \geq 0\}$ will be defined as a collection of independent normally distributed random variables with mean 0 and variance 1, where index t is for time which linearly moves forward. For each $\omega \in \Omega$, the mapping $t \rightarrow W_t(\omega)$ is called a sample path. W is a Brownian motion if the following conditions hold

(i) W is continuous,
(ii) $W_0 = 0$,
(iii) The random variables $W_{t_1} - W_{t_0}, W_{t_2} - W_{t_1}, \dots, W_{t_n} - W_{t_{n-1}}$ are independent for any sequence $0 = t_0 \leq t_1 \leq \cdots \leq t_n$,
(iv) For any $s \geq 0 \; and \; t > 0$, the increment $W_{s+t} - W_S$ is normally distributed with a mean of 0 and variance t.

With mean 0 the expected value at any future time is the same as it is currently, and because the random variables are independent the motion "starts afresh" at any time t.

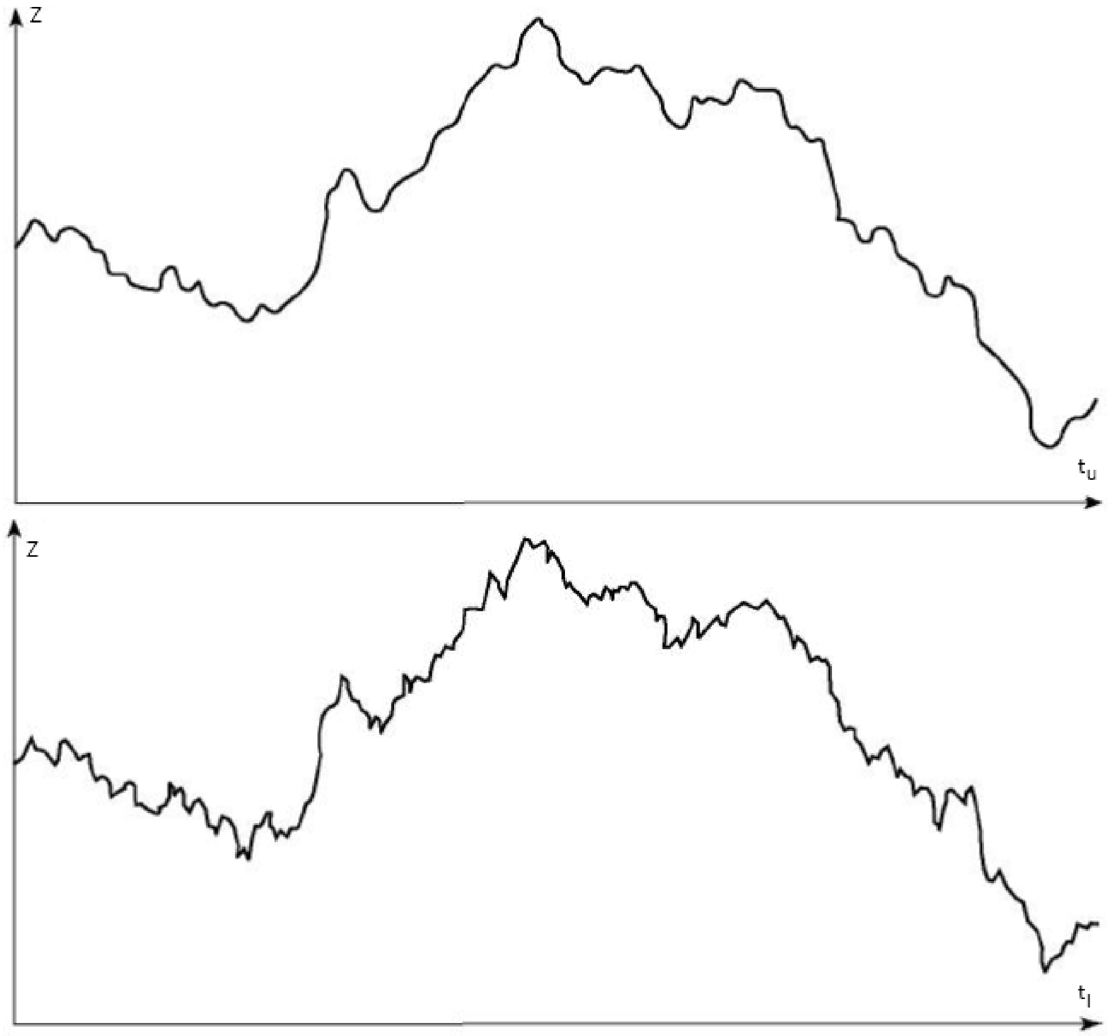

FIGURE 6 Examples of Brownian motion, with bigger time increments (upper graph) compared to smaller time increments (lower graph) Δt_u>Δt_l

To make this Brownian motion correspond better to reality it can be defined to have so-called drift. Hull (2013, p. 284) defines drift to be the mean change per unit time for a stochastic process. This would mean that the motion would have its random effects but in the long run, it would appear to have a general direction. When considering pricing models and their attributes, a risk-free rate is often one of the variables used in the models. Wang (2012, p. 36) notes this drift in the Brownian motion can be equated as the risk-free rate.

3.1.2 Itô's lemma

Itô's lemma is a much-needed result of mathematics, which will be needed when deriving the BSM model and the Monte Carlo simulation used in option pricing. The lemma was discovered by the mathematician Itô (1951), and it is based on the Brownian motion.

The next demonstration is given by Hull (2013, p. 291) concerning the lemma discovered by Itô (1951):

Suppose that the value of variable x follows the Itô process:

$$dx = a(x, t)dt + b(x, t)dz \tag{5}$$

Where the dz is the Brownian motion and a and b are functions of x and t. When a is the drift rate of variable x and b^2 is the variance rate of x, Itô's lemma shows that a function G of x and t follows the process:

$$dG = \left(\frac{\partial G}{\partial x}a + \frac{\partial G}{\partial t} + \frac{1}{2}\frac{\partial^2 G}{\partial x^2}b^2\right)dt + \frac{\partial G}{\partial x}b\,dz \tag{6}$$

Where dz is the Brownian motion. The proof can be found in Itô (1951), or as an appendix in Hull (2013, p. 297).

3.1.3 The law of large numbers

The law of large numbers is used as the base for the Monte Carlo simulation. Schinazi (2012, p. 103) and Wang (2012, p. 69) explain the strong law of large numbers to be as follows: with $X_1, X_2, \dots, X_n$ being a sequence of independent identically distributed random variables with mean μ and variance σ^2 there is

$$P\left(\lim_{n\to\infty}\frac{X_1 + X_2 + \cdots + X_n}{n} = \mu\right) = 1 \tag{7}$$

What this means is that as the sample size n approaches infinity the probability that $\bar{X} = \mu$ is 1. The proof of the law of large numbers can be found in Schinazi (2012).

3.2 Black-Scholes-Merton

The Black-Scholes-Merton model was first imposed by Black and Scholes (1973) for option pricing[5]. Black and Scholes started developing the model and later Merton (1973) continued and refined their work. It is considered one of the fundamental models for option pricing and is most likely the most used one. The Black-Scholes-Merton model will be referred to as BSM later in this thesis.

Let's define some variables used in the derivation of the BSM model:

μ = Expected return on stock per year
σ = Volatility of the stock price per year
r = Risk-free rate of interest

[5] Sometimes the model is referred to as just the Black-Scholes model.

t = Time, $t \in [0, T]$
S = Stock price
K = Delivery price
f = Price of the option
π = Value of the portfolio
$N(x)$ = The cumulative probability distribution function for a standardized normal distribution

To derive the BSM differential equation (BSMDE) the demonstration by Black and Scholes (1973) given by Hull (2013, p. 309-313) is followed.

Assumptions to derive the BSMDE:

1. The stock price follows the process of standard Brownian motion described in the earlier part with constant μ and σ.
2. Short selling is permitted.
3. No transaction costs or taxes. All securities are perfectly divisible.
4. No dividends.
5. No arbitrage opportunities.
6. Security trading is continuous.
7. r is constant.

Hull (2013, p. 309) notes that some of these assumptions can be relaxed. This yields some more complex models which enable more real definitions to be used. He adds that these complications can include, for example, the σ or r to be defined as stochastic processes themselves. This enables a more comprehensive consideration of real-life factors within the model, as the risk-free rate or the volatility is never constant over longer periods of time. However, as the maturities for option contracts are often quite short the risk-free rate or the volatility should not deviate too far from the reality when kept as constants, when compared to how complicated the model would become.

Next, the BSMDE will be derived on top of the assumptions and definitions given earlier. Let's assume a stock price process:

$$dS = \mu S dt + \sigma S dz \tag{8}$$

Where dz is a Brownian motion. With Itô's lemma, and with knowing that f is contingent on S the variable f must be some function of S and t. Hence:

$$df = \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right) dt + \frac{\partial f}{\partial S}\sigma S\, dz \tag{9}$$

The discrete versions of these two equations are

$$\Delta S = \mu S \Delta t + \sigma S \Delta z \tag{10}$$

And

$$\Delta f = \left(\frac{\partial f}{\partial S}\mu S + \frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)\Delta t + \frac{\partial f}{\partial S}\sigma S\,\Delta z \tag{11}$$

Where the Δ marks the changes in the corresponding variables. Now let's construct a portfolio (π) which consists of a derivative and the corresponding stock, to create a so-called delta-hedge portfolio[6]. By delta hedging the Brownian motion will be eliminated completely[7]. The portfolio is short one option ($-f$) and long the share an amount of $\frac{\partial f}{\partial S}$. The value of the portfolio can now be expressed as

$$\pi = -f + \frac{\partial f}{\partial S}S \tag{12}$$

And the change in the value of the portfolio is

$$\Delta\pi = -\Delta f + \frac{\partial f}{\partial S}\Delta S \tag{13}$$

We know Δf and ΔS from earlier, and if those are substituted to Equation 13, there is

$$\Delta\pi = \left(-\frac{\partial f}{\partial t} - \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)\Delta t \tag{14}$$

Because the Δz (and the μ) were removed from the portfolio, it is riskless during Δt. Because of the delta-hedge, there is no uncertainty in the portfolio and thus all the continuous risk is hedged. By combining the assumptions mentioned above, it is implied that the portfolio now always makes the same rate of return as other short-term risk-free securities. It can be written that

$$\Delta\pi = r\pi\,\Delta t \tag{15}$$

Now by again substituting the Equations 14 and 12 into this, there is

$$\left(\frac{\partial f}{\partial t} + \frac{1}{2}\frac{\partial^2 f}{\partial S^2}\sigma^2 S^2\right)\Delta t = r\left(f - \frac{\partial f}{\partial S}S\right)\Delta t \tag{16}$$

By reducing this equation there is

---

[6] Delta hedge is an options strategy which seeks to result in a neutral delta of the option.
[7] This portfolio would only stay delta-hedged if S and t would not change, because if they do then the share amount would also change. Obviously, S and t change constantly so constant weighing adjustments are required to keep the portfolio delta hedged.

$$\frac{\partial f}{\partial t} + rS\frac{\partial f}{\partial S} + \frac{1}{2}\frac{\partial f}{\partial S}\sigma^2 S^2 = rf \tag{17}$$

And this finally is the BSMDE. This equation has many solutions, and the most famous ones are for a call

$$c = S_0 N(d_1) - Ke^{-rT} N(d_2) \tag{18}$$

and for a put

$$p = Ke^{-rT} N(-d_2) - S_0 N(-d_1) \tag{19}$$

where

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} \tag{20}$$

$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \tag{21}$$

3.2.1 Alternative Black-Scholes-Merton models

There are many alternative BSM models, which usually have to relax some of the assumptions of the BSM. Malliaris & Salchenberger (1993a) list the major ones at that time to be the Cox and Ross pure jump model, Merton's mixed diffusion-jump model, Cox and Ross' constant elasticity of variance diffusion model, and Rubinstein's displaced diffusion process. There also exist several other alternative models that use different volatilities, such as a Generalized Autoregressive Conditional Heteroskedasticity (GARCH) volatility model, or different implied or stochastic volatility models (SVM). The Stochastic Alpha, Beta, Rho (SABR) Model by Hagan et al. (2002) is also one that is used a lot. The papers presented in the literature review part of this thesis often use different alternative BSM models as benchmarks.

Some exotic options can also be priced with alternative BSM models. There have for example been modifications done to the BSM to be able to value gap options.

3.3 Monte Carlo simulation

The Monte Carlo simulation (MCS) was first discovered by Stanislaw Ulam in 1940 for the use of nuclear physics. The project he was working on needed a code name. Because of the randomness included in the model Ulam named it after the Monte Carlo casino, where his uncle would go gambling. According to Hörfelt (2005), Boyle (1977) was the first one to use the MCS method in option pricing. Ross (2012, p. 41) and Wang (2012, p. 67) explain the core of the MCS to be in simulating a large number of random variables and calculating their function's average. They continue that this idea is based on the strong law of large numbers and is in its core an approach to approximating integrals.

The MCS in itself is explained by Ross (2012, p. 40-41) as follows. Let $g(x)$ be a function and let θ be the desired value for computation where

$$\theta = \int_0^1 g(x)dx \tag{22}$$

With X being uniformly distributed over (0,1), then θ can be expressed as

$$\theta = E[g(X)] \tag{23}$$

With $X_1, X_2, \dots, X_n$ independent uniform (0,1) random variables it follows that the random variables $g(X_1), \dots, g(X_n)$ are also independent and also identically distributed with mean μ. Now using the strong law of large numbers mentioned above it follows that with probability 1,

$$\sum_{i=1}^{n} \frac{g(X_i)}{n} \rightarrow E[g(X)] = \theta \; as \, n \rightarrow \infty \tag{24}$$

Ross (2012, p. 41) continues that this implies that by generating a large number of random numbers and taking the average of the function of the random numbers an integral can be estimated. This estimation is called the Monte Carlo simulation. Wang (2012, p. 67) explains this MCS scheme in short to be simply divided into two steps. With Equation 23 those steps would be

1. Generate independent identically distributed random variables $X_1, X_2, \dots, X_n$, that have the same distribution as X.
2. The estimate of the expected value θ is defined to be the sample average $\hat{\theta} = \frac{1}{n}[h(X_1) + h(X_2) + \cdots + h(X_n)]$

It should be noted that because $n \rightarrow \infty$ in practice means that a lot of random numbers have to be generated, this simulation is computationally intensive, and might require a lot of computing power and computing time. As can be seen with the MCS in option pricing when generating paths for the underlying the simulation will become even more time-consuming computationally.

In the option pricing framework Hull (2013, p. 446-448) as well as Boyle et al. (1997) give a good general demonstration of the MCS, which builds on top of the general MCS explained earlier. He explains the steps to the valuation of a derivative dependent on a market variable S (the underlying asset) that provides a payoff at time T when the interest rates are constant as:

1. Generate a random path for the price of the underlying S.
2. Calculate the payoff of the option.
3. Repeat the generation and calculation steps many times, to get many sample values from the option.
4. Calculate the mean of the sample payoffs to get the estimate of the expected payoff.
5. Discount the expected payoff at the risk-free rate to get the current value of the option.

This all has to happen in a risk-neutral world. Hull (2013, p. 447) continues his demonstration of the MCS in option pricing with familiar equations. Let's suppose that the process followed by the underlying assets price process to

$$dS = \mu S dt + \sigma S dz \tag{25}$$

Where dz is once again the Brownian motion, μ is the expected return in a risk-neutral world, and σ is the volatility. Now the goal is to simulate the paths followed by the underlying asset S. The life of the option can be divided into N short intervals of length Δt and approximate the Equation 25 as

$$S(t + \Delta t) - S(t) = \mu S(t)\Delta t + \sigma S(t)\epsilon\sqrt{\Delta t} \tag{26}$$

Where $S(t)$ is the value of S at time t, ϵ is a random sample from a normal distribution with mean zero and standard deviation of 1. Here it is possible for the value of S at time Δt to be calculated with the initial value of S, and the value of S at time $2\Delta t$ with the value at Δt, and so on. Hull (2013, p. 448) adds that working with ln S is in practice more accurate than with just S, and with Itô's lemma from earlier the process followed by ln S is

$$d \ln S = \left(\mu - \frac{\sigma^2}{2}\right) dt + \sigma dz \tag{27}$$

So with a few steps, it comes to

$$S(t + \Delta t) = S(t) exp\left[\left(\mu - \frac{\sigma^2}{2}\right)\Delta t \, + \sigma\epsilon\sqrt{\Delta t}\right] \tag{28}$$

This equation is used to construct a path for the underlying asset S. This equation can be simplified if μ and σ are constants.

Hull (2013, p. 448) explains that the limitations of the MCS are the earlier mentioned computational intensity and also the simulation's limitations in handling exercise opportunities. American options, which can be exercised not only at the end of the maturity but also before, are thus hard for the MCS to handle. However, Boyle et al. (1997) point out that there have been several papers about using the MCS also in American option pricing, such as Tilley (1993), Broadie and Glasserman (1997), and Barraquand and Martineau (1995). After that papers have come out on the same subject and for example Longstaff and Schwartz (2001) should be considered if the subject is of interest. These show that the MCS can be flexed also to American option pricing.

4 MACHINE LEARNING

The world of data-driven technology and models has in the last decades seen a remarkable evolution. There has been some transition from traditional statistical handcrafted models to more automated machine learning models, which can automatically learn and adapt to the data it is given. The opportunities that machine learning provides are still not completely clarified, and the optimal models or optimal model specifications have not yet been researched enough. In the future it seems like machine learning should be a part of every economist's or data analyst's toolbox and should at least be used as supporting models in addition to the traditional models.

This chapter's goal is to provide a good understanding of machine learning, its attributes, discuss the artificial neural network used later in this thesis more in depth, and explain how the ANN is used in option pricing. First machine learning will be defined, as it is often a difficult concept to understand. Machine learning has been around for longer than what people seem to give it credit for, but it has experienced a lot more attention lately mainly because of more computing power. It used to be difficult to run many layered data-intensive models as they would take older CPUs and GPUs a lot of time to run.

4.1 Defining machine learning

In economics and finance, machine learning reforms the way global trends and the markets are analyzed or how the economy is predicted. However, machine learning as a concept might be hard to grasp. Alpaydin (2020) explains that machine learning is mainly just pattern recognition (Nyholm, 2022). There are several pattern recognition processes that people do unconsciously, such as recognizing people's faces or driving cars, and it is difficult to define the algorithm that is used while doing these things. However, Alpaydin (2020, p. 3) explains that a machine can learn to do these things by analyzing sample data about earlier similar situations and then applying what it has learned to reality

(Nyholm, 2022). For example, it is possible to collect driving data for the machine and then let it learn from all of the earlier situations. The data has to be labeled to indicate for the machine to know which situations were handled right and which situations were handled wrong. Through this kind of teaching the machine might find instances where the driver consistently failed to stop at a stop sign were consistently classified as incorrect.

For anyone at least slightly concerned with statistics, this description should immediately bring to mind some of the more traditional statistical approaches. Bi et al. (2019) notes this and describe that there seems to be no clear boundary between the statistical approaches and pattern recognition machine learning. The difference between them is especially hard to determine in regression instances. Because statistics is used in combination with almost every other science, machine learning can also be applied to almost anywhere. It will find the patterns in the data in biology or finance.

Machine learning might bring to mind artificial intelligence and even ChatGPT, which has garnered significant attention lately. Artificial intelligence is considered a hypernym for machine learning, and Varian (2014) explains that hyponyms for machine learning include some of the methods such as neural networks or even deep learning (Nyholm, 2022). Deep learning is the method that is being used as the base of ChatGPT.

4.2 Artificial neural networks

This subchapter is based on Nyholm's (2022) chapter on neural networks. Rojas (1996) explains the research about artificial neural networks to have started in 1943 when Warren McCulloch and Walter Pitts presented the first model of artificial neurons. He continues that neural networks were originally meant to picture how human brains processed information. Neural networks used as a machine learning method often go by the name of artificial neural networks, as they are artificially constructed to replicate a cluster of brain neurons. Stergiou and Siganos (2006) explain that like people, neural networks learn best from examples, as the biological systems' learning also includes fitting to the synaptic connections between the neurons in the network. Since the research of artificial neurons started, neural networks have been applied to several problems in pattern recognition and are being used as a method of machine learning today. Training neural networks requires a lot of experience and experimenting and thus some people prefer something simpler, such as the decision tree, explains Nilsson (1998). Neural networks are also widely in use in nonlinear approximation and when neural networks are applied to practice, they can be somewhat of a black-box model, as it is given a set of inputs and learns from them and gives us an output (Bi et al., 2019; Butts et al., 2003). It seems like neural networks are a complex thing to understand as their "hidden" mid-layers between the inputs and outputs aren't visible to the naked eye, and thus what the network actually does cannot be observed.

A neural network consists of a big population of neurons interconnected through complex signaling pathways and this structure is used to analyze complex interactions between measurable covariates to eventually predict an outcome (Bi et al., 2019; Vähäkainu & Neittaanmäki, 2018). Nevasalmi (2020), as well as Vähäkainu and Neittaanmäki (2018), explain that neural networks possess different layers of these neurons, which are the input layer, the hidden layer or layers, and eventually the output layer. They continue that there can be more hidden layers than just one, making the network a more complex one. Every one of these hidden units is a linear combination of the input variables which have a link connecting them (Nevasalmi, 2020). Bi et al. (2019), Butts et al. (2003), Lai (2014), Patil and Subbaraman (2021), and Vähäkainu and Neittaanmäki (2018) state that there is a weighting system inside the network, to weigh different inputs or neurons differently. A weighting is used to define to the neural network how important a specific module of it is, whether a link or an input. The weights are determined based on how the errors can be reduced the most (Patil & Subbaraman, 2021).

Figure 7 is a demonstration of a simple neural network with an input layer, two hidden layers, and an output layer. The neurons are portrayed by the black circles, which are connected by links. It can be seen here how the hidden neurons are linear combinations of all the input values, as the links lead from each input to every hidden neuron in the first layer.

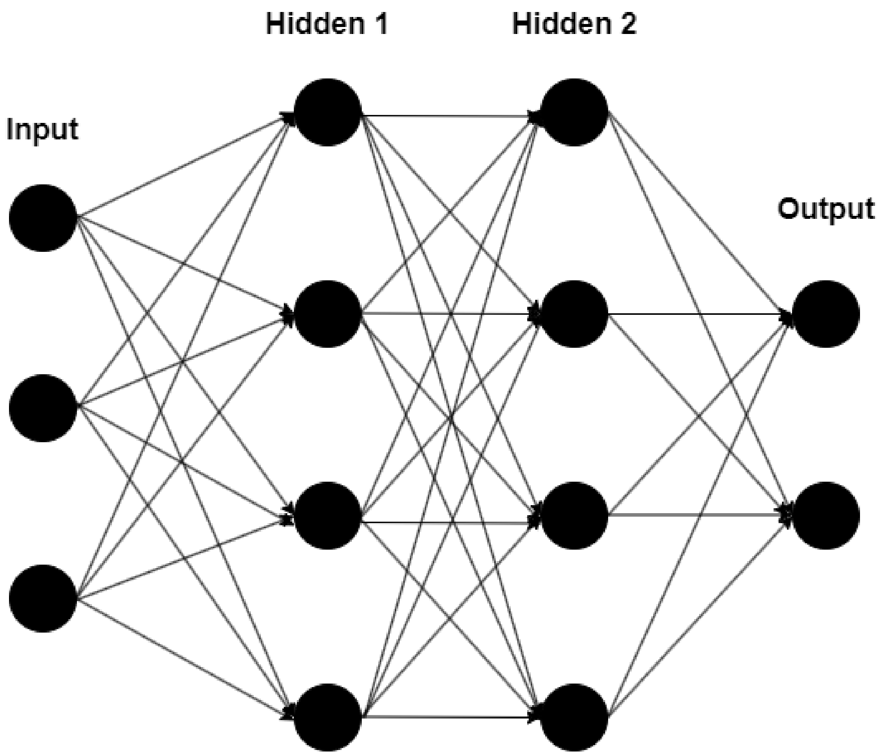

FIGURE 7 A neural network with four layers, where the two hidden layers contain 4 hidden neurons each

Neural networks or artificial neural networks -method is probably the one used most in research. Bi et al. (2019) define neural networks' strengths to be especially in how they accommodate to variable interactions and nonlinear associations without user specification. Citing Nyholm (2022) ends here.

As information flows through the network, an activation function is applied to the output after each neuron in the hidden and output layers (Tuominen & Neittaanmäki, 2019, p. 35). Tuominen and Neittaanmäki (2019, p. 35) continue that the activation function changes the output of the neuron from linear to nonlinear. Below is the ReLU presented as an equation in Equation 29 and as a Figure in Figure 8. The main idea in ReLU is that if the output of the neuron is negative ReLU outputs a negative number and if the value is positive, it outputs the value itself.

As the ReLU outputs zero when the value given by the neuron is negative it might cause some neurons' weights to be adjusted to zero. This problem is referred to as the specific neurons "dying", as a 0 weight does not allow the neuron to contribute anything to the network anymore. This problem can for example be addressed by using a ReLU variant a leaky ReLU. For more information about this activation function see for example Tuominen and Neittaanmäki (2019, p. 38-39).

$$f(x) = \max(0, x) \tag{29}$$

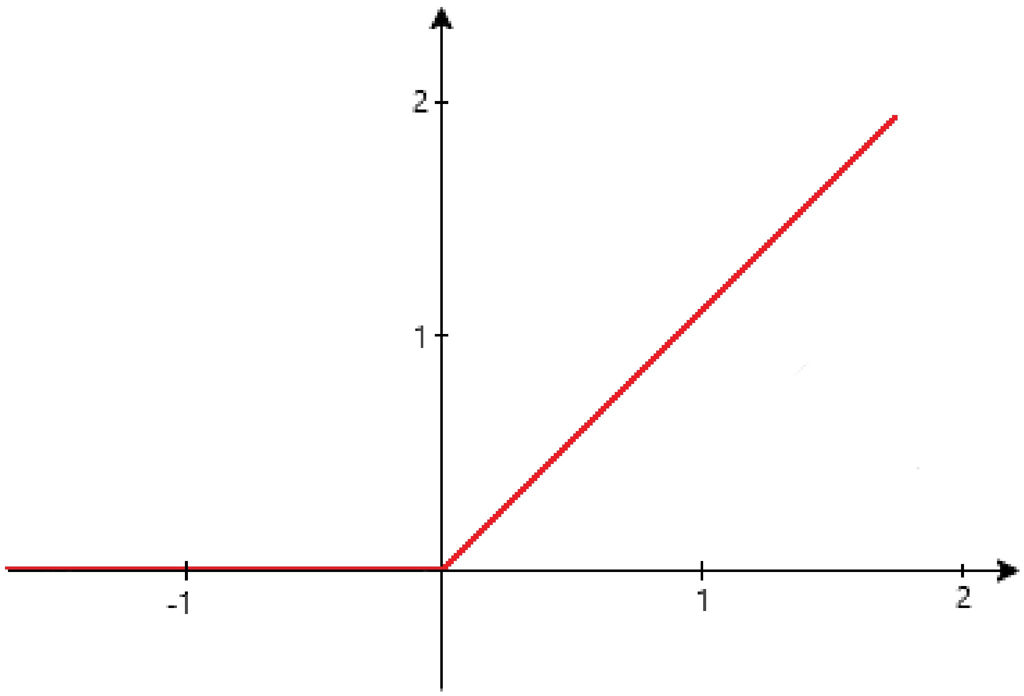

FIGURE 8 A rectified linear unit

The number of times data is passed through the network is called epochs (Lindholm et al., 2022, p. 125). Each time the data passes the weights of the neurons are adjusted to better fit the data according to the error function. The amount that data is passed through the network can be adjusted in the code, but increasing it is always time-consuming and computing power-consuming. This thesis also includes epoch comparisons, but the assumption already is that the higher the epochs more accurate the prediction.

ANN regression models can mainly be divided into two categories, the multilayer perceptrons (MLP)[8] and to radial basis functions (RBF) (Anders et al., 1998). Anders et al. (1998) explain that both of these are suitable for option pricing. However, for many (Amilon, 2003; Anders et al., 1998; Bennell & Sutcliffe, 2004; Garcia & Gençay, 2000; Jang & Lee, 2019; Liu et al., 2019; Ormoneit, 1999; Zapart, 2002) the MLP emerges as the preferred approach. The paper by Hornik et al. (1989) is considered an accurate demonstration of the MLP. Hornik et al. (1989) also go deep into the definitions and theorems of the MLP, but Amilon (2003) simply shows the basic functionality of the MLP. Figure 9 demonstrates an MLP network, and it has all of the notations of this text included in it. Amilon (2003) explains that each neuron sums the signals leading to it, adds a bias term, and makes a non-linear transformation $\left(g(\cdot)\right)$. The transfer function (or the activation function) is traditionally a smooth monotonically increasing function (Amilon, 2003). Amilon (2003) continues that the transformed signal of the neuron is then passed on to subsequent layers' neurons, and the process is again repeated, and that the connections between neurons are represented by the weights $\left(\omega_{ij}, W_{jk}\right)$. He shows that when the MLP is presented to the input vectors the inputs (x_k) are fed through the hidden layer(s) (h_j) all the way to the output layer via the neurons. Eventually, the network outputs

$$o_i = g\left(\sum_j \omega_{ij} g\left(\sum_k W_{jk} x_k\right)\right) \tag{30}$$

are compared to known targets (t_i) according to an error or loss function, which is usually the sum of squared errors

$$E = \frac{1}{2}\sum_i (t_i - o_i)^2 \tag{31}$$

is computed. Amilon (2003) explains the error is then propagated through the network backward and then the weights are adjusted to minimize the error function. The weights are adjusted, and the errors fed through the network as many times as the error function can still be minimized again.

[8] Some papers also refer to the MLP as a feedforward neural network.

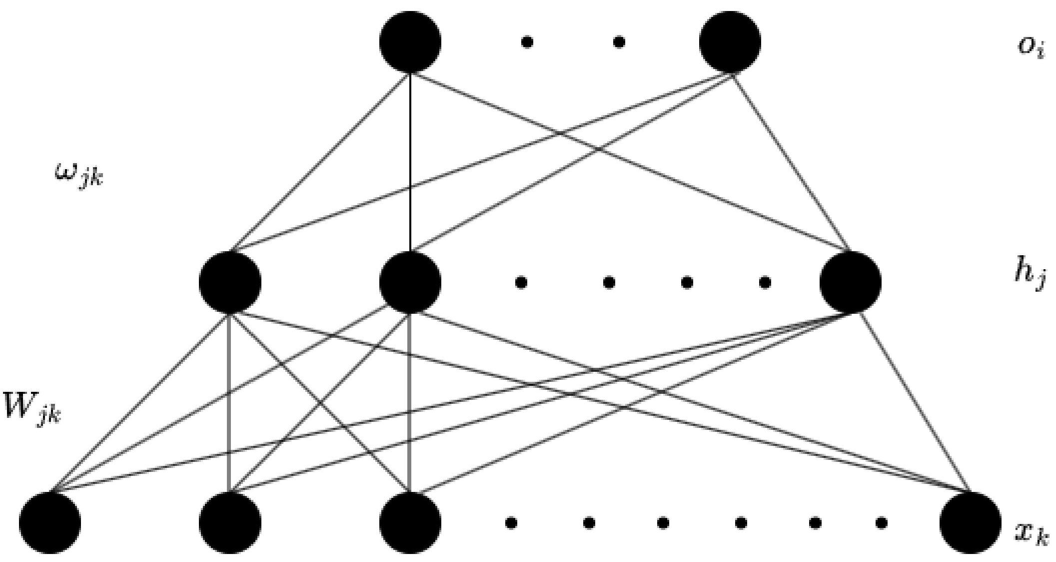

FIGURE 9 A single hidden layer ANN MLP architecture with several outputs, where the notations of the text are included in the picture

4.2.1 Overfitting

Overfitting is a problem that often also occurs while using basic statistical methods. As a model keeps fitting to data, it is important to test its performance on data outside of the training data set. If a model is fitted to training data too closely, it might not be able to generalize what it has learned to data outside of the training set. This problem is called overfitting (Lindholm et al., 2022, p. 18-19; Tuominen & Neittaanmäki, 2019, p. 52). Tuominen and Neittaanmäki (2019, p. 52) continue that the model learns the data "too well" as it adjusts the parameters according to specifications and also according to distortions of the training data. They note that overfitting is a common problem in networks that contain a lot of parameters, but the training data set is not big enough in relation to the size of the network. Malliaris and Salchenberger (1993a) also note that too many nodes in the middle layer (when using one hidden layer) lead to the model overfitting to the data. One might conclude that as the dataset size increases, the maximum size of the network that can be constructed also increases. According to Tuominen and Neittaanmäki (2019, p. 52-54) overfitting when using an ANN can be prevented using the following methods:

1. Increasing the size of the training dataset
2. Decreasing the size of the network
3. Stopping the learning process early
4. Partly dropping neurons from the network
5. Decreasing specific network weights

In practice, it might be easiest to decrease the number of layers or neurons in the network or decrease the number of epochs that the network runs. There also exist several ways for early stopping in ANN programming.

5 LITERATURE REVIEW OF NEURAL NETWORKS IN OPTION PRICING

This chapter aims to review the earlier literature done about option pricing with the neural networks and the Monte Carlo simulation and how these compare to other methods. For machine learning, most of the literature seems to revolve around artificial neural networks, but some also use gradient booster methods which will not be covered in this thesis. These machine learning models are usually compared to the BSM model and also to some volatility-enhanced BSM models such as one used with GARCH volatility or with a volatility index such as VIX. Some of the papers even focused on using different volatility estimates in the models, to see which volatility estimate performed the best. No papers came to the writers attention where the Monte Carlo simulation was used as a comparison method for the machine learning methods. However, the Monte Carlo simulation method has been used in the ANN and option pricing context, but for training the ANN (see for example Freitas et al. (2000)).

This branch of literature in option pricing seems to have been started by Malliaris and Salchenberger in 1993 with their two publishes, and a year after them with Hutchinson et al. (1994). A great review of the earlier literature has been constructed by Ruf and Wang (2020) which includes around 150 papers about option pricing using ANN before 2020. Their review includes a table that has the main papers in option pricing with ANN listed with different attributes of the research, and the table is of very good use when summarizing the literature done on the subject. A sample of this table can be found in part 5.1.5.

5.1 Artificial neural networks in option pricing

5.1.1 Inputs and features

There are a lot of possible parameters that can be used to try and price options, but the most common ones used seem to be the strike price, the underlying's price, volatility in one form or another, time to maturity, and the risk-free interest rate. These are also the inputs used in the baseline BSM model. On top of these, some papers use for example lagged parameters (Amilon, 2003; Malliaris & Salchenberger, 1993a; Zapart, 2002) or logarithmic variables (Buehler et al., 2019). There is also a lot of variation in which volatility estimate should be used, and these will be covered in the next subchapter.

The strike price and the underlying's price are the factors that show up almost always when trying to find a price for an option, and to no surprise. They are the variables that define the nature of the option, and if it has any value. When used in regressions, these are the factors that define most of the price variation in the option's price. When using an ANN for the pricing of the option, the strike price and the underlying's price show up in different formations, usually with the underlying's price divided by the strike price. Bennell and Sutcliffe (2004) note that the goal of using the S/Kvariable is for the moneyness in the variable itself to be accounted for, and thus the network wishes to learn to handle options whether ITM or OTM. However, later it can be seen that even when using the S/K and thus accounting for moneyness the network still struggles pricing deep ITM or OTM options. Anders et al. (1998), Bennell and Sutcliffe (2004), as well as Garcia and Gençay (2000), agree that the addition of individual variables of the strike price or the underlying's price on top of the S/K does not add enough value for them to be used. It seems that the underlying's price divided with the strike price is the most efficient and the most accurate way of including the two variables.

On top of the underlying's price and the strike price, the risk-free rate and the time to maturity seem to be included in a lot of networks. These two variables are quite self-explanatory, and there is not much to go over with them. The time to maturity should obviously be included, as the closer the maturity of the option is, the more the option's price should converge to its intrinsic value. The risk-free rate is important, and with it comes a decision of which rate to use. Amilon (2003) notes that if the risk-free rate is only considered as a discount factor, as it is the BSM model, the relevant input should be the risk-free rate times the time to maturity, as the time-to-maturity of a treasury bill is measured in calendar days. For the risk-free rate variable's data Amilon (2003) uses a continuously compounded 3-month treasury bill rate, Bennell and Sutcliffe (2004) follow this and also use the 3-month treasury bill rate whereas Liu et al. (2019) simply use the Euro LIBOR rate and Lai et al. (2014) use the 1-month Euribor rate.

5.1.1.1 Volatility estimates

Volatility reflects a lot of the uncertainty in the underlying assets' future price movements, and when trying to price options it is a crucial component. Many of the papers included in Ruf and Wang (2020) include a volatility estimate in the model, and almost all option pricing models usually include volatility in one form or another. However, volatility is not unambiguous, and the volatility of an asset can be expressed as several different volatility estimates. The models included in Table 3 include implied volatility, historical volatility, a volatility index such as VIX or volatility from calibration.

When comparing different ANN models to the BSM models, some papers also included some volatility comparisons. Amilon (2003) tested his ANN and BSM models with both historical price volatility and implied volatility. The historical volatilities tested were 30-day and 10-day standard deviations of the most recent continuously compounded daily returns of the underlying asset, in this case, the OMX index. Amilon (2003) found out that in pricing the options with the ANN including implied volatility it performed the best, but when trying to trade and hedge the mispriced options with the networks the model with historical volatility outperformed the model with implied volatility. Anders et al. (1998) also compared different volatility estimates, but this time the tested ones were historical price volatility for the last 30 days compared to a volatility index, in this case of pricing DAX options, the VDAX. According to Anders et al. (1998), the VDAX is simply a weighted average of volatilities implied by the different options traded at the Deutsche Börse AG. They found out that in in-sample pricing the historical volatility model performed about as well as the VDAX model, but it is important to note that these models compared were different also in other ways than just the volatility estimates. For out-of-sample pricing, the VDAX model clearly outperformed the one with historical volatility.

5.1.2 Outputs

The model outputs also vary between the different models, but obviously, the most common one used is just the options price (Liang et al., 2009; Malliaris & Salchenberger, 1993a; Teddy et al., 2008). Bennell and Sutcliffe (2004) state that by comparing the different models with individual option prices to the variable where the option price is divided by the strike price (C/K) it is evident that the C/K is one of the key components that help the ANN outperform the BSM models. The C/K has been used in a lot of the literature as well (Anders et al., 1998; Bennell & Sutcliffe, 2004; Garcia & Gençay, 2000; Hutchinson et al., 1994; Ormoneit, 1999).

5.1.3 Performance measures

The most common statistical performance measures in the literature are mean absolute error (MAE), mean absolute percentage error (MAPE) and mean squared error (MSE), which is in its core similar to the root mean squared error (RMSE) (Ruf & Wang, 2020). There are also other performance measures used in the papers. The most used performance measures are defined as follows:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}\left|C_t - \widehat{C}_t\right| \tag{32}$$

$$MAPE = \frac{\frac{1}{T}\sum_{t=1}^{T}\left|C_t - \widehat{C_t}\right|}{|C_t|} \quad (33)$$

$$MSE = \frac{1}{T}\sum_{t=1}^{T}\left(C_t - \widehat{C_t}\right)^2 \quad (34)$$

$$R^2 = 1 - \frac{\sum_{t=1}^{T}\left(C_t - \widehat{C_t}\right)^2}{\sum_{t=1}^{T}(C_t - \overline{C_t})^2} \quad (35)$$

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(C_t - \widehat{C_t}\right)^2} \quad (36)$$

Where C_t is the actual option price, $\widehat{C_t}$ is the price predicted by the model, $\overline{C_t}$ is the mean of the actual option prices and T is the quantity of the observations.

5.1.4 Network architecture

For the number of hidden layers and hidden neurons, Liu et al. (2019) and Bengio (2009) note that a single hidden layer is enough for mapping the inputs with the target function, and the existing literature, according to them, has reached a consensus that a single hidden layer is sufficient to make MLP a universal approximator for most problems. All of the other papers observed here have also ended up using only one hidden layer. However, for the number of hidden neurons, there seems to be some variation. Table 2 presents the different architectures regarding the hidden layers and neurons.

TABLE 2 A summary of the architecture of the networks from different studies

Authors & Year	*Hidden layers*	*Hidden neurons*
Malliaris & Salchenberger (1993a,b)	1	4
Hutchinson et al. (1994)	1	4
Anders et al. (1996)	1	3
Amilon (2003)	1	10-14
Bennell & Sutcliffe (2004)	1	3-5
Teddy et al. (2008)	1	8
Liu et al. (2019)	1	5-10

No papers other than Bennell & Sutcliffe (2004) stated the number of epochs they used in their network. Bennell & Sutcliffe (2004) used 1000 epochs, which seems to be a good starting number for the data iterations.

5.1.5 Summary

As mentioned at the beginning of this chapter Ruf and Wang (2020) have constructed a useful table with the different papers about option pricing with machine learning. This table includes the authors and year, features of the model with the input parameters, the outputs, benchmark models, performance measures, the partition method as well as the underlying asset. The table has been used in research about ANN in option pricing before (see for example Pohjonen (2022)). Table 3 is constructed from a sample of the comprehensive list by Ruf and Wang (2020) where the most impactful papers were included according to their citations or the publishing journal, as well as the ones with the most similarities or applications to this paper. In addition to the original most important columns, a column that presents the results can also be found in Table 3. For the abbreviations used in Table 3 please consult Appendix 1 for clarification.

TABLE 3 A summary of a sample of the literature on option pricing using artificial neural networks based on Ruf and Wang (2020).

Authors & Year	*Input parameters*	*Outputs*	*Underlying*	*Results*
Malliaris & Salchenberger (1993a,b)	$S, K, \tau, \sigma_{IM}, r$, lagged C and S	C	S&P100. 6M	ANN is more accurate OTM, whereas BSM is more accurate ITM
Hutchinson et al. (1994)	$S/K, \tau$	C/K	Simulation (BS); S&P500. 5Y	ANN is more accurate than BSM
Anders et al. (1998)	$S/K, S, \tau, \sigma_H, \sigma_V, r$	C/K, $(C - C_{BS-V})/K$	DAX. 3Y	ANN outperforms BSM in-sample and out-of-sample
Ormoneit (1999)	S/K	C/K	DAX. 9M	ANN performs similarly to BSM
Garcia & Gençay (2000)	$S/K, \tau$	C/K	Simulation (BS); S&P500. 8Y	ANN has smaller delta hedging

				errors than BSM
Zapart (2002)	Lagged wavelet coefficients	Wavelet coefficients	Individual stocks. 6M/1Y	ANN is as good and often better than BSM
Amilon (2003)	$S/K, \tau,$ $\sigma_H, r,$ lagged S	$C_{Ask}/K,$ C_{Bid}/K	OMX. 2Y	ANN is more accurate than BSM with historical volatility, ANN is as accurate as BSM with implied volatility
Bennell & Sutcliffe (2004)	$S, K, S/K, \tau, \sigma_{IM},$ open interest, volume	$C, C/K$	FTSE100. 1Y	ANN is more accurate than BSM OTM, BSM is more accurate than ANN ITM
Teddy et al. (2008)	$S - K, \tau, \sigma_H$	C	GBP-USD. 9M	ANN is more accurate than BSM, and also more accurate than any model in generalization pricing
Liang et al. (2009)	$\hat{C}$	C	Individual stocks. 2Y	ANN MLP is more accurate than conventional methods
Lai (2014)	$S/K, \tau, r$	σ	Simulation (BS, SV, SVJ)	N/A
Buehler et al. (2019)	log(S)	HR	Simulation (BS, SV); S&P500. 5Y	N/A
Jang & Lee (2019)	?	C	S&P100. 9Y	Bayesian neural networks perform better than

				classical option models in American option pricing
Liu et al. (2019)	$S/K, \tau, \sigma_{Cal}, r$	$(C - C_{BS-H})/K$	DAX. 4Y	ANN was the most accurate of many models overall

This study's main goal is to find out whether the ANN models price DAX options better than the BSM models or the MCS models. As there is no literature for the comparison of ANN models to the MCS, only the BSM models were used as a comparison. The MCS and the BSM function similarly and they are expected to yield similar results. In the literature about ANN in option pricing, it seems evident that artificial neural networks do seem useful in option pricing when compared to BSM models. Researchers tested the models in different situations, such as pricing ITM options vs OTM options or pricing in-sample vs out-of-sample and this part concludes the findings. The consensus seems to be that ANN models outperform the BSM models in at least one part of the testing, if not more when used right.

Liu et al. (2019) compared several different models, and one of them was the ANN, in out-of-sample price forecasting situations and in hedging. The models they tested were a wavelet-based option pricing model by Ma (2011), a stochastic volatility model with jumps by Bakshi et al. (1997), the practitioners BSM model by Anderou et al. (2014) and Dumas et al. (2002) and finally a hybrid ANN model by Anderou et al. (2008). They observed that from all of these models, the ANN model performed the best in most of the situations they tested, meaning that the ANN achieved the smallest forecasting errors. However, they did limit their sample to only OTM options and also noted that the ANN has limitations regarding data, as it is a very data-intensive model. Anders et al. (1998) compared the ANN to the BSM model both in-sample and out-of-sample. In their paper, they show similar results to those by Liu et al. (2019) in that the ANN they constructed was more accurate according to all of their performance measures out-of-sample and in Anders et al. (1998) case also for the in-sample testing. Amilon (2003) tested two different ANN models and compared them to two different BSM models. Similarly, to Liu et al. (2019) and Anders et al. (1998), Amilon (2003) shows that the constructed ANN is superior to the BSM models in all of the cases, although not always at a 5 percent statistical significance level. He did comparisons both for pricing and also for trading and hedging mispriced options, and the only difference was that the ANN using implied volatility was better for pricing and the ANN with historical volatility was better for trading and hedging the mispriced options.

A few papers argued that the ANN did not perform that much better than the compared BSM models. Hutchinson et al. (1994) argue that the ANN model should mainly be used when traditional parametric methods fail. Ormoneit (1999) shows that his ANN yields results comparable to those of the BSM model. The one factor combining these studies that do not show the ANN outperforming the BSM is that their models are very simplified. As Table 3 shows, these models lack in the number of factors the other papers have used in their models, and the results can to some extent be accredited to this. It can be seen that neither of these papers uses for example any volatility estimate in their model that is a crucial part of option pricing. Ormoneit (1999) also notes the lack of variables and states that a considerable performance improvement can be obtained by using a better architecture.

The additional research questions were about the pricing performance depending on moneyness and the ANN MLP model architecture regarding the number of hidden layers and the number of hidden neurons. Malliaris and Salchenberger (1993a) tested the models in OTM and ITM pricing for different weeks of the testing period. The ANN put together by them outperformed the BSM model used in 4/5 of the weeks tested for the OTM options, whereas the BSM worked better for ITM options as they priced them better in 3/5 weeks. Bennell and Sutcliffe (2004) found similar results, as they note that the ANN is clearly superior to the BSM model in OTM, and when switching to ITM options the performance weakened resulting in the BSM model outperforming the ANN. Malliaris and Salchenberger (1993a) note that for the OTM options, the BSM model overprices and the ANN underprices the options and thus it could be a good idea to take an average out of these two. They continue that for ITM both of the models seem to underprice them. Bennell and Sutcliffe (2004) noticed that for deep ITM options, the ANN has a lot of difficulty pricing them, as well as those with a long expiry date. At least for the deep ITM options the fault could be noted on to the lack of observations, as most of the options obviously are not deep ITM. Anders et al. (1998) also note that the trading volume for deep ITM and deep OTM options is very low and combined with the fact that Liu et al. (2019) point out that ANN is a data-intensive model it might just be that more observations would be needed to price these options accurately.

Not many papers included volatility comparisons. The main ones were Amilon (2003) and Anders et al. (1998). The conclusion regarding volatility is that in pricing the options a form of implied volatility works the best. This can be either the regular implied volatility or the VDAX index, which is derived from implied volatilities. The implied volatility performing the best is of no surprise, as the implied volatility already includes data from the option, as it is calculated by already knowing the option's current price.

For the ANN MLP architecture regarding the hidden layers, the results were mainly unequivocal, at least when considering the number of hidden layers. All of the papers included in this literature review ended up using one hidden layer, either as a result of earlier literature or as a result of testing. The differences in the architectures lie in the number of hidden neurons. These cannot be compared directly as the models include different numbers of inputs as well. As summarized in Table 2, the number of these neurons varies from 3 to 14. Malliaris and Salchenberger (1993a) compared models with 3, 4, and 5 neurons to each other, and the one with 4 neurons priced options most accurately. It seems like the amount of these neurons has to always be found out by testing, as literature does not give an easy answer to the number of hidden neurons.

The artificial neural networks seem to perform better than the Black-Scholes-Merton models, and also some alternative option pricing models when used correctly. The superior ANN models share some combining factors, which are mainly the 5 factors also used in the BSM model. These results can be further improved by using a so-called homogeneity hint to also measure the moneyness, which is the underlying asset's price divided by the strike price (S/K). The ANN shares some common factors with most other option pricing models, for example, that the closer the underlying's price is to the strike price the more accurate the model is, and also that the ANN has trouble pricing options that are very close to or very far from the maturity date. Liu et al. (2019) show that at least some of these limitations can be addressed by for example categorizing options based on their moneyness and time to maturity. There has also been regards that the ANN is very data-intensive when compared to other models, and thus might yield bad results if used with small amounts of data.

6 DATA AND METHODOLOGY

The data used in the quantitative analysis section of this thesis is daily data and is a set of the 1-month Euribor interest rates, the DAX stock market index closing prices, the corresponding DAX stock market index option attributes, and the VDAX volatility index. The options used in this thesis are European-style call options, traded primarily in the Eurex. The attributes of the option contracts are the strike price, the expiration date, and the close price. The VDAX is the implied volatility of the DAX stock index, which is implied by the derivatives on the index. The DAX stock market index in itself is assembled according to the free float market capitalization and includes 40 (primarily 30) stocks traded at the Frankfurt exchange (Deutsche Börse, 2024). The data was collected from the London Stock Exchange Group (LSEG) Eikon API, and the queries were made using Python.

The dataset consists of the latest 4 full years at the time of writing this thesis, meaning that the options that had their expiration dates during the period of January 2020 to December 2023 are included in this thesis. This full unfiltered dataset includes 4839 different option contracts with 628 992 observations for the closing prices of these option contracts. The filtering of this dataset is done in the next subchapter. However, at this point already the data was searched according to one of Anders et al. (1998) filtering, where the data that is extremely deep-in- or deep-out-of-the-money is excluded from the dataset. The measure for extremely deep-ITM or deep-OTM is according to Anders et al. (1998) when:

$$\frac{S}{K} < 0.85 \; or \; \frac{S}{K} > 1.15 \qquad (37)$$

This means that only contracts with strike prices according to the maximum and minimum value of the DAX index during the used time period are searched. The range of strike prices used for the data queries is thus 8283 to 19 376.

6.1 Filtering and partitioning the data

The raw option data includes a lot of uninformative observations, such as ones with too long time until expiration, or an option that is too far OTM. This can be addressed by filtering the data according to different filters, which remove these non-representative observations from the dataset. This should make the model more accurate at pricing options, as the observations not passing these filters are often not interesting as use cases for a pricing model such as this. Most of the filtering is done right after the data is queried from the API and it follows the same filtering of Anders et al. (1998) where the goal is to remove these uninformative and non-representative option observations. The cases in which the option is removed at that time point are as follows:

1. The call option is traded at less than 10 points

$$C_t < 10 \tag{38}$$

2. The option has less than 15 days or more than 2 years to maturity

$$T - t \leq \frac{15}{252} or\ T - t \geq 2 \tag{39}$$

3. The lower boundary condition for the value of European call option is violated

$$C < S - Xe^{-rr} \tag{40}$$

4. The option is extremely deep-in- or deep-out-of-the-money, as presented in Equation 37

After filtering the data according to these 4 filters, the dataset includes 3637 option contracts with 256 260 observations. This is the dataset used for all of the quantitative analysis.

The complete dataset is partitioned chronologically into a training dataset and a testing dataset. Ruf and Wang (2020) point out that most of the earlier research partitions data chronologically, as this does not violate the time series structure of the data. If the whole dataset was to be divided randomly it would break the structure and cause information leakage they add. This quantitative analysis focuses on how well the ANN model prices options using out-of-sample data, and thus it is always important to use the testing dataset. The training set includes 80 percent of the observations whereas the testing dataset includes 20 percent of the observations as this is the division used by for example Bennell and Sutcliffe (2004). This means that the training set includes all observations from the 1st of January 2020 to the 22nd of December 2022. The testing set includes all of the observations from the 23rd of December 2022 to the 31st of December 2023. The ANN training also uses so-called validation split, which divides the training data into the training set and the evaluation set. Between each epoch, the ANN will evaluate its performance on the validation set, and then adjust its weights based on the loss function. The ANN used in this thesis has a validation split of 25 percent meaning that the model will use 25 percent of its training data in the evaluation between each epoch.

6.2 The complete dataset and the time period

Table 4 gives an overview of the dataset, with the number of observations divided into different moneyness and TTM categories. In regards to TTM most, of the observations seem to have more than half a year to their maturity, whereas the smallest category in this regard is the observations that are really close to their maturity. Previously mentioned ATM options refer to options exactly at-the-money, so where the underlying's price equals the strike price, and this category would be extremely small because of its exactness. The ATM category is expanded to near-the-money (NTM), as shown in the table below to also include options that are near to ATM. The dataset is quite well balanced, as the sums of ITM, OTM, NTM options are quite close together.

TABLE 4 The complete dataset divided into different categories

No. observations	TTM<0.2	0.2<TTM<0.5	0.5<TTM	Sum
ITM (>1.05)	19 019	23 037	31 916	73 972
NTM (0.95-1.05)	28 371	36 087	41 742	106 200
OTM (<0.95)	10 364	26 052	39 672	76 088
Sum	57 754	85 176	113 330	256 260

Figures 10 and 11 highlight the whole dataset, and how moneyness and TTM affect the prices of the options. The main idea from the Figures is that the deeper the option is OTM the less value it has, and the closer the option is to its maturity the closer it is to its intrinsic value.

While observing Figures 10 and 11 it could be said that even though the data is filtered as mentioned before it still has some outliers, such as some deep OTM options having large option prices, that notably deviate from the ordinary curve of data points. The data was observed to address these outliers, but they seem to have been caused by the large increase in volatility, as can be seen in Figure 13 at the beginning of 2020. Increasing volatility levels signify increased fluctuations in asset prices, introducing a higher level of unpredictability into future market movements. These data points are crucial to the model as they include a lot of information about how the model should address the situation of heightened volatility.

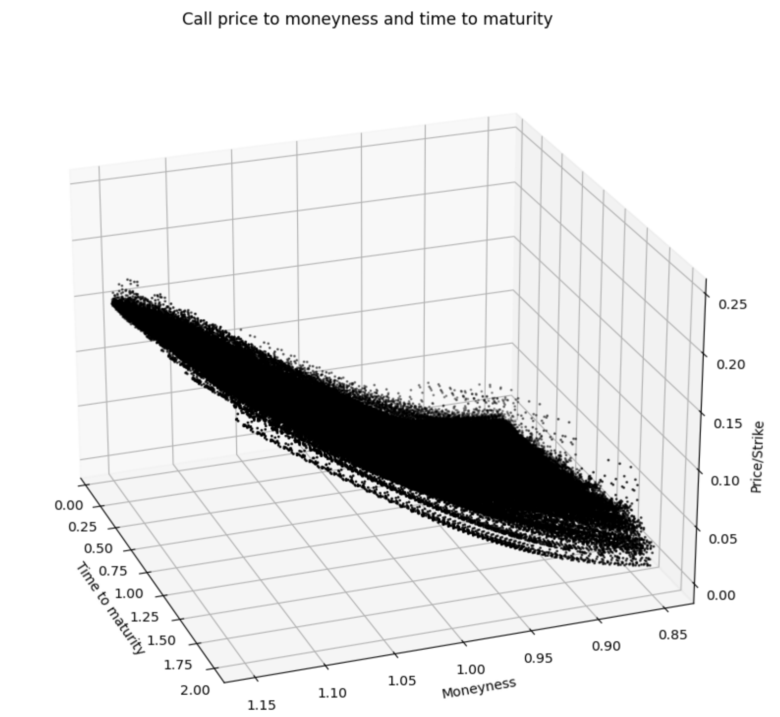

FIGURE 10 3-dimensional plot of the data points

There seems to form a gap without observations when the TTM is high, and the price of the option is zero or very close to zero. Because the TTM is so high, the uncertainty is high as well, as there exists more time for risks to realize. It is also observable that no data points exist where the option would be ITM, but the price would be lower than the intrinsic value of the option.

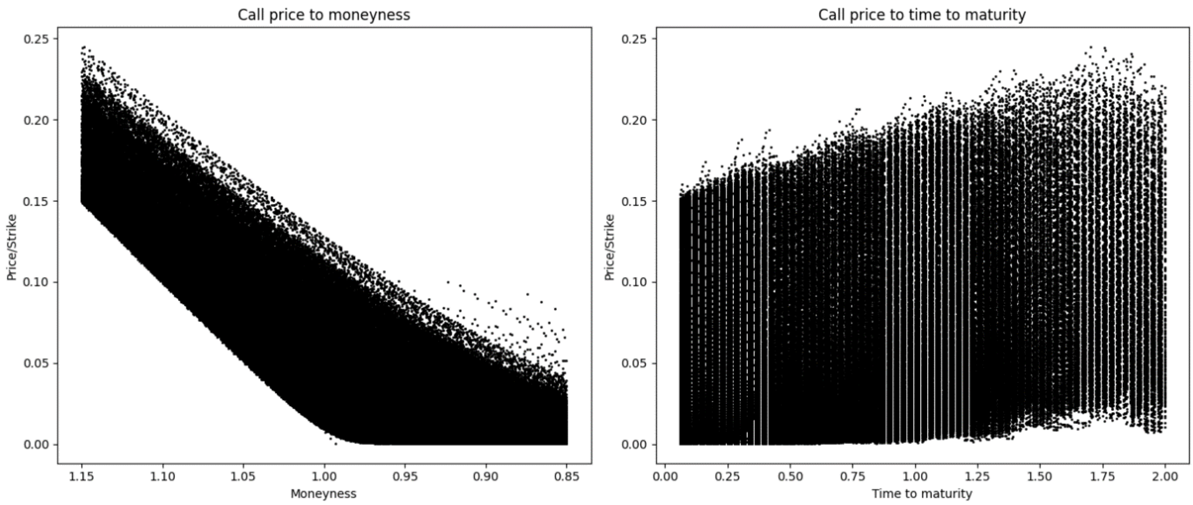

FIGURE 11 2-dimensional plots to demonstrate Figure 10 from different angles

The time frame of 4 complete years includes a few periods where the pricing is hard due to increased volatility. These increased volatility periods are caused by shocks in the economy, such as the COVID-19 pandemic and the Russo-Ukrainian war. In Figure 12 the COVID-19 pandemic's effects can be seen in early 2020, where the DAX index decreased to nearly 8000 index points level and here the index reached its lowest point in the whole time period in question. The VDAX volatility index shown in Figure 13 also highlights the COVID-19 pandemic, as the volatility reached its highest point in early 2020 as well. The increase in volatility from below 20 index points to almost 120 represents a substantial increase. The Russo-Ukrainian war's effects on the economy can be observed at the beginning of 2022, as an increase in volatility in the VDAX index in Figure 13 as well as a decrease in the DAX index in Figure 12.

FIGURE 12 DAX index from 2020 to the end of 2023

In finance when using volatility indices such as the VDAX index shown in Figure 13 it is important to mention that they already include information about the market at that time. As the market is the one pricing these indices, they are in a way forward-looking, as the market has already priced all the future information in them.

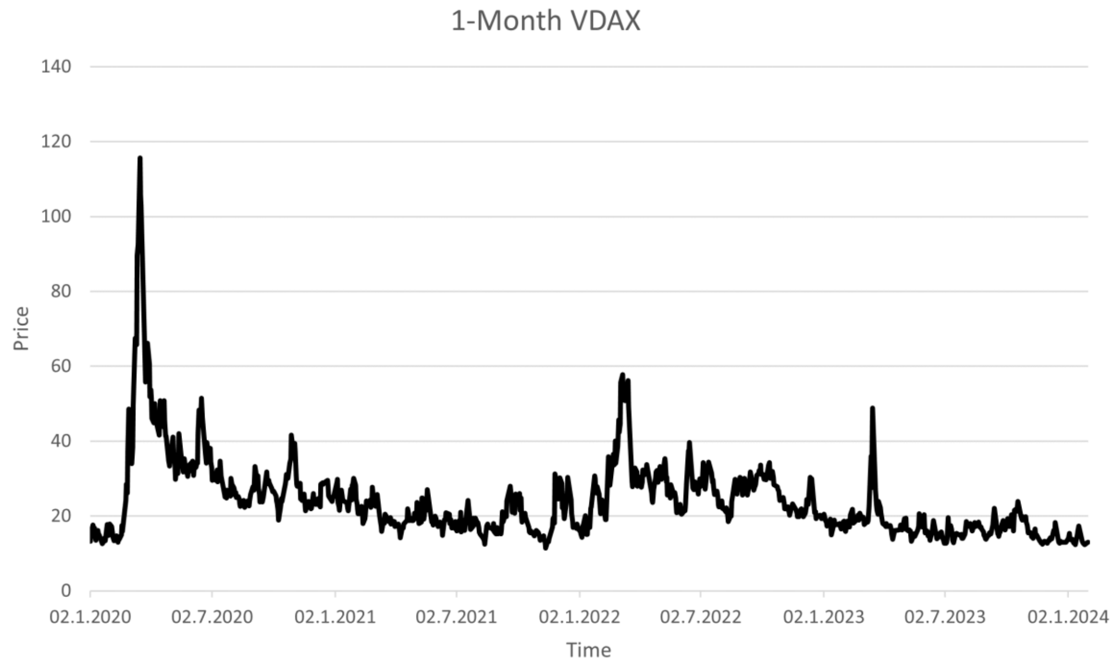

FIGURE 13 1-month VDAX volatility index from 2020 to the end of 2023

The time frame used in the analysis also includes large fluctuations in interest rates, where after a long period of zero or negative interest rates the market rates began increasing in the summer of 2022 as can be seen in Figure 14. The increases in the interest rates followed heightened inflation and the increases in central bank steering rates.

As most of the used time period includes negative interest rates, and only small fluctuations happened during zero interest rates, it is interesting to see whether the network has trouble pricing options depending on its interest rates or not. It might overweigh the interest rates' effects on the price of the option as the fluctuations in the rates are small. However, the later stage of the time period used in this study does include more interest rate changes so it might balance things out even though the majority of that period is included only in the testing dataset.

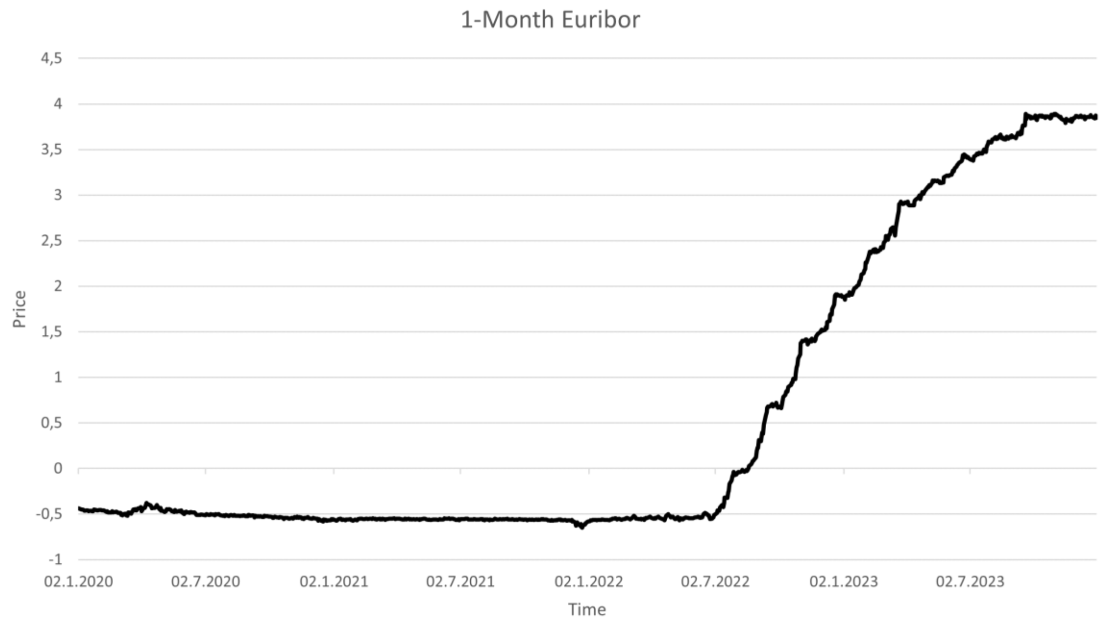

FIGURE 14 1-month Euribor from 2020 to the end of 2023

6.3 Research environment

The main tool used in this quantitative analysis is Python 3.11.6. Python is an open-source computer language, that seems to have the most possibilities for machine learning purposes due to its open-source nature. Python can be used for much more than data analysis as it can be tailored to fit many needs, whereas most programming languages only have some things where they excel. Python allows the use of many libraries, which include preprogrammed functions to be able to use it for the specific case. The libraries are helpful and easy to use, as they often have extensive documentation. The libraries used for the ANN are Keras integrated within Tensorflow 2.15.0. The models are run on an RTX 3060ti 8GB GPU, an Intel i5-10400F (2.9GHz) CPU, and 16 Gt of RAM.

6.4 Performance measures

The performance measures used in this analysis are the mean squared error (MSE) mean absolute error (MAE), mean absolute percentage error (MAPE) and R^2. These measures are defined as they are defined in subchapter 5.1.3. As the output of the model is the price of the option divided by its strike price, the C notation in the errors is also the price of the option divided by its strike price.

Anders et al. (1998) explain that the R^2 provides a measure of correlation between observed and fitted options prices, whereas the MAE measures absolute price discrepancies. They continue that MAPE judges the price differences relative to the price levels. The MSE measurement puts more weight on the large errors. To achieve a measure of the performance from different angles it is important that these three measurements measure different things in the pricing, as the measurements chosen here do.

6.5 Model and parameters for the neural network

The model that was used by the ANN was decided based on earlier literature. Several different models with different numbers of inputs were presented in Chapter 5, and conclusions were made about what the best inputs were. This chapter also concluded to use inputs as standardised features such as S/K and C/K. The question of whether this specific input structure is optimal according to different pricing error measurements is not in the scope of this study. The earlier literature does seem to conclude that the smallest errors can be achieved with the following model:

$$C/K \ = f(S/K, \tau, \sigma, r) \tag{41}$$

Where the variables are denoted according to Table 4. The volatility estimate is the 1-month VDAX index, as either implied volatility or the VDAX often yielded the smallest errors in volatility comparisons in the earlier literature. The estimate of the risk-free interest rate is the 1-month Euribor, as that is the most riskless interest rate there is for the Euro area.

A problem might arise from the interest rate fluctuations during the time period, as the training set only includes sub-2 percent interest rates, and as the testing set includes only noticeably higher interest rates than 2 percent. If the model given in Equation 41 fails to price options accurately also the following model will be estimated:

$$C/K = f(S/K, \tau, \sigma) \tag{42}$$

This model does not include the interest rate at all.

Other parameter choices for the neural network include the loss function, batch size, the activation function, and the optimizer. The loss function used in this thesis is the mean squared errors (MSE). A batch size of 64 is chosen and the activation function used is the rectified linear unit (ReLU) which according to Lindholm et al. (2022, p. 134) is a common choice. The optimizer is the Adam optimizer presented by Kingma and Ba (2014).

7 RESULTS

This chapter presents the results of the quantitative comparison in option pricing between the three different option pricing methods, the ANN, the MCS, and the BSM. First, an optimal structure for the ANN is presented based on the performance measures presented in Chapter 5.1.3. It is then compared to the other models using the whole dataset for the analysis. The models were also compared to each other using smaller sections of the data, to answer the research question based on the pricing performance depending on moneyness. These data categories were divided into 3 different sections for the moneyness. These comparisons also use the same performance measures as the comparison of different ANNs.

7.1 Optimal artificial neural network model

As the data partitioning is done chronologically it might cause problems based on the network's model. As can be seen in Figure 14, the 1-month Euribor increased consistently after the summer of 2022. As the data set's division point to the training and testing sets is in December 2022 the ANN might overestimate the effects the interest rate has on the prices of the options as the fluctuations in the interest rate were much smaller in the training set than in the testing set. The aim of this subchapter is to find out if the model with interest rate can price the options accurately. Additionally, if the model falls short in its accuracy, the aim is to find out whether this discrepancy can be attributed to fluctuations in the interest rate. If the model with interest rate is not sufficient, the two models will be compared according to their pricing accuracy. The architecture comparisons will be saved for later in this chapter.

The models have an architecture of 1 hidden layer and 6 neurons as well as 1000 epochs, as suggested by prior literature. Figure 15 compares the pricing errors to the level of the interest rate. According to Figure 15, it seems evident that the higher the interest rate, the higher the pricing error. The model estimated according to Equation 41 seems to have a problem with the interest rate as it overestimates the effects of the interest rate on the price of the options.

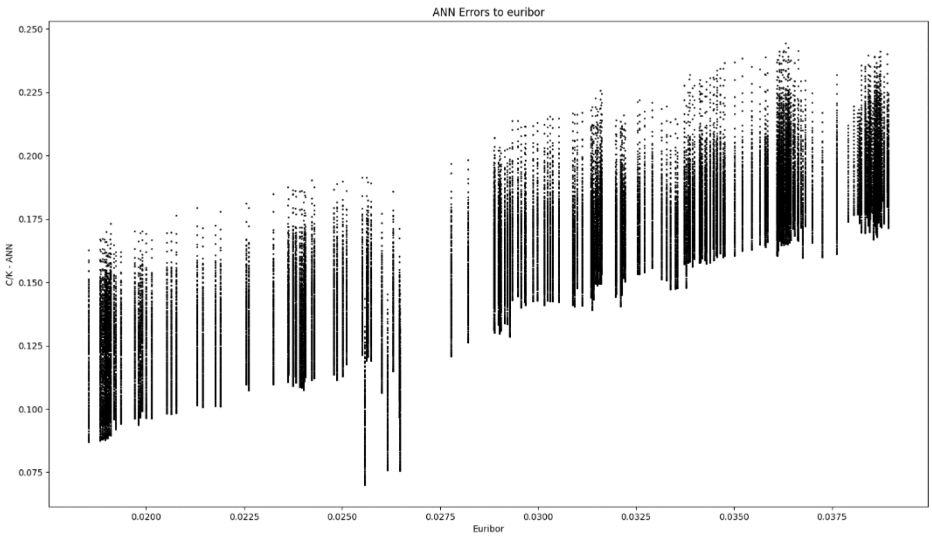

FIGURE 15 An error plot of the pricing errors to the level of Euribor

Excluding the interest rate from the model presents the simplest solution to addressing the difficulties posed by the rates fluctuations during the time period, as Equation 42 does. Table 5 compares the results of the two models, with and without the interest rate included. The model without interest rate outperforms significantly in terms of every performance measure used. The absolute errors are very small for the model without interest rate, as well as the percentage errors. The model with the interest rate that was constructed according to Equation 41 had a lot of trouble fitting to the data, as the R^2 does not give any reasonable results. However, without interest rates, the model fits the data fine and is able to explain 97 percent of the variance.

TABLE 5 Comparison of models with and without interest rate included

Model according to equation:	**MSE x 10^{-5}**	**MAE x 10^{-3}**	**MAPE**	R^2
41	2650	160	24.6	-9.92
42	6.53	5.76	0.647	0.9731

In conclusion, the model that estimates option prices with the interest rate according to Equation 41 cannot price them very accurately. There is a problem with the estimation and training of the model, as the training period only includes sub-2 percent interest rate which is a lot lower than the rate for the testing period. The interest rate seems to be the cause of large errors in option pricing as highlighted by Figure 15. When the interest rate is excluded from the model, the pricing accuracy improves a lot. Further analysis will be done with the model that corresponds to Equation 42.

7.2 Optimal artificial neural network structure

On top of deciding on the variables that are fed to the ANN, its different parameters are also important in obtaining the smallest errors in pricing performance. The aim of this subchapter is to find the optimal number of hidden layers, and hidden neurons as well as the optimal number of epochs used by the ANN. The number of the variables in question and their respective errors are presented in the tables below. These results are acquired with out-of-sample datasets. The results obviously might change depending on the data period. It is also important to note that the optimal structure may not stay optimal when using the model to price options only in a singular category, such as options near the maturity or options out-of-the-money.

The testing began with 1 hidden layer, 6 hidden neurons, and 1000 epochs and the variables used in the training and estimation are selected according to Equation 42. As Table 2 noted, 1 hidden layer is often sufficient according to the earlier literature, and the number of neurons is often below 10. Bennell and Sutcliffe (2004) used their ANN with 1000 epochs and other literature didn't note the number of epochs used so the testing will start with 1000.

According to Table 6, the accuracy of the model improves when moving up from only 1 hidden layer according to every performance measure. The optimal number seems to be 4, as before and after that all of the errors increase and the explanatory power of the model decreases. There isn't really a significant difference in the absolute errors between the models, but there does seem to be a notable decrease in percentage errors when the number of hidden layers is increased from 1. These results seem to be very different when compared to the neural network architectures presented in Table 2, as they all use only one hidden layer.

TABLE 6 Hidden layer comparison

No. hidden layers	**MSE x 10^{-5}**	**MAE x 10^{-3}**	**MAPE**	R^2
1	6.53	5.76	0.647	0.9731
2	5.70	5.36	0.266	0.9765
4	4.98	4.76	0.182	0.9795
6	5.83	5.42	0.198	0.9760
8	5.61	5.37	0.197	0.9769

Table 7 highlights the differences resulting from the different number of hidden neurons with 4 hidden layers and 1000 epochs. The number of hidden neurons is the number of them in each layer. All of the values seem to yield quite similar results, where 16 neurons is the best model when valued with squared errors, absolute errors or R^2 and the model with 56 neurons is the most accurate when measured with percentage errors. As the difference is small both models would work fine. As the model with 16 neurons per layer is closer to the prior literature and is superior to the model with 56 neurons when measured with three different performance measures (MSE, MAE, and R^2) it will be the model that is used in the later analysis.

TABLE 7 Hidden neuron comparison

No. hidden neurons	**MSE x 10^{-5}**	**MAE x 10^{-3}**	**MAPE**	R^2
6	4.98	4.76	0.182	0.9795
16	4.91	4.56	0.183	0.9798
26	5.52	5.05	0.178	0.9773
36	5.15	4.85	0.199	0.9788
46	5.80	5.56	0.232	0.9761
56	5.86	5.19	0.172	0.9759
66	7.79	6.57	0.222	0.9679

With 4 hidden layers and 16 neurons per each layer, the size of the network is large. However, as the results presented in these tables are out-of-sample there should not really exist overfitting problems which should be addressed as the size of the network is large. Testing with out-of-sample data proves that the model is competent also in generalizing the pricing even with a large network size.

Epoch comparisons will be done with 4 hidden layers with 16 neurons in each layer. Once again, a balance between not fitting the model to the data well enough and overfitting has to be found. Overfitting might be caused by a large number of epochs. According to Table 8, an optimal number according to every performance measurement seems to be 1000 epochs, which is the same number that Bennell and Sutcliffe (2004) were using who were the only ones to report the number of epochs.

Interestingly enough, fitting the model for more than 2000 epochs seems to increase the accuracy slightly when compared to 2000 epochs. It would be expected that if the accuracy starts decreasing because of overfitting, fitting the data more would cause larger errors because of even more overfitting.

TABLE 8 Epoch comparison

No. epochs	**MSE x 10^{-5}**	**MAE x 10^{-3}**	**MAPE**	R^2
500	5.03	4.98	0.202	0.9793
1000	4.91	4.56	0.183	0.9798
2000	5.53	5.18	0.199	0.9772
3000	5.29	4.99	0.193	0.9782

In conclusion, the optimal structure for the dataset seems to be 4 hidden layers, 16 hidden neurons per layer, and 1000 epochs. This structure will be used for the rest of the thesis, in the model comparisons. Testing several other network architectures also took place that were not reported, which did not perform as well as the models reported in tables 6-8. There might obviously be a very specific neural network architecture that performs better than the one found here. However, when testing for optimal architectures the model has to always be built and trained first, which uses a lot of computing power and thus a lot of time as well. There does not seem to be an easier way to come to a conclusion of the best model than to build and train them separately or let an algorithm do the building and training of several models. Iterating over hundreds of different models is not within the scope of interest for this study.

In comparison to earlier literature, this network architecture is substantially larger in size than any network used earlier. As Table 2 presents there are only single hidden layered networks with the number of neurons in the layer ranging from 3 to 14. It might be that a large dataset of over 200 000 observations allows for a larger network size without fear of overfitting. As mentioned, testing was done using only out-of-sample data and thus results are reliable as the network is able to generalize the estimations outside of their training dataset.

7.3 Number of simulations for the Monte Carlo simulation

The number of simulations that the Monte Carlo method runs should also be decided. This subchapter presents a comparison of a few different numbers of simulations. Usually, the pricing accuracy should always be better when the number of simulations is larger, as then it approaches the correct price of the option. However, the number of simulations always increases the computing power required and thus it is important to increase it until the pricing accuracy only increases in very small amounts. The analysis in this subchapter is done on the testing set only as that sample will be used for all of the other analyses as well.

Table 9 highlights how well the MCS prices options depending on the number of simulations used. It seems clear that increasing the number of simulations from 5 000 onwards does not provide significant increases in accuracy. A smaller number of simulations than those used in Table 9 might yield measurements that are significantly different, as there is some difference between 1000 and 5000 simulations. According to every performance measure, the pricing accuracy of the MCS improves when the number of simulations is increased. However, increasing the simulations past 5000 does not seem justifiable, as the required computing power increases, and the accuracy of the model does not seem to increase significantly. As the pricing values already exist for the most accurate MCS representation of 50 000 simulations those values will be used for the remaining analysis.

TABLE 9 Number of simulations and their measurements for the Monte Carlo simulation

No. simulations	MSE x 10^{-5}	MAE x 10^{-3}	MAPE	R^2
1 000	19.5	8.70	1.13	0.9194
5 000	19.0	8.44	1.13	0.9217
10 000	18.9	8.40	1.13	0.9222
50 000	18.8	8.37	1.13	0.9225

7.4 Pricing results

7.4.1 Pricing results for the whole dataset

Now that the optimal structure for the ANN is reached, and the number of simulations are decided for the MCS it is possible to compare these optimally structured models to each other. This subchapter presents the pricing results with the same dataset used earlier in this chapter, as that is not part of the training dataset used for the ANN. Later on, this subchapter will also present the results in model comparisons in different moneyness categories.

As a recap, the earlier studies found that generally, the ANN seems to outperform the BSM model. In some cases, the BSM was more accurate in its pricing and these cases were mainly found with ITM options. It is expected that the BSM and MCS perform equally, and that as the optimal structure for the ANN is reached it would perform the best out of these models.

Table 10 presents the performance metrics for the three different models when using the whole testing dataset. The ANN prices out-of-sample options, so no observations are used in testing which were used in its training. The same testing dataset that was used for the ANN is also used for the BSM and MCS.

For the whole dataset, the ANN clearly performs the best out of the three models as Table 10 presents. As expected, the BSM and the MCS yield almost identical results. In absolute terms, the errors of the ANN are around half of the errors of the BSM or the MCS. The ANN is also capable of explaining more of the price variability as shown by the larger R^2. In other words, the ANN model fits to the data a lot better than the stochastic models do. The percentages of the errors are also well in line with these findings, as the MAPE for the ANN is very small when compared to the MAPE of the BSM model or the MAPE of the MCS model.

An interesting observation is the large difference in MSE between the stochastic models when compared to the ANN while the difference in the MAE is not as large. A large difference in the MSE might implicate that the stochastic models have some extreme errors or outliers in their pricing. These errors are amplified by the MSE as the errors are squared.

TABLE 10 Pricing results for the whole testing dataset

	MSE x 10^{-5}	MAE x 10^{-3}	MAPE	R^2
ANN	4.91	4.56	0.183	0.9798
BSM	18.8	8.36	1.13	0.9225
MCS	18.8	8.37	1.13	0.9225

7.4.2 Pricing results in different moneyness categories

Tables 11, 12 and 13 present the pricing results for the different moneyness categories. The moneyness categories are defined as follows:

$$ITM\ when\ 1.15 \geq \frac{S}{K} > 1.05 \tag{43}$$

$$NTM\ when\ 1.05 \geq \frac{S}{K} \geq 0.95 \tag{44}$$

$$OTM\ when\ 0.95 > \frac{S}{K} \geq 0.85 \tag{45}$$

First, the pricing of ITM options is evaluated as shown in Table 11. When compared to the whole dataset the MAE of the ANN increased while the MAE for the stochastic models decreased to around half of what it was for the whole dataset. The earlier studies suggested that the ITM options have been more difficult for the ANN to price, and easier for the BSM to price. The findings of this study are consistent with prior research in this case.

According to Malliaris and Salchenberger (1993a) and Bennell and Sutcliffe (2004), the BSM model has even been able to outperform the ANN in this moneyness category. Table 11 showcases some similarities to this as according to the MAE and MAPE the stochastic models are smaller than the network in pricing ITM options. However, according to the MSE and R^2 the ANN outperforms the stochastic models. There does not seem to be a best model for pricing ITM options. The stochastic models and the ANN all performed similarly.

When compared to the whole dataset results the MAPE decreased a lot, even for the ANN which had an increase in MAE when moving to pricing ITM options rather than the whole dataset. As presented in Figures 10 and 11 the price of the option increases as it moves towards deeper ITM (moneyness increases). As shown in Equation 33 MAPE is calculated by dividing the error by the price of the option. As the denominator increases the value of the MAPE decreases and thus the higher the price of the option is the lower the MAPE is. This is why MAPE is not a suitable performance measure to compare different moneyness categories but rather compare the models to each other in the same moneyness category.

TABLE 11 Pricing results for ITM options

	MSE x 10^{-5}	MAE x 10^{-3}	MAPE	R^2
ANN	7.53	5.96	0.0480	0.9270
BSM	8.46	3.59	0.0315	0.9180
MCS	8.44	3.61	0.0317	0.9182

For near-the-money options, the pricing performance of the stochastic models as well as for the ANN model is nearly the same as for the whole dataset, as is explained by Table 12 according to all performance measures but the R^2. This is expected as Table 4 presents that the NTM has the most observations in the whole dataset and is thus weighed more in the whole dataset comparison as well. The network is able to price the options accurately and is superior to the stochastic models measured with every performance measure.

Table 12 shows that the R^2 values for the stochastic models have changed quite drastically to the whole dataset measured in Table 10. The low R^2 value indicates that the stochastic models are not able to explain a lot of the variance happening in the option prices, even though the absolute errors stay the same on average. As shown later in Figure 15 and noted by Malliaris and Salchenberger (1993a) the BSM overvalues OTM options a lot. Figure 15 also shows that NTM options are also often overpriced. This means that their pricing errors are often large which is also highlighted by the difference in MSE and MAE of the stochastic models when compared to the network as shown in Table 12. Equation 35 notes how the R^2 is calculated where the pricing error of the model that is squared is in the numerator of the fraction, and the fraction is subtracted from 1. Because the pricing errors of the stochastic models are large for OTM options, that means that the fraction is large and this in return means that a large number subtracted from 1 yields a small number as shown in Table 12 R^2 for the stochastic models. This same behavior of the R^2 should also come up during OTM option pricing and the effect should be even larger.

TABLE 12 Pricing results for NTM options

	MSE x 10^{-5}	MAE x 10^{-3}	MAPE	R^2
ANN	4.48	4.50	0.145	0.9299
BSM	18.6	9.35	0.612	0.7094
MCS	18.6	9.35	0.612	0.7094

The pricing of OTM options has been difficult for the stochastic models as noted by prior literature. Table 13 further demonstrates this difficulty, as the MSE and MAE increase a lot when compared to the whole dataset presented in Table 10. On the contrary, the ANN was able to price OTM options more accurately than the whole dataset of options according to MSE and MAE. When comparing the different models and their MSE, MAE, and MAPE it is clear that the performance of the ANN is superior to the performance of the stochastic models. The BSM and MCS once again do not show any difference in their performance. As mentioned before, the MAPE does not allow for moneyness category comparisons, as the denominator for the MAPE in Equation 33 decreases and thus drives up the value of the MAPE on its own.

The R^2 for the ANN is noticeably less than for the whole dataset meaning that even though the ANN yields lower errors in the OTM category than for the whole dataset, it still cannot explain as much of the variance in the OTM category. For the stochastic models, the R^2 behaves very differently in pricing OTM options as well, and it is a difficult measure in this case. As discussed earlier the overpricing of options in the OTM category causes the stochastic models to have a small R^2 value. This problem with the R^2 is a lot clearer in pricing OTM options in Table 13 than it is in NTM option pricing in Table 12. The pricing of OTM options is very difficult for the stochastic models measured with every performance measure and especially when measured with R^2.

TABLE 13 Pricing results for OTM options

	MSE x 10^{-5}	**MAE x 10^{-3}**	**MAPE**	R^2
ANN	2.00	2.67	0.455	0.8114
BSM	34.2	13.2	3.76	-2.219
MCS	34.2	13.2	3.76	-2.223

When comparing these different moneyness categories to each other, the ANN outperforms the stochastic models when pricing NTM and OTM options and all of the models seem to be about as good in pricing ITM options. When measured with squared and absolute errors the performance of the ANN is the best for the OTM options and the worst for ITM, whereas the performance of the stochastic models yielded the opposite results as their performance is the best for ITM options and the worst for OTM options. This was also found out by Bennell and Sutcliffe (2004), who found out that with their sample and network architecture, the pricing of ITM options also resulted in comparable results between the BSM and the ANN where the BSM might have been better.

When measured with R^2 the ANN performs the best in the NTM category and the worst in the OTM category. This means that the ANN is able to explain more of the variance of the option prices for NTM options than for other categories. The results for the stochastic models are similar to those results acquired when using MSE and MAE as they are best at pricing ITM options and worst at OTM option pricing.

Pricing errors of each model to their moneyness are highlighted in Figure 16, where the first plot represents the errors of the ANN, the second is the errors of the BSM model, and finally on the second row are the errors of the MCS. The pricing error is on the Y-axis and moneyness is on the X-axis with OTM options being closer to the Y-axis. The errors of the ANN seem to be more clustered together, and there are no extreme outliers in the whole data. It might be of interest to observe the error plot of the ANN with its Y-axis scaled based on its own errors and not by those it is compared to. Figure 17 represents a plot for the ANN's errors to moneyness, with the Y-axis (the error axis) scaled better.

As highlighted by Tables 12, and 13 already the ANN prices options more accurately than the stochastic models OTM and NTM and Figure 16 graphics are consistent with this finding. According to Figures 16 and 17 as well as Table 11, the errors of the ANN also spread out more when moving deeper into ITM, where the ANN performed the worst out of the three categories. A majority of these errors are above the zero line, which implies that the model underprices the ITM options. The ANN underpricing ITM options was also found out by Malliaris and Salchenberger (1993a). It could be said that systematically for ITM options the ANN makes more errors in its pricing whereas the stochastic models make more large errors.

The error plots for the BSM and MCS seem very identical as expected so they will be discussed as one. These stochastic models seem to have a lot more outliers in every moneyness category. These outliers are also shown in Tables 10, 11, 12, and 13 as the MSE for the stochastic models is large in comparison to the ANN in every single category. The MSE weighs extreme errors more than other error measurements. The error plots are also extremely one-sided, as there seem to mainly be pricing errors on the negative side of the Figure's Y-axis. This means that the stochastic models constantly overprice the prices of the options, and this is especially true for the OTM options. Nearly the whole cluster of errors in OTM is also below the zero line. This graphic is very much in line with Table 13 which also shows the difficulty of pricing OTM options by the stochastic models. This discovery also aligns with the results reported by Malliaris and Salchenberger (1993a), who also noted that OTM options are often overpriced by the BSM model. The pricing performance of the stochastic models does get better for ITM options, but the plot is still one-sided with the data points outside of the main cluster of pricing errors. If the stochastic models make a large pricing error it is consistently overpricing the options, rather than underpricing.

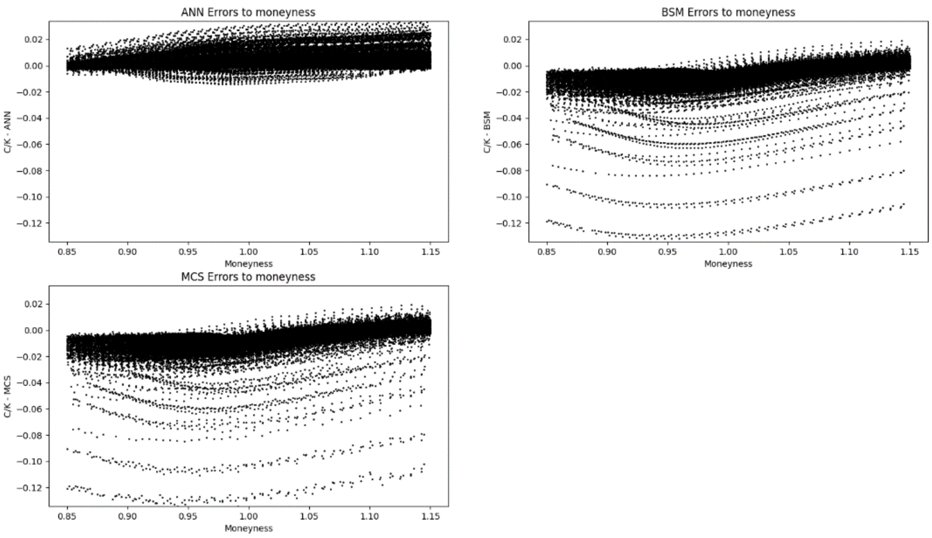

FIGURE 16 Pricing errors to moneyness of the three different pricing models

As mentioned, the unsuitably scaled Y-axis of the ANN in Figure 16 calls for closer inspection of the ANN model's error plotting. That is done in Figure 17. As the ANN's pricing results in Tables 11, 12, and 13 tell the ANN does have some difficulties in pricing ITM options. The largest cluster of errors and also the rest of the errors plotted in Figure 17 seem to be above the 0 line especially for other than OTM options. This is a sign that the model underprices these options. Malliaris and Salchenberger (1993a) noted the same observation that the ANN would consistently underprice ITM options. Based on Figure 17 this might be true for ITM and even NTM options, but the plot for OTM options is close to zero. It is also evident that the main cluster or the range of the error points expand as options deeper ITM are priced.

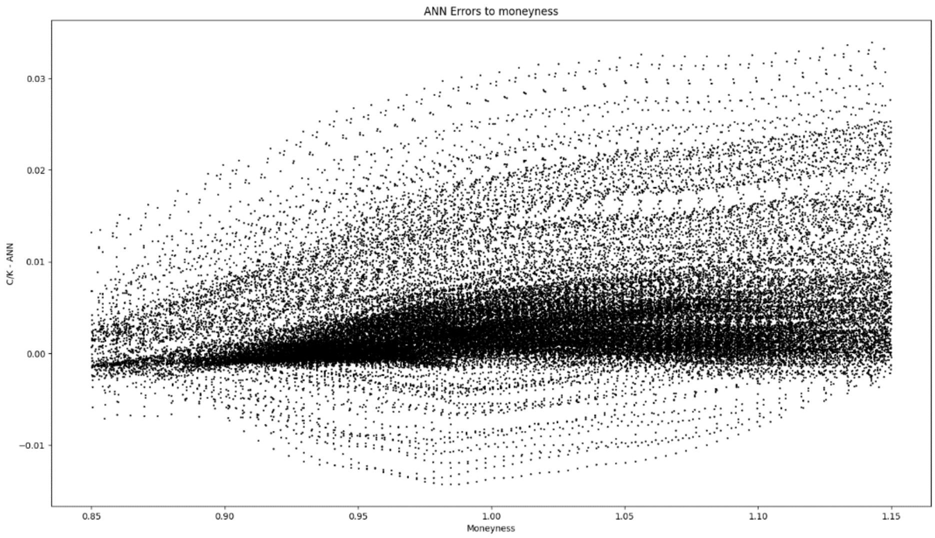

FIGURE 17 Pricing errors of the ANN to the moneyness with a scaled Y-axis

8 CONCLUSIONS

This thesis compared the artificial neural network option pricing model to two very similarly performing stochastic models the Black-Scholes-Merton model and the Monte Carlo simulation. This performance was compared for the whole dataset as well as for three different moneyness categories. The testing of the network was only done using out-of-sample data, and the three different models always used the same dataset in their comparisons. This thesis also researched and experimented with different artificial neural network architectures and compared several architectures to each other. The main results of this study are presented in Table 14.

The main goal of the study was to compare three different models to each other while the emphasis stayed on the neural network model. The primary objective was to answer the question "Is the artificial neural network better at pricing the DAX-index call options than the Monte Carlo simulation method, and the Black-Scholes-Merton option pricing model?". By all performance measures MSE, MAE, MAPE, and R^2 the artificial neural network is more accurate in pricing the European DAX-index call options using the whole testing dataset. Prior literature overall seems to agree that the artificial neural network is generally more accurate in its pricing so the findings of this study on this part align with previous research in the field.

Before doing these comparisons between the models an optimal neural network architecture had to be discovered for this dataset. This was done to answer the research question "What is the optimal number of hidden layers, hidden neurons, and epochs in the artificial neural network?". For this dataset, the optimal number of hidden layers was 4, with 16 neurons in each layer and with the neural network running 1000 epochs. This finding is somewhat contrary to prior research. There has not been too much discussion or research about optimal network architectures, but all of the papers referenced in this research only used 1 hidden layer and around 8 neurons.

To answer the third research question "Are the pricing performances of the artificial neural network, the Monte Carlo simulation, and the Black-Scholes-Merton model dependent on the moneyness of the option?" the dataset was divided into three different moneyness categories. The ANN was the most accurate out of the three models in the OTM and the NTM categories, where the difference between the stochastic models and the ANN was very significant in OTM option pricing. The discrepancies between the models were significantly diminishing the more ITM the options moved to the point of similar pricing accuracy in ITM option pricing between the models. This result is very much in line with prior literature. The ANN seems to systematically make more pricing errors when pricing ITM options, whereas the stochastic models make more large errors.

The pricing errors of the stochastic models seem to be dependent on different moneyness categories, as they seem to overprice out-of-the-money options. The stochastic models also seem to have a lot more extreme pricing errors in all of the moneyness categories when compared to the ANN and this can especially be observed in Figure 16. The network seems to underprice options that are NTM or ITM and this was also noted by other papers before, and these results can be drawn from Figure 17.

TABLE 14 The main results of the study

Research question	**The answer**
Is the artificial neural network better at pricing the DAX-index call options than the Monte Carlo simulation method, and the Black-Scholes-Merton option pricing model?	The ANN is better at pricing the DAX-index call options depending on the moneyness of the options. For the complete dataset, the ANN is more accurate in its pricing.
What is the optimal number of hidden layers, hidden neurons, and epochs in the artificial neural network?	The optimal architecture is 4 hidden layers, with 16 neurons in each layer and with 1000 epochs. This architecture might be different depending on other model specifications or the used dataset.
Are the pricing performances of the artificial neural network, the Monte Carlo simulation, and the Black-Scholes-Merton model dependent on the moneyness of the option?	The pricing performance of the three models is dependent on the moneyness of the options. The ANN is the best at pricing OTM and NTM options, whereas the stochastic models and the ANN are comparable in ITM option pricing.

Overall, the artificial neural network seems to perform well in option pricing when compared to the stochastic models. The BSM and the MCS performed nearly equivalently as expected. While measuring with MSE and MAE the ANN definitely excels in pricing out-of-the-money options, which are hard to price for the stochastic models. The search for an optimal network architecture is of essence while aiming for the lowest errors and the highest coefficient of determination, even though earlier literature showcases that even without it the network still often outperforms the BSM in most moneyness categories and for the whole dataset together. The ANN in this thesis might have performed even better, if there would have been no problems with the data partitioning because of the interest rates.

Training the ANN requires a large dataset and more work than the stochastic models. In situations where the models are comparable, such as ITM option pricing, it might be justified to just use the easiest-to-use model. For other moneyness categories where the differences between the models' pricing are large, the extra work to train an ANN should be considered as the stochastic models' performance is unsatisfactory.

The main limitations of this study lie in the absence of a strong theoretical background as an artificial neural network is a completely empirical model. Especially the Black-Scholes-Merton model is strongly derived from the theory, and thus the results it yields do have a strong foundation. There are also no mechanisms to explain the pricing decisions of the network. Other limitations might be attributed to there being seemingly no possibility of determining the optimal network architecture other than experimentation, which is very time and computing-power-consuming. It should also be said that setting up an artificial neural network option pricing model and training it does require some knowledge of programming, which is more than what is required by the Black-Scholes-Merton model or the Monte Carlo simulation.

There is still a lot to be covered about machine learning in the option pricing framework. This thesis examined the pricing performance of the ANN compared to other models but also investigated the different ANN architectures. However, there still exists a lot of customization options for the neural networks that would be beneficial to be researched more. This possible future research includes for example different model optimizers and different activation functions such as leaky rectified linear units. There also exist many alternative pricing models beyond the stochastic models used here, which the ANN could be compared to. It would also be interesting to see comparisons of computing time required to estimate the models, depending on the different parameters of the model. Artificial neural network's pricing performance could also be tested on more complex option contracts such as some exotic options. It would also be interesting to see whether any conclusions can be drawn on the optimal structure of the network and how the optimal architecture could be derived from the information of the dataset, and not only through trial and error.

REFERENCES

Alpaydin, E. (2020). *Introduction to Machine Learning, fourth edition*. MIT Press.

Amilon, H. (2003). A neural network versus Black–Scholes: A comparison of pricing and hedging performances. *Journal of Forecasting*, *22*(4), 317–335. https://doi.org/10.1002/for.867

Anders, U., Korn, O., & Schmitt, C. (1998). Improving the pricing of options: A neural network approach. *Journal of Forecasting*, *17*(5–6), 369–388. https://doi.org/10.1002/(SICI)1099-131X(1998090)17:5/6<369::AID-FOR702>3.0.CO;2-S

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. https://doi.org/10.1016/j.ejor.2005.03.081

Andreou, P., Charalambous, C., & Martzoukos, S. (2014). Assessing the performance of symmetric and asymmetric implied volatility functions. *Review of Quantitative Finance and Accounting*, *42*, 373–397. https://doi.org/10.1007/s11156-013-0346-z

Bachelier, L. (1900). *Théorie de la spéculation*. Gauthier-Villars.

Bakshi, G., Cao, C., & Chen, Z. (1997). Empirical Performance of Alternative Option Pricing Models. *The Journal of Finance*, *52*(5), 2003–2049. https://doi.org/10.1111/j.1540-6261.1997.tb02749.x

Barraquand, J., & Martineau, D. (1995). Numerical Valuation of High Dimensional Multivariate American Securities. *The Journal of Financial and Quantitative Analysis*, *30*(3), 383–405. https://doi.org/10.2307/2331347

Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends® in Machine Learning*, *2*(1), 1–127. https://doi.org/10.1561/2200000006

Bennell, J., & Sutcliffe, C. (2004). Black-Scholes Versus Neural Networks in Pricing FTSE 100 Options. *University of Southampton - Department of Accounting and Management Science, Papers*, *12*. https://doi.org/10.1002/isaf.254

Bi, Q., Goodman, K. E., Kaminsky, J., & Lessler, J. (2019b). What is Machine Learning? A Primer for the Epidemiologist. *American Journal of Epidemiology*, *188*(12), 2222–2239. https://doi.org/10.1093/aje/kwz189

Black, F., & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, *81*(3), 637–654.

Boyle, P., Broadie, M., & Glasserman, P. (1997). Monte Carlo methods for security pricing. *Journal of Economic Dynamics and Control*, *21*(8), 1267–1321. https://doi.org/10.1016/S0165-1889(97)00028-6

Boyle, P. P. (1977). Options: A Monte Carlo approach. *Journal of Financial Economics*, *4*(3), 323–338. https://doi.org/10.1016/0304-405X(77)90005-8

Broadie, M., & Glasserman, P. (1997). Pricing American-style securities using simulation. *Journal of Economic Dynamics and Control*, *21*(8), 1323–1352. https://doi.org/10.1016/S0165-1889(97)00029-8

Buehler, H., Gonon, L., Teichmann, J., & Wood, B. (2019). Deep hedging. *Quantitative Finance*, *19*, 1–21. https://doi.org/10.1080/14697688.2019.1571683

Butts, M. B., Hoest-Madsen, J., & Refsgaard, J. C. (2003). Hydrologic Forecasting. In R. A. Meyers (Ed.), *Encyclopedia of Physical Science and Technology (Third Edition)*. Academic Press. https://doi.org/10.1016/B0-12-227410-5/00325-2

Deutsche Börse. (2024). DAX. https://qontigo.com/index/dax/

Diebold, F., & Mariano, R. (1995). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

Dumas, B., Fleming, J., & Whaley, R. E. (1998). Implied Volatility Functions: Empirical Tests. *The Journal of Finance*, *53*(6), 2059–2106. https://doi.org/10.1111/0022-1082.00083

Freitas, J. F. G. de, Niranjan, M., Gee, A. H., & Doucet, A. (2000). Sequential Monte Carlo Methods to Train Neural Network Models. *Neural Computation*, *12*(4), 955–993. https://doi.org/10.1162/089976600300015664

Garcia, R., & Gençay, R. (2000). Pricing and hedging derivative securities with neural networks and a homogeneity hint. *Journal of Econometrics*, *94*(1), 93–115. https://doi.org/10.1016/S0304-4076(99)00018-4

Hagan, P., Kumar, D., Lesniewski, A., & Woodward, D. (2002). Managing Smile Risk. *Wilmott Magazine*, *1*, 84–108.

Hörfelt, P. (2005). On the Error in the Monte Carlo Pricing of Some Familiar European Path-Dependent Options. *Mathematical Finance*, *15*(2), 345–357. https://doi.org/10.1111/j.0960-1627.2005.00222.x

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hull, J. (2013). *Options, Futures and Other Derivatives: Global Edition: QMUL* (8th edition). Pearson Education UK. http://ebookcentral.proquest.com/lib/jyvaskyla-ebooks/detail.action?docID=5138073

Hutchinson, J. M., Lo, A., & Poggio, T. (1994). *A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks* (NBER Working

Paper 4718). National Bureau of Economic Research, Inc. https://econpapers.repec.org/paper/nbrnberwo/4718.htm

Itô, K. (1951). On Stochastic Differential Equations. *Memoirs of the American Mathematical Society*, *4*.

Jang, H., & Lee, J. (2019). Generative Bayesian neural network model for risk-neutral pricing of American index options. *Quantitative Finance*, *19*(4), 587–603. https://doi.org/10.1080/14697688.2018.1490807

Kac, M. (1947). Random Walk and the Theory of Brownian Motion. *The American Mathematical Monthly*, *54*(7), 369–391. https://doi.org/10.2307/2304386

Kingma, D. P., & Ba, J. (2014). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. https://doi.org/10.48550/arXiv.1412.6980

Lai, W.-N. (2014). Comparison of methods to estimate option implied risk-neutral densities. *Quantitative Finance*, *14*(10), 1839–1855. https://doi.org/10.1080/14697688.2011.606823

Liang, X., Zhang, H., Xiao, J., & Chen, Y. (2009). Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing*, *72*(13), 3055–3065. https://doi.org/10.1016/j.neucom.2009.03.015

Lindholm, A., Wahlström, N., Lindsten, F., & Schön, T. B. (2022). *Machine Learning: A First Course for Engineers and Scientists* (1st ed.). Cambridge University Press. https://doi.org/10.1017/9781108919371

Liu, X., Cao, Y., Ma, C., & Shen, L. (2019). Wavelet-based option pricing: An empirical study. *European Journal of Operational Research*, *272*(3), 1132–1142. https://doi.org/10.1016/j.ejor.2018.07.025

Longstaff, F. A., & Schwartz, E. S. (2001). Valuing American Options by Simulation: A Simple Least-Squares Approach. *The Review of Financial Studies*, *14*(1), 113–147.

Malliaris, M., & Salchenberger, L. (1993a). A neural network model for estimating option prices. *Applied Intelligence*, *3*(3), 193–206. https://doi.org/10.1007/BF00871937

Malliaris, M., & Salchenberger, L. (1993b). *Beating the best: A neural network challenges the Black-Scholes formula*. 445–449. https://doi.org/10.1109/CAIA.1993.366633

Merton, R. C. (1973). Theory of Rational Option Pricing. *The Bell Journal of Economics and Management Science*, *4*(1), 141–183. https://doi.org/10.2307/3003143

Nevasalmi, L. (2020). *Essays on economic forecasting using machine learning*. https://www.utupub.fi/handle/10024/150668

Nilsson, N. J. (1998). Neural Networks. In *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann. https://doi.org/10.1016/B978-0-08-049945-1.50008-3

Nyholm, S. (2022). Machine learning in macroeconomic forecasting. *Bachelor's Thesis*.

Ormoneit, D. (1999). A regularization approach to continuous learning with an application to financial derivatives pricing. *Neural Networks*, *12*(10), 1405–1412. https://doi.org/10.1016/S0893-6080(99)00078-7

Patil, A. R., & Subbaraman, S. (2021). Performance analysis of static hand gesture recognition approaches using artificial neural network, support vector machine and two stream based transfer learning approach. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-021-00831-7

Pohjonen, T. (2022). *Pricing and Hedging options with Neural Networks: A case of FTSE 100 index options*.

Rojas, R. (1996). The Biological Paradigm. In R. Rojas (Ed.), *Neural Networks: A Systematic Introduction*. Springer. https://doi.org/10.1007/978-3-642-61068-4_1

Ross, S. M. (2012). *Simulation*. Elsevier Science & Technology. http://ebookcentral.proquest.com/lib/jyvaskyla-ebooks/detail.action?docID=1044919

Ruf, J., & Wang, W. (2020). *Neural networks for option pricing and hedging: A literature review* (arXiv:1911.05620). arXiv. http://arxiv.org/abs/1911.05620

Schinazi, R. B. (2012). *Probability with Statistical Applications*. Birkhäuser Boston. https://doi.org/10.1007/978-0-8176-8250-7

Stergiou, C., & Siganos, D. (2006). Neural Networks. *Imperial College London, Department of Computing.*

Teddy, S. D., Lai, E. M.-K., & Quek, C. (2008). A cerebellar associative memory approach to option pricing and arbitrage trading. *Neurocomputing*, *71*(16), 3303–3315. https://doi.org/10.1016/j.neucom.2008.04.039

Tilley, J. (1993). 111 Valuing American Options in a Path Simulation Model. *Transactions of the Society of Actuaries*, *45*.

Tuominen, H., & Neittaanmäki, P. (2019). *Tekoälyn perusteita ja sovelluksia*. Jyväskylä: Jyväskylän yliopisto.

Tutorialspoint. (2017). *Artificial Intelligence—Neural Networks*. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_neural_networks.htm

Vähäkainu, P., & Neittaanmäki, P. (2018). *Tekoäly terveydenhuollossa.* Jyväskylä: Jyväskylän yliopisto.

Varian, H. (2014). Big Data: New Tricks for Econometrics. *American Economic Association*. https://www.aeaweb.org/articles?id=10.1257/jep.28.2.3

Wang, H. (2012). *Monte Carlo Simulation with Applications to Finance*. CRC Press LLC. http://ebookcentral.proquest.com/lib/jyvaskyla-ebooks/detail.action?docID=1648155

Zapart, C. (2002). Stochastic volatility options pricing with wavelets and artificial neural networks. *Quantitative Finance*, *2*, 487–495. https://doi.org/10.1080/14697688.2002.0000016

APPENDIX

APPENDIX 1 Notations and abbreviations of Table 3

C	Option price
C_{BS-X}	Option price given by the BSM. For X; H=BSM with historical volatility, V=BSM with volatility index such as VIX
HR	Hedging ratio
K	Strike price
S	Underlying's price
r	Interest rate
σ_{Cal}	Volatility from calibration
σ_H	Historical volatility
σ_{IM}	ATM implied volatility
σ_V	Volatility index such as VIX
τ	Time to maturity
$CVaR$	Conditional value at risk
DM	Diebold and Mariano test[9]
KS	Kolmogorov and Smirnov twosample test[10]
MAE	Mean absolute error
$MAPE$	Mean absolute percentage error
$MATE$	Mean absolute tracking error
ME	Mean error
MPE	Mean percentage error
MSE	Mean squared error
PE	Prediction error
R^2	Coefficient of determination

[9] See Diebold & Mariano (1995).
[10] See Lai (2014), where he explains the test in short.