Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Bjørn André Aaslund
Johannes Berge

X Hedging: An Explainable Artificial Intelligence Hedging Framework

Master's thesis in Industrial Economics and Technology Management
Supervisor: Rita Pimentel
Co-supervisor: Ying Ni
June 2022

NTNU
Kunnskap for en bedre verden

Bjørn André Aaslund
Johannes Berge

X Hedging: An Explainable Artificial Intelligence Hedging Framework

Master's thesis in Industrial Economics and Technology Management
Supervisor: Rita Pimentel
Co-supervisor: Ying Ni
June 2022

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Abstract

We develop a financial option hedging framework called *X Hedging* that utilises new Artificial Intelligence methods, is inherently explainable, and is adaptable to different market models, market frictions, and hedging instruments. The topic of pricing and hedging financial options is broadly studied in the financial literature. Recent methods use neural networks' ability to map complex non-linear relationships to create general and versatile methods, however, at the expense of explainability. We propose a hedging framework that uses gradient boosted decision trees to increase the explainability of the state-of-the-art frameworks without sacrificing performance. X Hedging is validated in experiments against the well-known option pricing and hedging model by Black and Scholes [1973], and the recent Deep Hedging model by Bühler et al. [2019], achieving the same performance. We exemplify how the method by Shapley [1953] achieves global explainability for X Hedging and "black-box" hedging models such as Deep Hedging. Thereafter, we discover that the derivative change of the underlying asset influence the final profit and loss along with the magnitude of the underlying asset value. Finally, we show that X Hedging complies with the newly proposed guidelines and regulations related to Explainable Artificial Intelligence through local explainability, highlighting the practical usability of the hedging framework in the industry.

Sammendrag

Vi utvikler et rammeverk for sikring *(eng: hedging)* av finansielle opsjoner. Rammeverket bruker nye metoder innen kunstig intelligens, er mulig å forklare, og tilpasselig til ulike markedsmodeller, markedsfriksjoner, og sikringsinstrument. Temaet som omhandler prising og sikring av finansielle opsjoner er omfattende studert i litteraturen. Nyere metoder bruker nevrale netts ferdigheter som omhandler å forbinde ikke-lineære forhold til å lage generelle og allsidige metoder. Dette kommer dog på bekostning av forklarbarheten *(eng: explainability)* til metoden. Vi foreslår et rammeverk for sikring som bruker gradientforsterkede beslutningstrær *(eng: gradient boosted decision trees)* til å øke forklarbarheten til de nyeste rammeverkene, uten reduksjon av ytelsen. X Hedging er validert i eksperimenter mot den velkjente opsjonsprisings- og sikringsmodellen av Black and Scholes [1973], og den nylige Deep Hedging-modellen utviklet av Bühler et al. [2019], hvor vi oppnår samme ytelse. Vi eksemplifiserer hvordan metoden til Shapley [1953] oppnår global forklaring for X Hedging og "black-box" sikringsmodeller som Deep Hedging. Deretter oppdager vi at den deriverte av det underliggende aktivumet påvirker det endelige resultatet sammen med størrelsen på den underliggende aktivumets verdi. Til slutt viser vi at X Hedging overholder de nylig foreslåtte retningslinjene og forskriftene knyttet til forklarbar kunstig intelligens *(eng: Explainable Artificial Intelligence)* gjennom lokal forklaring, noe som fremhever den praktiske anvendeligheten av sikringsrammeverket i bransjen.

Preface

This is a master's thesis in Industrial Economics and Technology Management (Indøk), written in the spring of 2022 for the course TIØ4900 - Financial Engineering, Master's Thesis at the Norwegian University of Science and Technology (NTNU).

In researching and developing financial models that use Artificial Intelligence, we have been able to combine two fields of great interest to us both, for which we are grateful. In addition, studying newly developed financial regulations related to Artificial Intelligence and implementing these in our model has been of great interest. We hope that the reader will appreciate the synergy of these topics as much as we have done.

We want to thank our supervisors, Rita Pimentel and Ying Ni. Their guidance throughout the entire process has been extremely valuable by proofreading, giving us detailed feedback, discussing complicated topics, and always being of help.

Trondheim, June 11th, 2022.

Bjørn André Aaslund
Johannes Berge

Contents

List of Figures

List of Tables

Abbreviations

The abbreviations are sorted in chronological order from when they appear in the thesis.

AI Artificial Intelligence

OTC Over-The-Counter

ML Machine Learning

XAI Explainable Artificial Intelligence

BIS Bank for International Settlements

OECD Organisation for Economic Cooperation and Development

CBOE Chicago Board Options Exchange

ISSM Institute for the Study of Security Markets

Quadratic CVaR Quadratic Conditional Value At Risk

CVaR Conditional Value At Risk

GBM Geometric Brownian Motion

SDE Stochastic Differential Equation

SGD Stochastic Gradient Descent

GBDT Gradient Boosting Decision Tree

GOSS Gradient-Based One-Side Sampling

EFB Exclusive Feature Bundling

MSE Mean Squared Error

SHAP SHapley Additive exPlanations

1 Introduction

Artificial Intelligence (AI) methods used in finance show great performance, speed, and versatility, but lack explainability. Explainability is especially important for regulatory oversight, risk management models, and ethical decision making. We contribute to the literature in the intersection between finance, AI, and regulations within the two fields. This is done by developing a financial model that utilises new AI methods, is inherently explainable, and is specifically developed for risk-takers such as market makers and *over-the-counter (OTC)* traders that trade in financial derivatives.

Options are popular financial derivatives that grant the holder the right to buy (call) or sell (put) an underlying asset at a predetermined price until a specific point in time. Market makers provide liquidity to options markets by buying and selling options from and to the participants in the market, even when there are no direct counterparts for the participants. The consequence is that the market makers take on risk associated with being obligated to deliver the underlying asset at a specific point in time. For options, the magnitude of this risk will depend on the option's maturity and how volatile the underlying asset is. The market makers are compensated for this risk by selling the option at a higher price than they buy it, thus profiting from the bid-ask spread. However, the market maker still wishes to offset the risk as much as possible and increase the possibility of profiting. They do this by *hedging* their short option positions; buying or selling a certain amount of the opposite position via a hedging instrument. The same is true for OTC traders, as the seller (writer) in an OTC option trade also wishes to hedge their short option position and mitigate the risk.

A considerable body of research has developed hedging models, or hedging frameworks[1], to define how much a hedger should invest in a specific hedging instrument at a particular point in time to neutralise the risk associated with the position. To produce hedging strategies, strict assumptions are commonly considered for the behaviour of the market, the input signals for the hedging model, and the hedging instruments. For instance, the famous option pricing and hedging model by Black and Scholes [1973] *(Black-Scholes model)* assumes zero market frictions, only the underlying asset price as input, and specific greeks as hedging instruments. Since the introduction of the Black-Scholes model, researchers have generalised and extended it to create frameworks that can handle a more realistic set of assumptions. The introduction and advancements of *machine learning (ML)* and AI in finance have coincided with the development of such frameworks. This ranges from neural network models that directly learn the Black-Scholes option prices and hedging ratios, e.g. Hutchinson et al. [1994], to deep reinforcement learning frameworks that learn the market dynamics without any prior assumptions and produce hedging strategies therefrom, such as Bühler et al. [2019].

As hedging models have moved from analytical formulas to more sophisticated AI models, the trend is increased generalisability but decreased explainability. Neural networks are well understood in terms of how they can approximate any continuous function [Hornik et al., 1989], but studying the neural network does not give insights into the characteristics of the function being approximated. Due to this, they are usually called "black-box" models, and it raises challenges related to regulatory demands on explainability through transparency for financial firms. Even though these models might produce accurate results, if the companies cannot report how the models made specific decisions, they may not be able to use them.

The field of developing AI models that can be explained is called *Explainable Artificial Intelligence (XAI)*. The *Bank for International Settlements (BIS)* presents in Prenio and Yong [2021] that several entities have recently issued guidelines and regulatory frameworks on AI governance in general terms and specifically within the financial sector, where XAI is highlighted as necessary. These include international organisations such as the *Organisation for Economic Cooperation and Development (OECD)* and G20, and governments of countries including the United States and Germany. Common for all the guidelines is that explainability through transparency and interpretability is a key issue for decision-making models that could require supervisory oversight and accountability. This highlights the relevance of studying XAI together with risk management and specifically option hedging frameworks.

We propose an explainable option hedging framework termed *X Hedging* that meets the demands of the most important regulations and guidelines on XAI. X Hedging uses inherently explainable ensemble

[1]The term *hedging model* relates to any method that produces hedging strategies, while the term *hedging framework* (or abbreviated *framework*) specifically denotes a more general hedging model. For instance, X Hedging and the model by Black and Scholes [1973] are hedging models, but only X Hedging is labelled as a hedging framework.

tree methods that recently have proven to match the performance of neural networks [Makridakis et al., 2022]. Furthermore, we apply an explainable framework to X Hedging and the current state-of-the-art *Deep Hedging* framework proposed by Bühler et al. [2019], and highlight the benefits of the former.

The thesis is structured as follows. A review of the current literature on hedging, AI in finance, and regulatory frameworks related to XAI is presented in Section 2. In Section 3 we present X Hedging along with relevant methods used for the experiments. The experimental results, along with an interpretation and discussion, are presented in Section 4. Section 5 concludes the thesis.

2 Literature Review

Black and Scholes [1973] propose in their well-known model, an analytical solution to pricing and hedging vanilla European options. They calculate the option greeks, which are the partial derivatives of the option value with respect to different variables, such as the underlying asset price (Δ) and volatility ($\mathcal{V}$). These greeks can be used to offset the risk of the option by trading the greek amount of the corresponding hedging instrument. A shortfall of the Black-Scholes model is that the strict assumptions it makes about the market often do not hold in reality, which can produce inaccurate pricing and hedging results. For instance, the model assumes no market frictions, but market frictions such as transaction costs are expected for real-world asset and options trading. This was already noted in the original paper by Black and Scholes [1973]. Furthermore, the model assumes constant volatility, but this is seldom true for real markets. A negative correlation between the volatility and the underlying asset price was first observed by Black [1976]. This was later confirmed by Swidler and Diltz [1992] in an empirical study of data from the *Chicago Board Options Exchange (CBOE)* and *Institute for the Study of Security Markets (ISSM)*. Via the same empirical study, the authors also show that transaction costs are present. Neither the Black-Scholes assumption of normally distributed asset returns holds because most markets have frictions and non-constant volatility. Instead, as noted by McDonald [2014] in Chapter 18.6 and papers like Mandelbrot [1963], Fama [1965], and Baillie and DeGennaro [1990], the distributions of asset returns frequently have leptokurtosis and fat tails contrary to the thin tail of a normal distribution.

In an attempt to make the Black-Scholes model more general and suitable in real-world market situations such that pricing and hedging can be done as accurately as possible, researchers have proposed relaxations of its strict assumptions. Transaction costs have been introduced by Leland [1985], Hodges and Neuberger [1989], Davis et al. [1993], and Atkinson and Alexandropoulos [2006]. Leland [1985] adds proportional transaction costs by discretising and adjusting the magnitude of the volatility parameter. The volatility increases with a positive proportional transaction cost, affecting the option price and the greeks used to hedge. Hodges and Neuberger [1989] state that the method proposed by Leland [1985] is not optimal and suggests using dynamic programming to offset the risk of an option with proportional transaction costs or other cost structures. This method maximises the expected utility of the terminal wealth. They are able to produce smaller hedging errors than Leland [1985]. Davis et al. [1993] use a similar approach to Hodges and Neuberger [1989] by ways of dynamic programming, but their method does not assume any particular market model, making it more general. Davis et al. [1993] prove that when they limit the transaction costs in the model to zero, the option values approach the Black-Scholes values. Atkinson and Alexandropoulos [2006] extend both Hodges and Neuberger [1989] and Davis et al. [1993] by increasing the number of underlying assets of the options while still allowing for transaction costs. They also provide an analytical solution to the difference in hedging strategies with and without transaction costs. Researchers have also extended the Black-Scholes model with other market frictions. Rogers and Singh [2010] address illiquidity, modelling it as a temporary impact on the underlying asset price based on the trading volume. This translates to a non-linear transaction cost, and they can still find the optimal hedging strategies. Bank et al. [2017] extend the temporary price impacts introduced by Rogers and Singh [2010], and propose to use the model by Bachelier [1900] instead of the Black-Scholes model, arguing that the former generalises better whenever market frictions are present.

To further generalise the Black-Scholes model, researchers have introduced non-constant volatility. This allows for the creation of a certain type of skew of the implied volatilities, a *volatility smile*. This is observed in real markets, and it is common knowledge for researchers, as noted by McDonald [2014], Chapter 24.1. Merton [1973] highlights that the original Black-Scholes model can be adjusted for time-varying volatility, but this assumes a time deterministic function for the volatility. The same author proposes a model in Merton [1976] where the asset prices follow a stochastic jump process. This is known as the Merton jump-diffusion model, where adding jumps creates volatility smiles. To obtain a similar volatility smile, a different approach is to allow for stochastic volatility and let there be a correlation coefficient between the underlying asset price and the volatility itself. Papers that have extended the Black-Scholes model to incorporate this stochastic volatility include Cox [1975], Hull and White [1987], Wiggins [1987], Scott [1987], and Heston [1993]. Cox [1975] propose a model where the volatility directly varies with the stochastic movement of the asset value. The model allows for negative correlation between asset price and volatility, which will result in a more realistic volatility skew. Hull and White [1987], Wiggins [1987], and Scott [1987] extend the work of Cox [1975] by allowing the correlated volatility to follow its own stochastic process. Heston [1993] comments that the disadvantage of these papers is that

the option valuation and hedging ratios need to be found via extensive numerical approximations. He rather suggests a closed-form formula for option valuation and hedging when the correlated volatility follows its own stochastic process.

The introduction of ML and AI into finance has allowed for greater generalisation of models, and a key contributor has been the neural network. Rosenblatt [1958] laid the foundation of the modern neural network as he developed the single-layer perceptron. This was later to become the multi-layer perceptron (MLP), what we know today as the modern neural network. Among the first papers to introduce these neural networks into finance are Felsen [1975] and Felsen [1976]. The author uses neural networks to recognise investment decision patterns and forecast stock price movements. He obtains forecasting accuracy better than the average of the best methods that existed at the time, but he states that his models should be used in combination with human decision making. Kimoto et al. [1990] extend the previous works of Felsen [1975] and Felsen [1976] by using modular neural networks to forecast stock movements on the Tokyo stock exchange. They are able to produce very accurate results, which at the time led to an increased interest in the use of neural networks for financial forecasting. Some later notable papers on this subject include Donaldson and Kamstra [1996], Huarng and Yu [2006], and Bao et al. [2017]. These papers specifically comment on the neural networks' ability to map non-linear relationships as a benefit for forecasting financial time series. According to Buchanan [2019], a report issued by the Alan Turing Institute, several other domains within finance have been able to utilise the performance benefits of neural networks. This includes algorithmic trading, where Sezer and Ozbayoglu [2018] notably propose a model that uses convolutional neural networks to treat financial time series as images. Also fraud detection, advisory services, and loan approvals are mentioned in the report. In those circumstances, classification algorithms that use neural networks to detect outliers have been popularised.

Hutchinson et al. [1994] introduce neural networks to hedging by proposing a non-parametric method that simulates GBMs and learns the Black-Scholes option prices and followingly solves for the hedge ratios. They apply their method out-of-sample to delta hedge S&P500 options, and compare their results to the corresponding Black-Scholes values. Carverhill and Cheuk [2003] extend Hutchinson et al. [1994] by investigating the best way to set up a neural network for option pricing and hedging. They argue that it is better to train the neural networks on option price changes instead of the actual prices and from there derive the hedge ratios. Amilon [2003] is also similar to Hutchinson et al. [1994] in that he uses the Black-Scholes model to compare his neural network model for pricing and hedging. However, he extends Hutchinson et al. [1994] by adding multiple input variables to the neural network in order to capture better the relationship between the prices of the underlying asset and the options. He concludes that his model is able to outperform the Black-Scholes model in terms of hedging and pricing of Swedish stock index call options. Further applying neural networks to hedging is Sutcliffe and Chen [2011]. They use deep neural networks to price and hedge short Sterling options. They conclude that their hybrid neural networks outperform both simpler neural networks and the Black-Scholes model on real-world market data. The paper highlights the advantages of neural networks when it comes to pricing and hedging non-vanilla options since they are more complex. Von Spreckelsen et al. [2014] is similar to the previous papers in using neural networks for pricing and hedging, but their model can be used in real-time for applications in high frequency trading.

Reinforcement learning, sometimes used with neural networks, has also allowed for greater generalisation of models in finance. Charpentier et al. [2021] give an overview of applications for reinforcement learning in finance, and including options pricing and hedging (risk management), there are mainly two other areas; optimal asset allocation (portfolio optimisation), and market impact modelling. Moody et al. [1998] and Moody and Saffell [2001] were among the first to apply reinforcement learning to portfolio optimisation as they use recurrent reinforcement learning to discover investment policies. Since the model is recurrent, previous investment decisions describe the current investment decision. At the time, they showed improved results compared to other trading strategies using real-world data. Deng et al. [2017] extend the work by Moody et al. [1998] and Moody and Saffell [2001] by introducing deep learning to their reinforcement learning model. Additionally, they describe their approach as a first-time using deep learning in combination with reinforcement learning for real-time trading. Almahdi and Yang [2017] extend the work on recurrent reinforcement learning, but differ from the beforementioned papers as they allocate assets based on the expected maximum drawdown risk measure. They conclude that they outperform current hedge fund benchmark in portfolio performance. Wang and Zhou [2020] develop a reinforcement learning-based portfolio selection model based on mean-variance selection in continuous time. They state that their approach is almost model-free and works with both real data and simulated

data. Interestingly, they choose a simpler reinforcement learning model compared to one with neural networks, due to the latter's "black-box" nature and difficulties with small data sets[2]. This is also remarked in our work. In market impact modelling, some notable papers include Spooner et al. [2018], Guéant and Manziuk [2019], and Baldacci et al. [2019]. They all use reinforcement learning to improve market maker decisions based on the information in order books, bid-ask spreads and other market factors. These papers advocate for the benefits of using reinforcement learning to model market maker behaviour.

Halperin [2017] is among the first to propose deep reinforcement learning to price and hedge options. Their Q-learning algorithm learns the hedging strategies and prices based on trading data for a replicating portfolio of a stock and a cash amount. It is first created in a parametric setting, but then it is able to learn the strategies without any assumptions about the market, i.e. it becomes model-free. The model is extended in Halperin [2019] on three specific points which include more comprehensive testing of the performance of the model, inverse reinforcement learning, and a model to price and hedge a portfolio of options instead of just a single option. Bühler et al. [2019] propose a general hedging framework based on reinforcement learning and neural networks termed Deep Hedging which is regarded as the state-of-the-art of hedging model. It is developed by researchers from the investment bank JP Morgan with the aim of creating a hedging framework that can be applied in practice without the need for strict assumptions or frequent human intervention. The framework does not assume any particular process for the underlying asset (model-free), neither does it need to compute the traditional greeks (greek-free), and it does not require any pricing model for the option. Compared to Halperin [2017] it also does not require any parametric initial setup. The authors state that their framework can be used with high dimensional derivatives, for any kind of liability, market dynamic, and market friction. Bühler et al. [2019] specifically build on the ideas of Dupire et al. [1994], who propose a model that is able to compute greeks that are not only dependent on the factors asset price and volatility, but also forward rate and implied volatilities. The paper also extends the previous work by Bühler [2019], where the author discusses how one may hedge a portfolio of derivatives for one step in the presence of transaction costs. Additionally, he advocates for the use of *quadratic conditional value at risk (Quadratic CVaR)*, a convex risk measure more suitable for optimisation in the hedging framework than regular *conditional value at risk (CVaR)*. This remark is taken into account in our work.

Bühler et al. [2019] show that their hedging framework produces accurate hedging results alongside generability and flexibility. However, we consider two drawbacks of the algorithm; the neural networks do not perform well with small data sets, and the neural networks provide low explainability. Financial datasets are often small (see for instance Lommers et al. [2021]), and some attempts have been made to address this issue within Deep Hedging. The original authors admit to the issue related to small data sets, and propose in Bühler et al. [2020] to use Variational Autoencoders to produce synthetic data that accompany real data. In a similar fashion, Cuchiero et al. [2020] use generative adversarial neural networks to calibrate local stochastic volatility models, with potential applications in Deep Hedging. Besides the issue of small data sets, the neural networks are called "black-boxes", meaning they do not provide any direct insight into how the decisions are made, i.e. they lack direct transparency and are thus not explainable by nature. This is partly due to the composition of multiple ridge functions within the neural networks. Neither are they neural networks equipped with interpretability, meaning it is difficult to logically interpret the relationship between inputs and outputs[3]. These are topics of XAI, an increasingly important part of regulatory frameworks in finance. In this thesis, we address both the shortcomings of Bühler et al. [2019], emphasising the latter.

The BIS report Prenio and Yong [2021] studies the current guidelines and regulations that financial institutions have to adhere to when it comes to AI governance, emphasising the importance of XAI. They identify explainability through transparency and interpretability as essential factors for financial institutions to implement in their services to provide insights to supervisors and customers. This means that financial institutions should easily be able to explain how decisions are made, even if they are performed by an algorithm. A practical test to verify the transparency of models is if the way they work can be understood by non-technical board members, then this would imply that they can be understood by supervisors as well. There are two main approaches to achieve this; use inherently explainable models (see

[2]Small data sets refers to datasets that are small in size, meaning lack of training samples.

[3]As the literature frequently mix the terms explainability, transparancy, and interpretability, we will for the purpose of clarity relate transparency to insights to the inner workings of a model, interpretability to the relationship between input and output, and explainability as the collection of the two terms.

Rudin [2019]), or use explainable frameworks around "black-box" models. The two approaches relate to two different types of explainability. It is important to distinguish between local and global explainability, as noted by Molnar [2022], Chapter 3. Local explainability concerns how a model explicitly reaches a particular decision by looking at individual components, while global explainability explains the overall workings of a model. To further clarify the terms, we relate transparency to local explainability, while interpretability helps explain a model locally and globally. Our work investigates a particular financial model with both of these facets in mind.

Prenio and Yong [2021] further state that explainability and specifically transparency is a prerequisite for some of the other guidelines on AI governance, such as reliability and accountability. The reason is that without explainability, it might be challenging to track and solve errors and hold anyone accountable for these errors. Prenio and Yong [2021] reference a report by the Financial Sector Supervisory Commission of Luxembourg, Curridori [2018], and explain that explainability is particularly important when models are "off-the-shelf". That is, models and frameworks that are not developed in-house but rather acquired from another entity or are open-source. It could be argued that this is case for general and open models such as Deep Hedging by Bühler et al. [2019]. Prenio and Yong [2021] specifically refer to the OECD AI Principles[4] and G20 AI Principles[5] as important contributions to the regulatory framework on XAI. These guidelines mention transparency as an important general factor for the development of benefitial AI. The European Union[6], Germany[7], Hong Kong[8], the Netherlands[9], Singapore[10] and United States[11] have all issued AI principles for their respective financial sectors, and they list explainability as an important factor for models whenever monetary gains are the objective.

We develop a general explainable hedging framework inspired by the work of Bühler et al. [2019], where we use tree ensemble methods instead of neural networks for estimation of the hedging strategies. To the best of our knowledge, this is the first time such an approach is proposed in the literature, and it neatly combines subjects of AI, finance, and regulations. A major benefit of tree ensemble methods is that they provide direct insight and transparency towards the decisions that are made within the model, while still being able to map complex relationships between input and output (non-linearities). Additionally, tree methods usually work better on small data sets compared to neural networks [Treboux et al., 2018], and depending on the structure of the data, they might outperform neural networks as in the medical paper Lundberg et al. [2020]. In finance, Krauss et al. [2017] combine gradient boosted trees with neural networks to improve out-of-sample predictions of statistical arbitrage. Furthermore, Gradojevic and Kukolj [2022] state that they address the interpretability of non-parametric option pricing models by use of random forest and the extreme gradient boosting models for different market regimes. They show the superiority of these explainable ML models in terms of accuracy and explainability compared to the neural networks pricing model presented by Culkin and Das [2017].

In this thesis we adopt the LightGBM model introduced by Ke et al. [2017]. LightGBM extends the primary proposal on Gradient Boosting Decision Trees by Friedman [2001], but also extend later models such as pGBRT by Tyree et al. [2011], and XGBoost by Chen and Guestrin [2016]. It differs from the previous ones because it delivers substantially higher speed at approximately the same accuracy. Papers applying LightGBM and XGBoost in finance are primarily concerned with classification tasks such as default predictions and customer loyalty predictions. These methods preferably require large datasets,

[4]Organisation for Economic Cooperation and Development (2019): *OECD Principles on Artificial Intelligence*: `https://oecd.ai/en/ai-principles`

[5]G20 (2019): *G20 ministerial statement on trade and digital economy*: `https://wp.oecd.ai/app/uploads/2021/06/G20-AI-Principles.pdf`

[6]European Insurance and Occupational Pensions Authority (2021): *Artificial intelligence governance principles: towards ethical and trustworthy artificial intelligence in the European insurance sector*: `https://www.eiopa.europa.eu/document-library/report/artificial-intelligence-governance-principles-towards-ethical-and_en`

[7]Federal Financial Supervisory Authority of Germany (2021): *Big data and artificial intelligence: Principles for the use of algorithms in decision-making processes*: `https://www.bafin.de/SharedDocs/Downloads/EN/Aufsichtsrecht/dl_Prinzipienpapier_BDAI_en.html`

[8]Hong Kong Monetary Authority (2019): *High-level Principles on Artificial Intelligence*: `https://www.hkma.gov.hk/media/eng/doc/key-information/guidelines-and-circular/2019/20191101e1.pdf`

[9]Netherlands Bank (2019): *General principles for the use of AI in the financial sector*: `https://www.dnb.nl/media/voffsric/general-principles-for-the-use-of-artificial-intelligence-in-the-financial-sector.pdf`

[10]Monetary Authority of Singapore (2018): *Principles to promote fairness, ethics, accountability and transparency in the use of AI and data analytics in Singapore's financial sector*: `https://www.mas.gov.sg/~/media/MAS/News%20and%20Publications/Monographs%20and%20Information%20Papers/FEAT%20Principles%20Final.pdf`

[11]National Association of Insurance Commissioners (2020): *Principles on Artificial Intelligence*: `https://content.naic.org/sites/default/files/inline-files/AI%20principles%20as%20Adopted%20by%20the%20TF_0807.pdf`

and a couple of notable papers include Zhou et al. [2019] and Machado et al. [2019]. In addition to using a more explainable model, we calculate Shapley values to improve the interpretability of X Hedging and Deep Hedging, and to highlight the local explainability of X Hedging that Deep Hedging does not have. In line with this work is Buckmann et al. [2021], who use Shapley values to explain neural networks that forecast financial time series. They are able to explain feature importance akin to conventional methods that are not "black-box".

3 Methodology

Our hedging framework, X Hedging, is inspired by Bühler et al. [2019], with some important new proposals that we highlight in this section. The main motivation for a market maker or an OTC trader that hedges against a liability Z is to minimize the risk of their total position to profit from the bid-ask spread. Hedging occurs at discrete time steps t_k, where the time index $k = 0, 1, 2, \dots n$. Further are $t_0 = 0$ and $t_n = T$, where T is the maturity of the liability Z. The purpose is to find the optimal strategy $\delta_k \in \mathbb{R}$ of the hedging instrument S_k at each discrete time step t_k[12]. The hedging positions are defined as

$$(\delta \cdot S)_T := \sum_{k=0}^{n-1} \delta_k \cdot (S_{k+1} - S_k). \tag{1}$$

One of the main advantages of a non-parametric hedging model is that it can account for market frictions such as transaction costs and liquidity constraints. To simplify the notation, we define $\delta_{-1} = \delta_n = 0$. The latter condition will also imply full liquidation in T. The market frictions' impact will depend on the change in the hedging strategies, and the total sum of the market frictions may be written generally as

$$C_T(\delta) := \sum_{k=0}^{n} c_k(\delta_k - \delta_{k-1}), \tag{2}$$

where $c_k : \mathbb{R} \to \mathbb{R}$, and $\delta_k - \delta_{k-1}$ is the change in the hedging strategy between two time steps. We focus on two types of market frictions, namely fixed and proportional transaction costs. As we have seen, these transaction costs have been extensively studied in the literature by Leland [1985], Hodges and Neuberger [1989], Davis et al. [1993], and even Bühler et al. [2019]. Given a cost constant $\kappa > 0$, fixed transaction costs are defined as

$$c_k(x) := \kappa \mathbb{1}_{|x| \geq \epsilon}, \tag{3}$$

where $\mathbb{1}$ is the identicator function which is zero for inputs below ϵ and one for inputs above ϵ, and ϵ is the hedging strategy change threshold. Proportional transaction costs are given by,

$$c_k(x) := \kappa S_k |x|. \tag{4}$$

The fixed transaction cost will impose the same cost for every change in hedging strategy that is larger than the threshold ϵ, and zero otherwise. The proportional transaction cost will impose a cost proportional to the amount of change in the hedging strategy and proportional to the value of the hedging instrument. That is, a higher value of the hedging instrument, and a higher change in hedging strategy will impose a greater cost compared to a smaller value of the hedging instrument and a smaller change in the hedging strategy.

The total portfolio position of the hedger at time step T, which translates to the final profit and loss (P&L), is given by

$$\text{P\&L}_T(Z, p, \delta) := -Z + p + (\delta \cdot S)_T - C_T(\delta), \tag{5}$$

where p amounts to the initial cash injection, and the premium of the liability is Z. Negative values of the $\text{P\&L}_T$ relate to losses while positive values relate to gains. The goal of the hedger is to minimize the expected value of a loss function ℓ associated with the P&L, i.e.

[12] The framework can be extended to d hedging instruments where the optimal hedging strategy for the d'th instrument $S_k^{(d)}$ is $\delta_k^{(d)} \in \mathbb{R}^d$.

$$\pi := \inf_{\delta} \mathbb{E}[\ell(\mathrm{P\&L}_T(Z, p, \delta))] \tag{6}$$

The optimal choice of δ in Equation (6) is based on the information available to the hedger at each time step t_k, denoted by I_k. We use $I_k = S_k$. Since we define the hedging instrument S_k as the value of an asset at time t_k, I_k may be interpreted as the information about the movement of the underlying asset at time step t_k[13].

We define the liability Z as a European vanilla call option with payoff $\max\{S_T - K, 0\}$, where S_T is the value of the underlying asset at maturity, and K is the option strike price[14]. In summary, the framework hedges a vanilla call option by trading the underlying asset, based on the underlying asset's movement. This is equivalent to standard delta-hedging by Black and Scholes [1973], providing a solid reference point to benchmark our hedging model. Under the risk-neutral measure, we assume the underlying asset value S follows a *Geometric Brownian Motion (GBM)*,

$$dS_t = (r - \lambda)S_t dt + \sigma S_t dW_t, \tag{7}$$

where S_t is the continuous price of the underlying asset, dS_t is the incremental change in the underlying asset, r is the constant interest rate, λ is the continuous dividend yield, σ is the annual volatility, and $\{W_t : t \geq 0\}$ is a standard Brownian motion. The analytical solution to the *stochastic differential equation (SDE)* in Equation (7) is given by

$$S_t = S_0 e^{(r - \lambda - \frac{\sigma^2}{2})t - \sigma W_t}, \tag{8}$$

where $S(0)$ represents the initial asset price. To simulate the evolution of the asset price, we discretize Equation (8) for the discrete time steps t_k as follows,

$$S_{k+1} = S_k e^{(r - \lambda - \frac{\sigma^2}{2})\Delta t - \sigma\sqrt{\Delta t} W_k}, \tag{9}$$

where $\Delta t = t_{k+1} - t_k$ is the incremental time change.

Deep Hedging by Bühler et al. [2019] uses a set of deep neural networks to map the information set $\{I_0, I_1, \ldots, I_n\}$ to the hedging strategies δ_k. A feed-forward neural network $\delta_k := F_k(I_k; \theta_k)$ for each time step t_k will be a function of the information at time t_k given the neural network parameters θ_k. The objective of the neural networks at each time step t_k is to optimize the parameters θ_k such that π in Equation (6) is obtained. This is achieved by obtaining a loss for each input and using backpropagation and *stochastic gradient descent (SGD)* to update the weights. For an in-depth explanation of these key concepts of neural networks we refer to Goodfellow et al. [2016]. Bühler et al. [2019] prove that the neural networks theoretically find the optimal hedging strategy by applying the universal approximation theorem. We use Deep Hedging to benchmark the performance of X Hedging, and to highlight the advantages of X Hedging.

To make Deep Hedging more explainable we propose to replace the neural networks and followingly the loss gradient calculations with more explainable tree ensemble methods and a custom loss function setup and optimization. This is the basis for X Hedging. We replace the neural networks $F_k(I_k; \theta_k)$ with a specific *gradient boosting decision tree (GBDT)* method $G_k(I_k; \dot{\theta}_k)$ for each time step t_k where $\dot{\theta}_k$ are the parameters of the GBDTs. GBDT is a class of tree ensemble method first proposed by Friedman [2001], and works by sequentially creating weak decision trees $g_{km}(I_k; \ddot{\theta}_k)$ for $m \in \mathbb{N}_0$ where the model parameters $\ddot{\theta}_k$ are selected to improve the loss of the previous weak decision tree. This is called boosting, and the final ensemble is a combination of the weak decision-trees, $G_k(I_k; \dot{\theta}_k) = \sum_m w_{km} g_{km}(I_k; \ddot{\theta}_k)$, where w_{km} is the importance weight associated with the m-th weak decision tree at time step t_k. An

[13] The framework can be extended to include multiple information sources, e.g. news sentiment analyses or trader opinions. In this case, $I_k^{(u)} \in \mathbb{R}^u$ would denote the u-th information source at time t_k.

[14] Any type of liability, or financial derivative, may be used with this hedging framework as long as the payoff is well-defined.

important attribute of these tree methods is that they provide explainability of each output by visually tracking every decision the model makes along the tree.

The specific GBDT method we implement is LightGBM proposed by Ke et al. [2017]. The key contribution of LightGBM is that it is more efficient than other GBDT models while it still produces accurate results. Ke et al. [2017] attributes this to their two novel techniques; *Gradient-Based One-Side Sampling (GOSS)* and *Exclusive Feature Bundling (EFB)*. GOSS works by using all the instances with large gradients and randomly sampling the ones with low gradients. This is an alternative to simply using a subset of the instances. EFB reduces the number of features by bundling the most important ones together.

Figure 1 shows both X Hedging and Deep Hedging for the first time step, an arbitrary time step, and the last time step.

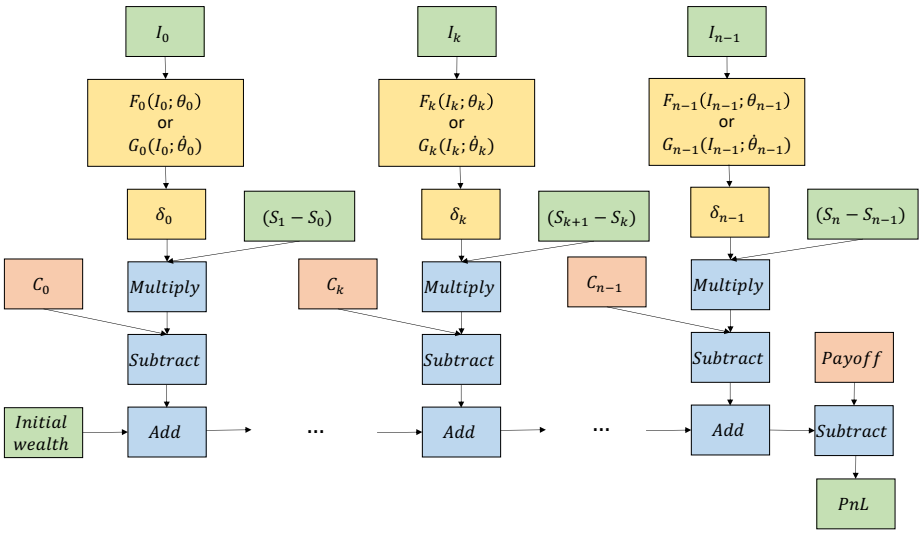


Figure 1: Visual overview of X Hedging and Deep Hedging frameworks. Green boxes contain inputs and outputs. The blue boxes represent mathematical operations. The orange boxes represent all the terms that are subtracted in Equation (5), and the yellow boxes are the neural network or tree-based model and their output.

An important difference between X Hedging and Deep Hedging is that the former cannot use SGD and backpropagation of the loss function $\ell(\text{P\&L}_T)$ to obtain optimal values of $\dot{\theta}_k$ for Equation (6). This is due to how the parameters are updated in the trees in LightGBM. Instead, each LightGBM model G_k needs to be fitted individually based on the global loss function $\ell(\text{P\&L}_T)$. After fitting all the LightGBM models sequentially for one iteration, we train each model in random order to introduce stochasticity. The stochasticity is introduced to reduce the probability of finding a local minimum, instead of a global one. This uses the same principles as SGB to avoid ending up in local minima. As far as we know, this is a novel approach to optimizing a reinforcement learning model, consisting of multiple tree ensemble models.

Furthermore, compared to the neural networks, the LightGBM models need not only the gradients of the loss function, but also the Hessians. That is, the second derivate of the loss function $\ell(\text{P\&L}_T)$ with respect to δ_k. The complexity of the loss function $\ell(\text{P\&L}_T)$ makes it difficult to analytically derive a formula for the gradients and the Hessians. Therefore, a computational graph of $\ell(\text{P\&L}_T)$ is created to simplify the calculation process. The graph uses automatic differentiation, more specifically reverse-mode auto differentiation, for fast computation of the first and second order derivatives. Assume that we want to calculate the gradients and the Hessians for $G_k(I_k;\dot{\theta}_k)$. For Figure 1, the final green box with $\text{P\&L}_T$ needs to be differentiated with respect to the yellow boxes with δ_k. This can be efficiently computed by using the chain rule.

LightGBM regressions are constructed by multiple trees, where each node in the trees divide the data based on a greater or equal condition for one feature. Therefore, the final regression line will consist of multiple orthogonal lines, making it non-smooth and noisy. However, the hedging strategies δ_k are

assumed to be smooth functions since the setup of the framework resembles delta hedging by Black and Scholes [1973]. Based on this assumption, a Savitzky–Golay filter, proposed by Savitzky and Golay [1964], is used to smooth the regression line. The smoothing is performed by a convolution, where successive adjacent data points are fitted to a second order polynomial with linear least squares. This step is only used for inference after the models are trained. The smoothing increases the performance of X Hedging without losing the benefit of explainability and transparency of the framework. Smoothing is a common approach to preprocess financial data. Meinl and Sun [2015] review some methods that are used to smooth financial data, and explain the importance of this process.

The type of loss function ℓ in Equation (6) inherently decides what the hedger wants to achieve. We consider three different loss functions. The first is the *mean squared error (MSE)*, defined as

$$\text{MSE}(Y) = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y}_i - Y_i)^2, \tag{10}$$

where N is the number of samples, $\hat{Y}_i$ are the estimated target values, and Y_i are the true target values. In our case, with MSE as the loss function, we are minimizing the mean squared error between the calculated $\text{P\&L}_T$ in Equation (6) and zero, which is our target. Zero is the target such that the bid-ask spread, which is not included in Equation (5), will be the profit of the hedger. The second loss function we consider is the Quadratic CVaR, defined by Bühler [2019], and derived from the CVaR. CVaR is a convex risk measure, and has some properties found to be useful for financial portfolio optimization. These properties include convexity (diversification of assets), monotonocity (less cash injection is needed in a better portfolio), and cash-invariance (cash deposits reduce the risk equivalent to the deposit value). Convex risk measures are studied in the literature by Xu [2006], Föllmer and Schied [2008], İlhan et al. [2009], and Bühler et al. [2019], among others. The CVaR at $\alpha\%$ level is the expected return on the portfolio in the worst $\alpha\%$ of the cases, defined for any random variable Y as

$$\text{CVaR}_\alpha(Y) = \frac{1}{\alpha}\int_0^\alpha \text{VaR}_\gamma(Y)d\gamma, \tag{11}$$

where $\text{VaR}_\gamma(Y)$ is the value at risk at the $\alpha\%$ level. Using CVaR as the loss function means that we are specifically punishing the hedging decisions that result in very large losses. We are minimizing the risk of the tails in the distribution of $\text{P\&L}_T$. This transforms the learning problem from a supervised learning problem when Equation (10) is used, to a reinforcement learning problem. This is characterized as a reinforcement learning problem since there are no targets, and the reward function is in our case the convex risk measure. Let Y' be the set with the $\alpha\%$ smallest values, then the discrete CVaR is calculated by,

$$\text{CVaR}_\alpha = \frac{1}{|Y'|}\sum_{y'\in Y'} y', \tag{12}$$

where y' is the $\alpha\%$ lowest $\text{P\&L}_T$. The second derivative of Equation (12) with respect to y' is zero. Since LightGBM needs both the gradients and the Hessians of ℓ, we use Quadratic CVaR, also proved to be a convex risk measure. Bühler [2019] argues that Quadratic CVaR is an efficient risk measure for optimization, and that it is easy to interpret the result because of the relation to the traditional CVaR. The second derivative of Quadratic CVaR is non-zero, thus applicable for X Hedging. The discrete Quadratic CVaR is given by,

$$\text{Quadratic CVaR}_\alpha = \frac{1}{|Y'|}\sum_{y'\in Y'} (y')^2. \tag{13}$$

The proposed algorithm for X Hedging is described in Algorithm 1.

Algorithm 1 X Hedging

T : number of time steps
N : number of asset paths
K : options strike price
$S : T \times N$ simulated asset prices
Z : European vanilla call payoff, $\max\{S_T - K, 0\}$
c : market friction for each time step t_k. Either 0, fixed, or proportional.
ℓ : loss function, MSE or Quadratic CVaR
G : n-1 LightGBM models
J : Number of iterations for random training
δ_0 : Black-Scholes delta
for t from n to 1 **do**
    $\delta_t \leftarrow G_k(S_k)$ for all N paths
    Compute loss ℓ between output and target
    Compute gradient $\frac{\partial \ell(\text{P\&L}_T)}{\partial \delta_k}$ and Hessian $\frac{\partial^2 \ell(\text{P\&L}_T)}{\partial \delta_k^2}$
    Update model weights of G_k
end for
R : Sample n random values from 1 to n
for j in J **do**
    for k in R **do**
        $\delta_k \leftarrow G_k(S_k)$ for all N paths
        Compute loss ℓ between output and target
        Compute gradient $\frac{\partial \ell(\text{P\&L}_T)}{\partial \delta_k}$ and Hessian $\frac{\partial^2 \ell(\text{P\&L}_T)}{\partial \delta_k^2}$
        Update model weights of G_k
    end for
end for
Calculate $\text{P\&L}_T(Z, p, G_k)$, and apply Savitzky–Golay filter.

In our experiments, we consider as a baseline hedging a European vanilla call option with an underlying asset that follows a GBM, and we assume no market frictions. The Black-Scholes model can be used to solve for the hedging strategies in this setup. To offset the risk produced by the movement of the underlying asset at time t, Δ_t underlying assets are purchased, where

$$\Delta_t = e^{-\lambda(T-t)} N(d_1), \tag{14}$$

where $N(x)$ is the cumulative function of a standard normal distribution, and d_1 is defined by

$$d_1 = \frac{\ln(S/K) + (r - \lambda + 0.5\sigma^2)}{\sigma\sqrt{T}}. \tag{15}$$

Replacing δ with Δ in Equation (5) we get the baseline model $\text{P\&L}_T(Z, p, \Delta)$.

To benchmark X Hedging when we add transaction costs, we use the extended Black-Scholes model by Leland [1985]. This particular extended model is selected due to its simplicity and wide adoption in the literature. Leland [1985] shows that the Black-Scholes delta hedging with proportional transaction costs c can be obtained by discretizing the Black-Scholes equation and transforming the variance σ^2 into

$$\hat{\sigma}^2(\sigma^2, k, \Delta t) = \sigma^2 \left[1 + \kappa \frac{\sqrt{2/\pi}}{\sigma\sqrt{\Delta t}}\right]. \tag{16}$$

We want to compare the hedging strategy obtained by the Black-Scholes model without and with transaction costs to the hedging strategies from Deep Hedging and X Hedging. For clarity, we term these

hedging models *BS Hedging* and *BS-L Hedging*. BS Hedging refers to the hedging model that uses the Black-Scholes model, while BS-L Hedging refers the extended Black-Scholes model by Leland [1985].

To achieve global explanation of X Hedging, as well as Deep Hedging and the benchmark Black-Scholes hedging models, we use Shapley values to interpret the relationship between input and output in the model. *SHapley Additive exPlanations (SHAP)*[15] performs a complex sensitivity analysis which is based on Shapley values. SHAP only requires a model that maps inputs to outputs and may be used to interpret all prediction models. Lundberg and Lee [2017] explains how Shapely values are calculated for machine learning models. Shapley values are a solution concept from cooperative game theory, first introduced by Shapley [1953]. The concept works by assigning each cooperative game a unique distribution of a total surplus generated by the coalition of all players. The Shapley values are calculated by

$$\phi_i(v) := \sum_{U \subseteq M \setminus i} \frac{|U|!(m - |U| - 1)!}{m!}(v(U \cup i) - v(U)), \tag{17}$$

where m is the total number of players, M is the set of all players, U is a coalition of players, $v(U)$ is the worth of the coalition, and i is one player. This concept has become a valuable theoretical principle for XAI. It transforms the setting by treating each feature as a player, the set of features as a set of players, and the model as a game. The calculated Shapley values are interpreted as the contribution of each feature to the prediction.

To compare the different hedging models, we need a statistical distance between the resulting histograms of the final P&L$_T$. We use Kullback–Leibler divergence or relative entropy, proposed by Kullback and Leibler [1951], which is given by,

$$D_{KL}(P||Q) = \sum_{x \in X} P(x) \log\left(\frac{P(x)}{Q(x)}\right), \tag{18}$$

where P and Q are two discrete probability distributions, and X is the set of all values of x with non-zero probability for P and Q distributions. The KL-divergence is non-negative and does not have any upper limit. A KL-divergence of zero indicates that the two probability distributions are identical. A disadvantage of the KL-divergence is that it is not symmetric, meaning, $D_{KL}(P||Q) \neq D_{KL}(Q||P)$. This is a problem when we want to compare models without any benchmark model. Therefore, we need a symmetric statistical distance to compare probability distributions when a reference distribution does not exist. The Jenssen-Shannon divergence (JS-divergence), defined by Lin [1991], is a statistical distance that is based on KL-divergence, and given by,

$$D_{JS}(P||Q) = \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M), \tag{19}$$

where $M = \frac{1}{2}(P + Q)$. Since JS-divergence is symmetric, we have $D_{JS}(P||Q) = D_{JS}(Q||P)$. As we move from the baseline models in our experiments to add more complexity, this is an advantageous feature. As with KL-divergence, a JS-divergence value of zero indicated that the two probability distributions being compared are identical.

[15] Documentation: `https://shap.readthedocs.io/en/latest/index.html`

4 Results and Discussion

Various numerical experiments[16] of X Hedging and Deep Hedging are considered. At first, we validate the performance of X Hedging in Section 4.1 to show that the framework produces accurate hedging strategies in various configurations, and that it is on-par with Deep Hedging and the Black-Scholes benchmarks. The configurations include different market frictions, including proportional and fixed transaction costs, no transaction costs, and two types of loss functions, namely MSE and Quadratic CVaR. Then, in Section 4.2 we comment on the small data issue raised in Section 2 and Section 3 by analysing X Hedging and Deep Hedging for a varying number of in-sample training paths. Finally, in Section 4.3 we discuss the explainability of X Hedging and Deep Hedging based on the SHAP framework, and the inherent benefits of the tree ensemble method in X Hedging.

The parameters used to simulate the underlying asset for the vanilla European call options are presented in Table 1. In Sections 4.1 and 4.3 the hedging models are trained in-sample on 20 000 simulated GBM paths and tested out-of-sample on 10 000 other simulated GBM paths. In Section 4.2 the in-sample training paths varies, but the number of out-of-sample test paths remains the same. The parameters used for the LightGBM models in X Hedging[17] and for the neural networks in Deep Hedging[18] are given in Tables 2 and 3, respectively. Appendix A provides an overview of all experiments conducted in this entire section along with their different configurations.

S_0	K	r	λ	σ	T	n	N
1	1	0.0	0.0	0.2	1	10	20 000

Table 1: GBM and option parameters used to simulate the underlying asset prices.

N. data leaf	N. leaves	N. boost rounds	Learning rate	Early stopping	Iterations
5	$10 + j$	$10 + 5j$	0.1	10	15

Table 2: Parameters used for the LightGBM models in X Hedging.

Activation function	Optimizer	Initial learning rate	Neurons	Batch size	Epochs
tanh	Adam	0.01	32	1024	10000

Table 3: Parameters used for the neural networks in Deep Hedging.

4.1 Performance Validation

We start by validating X Hedging with MSE defined in Equation (10) for these scenarios: no market frictions, proportional transaction costs, and fixed transaction costs. Table 4 summarises the results from experiments with X Hedging, Deep Hedging, and BS Hedging when there are no market frictions, and MSE defined is the loss function ℓ. In this base-case, BS Hedging, with deltas shown in Equation (14), is used as the benchmark because the Black-Scholes assumptions hold. Observable from the means, standard deviations and JS-Divergence values, the results show that both X Hedging and Deep Hedging

[16]The computations were performed on a MacBook Pro with 2.2GHz Intel Core i7 CPU - 4 core, and 16Gb RAM.

[17]The minimum amount of data points in each leaf node is given by *N. data leaf*. Larger values will result in lower variance. The maximum number of leaves for each tree, where j is the current iteration, is given by *N. leaves*. For every iteration the maximum number of leaves will increase in order to create a more complex model without overfitting in the beginning. The number of boosting rounds used to reduce the loss of the model output, where j is the current iteration, is given by *N. boost rounds*. For every iteration the number of boosting rounds will increase to create a more complex model. The learning rate multiplier for each weak learner is the *learning rate*. The maximum of boosting rounds without loss improvement is set by *early stopping*. The number of times we randomly iterate over each model at each time step is given by *iterations*.

[18]The *activation function* tanh is used due to its similarities with the Black-Scholes hedging strategies, which should make finding the correct strategies more efficient. The stochastic gradient descent *optimizer* Adam is chosen due to its well-documented performance. The initial learning rate 0.01 is chosen as a standard value. The size of the neural networks, the number of *neurons* is set to 32 as experiments showed that this gave good results. The same argument holds for the *batch size* and the number of *epochs*.

are similar to BS Hedging. All the hedging models have a mean close to the target value zero, which is a good indication that they produce the optimal hedging strategies. Both X Hedging and Deep Hedging produce JS-divergence values (see Equation (19)) with BS Hedging lower than 0.001[19]. The values are not exactly zero, but some of the error in the JS-divergence might be a result of discretizing the P&L$_T$ into 50 bins. The high similarity in the P&L$_T$ distributions indicates that both frameworks are good estimators in this base-case[20]. Observing the mean values, and comparing the JS-divergence values between X Hedging and BS Hedging, and Deep Hedging and BS Hedging, we that X Hedging is closer to BS Hedging and has a higher mean. However, X Hedging has a slightly higher standard deviation. These differences between X Hedging and Deep Hedging are not large enough to conclude that one of the frameworks outperforms the other. Noting the different computational times in Table 4, it is as expected that the analytical solution of BS Hedging uses significantly less time than the two AI models. What is more interesting is that the computational times of X Hedging and Deep Hedging are fairly similar. This similarity could be interpreted as advantageous for X Hedging as it does not limit its practical applicability compared to Deep Hedging. However, it should be noted that more sophisticated hyperparameter optimization could lead to faster convergence for the neural networks in Deep Hedging. The histograms in Figure 2 reveal not only the similarities of the three P&L$_T$ distributions, but also the characteristics of them. All histograms are similar to a normal distribution, but with a longer tail for negative values. This implies that the hedging strategy will give big losses in some edge cases, but that most of the strategies produced are acceptable.

	Mean	**St.Dev**	**JS(DH||z)**	**JS(XH||z)**	**Time**
BS	-0.000192	0.021327	0.000437	0.000398	10.0
DH	-0.000152	0.021415	-	0.000632	803.0
XH	-0.000146	0.021482	-	-	872.0

Table 4: Summary of results from X Hedging (XH), Deep Hedging (DH), and BS Hedging (BS) with no market frictions and MSE as the loss function. Presented is the mean and the standard deviation for each P&L$_T$ distribution together with the JS-divergence values between them. The computational time is given in seconds, and z is a placeholder.

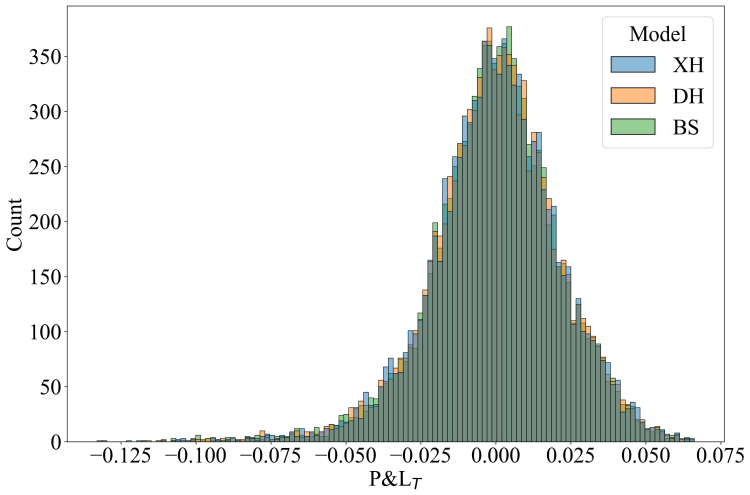


Figure 2: Histograms for X Hedging (XH), Deep Hedging (DH), and BS Hedging (BS) with no market frictions and MSE as the loss function.

[19] In the development of these experiments we have analysed histograms of multiple P&L$_T$ distributions, and come to the conclusion that a JS-divergence value of 0.001 is a reasonable performance threshold to indicate adequate similarity between two distributions.

[20] We report results related to the P&L$_T$ as opposed to ℓ(P&L$_T$) to compare results between experiments that use different loss functions.

Figure 3 shows the hedging strategies for X Hedging, Deep Hedging, and BS Hedging with no market frictions and MSE as the loss function ℓ, at each time step t_k. The hedging strategies for $k = 0$ is not visualised since S_0 has the same value for all simulated paths, and hence the strategies will only consist of one point at S_0. Neither do we show the strategy for $k = 10$ since hedging does not occur at the last time index, indeed it is where the P&L$_T$ is calculated. From $k = 1$ to $k = 9$, the hedging strategies from X Hedging and Deep Hedging approach the delta strategies from BS Hedging. One can observe that X Hedging and Deep Hedging are most different from BS Hedging for small and large values of S when $k = 1$ and $k = 2$. This is expected since the simulated data does not reach these ranges for small k values. X Hedging has an almost horizontal line in these intervals, which correspond to the leaves with the lowest and highest hedging strategies. This lets us observe which ranges the simulated underlying asset prices are within. It is important to notice that the seemingly inaccurate strategies for the early time steps in Figure 3 do not particularly affect the final P&L$_T$ negatively because very few paths will reach the low and high values. It is also evident that the shapes of the hedging strategies are approximately linear at the first time steps, but then they approach step functions towards $k = 9$. This is natural since it is harder to predict where each path will end up at earlier time steps compared to the ones closer to maturity. Meaning, at early time steps few decisive hedging decisions are made, while at the later time steps the hedging models either hedge a lot or nothing. One final observation about the strategies is the difference in smoothness. X Hedging produce less smooth strategies than the other two hedging models. This is because the decision trees in X Hedging produce uneven strategies, and even applying the Savitzky–Golay filter does not make the strategies completely smooth.

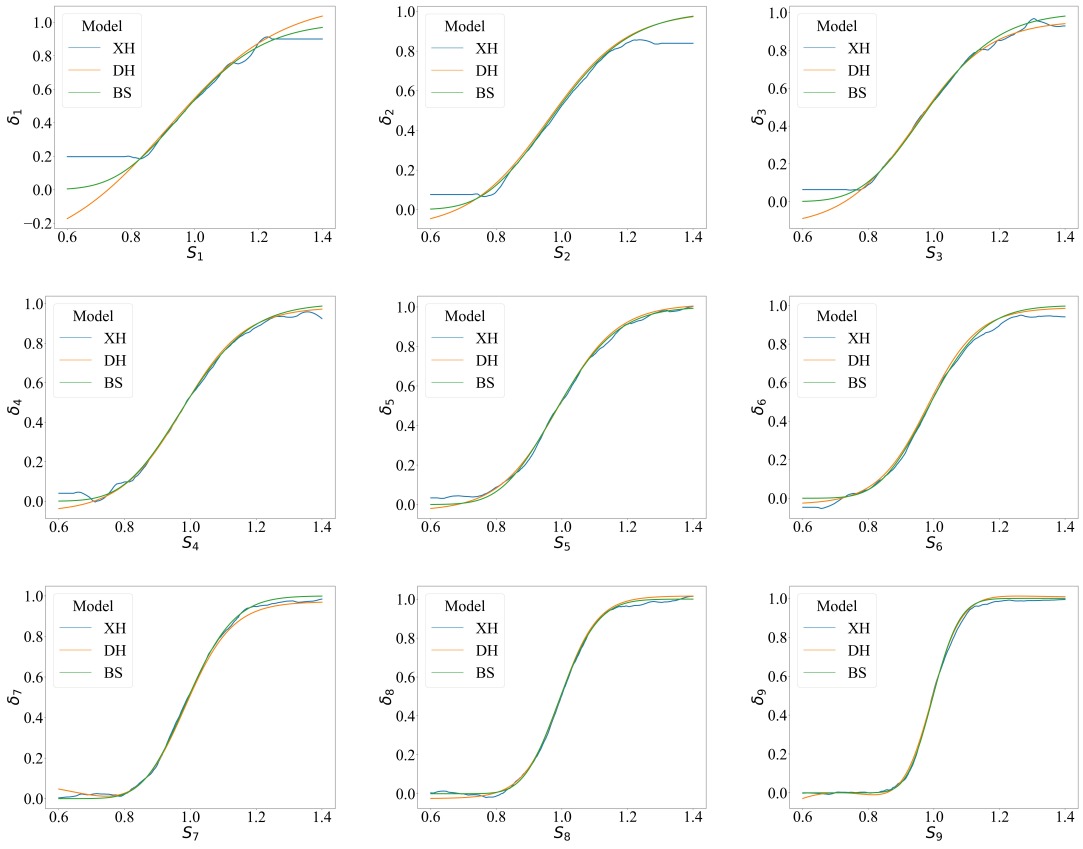


Figure 3: Hedging strategies for X Hedging (XH), Deep Hedging (DH), and Black-Scholes (BS) from $k = 1$ to $k = 9$. No market frictions are considered and MSE is the loss metric. S_k denotes the underlying asset value, and δ_k denotes the corresponding hedging strategy at time index k.

Table 5 summarises the results from experiments with X Hedging, Deep Hedging, and BS-L Hedging when proportional transaction costs defined in Equation (4) are considered and MSE is the loss function ℓ. BS-L Hedging is used to benchmark X Hedging and Deep Hedging by adjusting BS Hedging with a different volatility, as shown in Equation (16). In these experiments, we consider the proportional

transaction cost constant $\kappa = 0.005$. This value is realistic in terms of applications for real markets, and it is in the interval of values that Leland [1985] considers. The mean and standard deviations in Table 4 are very similar for the three hedging models. The JS-divergence value between X Hedging and Deep Hedging once again highlights that our novel approach is able to produce similar results to Deep Hedging. However, comparing X Hedging and Deep Hedging to the benchmark BS-L Hedging, we see that the JS-divergence values are significantly higher than without transaction costs in Table 4. In this instance it is worth noting that the values are still roughly at the performance threshold of 0.001, indicating good performance towards the benchmark. We should also mention that X Hedging and Deep Hedging handle the transaction costs in a different manner to BS-L Hedging, which adjusts the magnitude of the volatility parameter. As a consequence, we can hypothesize that X Hedging and Deep Hedging in fact should be the preferred methods since they handle the transaction costs directly and in a general sense, instead of indirectly via the specific volatility parameter. Comparing Table 4 and Table 5 it should also be mentioned that after adding the proportional transaction cost to the hedging models, their mean values decrease. This is expected since the market frictions make it more expensive to hedge the options.

	Mean	**St.Dev**	**JS(DH||z)**	**JS(XH||z)**	**Time**
BS-L	-0.007318	0.022197	0.001039	0.001285	10.0
DH	-0.007338	0.021980	-	0.000651	850.0
XH	-0.007299	0.022011	-	-	967.0

Table 5: Summary of results from X Hedging (XH), Deep Hedging (DH), and BS-L Hedging (BS-L) with proportional transaction costs and MSE as the loss function. Presented is the mean and the standard deviation for each P&L$_T$ distribution together with the JS-divergence values between them. The computational time is given in seconds, and z is a placeholder.

The histograms of the P&L$_T$ distributions of X Hedging, Deep Hedging and BS-L Hedging in Figure 4 show how similar results the three hedging models produce. The characteristics of the distributions are similar to the ones in Figure 2, but there are a few differences. For one, the distributions are shifted to the left. As mentioned, this implies a more negative mean, but one can observe that this indicates higher values for the left-most outliers. Also, BS-L Hedging separates itself from X Hedging and Deep Hedging in a more prominent way around the mean. The standard deviations from this experiment are higher than the experiment without any market frictions. A higher standard deviation may indicate a more difficult hedging problem since the hedging model is not able to remove as much variation in the P&L$_T$. Therefore, it can be argued that it is harder to produce accurate hedging strategies when market frictions are added.

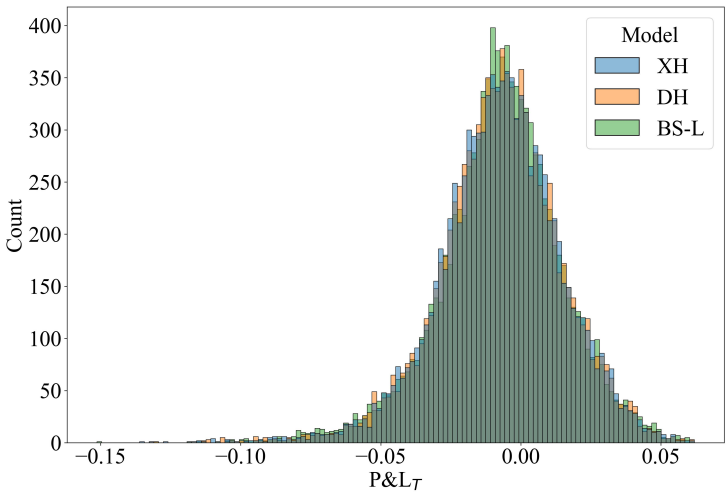


Figure 4: Histograms for X Hedging (XH), Deep Hedging (DH), and BS-L Hedging (BS-L) with proportional transaction costs and MSE as the loss function.

The hedging strategies for X Hedging, Deep Hedging, and BS-L Hedging are presented in Figure 5 when proportional transaction costs are considered and MSE is the loss function. Equivalent to Figure 3, the strategies are the most dissimilar at early time steps, before they approach almost the same strategy towards the end. As in the previous experiment, the strategies from X Hedging approximate well the benchmark, even with the horizontal lines at the early time steps.

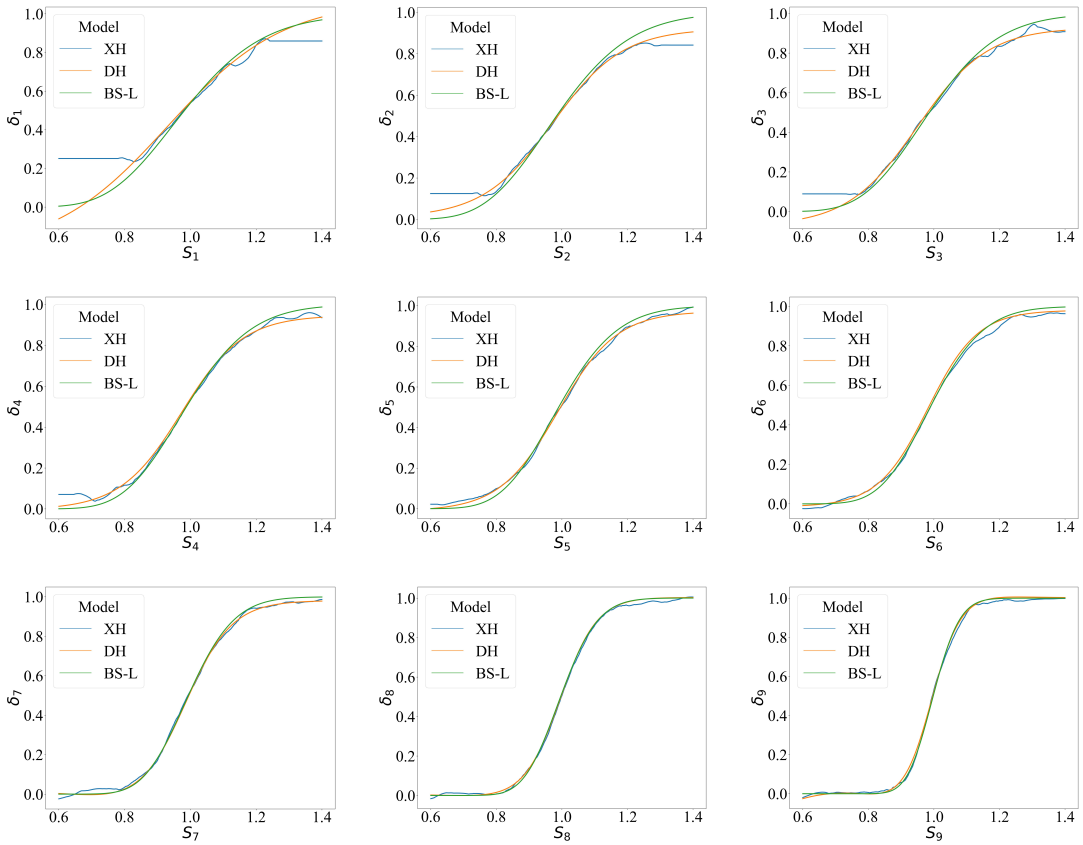


Figure 5: Hedging strategies for X Hedging (XH), Deep Hedging (DH), and Black-Scholes Leland [1985] (BS-L) from $k = 1$ to $k = 9$. Proportional transaction costs are considered and MSE is the loss metric. S_k denotes the underlying asset value, and δ_k denotes the corresponding hedging strategy at time index k.

Table 6 summarises the results from experiments with X Hedging and Deep Hedging when fixed transaction costs as defined in Equation (3) are considered and MSE is the loss function ℓ. The fixed transaction cost constant is set to $\kappa = 0.001$ and the change threshold is $\epsilon = 0.0001$. For this case only X Hedging and Deep Hedging are compared to each other, since as far as our research indicates there are no analytical solutions for this type of fixed transaction costs in this specific setting. The results show that X Hedging and Deep Hedging have similar performance under these conditions. The means and standard deviations are very close, and the JS-divergence value between the two frameworks is as low as the previous experiments. Compared to Table 4 it is evident that the fixed transaction costs affect the final P&L$_T$. The mean values in Table 6 are approximately two orders of magnitude more negative than the corresponding mean values in Table 4, which is expected since this is equivalent to the number of hedging opportunities multiplied with the fixed transaction cost. These results show that the frameworks consider it more profitable to hedge and pay the cost of the transaction, as opposed to not hedging and avoiding the transaction costs. Since the standard deviations are approximately the same in Tables 4 and 6, we can further argue that the P&L$_T$ distributions are similar, but simply shifted by a factor of one hundred to the left. The histograms in Figure 6 support this claim, as it is evident that the value for κ determines how much the distributions will be shifted. An increase in κ results in a bigger shift towards the left. For small values of ϵ the general observation of shift would still be the same. This might be due to the gradient of the identicator function in the fixed transaction cost function (see Equation (3)). The

gradient is one if the underlying asset is bought, and zero if it is not bought. A constant gradient will not affect the learning, and thus have a small influence on the strategies. It could be argued that it would be reasonable to include a constructed gradient that reflects the bias of the traders. This is because a trader does not buy a small position when there exists a fixed transaction cost. Instead, the trader buys less frequent and larger positions to reduce the sum of transaction costs. This bias could be created by introducing a constructed gradient which shifts the hedging strategies to the right and make them more similar to a step function.

	Mean	**St.Dev**	**JS(DH||X)**	**Time**
DH	-0.010187	0.021443	0.000682	888.0
XH	-0.010106	0.021581	-	1012.0

Table 6: Summary of results from X Hedging (XH), and Deep Hedging (DH) with fixed transaction costs and MSE as the loss function. Presented is the mean and the standard deviation for each P&L$_T$ distribution together with the JS-divergence values between them. The computational time is given in seconds, and z is a placeholder.

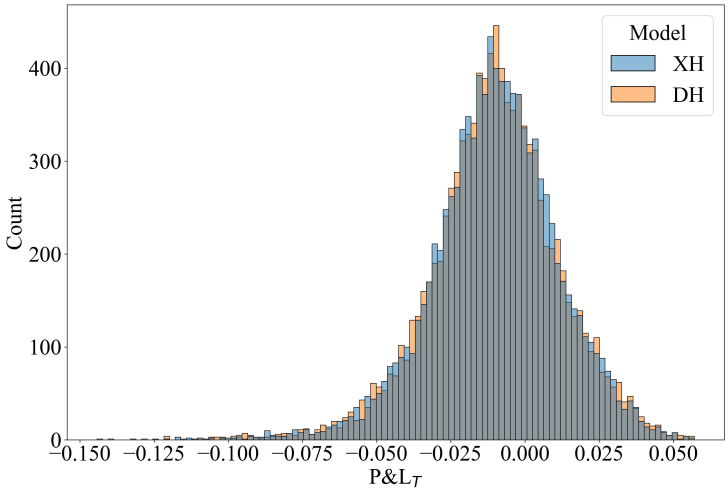


Figure 6: Histograms for X Hedging (XH), and Deep Hedging (DH) with fixed transaction costs and MSE as the loss function.

Figure 7 visualises the hedging strategies for X Hedging and Deep Hedging when fixed transaction costs are considered. Similar to the previous experiments, the strategies of the two frameworks converge towards each other as the option approaches maturity, but are particularly different for the early time steps. The strategies are especially different for small and large values of the underlying asset S, where X Hedging will assume the same strategy, but Deep Hedging continuously changes strategy. Again, since few asset values are in these ranges, their difference will not be notable in the final P&L$_T$ distributions. Upon further inspection, it is observable that the strategies are similar to the strategies in Figure 3, where no market frictions are considered. This is probably due to to the small effect from the constant gradients of Equation (3).

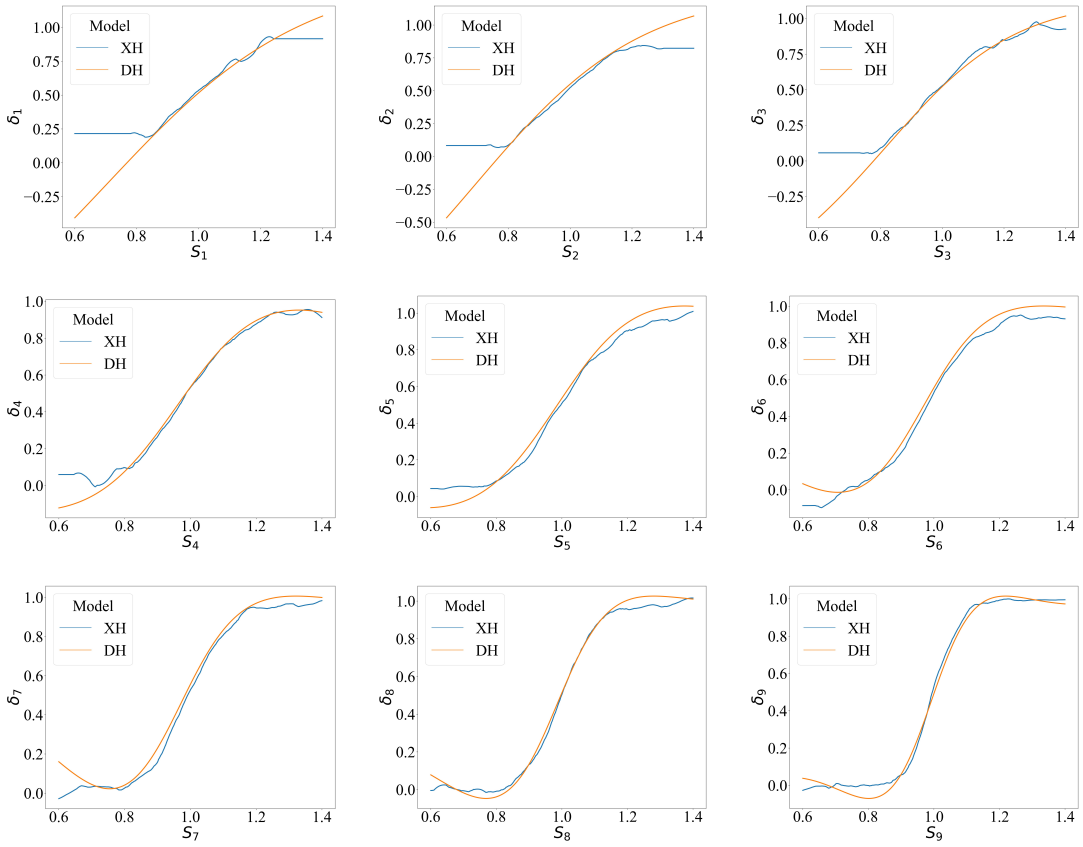


Figure 7: Hedging strategies for X Hedging (XH), and Deep Hedging (DH) from $k = 1$ to $k = 9$. Fixed transaction costs are considered and MSE is the loss metric. S_k denotes the underlying asset value, and δ_k denotes the corresponding hedging strategy at time index k.

The preceding experiments have validated X Hedging with MSE as the loss function for various market frictions. The next case considers Quadratic CVaR defined in Equation (13) as the loss function, but only with proportional transaction costs as this is the most complex setting for the hedging models. Table 7 summarises the results from experiments with X Hedging, Deep Hedging, and BS-L Hedging when proportional transaction costs are considered and Quadratic CVaR is the loss function ℓ. Bühler [2019] argues that a small quantile α for Quadratic CVaR will lead to a sparse gradient, and therefore a slow convergence when we optimize using this loss function. Therefore, there is a tradeoff between the computational time and the size of α, and we select $\alpha = 0.5$ to increase the convergence speed while still reducing the tail loss. Table 7 reveals that the mean values of the distributions for Deep Hedging and X Hedging are higher when Quadratic CVaR is used compared to MSE (see Table 5). Thus for minimizing losses, this indicates that Quadratic CVaR should be the preferred loss function ℓ. We observe that the JS-divergence values between all the models are notably higher than in the previous experiments. This might be because Quadratic CVaR is a more complex function to minimize due to the sparse gradients. The increased JS-divergence between X Hedging and Deep Hedging shows that maybe the frameworks could have been trained longer when Quadratic CVaR is used. Table 7 also shows that X Hedging has a lower mean and a lower JS-divergece against BS-L Hedging than Deep Hedging, which proves that our proposed hedging framework performs better in this setup.

	Mean	Std	JS(DH||z)	JS(XH||z)	Time
BS-L	-0.006963	0.022263	0.00294	0.002170	10.0
DH	-0.006747	0.022440	-	0.001093	860.0
XH	-0.006626	0.022412	-	-	833.0

Table 7: Summary of results from X Hedging (XH), Deep Hedging (DH), and BS-L Hedging (BS-L) with proportional transaction costs and Quadratic CVaR as the loss function. Presented is the mean and the standard deviation for each P&L$_T$ distribution together with the JS-divergence values between them. The computational time is given in seconds, and z is a placeholder.

Figure 8 shows the histograms of the different P&L$_T$ distributions when proportional transaction costs and Quadratic CVaR is considered. Comparing Figure 8 to Figure 4 one can see that the left tail is slightly smaller when Quadratic CVaR is used, even though BS-L Hedging produces one great outlier at approximatly -0.19. This behavior is expected since Quadratic CVaR minimize with respect to the greatest losses. If we had reduced α from 0.5 to e.g. 0.05, even more outliers on the left side of the histograms would disappear, reducing the largest losses of the hedger.

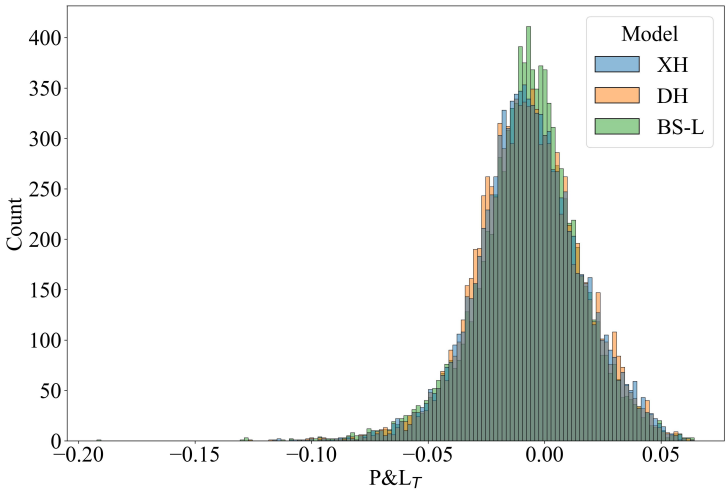


Figure 8: Histograms for X Hedging (XH), Deep Hedging (DH), and BS-L Hedging (BS-L) with proportional transaction costs and Quadratic CVaR as the loss function.

The hedging strategies for the models with proportional transaction cost and Quadratic CVaR in Figure 9 follow the same pattern that we have seen in the previous experiments. That is, the strategies for X Hedging seem to converge towards the BS-L Hedging as t approaches maturity. Even though there are some differences at the smallest and largest values of S_k, X Hedging and Deep Hedging produce accurate strategies. One difference between BS-L Hedging and the other two hedging models, is that the former seems to have a steeper slope around $S_k = 1.0$. Both strategies of X Hedging and Deep Hedging intersects the hedging strategies produced by BS-L Hedging around $S_k = 1.0$. This means that a small change in the underlying asset price affects δ_k less in X Hedging or Deep Hedging compared to BS-L Hedging.

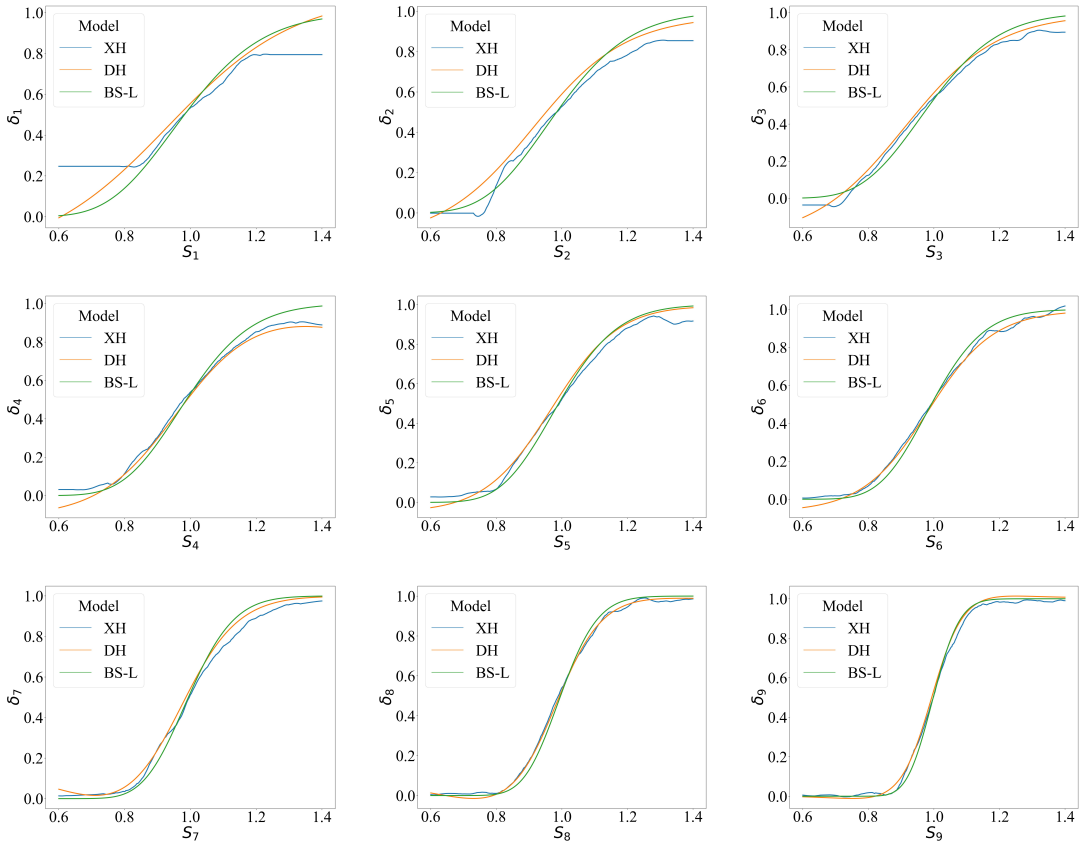


Figure 9: Hedging strategies for X Hedging (XH), Deep Hedging (DH), and Black-Scholes Leland [1985] (BS-L) from $k = 1$ to $k = 9$. Proportional transaction costs are considered and Quadratic CVaR is the loss metric. S_k denotes the underlying asset value, and δ_k denotes the corresponding hedging strategy at time index k.

Throughout the preceding experiments, we have observed that X hedging performs on-par with Deep Hedging and the Black-Scholes based benchmarks. In fact, nothing indicates that Deep Hedging should be preferred over X Hedging with the setups we have considered. This lays a good foundation for the performance of X Hedging, which makes it even more relevant to adress issues related to small data sets and explainability.

4.2 Impact of Training Set Size Reduction

As noted, Bühler et al. [2020] highlight that one of the limitations of Deep Hedging is that it needs a large training set to find the correct hedging strategies because of the neural networks' many degrees of freedom. Our aim is therefore to investigate if X Hedging with its tree ensemble methods, known to better handle small data sets, outperforms Deep Hedging on a small data set. Specifically, we test how a reduction in the number of simulated in-sample paths affects X Hedging and Deep Hedging. The setup is the same as in the first experiment, which means that we do not consider any market frictions, and MSE is the loss function ℓ. X Hedging and Deep Hedging are considered for an increasing number of in-sample training paths from 2000 to 20 000, with an interval of 1000, and compared against BS Hedging. The resulting JS-divergence values between X Hedging and BS Hedging, and Deep Hedging and BS Hedging, are shown as a function of the number of paths in Figure 10. Additionally, the difference between the JS-divergence values for the two frameworks is plotted to show which is affected the most by the number of paths.

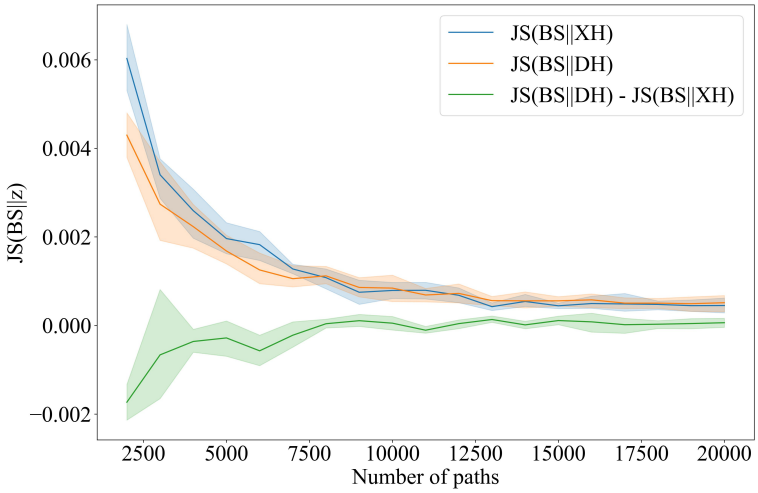


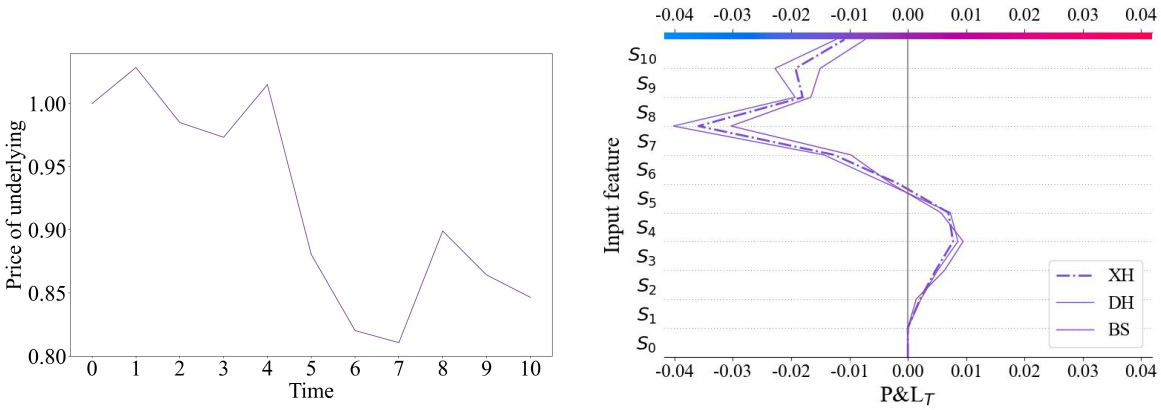
Figure 10: The JS-divergence between BS Hedging (BS), and X Hedging (XH) and Deep Hedging (DH) as a function of the number of paths, together with the difference between them. The shadows around the lines illustrate the 95% confidence interval of the JS-divergence.

The highest JS-divergence values for both frameworks appear from 2000 to 8000 paths, which indicates that both hedging frameworks struggle with a small data set. Based on the JS-divergence values with their confidence intervals, it could be argued the X Hedging does not have any clear benefit over Deep Hedging. However, the 95% confidence intervals of the two frameworks overlap in some intervals, which makes it difficult to conclude that any of the two outperform the other. It is also worth noting that the two frameworks have different starting points for learning the hedging strategies. Deep Hedging uses tanh as the activation function, a function that closely resembles the hedging strategies the hedging models are trying to learn, as in Figure 3. So even though neural networks are universal approximators and should be able to fit any continuous function regardless of the chosen activation function, this bias could be of help when there is few training data. X Hedging does not have any such bias towards the function it tries to approximate. This could be an explanation for why the two frameworks perform almost equally for a small data set; Deep Hedging benefits from the bias it has towards the correct strategy while X Hedging might adapt to the a small data set somewhat better. This unsatisfactory performance is at least expected for Deep Hedging since neural networks are known for requiring large datasets to generalize well [Linjordet and Balog, 2019]. It could be argued that both hedging models have the need for methods that expand the datasets, as demonstrated by Bühler et al. [2020] and Cuchiero et al. [2020]. Moreover, the introduction of an even simpler hedging strategy approximator could prove to outperform both LightGBM and the neural networks in this experiment, but the simpler model would probably not be able to perform well with added complexity such as different market frictions and multiple inputs. Considering a JS-divergence value of 0.001, both frameworks work well when the number of paths is above 8000. The reduction in JS-divergence is small when the number of paths increases from 8000 to 20 000, and the difference between X Hedging and Deep Hedging is almost also constant in this range.

4.3 Explainability

As previously mentioned, there is a distinction between local and global explainability. We first discuss global explainability of X Hedging, Deep Hedging and BS Hedging using SHAP, and then local explainability of X Hedging by interpretation of the trees in the hedging model. *Decision plots* from the SHAP framework are used to explain and interpret the three hedging models. These plots show the Shapley values ϕ_i, as defined in Equation (17), for all the input features i. In the context of this thesis, the input feature i is the underlying asset price S_k at time step t_k. Meaning, the input features are the underlying asset prices at each time index k. The paths that occur in the decision plots are called *decision paths*, and they show how each input feature contribute to individual predictions, and adding all the contributions together results in the final P&L$_T$. This makes it easier to understand how complex models arrive at their final predictions.

Figure 11 explains how X Hedging, Deep Hedging and BS Hedging arrive at the P&L$_T$ outputs for one specific underlying asset price path. Figure 11a illustrates the movements of one underlying asset S_k for each time step t_k that is used as input to each of the hedging models, while Figure 11b contains decision paths that show how much each of the input features contribute to the final output based on the movement of the underlying asset. The decision paths for all hedging models have a vertical line for input feature S_0 at zero because all the underlying paths have $S_0 = 1.0$, which means that S_0 does not change the final P&L$_T$. An analysis of the two plots reveals which movements of the underlying asset contribute to a negative and positive P&L$_T$. The underlying asset price experience a sudden decrease from approximately 1.0 to 0.8 from time index $k = 4$ to $k = 7$. The decision paths reveal that this sudden decrease moves the P&L$_T$ down starting at S_4. The P&L$_T$ recover after time index $k = 7$, where the underlying asset has smaller changes. Based on the behaviour of the underlying asset and the decision paths, one can observe that the hedging models perform better when the underlying asset exhibit small changes as opposed to large changes. When the changes of the underlying asset are low to moderate, the impact on the P&L$_T$ is relatively low, while for large changes it is high. Figure 11 also highlights that the decision paths for all three hedging models are quite similar and arrive at similar values for the P&L$_T$, which matches the results from Section 4.1.



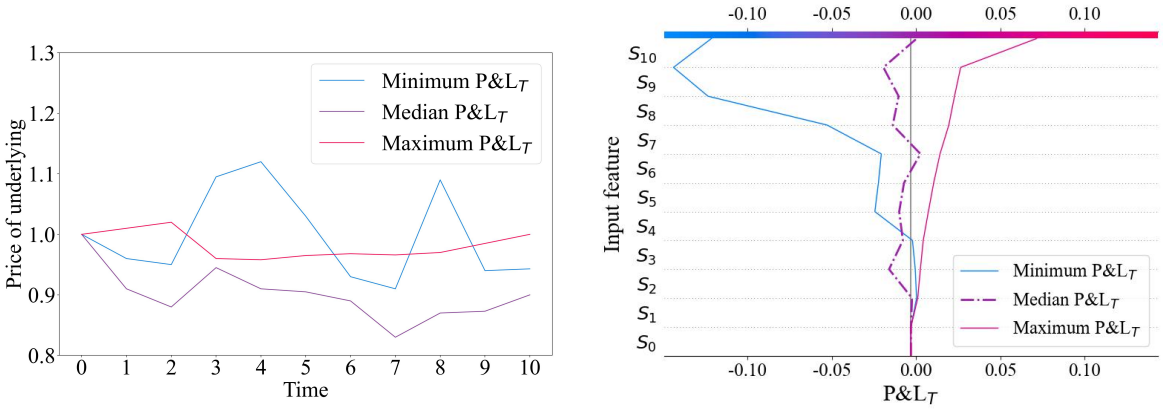
(a) Underlying asset path movements for time index k.

(b) SHAP decision plot.

Figure 11: Underlying asset path movement to the left together with a decision plot for X Hedging (XH), Deep Hedging (DH), and BS Hedging (BS) to the right. The decision paths show how each input feature contribute to the final output.

For the subsequent analyses using SHAP, only Deep Hedging is selected as the hedging model of study to avoid redundancy. Deep Hedging may only be explained through global explainability, and is selected to later emphasize the advantage of local explainability in X Hedging.

Hedging aims at reducing the risk of excessive losses. Therefore, it is valuable for a hedger to understand which underlying asset paths that makes the hedging framework trend towards negative P&L$_T$. Figure 12 shows three different underlying asset paths and how the resulting outputs are explained. The underlying asset paths in Figure 12a result in the minimum, the maximum, and the median of the predicted P&L$_T$ outputs in Figure 12b. The path that creates the minimum P&L$_T$ has larger fluctuations relative to the two others. This is because the GBM simulation for this underlying asset more frequently draws W_k close to ± 1 which makes the stochastic term in Equation (7) close to $\pm\sigma\sqrt{\Delta t}$. That is, it inhibits the largest possible movements for the GBM simulation. The decision plot shows that for the minimum P&L$_T$ Deep Hedging experience significant loss after S_7, which corresponds to highly variable movements of the underlying asset. It is also evident that the underlying asset path that yields the maximum P&L$_T$ in Figure 12a has small movements. It is easier to hedge an option with an underlying asset that has a stable behaviour. This can also be seen in the decision plot in Figure 12b where all the input features contribute to a positive P&L$_T$ in this case. The underlying asset paths that create the minimum and maximum P&L$_T$ values are both extraordinary in the sense that they inherit very large and very small fluctuations. The remaining underlying asset paths simulated via GBM have movements somewhere in between the two outliers, as illustrated by the underlying asset path that results in the median P&L$_T$, it has moderate movements. It is reasonable that the Deep Hedging model hedges the option with these paths more reliably, since this is an expected behaviour from the GBM simulations.

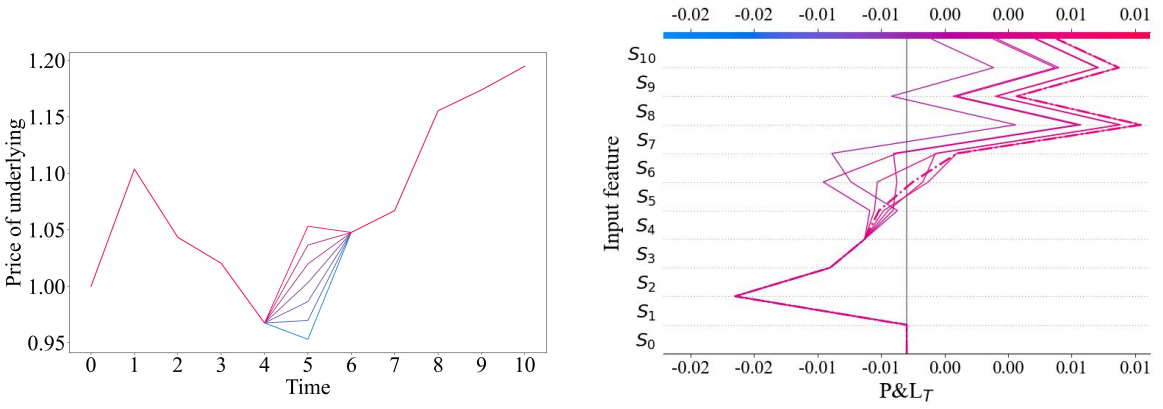


(a) Underlying asset paths movements for time index k.

(b) SHAP decision plot.

Figure 12: Three underlying assets paths to the left together with a corresponding decision plot for Deep Hedging to the right. These specific underlying asset paths result in the minimum, median, and maximum P&L$_T$.

SHAP provides the opportunity to perform sensitivity analyses. This is done by selecting one underlying asset path, and adding and subtracting small values to one of the time steps, or input features. Figure 13 shows a sensitivity analysis of S_5, with the underlying asset paths in Figure 13a and the corresponding decision plot in Figure 13b. One underlying asset path is selected, and six new paths are created from this selected path. Three equally distributed values between zero and 0.05 are added and subtracted to S_5. The derivative of the underlying asset path is changed between S_4 and S_5, and between S_5 and S_6. The paths are again equal after S_6. This process results in the seven paths plotted in Figure 13a. The decision plot in Figure 13b shows the original trajectory of the P&L$_T$ as a dotted line, without the small adjustments in the underlying asset path. All the decision paths in Figure 13b are as expected on top of each other until the change occurs at S_4, before they start to deviate between S_4 and S_5. The deviation is not considerable in this interval, but it is more significant in the next interval. The decision paths where the derivatives of the underlying asset paths have changed the most move towards negative P&L$_T$, while the other decision paths move towards positive P&L$_T$. The decision paths which move towards negative P&L$_T$ at S_5 correspond to the underlying asset path at the top and at the bottom of Figure 13a. This implies that if the absolute change in the derivative is above some threshold, then the hedging model will end up with a decrease in P&L$_T$. The decision paths are not parallel between S_6 and S_7 even though all the paths are equal in this interval. All the decision paths are parallel after S_7. These observations indicate that there is a lag of one interval in the hedging model.

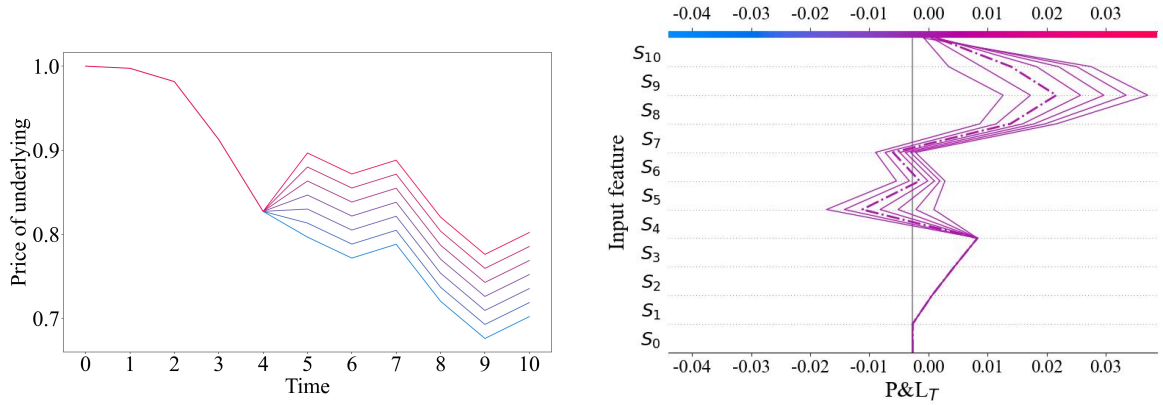


(a) Underlying asset path movements for time index k.

(b) SHAP decision plot.

Figure 13: Sensitivity analysis on a small change in S_5 by using a SHAP decision plot along with its corresponding underlying asset path. One random path is selected, and this path is adjusted such that three values of S_5 are higher, and three values are lower than the original value of S_5. The decision path with a highlighted dotted line corresponds to the original underlying asset path.

Another way to perform a sensitivity analysis using SHAP is to introduce different changes to the underlying asset at one time step, and thereafter keep the resulting paths parallel. Figure 14 illustrates the result from this sensitivity analysis. One underlying asset path is selected, and six new paths are created from the selected path. Three equally distributed values between zero and 0.05 are added and subtracted to all time indexes after $k = 4$. This process results in the seven decision paths in Figure 14a, and aims at revealing how the magnitude of the underlying asset value influence the P&L$_T$. Once again, Figure 14b shows that the decision paths are on top of each other when the corresponding underlying paths are equal. The change of the underlying asset in S_5 makes the decision paths deviate. The decision paths are not parallel after this point, which implies that the magnitude of the underlying asset will influence the final P&L$_T$. The decision paths in are further apart for S_5, S_8 and S_9 than the other input features after the underlying asset paths starts to deviate at $k = 5$. The prominent deviation between the decision paths at S_5 can be explained by the underlying asset paths having different derivative changes, and hence moving in different directions. This finding was also observed in the previous sensitivity analysis, where the underlying asset paths with the biggest absolute change in derivative influenced the decision path the most towards lower P&L$_T$. This is however not the case for S_8 and S_9, where all the paths have the same change in derivatives. For these input features, the underlying asset values are small, which might be an explanation for the large deviations in P&L$_T$ at these values. This sensitivity analysis indicates that the decision paths depend on the magnitude of the underlying asset paths together with the change of derivative of the same underlying asset paths.



(a) Underlying asset paths movements for time index k.

(b) SHAP decision plot.

Figure 14: Sensitivity analysis on a small change in S_5 that is not reverted, by using a SHAP decision plot along with its corresponding underlying asset path. One random path is selected, and this path is adjusted at S_5 such that three paths are higher, and three paths are lower than the original path. The decision path with a highlighted dotted line corresponds to the original underlying asset path.

Interestingly, all the decision paths in Figure 14b end up at approximately the same P&L$_T$ of zero. This is a good sign for the performance of the framework, as it seems to be robust against an upward or downward shift in the underlying asset path. However, the final value of the P&L$_T$ is not the only characteristic of a good hedging strategy. It is also beneficial that the decision path is close to the expected model output for all the input features. This implies that P&L$_T$ is close to zero throughout the entire time period since we in this instance use MSE as the loss function and target zero. This corresponds to having a lower risk for deviating from the expected model output at maturity. The leftmost decision path in Figure 14b has the minimum absolute deviation from zero, and hence has a lower risk of ending far away from zero. As a result, this implies that it is possible to use decision plots to analyse which paths of the underlying asset provide the greatest risk.

It is possible to identify typical decision paths by analysing the model behaviour for a greater number of underlying asset paths. Figure 15 reveals how the typical decision paths look for all out-of-sample underlying asset paths. The purple decision paths show that most of the predictions stay between -0.03 and 0.03 for all the features, which indicates that the hedging model is quite robust. We observe some outlier paths that provide quite different P&L$_T$. In fact, the leftmost outlier in Figure 15 is equivalent to the minimum P&L$_T$ path in Figure 12.

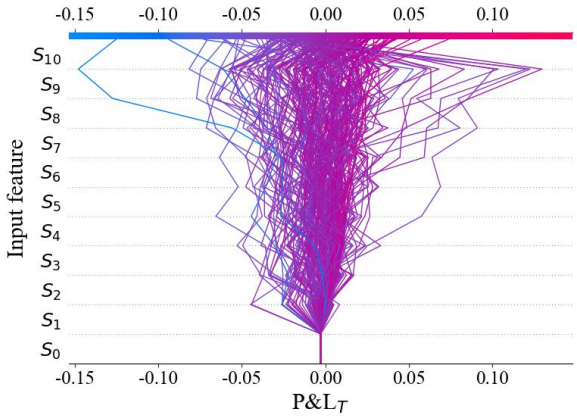


Figure 15: SHAP decision plots for all underlying asset paths. This illustrates the typical decision paths of the model together with decision path outliers.

The previous analyses using decision plots show that it is possible to gain some level of global explainability from AI models using SHAP. However, it is not possible to obtain local explainability of Deep Hedging since it is based on neural networks. This is where X Hedging prosper since tree models are locally explainable by design. Moreover, one common critique of SHAP is that a "black-box" is used to explain a "black-box" [Kumar et al., 2020]. One may consider SHAP a "black-box" since it uses advanced techniques to approximate Shapley values.

For X Hedging, each node in the tree contains a condition on one of the input features, and the child node depends on whether the condition is fulfilled or not. Meaning, every node relates to a condition on S_k, and the proceeding nodes depend on if the condition is met by the value of S_k. This structure makes it easy to explain a hedging strategy at a time step t_k. The hedging strategy is obtained by traversing all the trees in G_k until a leaf node is reached. Ultimately, the final decision of the model will be a linear combination of multiple trees in the ensemble, but each decision is still possible to backtrack. An example of one of the fitted trees is shown in Figure 16. This shows the transparency the model offers. It is possible to explain all the intermediate decisions the model makes from the input to the output. Additionally, tree models offer the opportunity to understand what happens if the input changes. With local explainability the trust and confidence in the hedging model increases, and it is possible to meet the demands of regulations and guidelines on explainability and XAI.

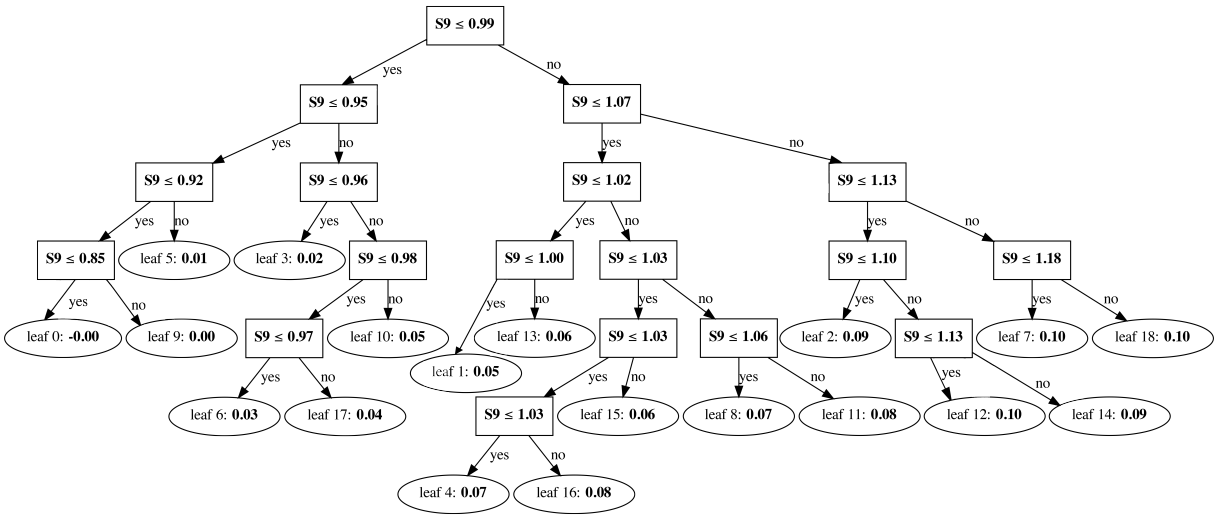


Figure 16: One of the X Hedging LightGBM decision trees visualised at $k = 9$. Binary splits of S_9 happen at each node, and values in the nodes meet the condition in the preceding node. The leaf nodes are the final values of δ_9. A linear combination of the leaf node values for multiple trees at $k = 9$ gives the true prediction of X Hedging.

In Figure 17 the python package dtreeviz[21] is used to visualise how decisions are made within a section of one of the trees in X Hedging at time index $k = 8$, along with information about how the hedging strategies are distributed along the tree. Appendix B visualises the entire tree. The hedging strategy curve is split throughout the tree, and each data point in the curve end up in a leaf node, where it is associated with a specific hedging strategy δ_8. The data points are underlying asset path values. It is worth noting that the hedging strategy curve in the root of tree strongly resembles the hedging strategies presented in Section 4.1.Figure 17 also shows how many strategies that belong to each leaf node, which is useful information to characterize common prediction paths throughout the tree. The leaves in Figure 17 contain between 21 and 1952 data points. The leaves with most data points are located around $S_8 = 1.0$, which reflect the zero interest rate. The leaves with few data points can be treated as outliers, and hedgers should pay attention to these. A smaller change to S_8 might change which leaf the data point ends up in. Hence, tree based models offer a way to explain the certainty of the output. Leaves that contain more training data offer more trusted predictions than leaves with few data samples.

To demonstrate the local explainability of X Hedging, we have highlighted a prediction path for $S_8 = 1.03$ with orange boxes and arrows in Figure 17. This shows all the decisions the tree does to end up with a prediction of 0.06. It is easy to follow all the decisions, since they only consist of comparing numerical values. Such easy interpretation of decisions is not possible in Deep Hedging due to the "black-box" neural networks. Since strategy curves are plotted together with the vertical decision line, it is also possible to see how close the data point is to end up in another leaf. This is the case in the third node of the highlighted prediction path. A small increase of S_8 will make the data point end up in the other successive node. As have been shown in this example, trees offer an intuitive way to explain the decisions of the hedging model.

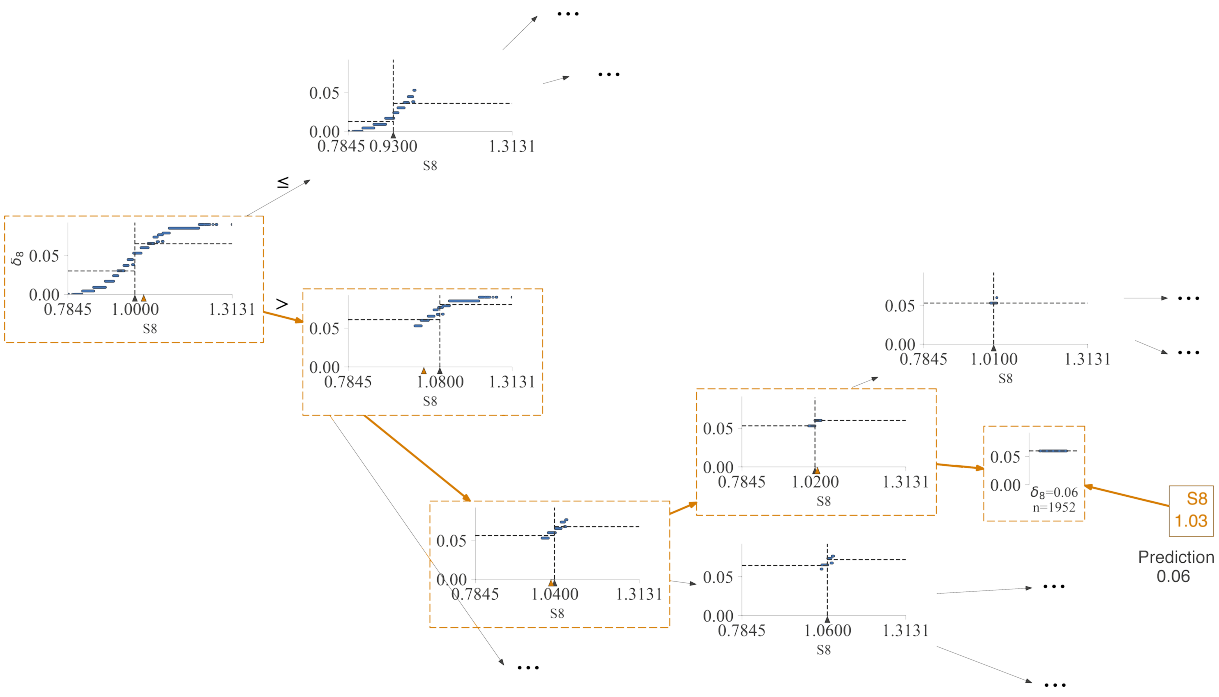


Figure 17: Prediction path in one tree for $S_8 = 1.03$.

Throughout this thesis, only small values of the underlying asset path have been considered, with $S_0 = 1$ and $\sigma = 0.2$. Neural networks converge faster for input values close to zero, but trees are indifferent to the magnitude of the input values. This is due to the fact that tree based models only perform simple numerical comparisons. It is adviced to scale the data such that the mean is close to zero when using neural networks [Koprinkova and Petrova, 1999]. As a consequence, it is recommended to downscaling, or normalising, the data if a GBM with large values such as $S_0 = 100$ is used. This scaling drastically reduces the explainability of the model because the numerical values loose their expressiveness. After scaling the data, it is not possible to directly relate the scaled input to the original data. This also affects

[21] Dtreeviz documentation: https://github.com/parrt/dtreeviz

the global explainability since the decision plot in this case disclose how the scaled input contributes to the final prediction. High input features does not affect the local or the global explainability of tree models since scaling does not affect the performance of trees. As such, this is a substantial positive for X Hedging.

We have demonstrated the benefits of local explainability in X Hedging with concrete numerical examples, but it is also worth emphazising the more practical use cases for the framework in the industry. As we have previously highlighted, there are no restrictions on what information X Hedging need in order to produce hedging strategies. Meaning, the information could just as well be news sentiment (see Shapiro et al. [2022]), as information about the movement of an asset (as we have demonstrated). If we consider an example where a market maker wishes to hedge an option on Brent Crude Oil[22], X Hedging could be used with various inputs including news sentiment to offset the risk as much as possible. Considering a situation where Crude Brent Oil experience great volatility due to the outbreak of a war between two countries, the market maker may be able to offset the risk using news information about the war. This could raise questions by supervisors or customers regarding how such good performance could be achieved amidst an unstable political situation. The market maker could then backtrack the X Hedging, and show how each intermediate decision was made, and if any were based on news from sources worthy of critique.

[22]Such financial derivates are for instance provided by the CME Group, `https://www.cmegroup.com/trading/energy/brent-crude-oil.html`

5 Conclusion

In this thesis, we present a novel hedging framework termed X Hedging that is inherently explainable. The framework hedges financial options, helping to offset the portfolio risk for market makers and OTC traders. It complies with the newly developed regulations and guidelines that concerns Explainable Artificial Intelligence within finance. As such, the framework is developed to be used by practitioners in the industry, as well as to expand the academic literature. X Hedging is a general framework in the sense that it handles different market frictions, different hedging instruments, and it does not assume a specific behaviour for the underlying asset.

Since the introduction of the famous option pricing and hedging model by Black and Scholes [1973], researchers have developed more general pricing and hedging models, culminating in the Deep Hedging framework by Bühler et al. [2019]. We provide evidence that X Hedging equals the performance of Deep Hedging by validating it for market frictions such as fixed and proportional transaction costs, and for the loss functions Mean Squared Error and Quadratic Conditional Value at Risk. The type of loss function determines if X Hedging is classified as supervised learning or reinforcement learning. The key difference between X Hedging and Deep Hedging is that the former uses gradient boosted decision trees, namely the popular LightGBM by Ke et al. [2017], instead of neural networks to approximate the hedging strategies as in the latter. LightGBM benefits from being able to map complex non-linear relationships, the same as neural networks, but is locally explainable in the sense that it is possible to observe how individual decisions are made within the hedging model. To the best of our knowledge, this is the first time a model that optimizes a group of connected LightGBM models is proposed. To implement the connected models in X Hedging, we design an innovative algorithm that uses the gradients and the Hessians of the total loss function. This approach introduces stochasticity in the training of X Hedging, which neural networks are already equipped with in form of stochastic gradient descent.

Due to the known drawback of neural networks when it comes to small training data sets, we investigate whether or not the tree methods in X Hedging are better suited in this regard. We observe that X Hedging and Deep Hedging perform similarly on small training data sets. Ultimately, we do not find any major performance benefits with of the tree models in X Hedging, and thus recommend solutions like Cuchiero et al. [2020] and Bühler et al. [2020] to simulate market data. However, we briefly mention that this could be due to the bias of neural networks' tanh activation functions. We also discuss that even simpler models could handle small training data sets better, like a single decision tree, but these models probably struggle with increased complexity of the market environment.

To highlight X Hedging's practical use case, we discuss how one may achieve local explanation of X Hedging through a specific example from the industry. In addition, we illustrate how one may achieve global explainability of X Hedging and "black-box" hedging models such as Deep Hedging using Shapley values. We discover that the change in the derivative of the underlying path influences the final profit along with the magnitude of the underlying asset value.

For future work, we propose to further validate the generality of X Hedging by applying it with different settings and investigate the possibility to replace LightGBM with other Artificial Intelligence methods that are explainable. Specifically, we suggest to validate X Hedging for an increasing number of features (e.g. asset volatilities, news sentiments), and introduce more hedging instruments to offset the risk of the hedger's portfolio. The latter proposal would imply an update to LightGBM for multioutput regression, or the introduction of a different explainable method that handles multioutput regression such as CatBoost[23]. We also recommend testing different stochastic processes such as the models proposed by Merton [1976] and Heston [1993] to simulate the underlying asset, and to test the framework with real market data. Even though such experiments may be conducted, it is worth emphasizing that the performance validation presented in this thesis already proves the practical usability of X Hedging.

[23] CatBoost documentation: https://catboost.ai

Bibliography

Almahdi, S. & Yang, S. Y. (2017). An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. *Expert Systems with Applications*, *87*, 267–279, https://doi.org/10.1016/j.eswa.2017.06.023.

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.

Atkinson, C. & Alexandropoulos, C. A. (2006). Pricing a European Basket Option in the Presence of Proportional Transaction Costs. *Applied Mathematical Finance*, *13*(3), 191–214, https://doi.org/10.1080/13504860600563184.

Bachelier, L. (1900). Théorie de la spéculation. *Annales scientifiques de l'École Normale Supérieure*, *3e série, 17*, 21–86, https://doi.org/10.24033/asens.476.

Baillie, R. T. & DeGennaro, R. P. (1990). Stock returns and volatility. *Journal of financial and Quantitative Analysis*, *25*(2), 203–214, https://doi.org/10.2307/2330824.

Baldacci, B., Manziuk, I., Mastrolia, T., & Rosenbaum, M. (2019). Market making and incentives design in the presence of a dark pool: a deep reinforcement learning approach. *arXiv*, https://doi.org/10.48550/ARXIV.1912.01129.

Bank, P., Soner, H. M., & Voß, M. (2017). Hedging with temporary price impact. *Mathematics and Financial Economics*, *11*(2), 215–239, https://doi.org/10.1007/s11579-016-0178-4.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE*, *12*, https://doi.org/10.1371/journal.pone.0180944.

Black, F. (1976). Studies of Stock Price Volatility Changes. *Proceedings of the 1976 Meetings of the American Statistical Association, Business and Economics Statistics Section*, (pp. 177–181).

Black, F. & Scholes, M. (1973). The Pricing of Options and Corporate Liabilities. *Journal of Political Economy*, *81*(3), 637–654, https://doi.org/10.1086/260062.

Buchanan, B. G. (2019). Artificial intelligence in finance. *The Alan Turing Institute, Zenodo*, https://doi.org/10.5281/zenodo.2626454.

Buckmann, M., Joseph, A., & Robertson, H. (2021). An interpretable machine learning workflow with an application to economic forecasting. *Bank of England* https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2022/an-interpretable-machine-learning-workflow-with-an-application-to-economic-forecasting.pdf?la=en&hash=829CF4A26BD34A1176432F07385890EB02A5EEB8.

Bühler, H. (2019). Statistical Hedging. *Available at SSRN 2913250*, https://doi.org/10.2139/ssrn.2913250.

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

Bühler, H., Horvath, B., Lyons, T., Arribas, I. P., & Wood, B. (2020). A Data-driven Market Simulator for Small Data Environments. https://doi.org/10.48550/ARXIV.2006.14498.

Carverhill, A. P. & Cheuk, T. H. F. (2003). Alternative Neural Network Approach for Option Pricing and Hedging. *Risk Management*, https://doi.org/10.2139/ssrn.480562.

Charpentier, A., Elie, R., & Remlinger, C. (2021). Reinforcement learning in economics and finance. *Computational Economics*, (pp. 1–38)., https://doi.org/10.1007/s10614-021-10119-4.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 785–794)., https://doi.org/10.1145/2939672.2939785.

Cox, J. (1975). Notes on Option Pricing I: Constant Elasticity of Variance Diffusions. *J. Portfol. Manage. Dec. 1996*, *23*.

Cuchiero, C., Khosrawi, W., & Teichmann, J. (2020). A Generative Adversarial Network Approach to Calibration of Local Stochastic Volatility Models. *Risks*, *8*, 101, https://doi.org/10.3390/risks8040101.

Culkin, R. & Das, S. R. (2017). Machine learning in finance: the case of deep learning for option pricing. *Journal of Investment Management*, *15*(4), 92–100 https://srdas.github.io/Papers/BlackScholesNN.pdf.

Curridori, A. (2018). Artificial Intelligence: Opportunities, Risks and Recommendations for the Financial Sector. *Commision de Surveillance du Secteur Financier* https://www.cssf.lu/wp-content/uploads/files/Publications/Rapports_ponctuels/CSSF_White_Paper_Artificial_Intelligence_201218.pdf.

Davis, M. H., Panas, V. G., & Zariphopoulou, T. (1993). European option pricing with transaction costs. *SIAM Journal on Control and Optimization*, *31*(2), 470–493, https://doi.org/10.1137/0331022.

Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(3), 653–664, https://doi.org/10.1109/TNNLS.2016.2522401.

Donaldson, R. G. & Kamstra, M. (1996). Forecast combining with neural networks. *Journal of Forecasting*, *15*(1), 49–61, https://doi.org/10.1002/(SICI)1099-131X(199601)15:1<49::AID-FOR604>3.0.CO;2-2.

Dupire, B. et al. (1994). Pricing with a smile. *Risk*, *7*(1), 18–20 http://spekulant.com.pl/article/Volatility%20Surface%20Modeling/dupire%20local%20vol.pdf.

Fama, E. F. (1965). The behavior of stock-market prices. *The journal of Business*, *38*(1), 34–105, https://doi.org/10.1086/294743.

Felsen, J. (1975). Learning Pattern Recognition Techniques Applied to Stock Market Forecasting. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-5*(6), 583–594, https://doi.org/10.1109/TSMC.1975.4309399.

Felsen, J. (1976). A man-machine investment decision system. *International Journal of Man-Machine Studies*, *8*(2), 169–193, https://doi.org/10.1016/S0020-7373(76)80042-9.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189 – 1232, https://doi.org/10.1214/aos/1013203451.

Föllmer, H. & Schied, A. (2008). *Stochastic Finance: An Introduction in Discrete Time*. De Gruyter https://doi.org/10.1515/9783110212075.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Gradojevic, N. & Kukolj, D. (2022). Unlocking the black box: Non-parametric option pricing before and during COVID-19. *Annals of Operations Research*, (pp. 1–24)., https://doi.org/10.1007/s10479-022-04578-7.

Guéant, O. & Manziuk, I. (2019). Deep Reinforcement Learning for Market Making in Corporate Bonds: Beating the Curse of Dimensionality. *Applied Mathematical Finance*, *26*(5), 387–452, https://doi.org/10.1080/1350486X.2020.1714455.

Halperin, I. (2017). QLBS: Q-Learner in the Black-Scholes-Merton worlds. *SSRN Electronic Journal*, https://doi.org/10.2139/ssrn.3087076.

Halperin, I. (2019). The QLBS Q-learner goes NuQlear: fitted Q iteration, inverse RL, and option portfolios. *Quantitative Finance*, *19*(9), 1543–1553, https://doi.org/10.1080/14697688.2019.1622302.

Heston, S. L. (1993). A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options. *The Review of Financial Studies*, *6*(2), 327–343, https://doi.org/10.1093/rfs/6.2.327.

Hodges, S. & Neuberger, A. (1989). Optimal Replication of Contingent Claims Under Transaction Costs. *The Review of Future Markets Vol 8, No 2.* https://warwick.ac.uk/fac/soc/wbs/subjects/finance/research/wpaperseries/1989/89-07.pdf.

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.

Huarng, K. & Yu, T. H.-K. (2006). The application of neural networks to forecast fuzzy time series. *Physica A: Statistical Mechanics and its Applications*, *363*(2), 481–491, https://doi.org/10.1016/j.physa.2005.08.014.

Hull, J. & White, A. (1987). The Pricing of Options on Assets with Stochastic Volatilities. *The Journal of Finance*, *42*(2), 281–300, https://doi.org/10.1111/j.1540-6261.1987.tb02568.x.

Hutchinson, J. M., Lo, A. W., & Poggio, T. (1994). A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks. *The Journal of Finance*, *49*(3), 851–889, https://doi.org/10.1111/j.1540-6261.1994.tb00081.x.

Ke, G., et al. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *Advances in Neural Information Processing Systems 30 (NIP 2017)* https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/.

Kimoto, T., Asakawa, K., Yoda, M., & Takeoka, M. (1990). Stock market prediction system with modular neural networks. *1990 IJCNN International Joint Conference on Neural Networks*, (pp. 1–6 vol.1)., https://doi.org/10.1109/IJCNN.1990.137535.

Koprinkova, P. & Petrova, M. (1999). Data-scaling problems in neural-network training. *Engineering Applications of Artificial Intelligence*, *12*(3), 281–296, https://doi.org/10.1016/S0952-1976(99)00008-1.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the SP 500. *European Journal of Operational Research*, *259*(2), 689–702, https://doi.org/10.1016/j.ejor.2016.10.031.

Kullback, S. & Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79 – 86, https://doi.org/10.1214/aoms/1177729694.

Kumar, I. E., Venkatasubramanian, S., Scheidegger, C., & Friedler, S. (2020). Problems with Shapley-value-based explanations as feature importance measures. In H. D. III & A. Singh (Eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research* (pp. 5491–5500).: PMLR https://proceedings.mlr.press/v119/kumar20e.html.

Leland, H. E. (1985). Option Pricing and Replication with Transactions Costs. *The Journal of Finance*, *40*(5), 1283–1301, https://doi.org/10.2307/2328113.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, *37*(1), 145–151, https://doi.org/10.1109/18.61115.

Linjordet, T. & Balog, K. (2019). Impact of Training Dataset Size on Neural Answer Selection Models. In L. Azzopardi, B. Stein, N. Fuhr, P. Mayr, C. Hauff, & D. Hiemstra (Eds.), *Advances in Information Retrieval* (pp. 828–835).: Springer International Publishing https://www.springerprofessional.de/en/impact-of-training-dataset-size-on-neural-answer-selection-model/16626666.

Lommers, K., Harzli, O. E., & Kim, J. (2021). Confronting Machine Learning with Financial Research. *The Journal of Financial Data Science*, https://doi.org/10.3905/jfds.2021.1.068.

Lundberg, S. M., et al. (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machinne Intelligence 2*, (pp. 56–67)., https://doi.org/10.1038/s42256-019-0138-9.

Lundberg, S. M. & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Proceedings of the 31st International Conference on Neural Information Processing Systems*, (pp. 4768–4777)., https://doi.org/10.5555/3295222.3295230.

Machado, M. R., Karray, S., & de Sousa, I. T. (2019). LightGBM: an Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry. *2019 14th International Conference on Computer Science Education (ICCSE)*, (pp. 1111–1116)., https://doi.org/10.1109/ICCSE.2019.8845529.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, https://doi.org/10.1016/j.ijforecast.2021.11.013.

Mandelbrot, B. (1963). The Variation of Certain Speculative Prices. *The Journal of Business*, *36*, https://doi.org/10.1086/294632.

McDonald, R. L. (2014). *Derivatives Markets*. Pearson New International Edition, 3rd edition https://www.pearson.com/store/p/derivatives-markets/P100001564995/9780133251593.

Meinl, T. & Sun, E. (2015). Methods of Denoising Financial Data. *Handbook of Financial Econometrics and Statistics*, (pp. 519–538)., https://doi.org/10.1007/978-1-4614-7750-1_18.

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

Merton, R. (1976). Option Prices When Underlying Stock Returns Are Discontinuous. *Journal of Financial Economics*, *3*, 125–144, https://doi.org/10.1016/0304-405X(76)90022-2.

Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. 2nd edition https://christophm.github.io/interpretable-ml-book.

Moody, J. & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, *12*(4), 875–889, https://doi.org/10.1109/72.935097.

Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting*, *17*(5-6), 441–470, https://doi.org/10.1002/(SICI)1099-131X(1998090)17:5/6<441::AID-FOR707>3.0.CO;2-%23.

Prenio, J. & Yong, J. (2021). Humans keeping AI in check – emerging regulatory expectations in the financial sector. Bank for International Settlements https://www.bis.org/fsi/publ/insights35.pdf.

Rogers, L. C. G. & Singh, S. (2010). The Cost of Illiguidity and its Effects in Hedging. *Mathematical Finance*, *20*(4), 597–615, https://doi.org/10.1111/j.1467-9965.2010.00413.x.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65 6*, 386–408, https://doi.org/10.1037/H0042519.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, *1*(5), 206–215, https://doi.org/10.1038/s42256-019-0048-x.

Savitzky, A. & Golay, M. J. E. (1964). Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, *36*, 1627–1639, https://doi.org/10.1021/ac60214a047.

Scott, L. O. (1987). Option Pricing when the Variance Changes Randomly: Theory, Estimation, and an Application. *The Journal of Financial and Quantitative Analysis*, *22*(4), 419–438, https://doi.org/10.2307/2330793.

Sezer, O. B. & Ozbayoglu, A. M. (2018). Algorithmic financial trading with deep convolutional neural networks: Time series to image conversion approach. *Applied Soft Computing*, *70*, 525–538, https://doi.org/10.1016/j.asoc.2018.04.024.

Shapiro, A. H., Sudhof, M., & Wilson, D. J. (2022). Measuring news sentiment. *Journal of Econometrics*, *228*(2), 221–243, https://doi.org/10.1016/j.jeconom.2020.07.053.

Shapley, L. S. (1953). A Value for n-Person Games. *Princeton University Press*, (pp. 307–318)., https://doi.org/10.1515/9781400881970-018.

Spooner, T., Fearnley, J., Savani, R., & Koukorinis, A. (2018). Market Making via Reinforcement Learning. *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, (pp. 434–442)., https://doi.org/10.5555/3237383.3237450.

Sutcliffe, C. & Chen, F. (2011). Pricing and Hedging Short Sterling Options Using Neural Networks. *Intelligent Systems in Accounting, Finance and Management*, *19*, https://doi.org/10.2139/ssrn.1929767.

Swidler, S. & Diltz, J. D. (1992). Implied Volatilities and Transaction Costs. *The Journal of Financial and Quantitative Analysis*, *27*(3), 437–447, https://doi.org/10.2307/2331329.

Treboux, J., Genoud, D., & Ingold, R. (2018). Decision Tree Ensemble Vs. N.N. Deep Learning: Efficiency Comparison For A Small Image Dataset. *2018 International Workshop on Big Data and Information Security (IWBIS)*, (pp. 25–30)., https://doi.org/10.1109/IWBIS.2018.8471704.

Tyree, S., Weinberger, K. Q., Agrawal, K., & Paykin, J. (2011). Parallel Boosted Regression Trees for Web Search Ranking. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11 (pp. 387–396). New York, NY, USA: Association for Computing Machinery https://doi.org/10.1145/1963405.1963461.

von Spreckelsen, C., von Mettenheim, H.-J., & Breitner, M. H. (2014). Real-Time Pricing and Hedging of Options on Currency Futures with Artificial Neural Networks. *Journal of Forecasting*, *33*(6), 419–432, https://doi.org/10.1002/for.2311.

Wang, H. & Zhou, X. Y. (2020). Continuous-time mean–variance portfolio selection: A reinforcement learning framework. *Mathematical Finance*, *30*(4), 1273–1308, https://doi.org/10.1111/mafi.12281.

Wiggins, J. B. (1987). Option values under stochastic volatility: Theory and empirical estimates. *Journal of Financial Economics*, *19*(2), 351–372, https://doi.org/10.1016/0304-405X(87)90009-2.

Xu, M. (2006). Risk Measure Pricing and Hedging in Incomplete Markets. *Annals of Finance*, *2*, 51–71, https://doi.org/10.1007/s10436-005-0023-x.

Zhou, J., Li, W., Wang, J., Ding, S., & Xia, C. (2019). Default prediction in P2P lending from high-dimensional data based on machine learning. *Physica A: Statistical Mechanics and its Applications*, *534*, 122370, https://doi.org/10.1016/j.physa.2019.122370.

İlhan, A., Jonsson, M., & Sircar, R. (2009). Optimal static-dynamic hedges for exotic options under convex risk measures. *Stochastic Processes and their Applications*, *119*(10), 3608–3632, https://doi.org/10.1016/j.spa.2009.06.009.

Appendix

A Experiments Overview

	Market frictions			Loss function		Cat.			Model			
no.	**Zero**	**PTC**	**FTC**	**MSE**	**QCVaR**	**Data**	**Shap**	**TV**	**XH**	**DH**	**BS**	**BS-L**
1	✓			✓					✓	✓	✓	
2		✓		✓					✓	✓		✓
3			✓	✓					✓	✓		
4		✓			✓				✓	✓		✓
5	✓			✓		✓			✓	✓	✓	
6	✓			✓			✓		✓	✓	✓	
7	✓			✓			✓			✓		
8	✓			✓			✓			✓		
9	✓			✓			✓			✓		
10	✓			✓			✓			✓		
11	✓			✓				✓	✓			
12	✓			✓				✓	✓			

Table 8: Sequential overview of the experiments as they appear in Section 4. The experiments either contain no market frictions, proportional transaction costs (PTC), or fixed transaction costs (FTC). The loss function is either MSE or Quadratic CVaR (QCVaR). The experiments are divided into three categories (Cat.), called Data, Shap, and tree visualization (TV). The experiments that do not have a checkmark for category is used for performance validation.

B X Hedging Tree Visualisation

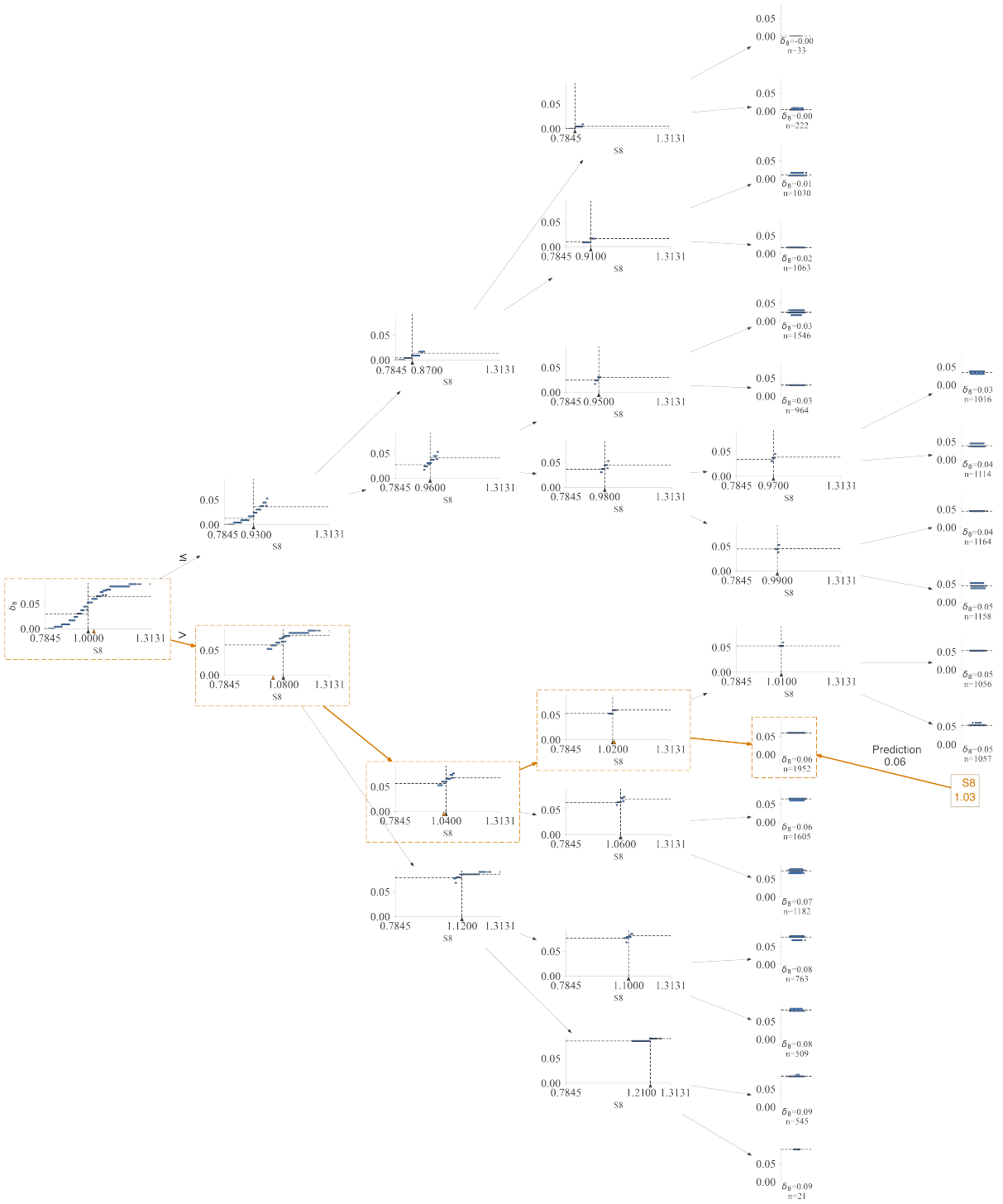
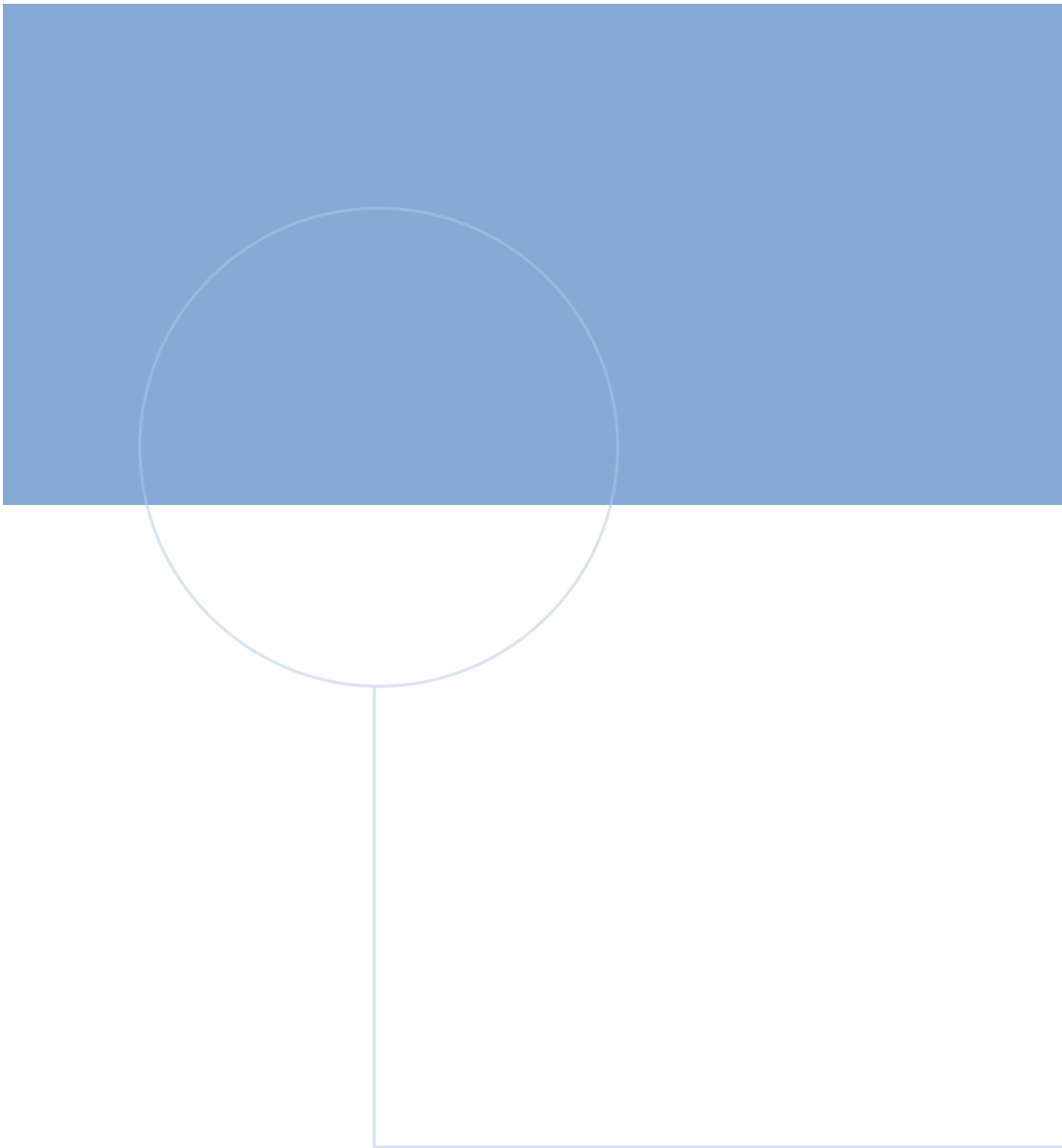


Figure 18: Prediction path in one tree for $S_8 = 1.03$. Entire tree.



NTNU

Kunnskap for en bedre verden