Statistical Hedging with Neural Networks

Weiguan Wang

Supervisor: Professor Johannes Ruf

A thesis submitted to the Department of Mathematics and Superstition of the London School of Economics and Political Science for the degree of

Doctor of Philosophy

London, April 2021

Dedicated to my loving parents

DECLARATION

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my belief, infringe the rights of any third party.

I declare that my thesis consists of 47,762 words.

Statement of co-authored work

Chapters 2, 3, and 4 are joint work with my supervisor Johannes Ruf. In specific, Chapter 2 concerns the paper "Neural networks for option pricing and hedging: A literature review", published on *Journal of Computational Finance*. Chapter 3 and 4 concerns the paper "Hedging with linear regressions and neural networks", accepted for publication on *Journal of Business and Economic Statistics*, subject to minor corrections, and a working paper "Information leakage in backtesting".

Weiguan Wang
London
April 2021

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor Professor Johannes Ruf, without whom my Ph.D. and this thesis would not be possible. Doing the Ph.D. is the toughest thing I have done so far in my life. During this journey, there have been so many setbacks and frustrations. Johannes has encouraged me a great deal in such times, and given me precious advices to help me work through. He has become my close friend, and I owe him my greatest gratitude.

Secondly, I would like to thank my parents. For the past almost thirty years, my parents have immersed me with love. Being middle-school graduates, they started their business from scratch and worked extremely hard, while raising me up. They care about my educations, and have always tried to give me the best they can. Their emotional and financial support have allowed me to lead a carefree life all the way to this Ph.D. without distraction.

My undergraduate major is Automation. I first came to know financial mathematics when I was doing a minor in finance. I got to know a bit about derivatives, and I was intrigued by those products. They are like a coin having two sides, managing risks as well as causing the financial crisis. It is a real challenge to switch to financial mathematics (although this thesis contains little maths). I took initiative and spent lots of efforts when doing my postgraduate at UCL. There, I met Johannes who later supervised my Ph.D., and other wonderful friends. Never for a single day have I regretted taking the decision to learn financial mathematics. Luckily, this field evolves with the advance of technology, and I took the chance to combine machine learning and option hedging as my research topic.

In the end, I would like to thank Johannes Muhle-Karbe and Mihail Zervos for being my examiners. Also I enjoyed my time teaching with Tuğkan Batu, Ioannis Kouletsis, and Luitgard Veraart. I thank the administration team of the Department of Mathematics for their support; in particular, Kate Barker, Enfale Farooq, Rebecca Lumb, Sarah Massey and Edward Perrin. Along this tough but fruitful journey in retrospect, I would like to thank my friends who have supported me and made the journey colorful. They are Thomas Bernhardt, Yiwei Guan, Yang Guo, Yupeng Jiang, Jiaqi Liang, Shujian Liao, Jialiang Luo, Raymond Pang, Denis Schelling, Yuyan Wang, Kangjianan Xie, Junwei Xu, Qianling Zheng and Xiaolin Zhu. In addition, I would like to thank my long time friends; in particular, Anlun Jia, Xiaofang Lin,

Shengliang Ou, and Fan Wang. The distance in space and time between London and China have never set us apart.

ABSTRACT

This thesis investigates the problem of statistical hedging with artificial neural networks (ANNs). The statistical hedging is a data-driven approach that derives hedging strategy from data and hence does not rely on making assumptions of the underlying asset. Consider an investor who sells an option and wishes to hedge it with some amount of underlying asset. ANNs can be used to determine this number by minimising the discrete hedging error.

In the first chapter, we provide a comprehensive literature review of papers on the topic of using ANNs for option pricing and hedging, as well as other related ones. Based on our research experience and summary of papers, we provide several advices that we believe are critical in using ANNs for option pricing and hedging problem. In particular, we point out an existing information leakage issue in the literature when preparing data. This review is invaluable for future researchers who are wish to work in this topic.

In the second chapter, we consider the hedging problem in the single period case. The ANN is designed to output a hedging ratio directly, instead of first learning to prices. The experiments are taken on simulated Black-Scholes (BS), Heston, end-of-day S&P 500, and tick Euro Stoxx 50 datasets. The results show the ANN can significantly outperform the BS benchmark, but is only comparable to linear regressions on sensitivities. Hence, we illustrate that the edge of the two statistical hedging methods arises mainly from the existence of the leverage effect. Moreover, the information leakage found in the literature is reproduced. It's shown that a wrong in- and out-of-sample split can overestimate the performance of statistical hedging methods. This leakage can be further exploited by tagging independent variables.

Building on the previous chapter, sensitivity analysis are given in the third chapter. They concern data cleaning on the two historical datasets, different simulation parameters of the two simulated datasets, and data preparations. In particular, we show that the statistical hedging methods can also exploit drift and convexity apart from the leverage effect.

In the last chapter, our model is extended to multiple periods on the Black-Scholes data. The replicating portfolio is rebalanced with a fixed frequency over the option's life. We show again the ANN and linear regression methods outperform the BS benchmark, and their performance are comparable.

Table of contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1

Introduction

1.1 Motivation and contribution

Artificial neural networks (ANNs) have been used as a nonparametric method for option pricing and hedging since the early 1990s. In recent years, particularly after 2016, using ANNs for quantitative finance has gained renewed attention from both the academia and industry. The applications of ANN have also extended to model calibration, solving partial differential equations, market generation, and so on.

The first publications on using ANNs for option pricing are Malliaris and Salchenberger [1993a] and Hutchinson et al. [1994]. They approximate the pricing function for European options by fitting an ANN on historical or simulated data. This approach is called nonparametric pricing approach, since it does not assume an underlying stochastic model. It has several advantages. First, it removes unrealistic assumptions existing in pricing methods that are based on modelling the movement of underlying assets. Instead, the ANNs are used to learn the pricing function directly from data. Secondly, the ANN can give the option price instantly once it completes the time-consuming training process. And the training process can be put offline. This is a huge computational advantage over traditional numerical methods such as Monte Carlo and partial differential equations, when analytic pricing formulas are not available.

The rising popularity of ANNs in quantitative finance has several reasons. First, complex nonlinear functions can be learned by ANNs from historical data, which meets the need of many finance applications. Secondly, the developments of powerful hardware (e.g. GPUs) and friendly software platforms (e.g. Tensorflow and PyTorch) are making ANNs accessible to a broader area of researchers other than computer scientists. Lastly, more and more data are becoming available to ANNs, whose high degree of freedom requires sufficient data to generalize well.

This thesis has three major contributions. They are:

1. It provides a comprehensive literature review on papers that use ANNs for option pricing and hedging. By summarising these more than 150 papers, we provide several advices on implementing ANNs for this task. Hence, this review serves as a first reading for researchers who are interested in working on machine learning for financial engineering.
2. It carefully investigates the usefulness of ANNs in the single period option hedging task. We design an ANN that outputs the hedging ratio directly instead of first estimating the option price. We show that this ANN can significantly outperform the Black-Scholes Delta on both simulation and historical datasets. However, such an advantage can also be achieved by linear regressions on option Greeks. These linear models, although simple, have never been proposed in the literature, and ANNs have never been benchmarked to them.
3. It discovers and verifies an information leakage long existing in the literature. Exploiting such a leakage leads to an overestimate of the performance of statistical hedging, hence claiming spurious outperformance.

The remainder of this subsection elaborates on these contributions.

Chapter 2 provides a comprehensive **literature review** of relevant papers in Chapter 2. Papers are compared in terms of input features, output variables, benchmark models, performance measures, data partition methods, and underlying assets. Most of the papers focus on the pricing task, and a large part of them report outperformance of ANNs against parametric or non-parametric benchmarks. Only very few papers concerns the hedging task. To hedge an option, we need to know the numbers of hedging instruments to hold, and these numbers are usually given by sensitivities of option price with respect to the underlying price in parametric models. Sensitivities can be calculated from ANNs that are trained to option prices, by taking derivatives with respect to certain variables. Instead, ANNs can also be trained to directly output hedging ratios, and this topic has not been well studied. Only a few papers discuss such an architecture for ANNs. The first papers are Carverhill and Cheuk [2003], Chen and Sutcliffe [2012], and Shin and Ryu [2012]. Shin and Ryu [2012] learn to optimal hedge values derived from a hedging cost minimisation problem. Chen and Sutcliffe [2012] learn to the “actual” delta, i.e. the ratio of option price change to underlying’s in a high frequency data set. More recently, Buehler et al. [2019a,b] include market frictions and use ANN to output hedge ratio in a risk minimization framework.

Let us now compare this thesis to Carverhill and Cheuk [2003]. They use an ANN with two outputs to explain the change of option price in terms of changes of the underlying price and implied volatility. Denoting the two outputs by y_1 and y_2, they approximate the change of option price ΔC with the expression

$$\Delta C \approx y_1 \Delta S + y_2 \Delta \mathrm{IV}, \tag{1.1}$$

where ΔS and ΔIV are changes of the underlying and the implied volatility of at-the-money options. The two output quantities thus form a linear composition of the two predictors to learn to the observed option price change. In view of (1.1), one should not interpret the two predictors ΔS and ΔIV as two hedging instruments, nor can one interpret the two outputs as hedging ratios, since implied volatility is not tradable. This is a linear two-factor model for explaining how much of the change of the option price is caused by changes of the stock and implied volatility, and the two outputs y_1 and y_2 are factor loadings. Instead, we build our statistical hedging from a practical trading point of view. We use the underlying or corresponding futures as the only hedging instrument, and the ANN outputs the hedging ratio. We also innovatively add option Greeks to the feature set of the regressions. Using only one hedging instrument allows for a partial hedge of the non-tradable volatility change by the underlying.

The motivation to replace sensitivity by hedging ratio is the following. First, from a risk-management point of view the hedging ratio is the main quantity of interest. Different models might yield similar option prices but completely different hedging strategies, see Lyons [1995]. Secondly, outputting hedging ratio can skip the step to differentiate, possibly numerically, the trained ANN. Lastly, learning the hedging ratio directly can incorporate the correlation between the underlying price and other parameters. This correlation is usually not taken into account when computing sensitivity.

Chapter 3 investigates the usefulness of **ANNs for option hedging**. To do this, we design an ANN architecture that directly outputs hedging ratio. The ANN is trained to minimise the mean squared hedging error (MSHE) over a single period. This experiment is taken on datasets simulated under the Black-Scholes and Heston models, end-of-day option data on S&P 500, and tick option data on Euro Stoxx 50. On the first three datasets, the experiment is taken on one-day and two-day hedging. On the last dataset, hourly hedging is investigated additionally. We compare the performance of ANNs against the benchmark Black-Scholes strategy, and discover that ANNs are able to outperform the benchmark in most scenarios.

To understand the advantage of ANNs, simple interpretable linear regressions on sensitivities are introduced as additional benchmarks. These kinds of benchmarks have not been well studied yet in the literature. However, they perform comparably to, sometimes better than, ANNs and the state-of-the-art Hull-White regression model (see Hull and White [2017]). By investigating the coefficients of linear regressions, we realise that the major advantage of statistical hedging (including linear regression and ANNs) over the Black-Scholes is due to its capture of leverage effect existing in S&P 500 and Euro Stoxx 50 datasets. Further studies (provided also in Section 4.5.7) on data simulated under the Black-Scholes and Heston models with various parameters show the statistical hedging methods can learn also drift and convexity effects. These effects can reinforce or cancel each other, hence making the Black-Scholes benchmark suboptimal and statistical hedging outperform, even in the Black-Scholes model.

In addition, we discover and verify an **information leakage** issue long existing in the literature. This leakage is introduced by randomly shuffling and splitting data into in-sample and out-of-sample data, thus breaking the time structure. In details, we hedge a cross-section of options with different maturities and strikes. These options share the same underlying price, hence a cross-section dependence exists. The random shuffling approach ensures the sample-wise independence, however, it overlooks the cross-section dependence. It allows the underlying price of the same day to appear both in the in-sample and out-of-sample datasets. Hence, it allows a regression model to 'memorize' whether on a specific day the underlying price goes up or down in the in-sample set, hence looking into the future when backtesting on the out-of-sample data. This alone leads to an overestimate of the relative performance of statistical hedging methods, and it can be further exploited by tagging each sample with additional features just like the underlying prices. This information leakage is not uncommon in the literature, and is more disturbing nowadays, as the capacity of ANNs is getting increasingly larger. Table 2.1 offers a summary of data partition methods in the literature. In particular, we use the experimental setup of Cao et al. [2020] to show that their significant outperformance of VIX is largely due to the information leakage issue. To reproduce this information leakage, three extra experiments are taken on the Black-Scholes and S&P 500 data. The four experiments, including the original one, are called 'Baseline', 'VIX', 'Permute', and 'Permute + VIX' experiments. In the 'VIX' experiment, we add independently simulated random numbers[1] to each sample, and split the in-sample and out-of-sample data chronologically. The performance of regressions worsens compared to that of regressions in the 'Baseline' experiment, since the VIX is a meaningless feature. In the 'Permute' experiment, data is randomly shuffled and split. The information leakage, as we explained, improves the relative performance of regressions against the BS benchmark compared to that in the 'Baseline'. In the 'Permute + VIX' experiment, the fake VIX values are first added, and then data is shuffled and split randomly. Now the ANN can memorize two features, i.e. the underlying price and the fake VIX. We show that the overestimate of regression performance can be even more remarkable.

Chapter 4 contains additional diagnostics and sensitivity analysis that complement Chapter 3. Three kinds of sensitivities are investigated. They concern data cleaning on the two historical datasets, different simulation parameters of the two simulated datasets, and data

[1] They serve as the fake VIX, the Chicago Board Options Exchange's CBOE Volatility Index.

preparations. Here are a few observations. Removing short-maturity options or using a single window data preparation improves the performance of statistical hedging. Other changes have mixed effects on the performance, hence we refer to the corresponding text for more details.

Chapter 5 extends the single-period hedging framework to multiple periods. Here, we use a recurrent neural network to hedge an at-the-money option over its entire life, by rebalancing the replicating portfolio every one or two days. As before, the RNN is compared with linear regressions on sensitivities. We show that the two methods have similar performance. In addition, we investigate the performance of statistical methods that are trainded on single period and then applied on the multiple periods situation. Unfortunately, there is little advantage of using recurrent models on multiple periods within our simple setup over using a non-recurrent model trained in the single-period setup.

1.2 Primer: neural networks

Fully-connected neural networks

A neural network is a composition of simple elements called neurons, which maps a certain input to an output. Such a network then forms a directed, weighted graph. A neural network can also be thought of as a repeated composition of linear and non-linear transformations. Figure 1.1 illustrates a two-hidden-layer feed-forward neural network, with four inputs and one output.

As nonlinear transformation we shall rely on the ReLU (rectified linear unit) activation function, given by $\phi(x) = \max(0, x)$. The benefits of using ReLU activation is addressed in Glorot et al. [2011] and Section 3.1 of Krizhevsky et al. [2012]. Then each node in the hidden layer performs an operation of the form $\phi(w \cdot x + b)$, where w denotes an adjustable weight matrix, x is the input of the node, and b denotes the bias term. This operation returns a real value, called the activation, which then serves as the input to the nodes in the next layer. As an illustration, the network of Figure 1.1 can then be represented by the function

$$\mathbb{R}^3 \to \mathbb{R}, \; F : x \mapsto \widehat{\phi}\left(w_3 \cdot \phi\left(w_2 \cdot \phi\left(w_1 \cdot x + b_1\right) + b_2\right) + b_3\right), \tag{1.2}$$

where $b_1, b_2 \in \mathbb{R}^4$ and $b_3 \in \mathbb{R}$ are adjustable bias terms, $w_1 \in \mathbb{R}^{4\times3}$, $w_2 \in \mathbb{R}^{4\times4}$, and $w_3 \in \mathbb{R}^{4\times1}$ are adjustable ANN weight matrices.

Recurrent neural network (RNN)

Following the formulation in Goodfellow et al. [2016], Figure 1.2 shows an RNN with no intermediate outputs. At time t, this RNN takes the input δ_t and its hidden state u_{t-1}

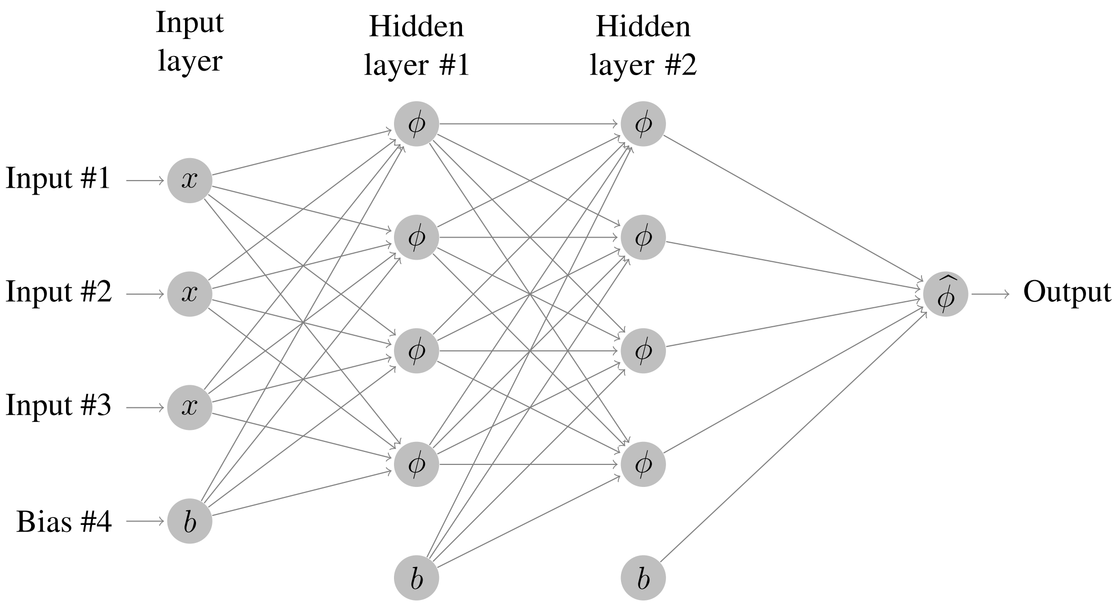


Fig. 1.1 The structure of a two-hidden-layer feed-forward neural network. Here, ϕ denotes the activation function for the hidden layers, $\widehat{\phi}$ the activation for the output layer, x the input, and b the bias. This network has 4 nodes for each hidden layer.

of previous step to produce the new hidden state u_t. Hence, the left side of Figure 1.2 is represented by

$$u_t = F(u_{t-1}, \delta_t; W),$$

where W is the set of trainable weights. Any state of the RNN depend on its current state and hence on all its previous states and inputs. The equivalent unfolded representation on the right of Figure 1.2 shows exactly this dependence of the state u_t on the entire sequence $(\delta_0, ..., \delta_t)$.

A simple fully-connected RNN has two sets of weights, a feed-forward kernel $\widehat{w}$ and a recurrent kernel $\widetilde{w}$. They are applied on the input δ_t and hidden state u_{t-1}, respectively. The sum of the two products is the new hidden state, given by

$$u_t = \delta_t \cdot \widehat{w} + u_{t-1} \cdot \widetilde{w}.$$

The initial state of the RNN can be specified as an input, or left as a part of trainable weight.

Loss function and regularisation

Taking L^2 loss for example, the training of ANN is to minimise the loss function

$$L(W|x,t) = \frac{1}{N}\sum_i (y_i - t_i)^2. \tag{1.3}$$

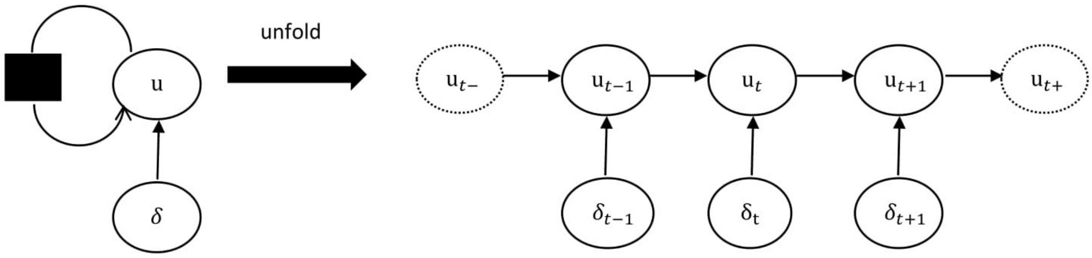


Fig. 1.2 The illustration of an RNN. The left and right sides are folded and unfolded representations of the RNN.

Here, the input feature is denoted by x, the target by t, the number of samples in the training set by N, and the predicted value for sample i by y_i.

Regularisation is one of the most important techniques of deep learning to address the over-fitting issue when a large model is used. Goodfellow et al. [2016] (Section 5.2.2) define regularisation as "any modification to a learning algorithm that is intended to reduce its generalization error but not its training error". A widely used regularisation is the Tikhonov regularisation (Tikhonov et al. [1995]), or the so-called L^2 regularisation. It adds a penalty to the (1.3), and loss function becomes the following

$$L(W|x,t) = \frac{1}{N}\sum_i (y_i - t_i)^2 + \alpha||w||_2^2, \tag{1.4}$$

where

$$||w||_2^2 = \sum w_k^2,\ w_k \in W.$$

Here, α is the regularisation strength. The penalty term shrinks the model by pushing weights to zero. It effectively trims the ANN by removing insignificant links between nodes. A very large α makes the ANN essentially a linear model. The value α is an important hyperparameter besides the initial learning rate. Bengio [2012] provide some practical guidelines on tuning this hyperparameter.

Another kind of widely used regularisation is early stopping. During the training of a machine learning model, the generalisation performance on the validation set first improves and then deteriorates as the training continues. Hence, stopping the training when the validation performance deteriorates offers a better performance on the test set. In practice, one can save the network weights and update them only when the validation performance improves. At the test time, the recorded weights are retrieved. This regularisation method does not need tuning hyperparameter, which is a computational advantage over many other methods.

Optimiser

The algorithms to minimise the loss function are called optimisers in computer science language. The most popular family of optimisers are gradient descent algorithms. They include stochastic gradient descent (SGD), ‘Adam’ (Kingma and Ba [2015]), and others. Throughout this thesis, we use ‘Adam’ optimiser. It extends the traditional SGD optimisers by including adaptive moments estimations, hence being more computational efficient. This optimiser has been implemented by the deep learning package ‘Tensorflow’, which we use for all the experiment implementations of the thesis.

Chapter 2

Neural Networks for Option Pricing and Hedging: A Literature Review

This chapter is based on join work with Prof. Johannes Ruf, and contains the paper

Ruf, J and Wang, W, Neural networks for option pricing and hedging: A literature review, Journal of Computational Finance, volume 24, number 1, pages 1–45.

2.1 Introduction

Beginning with Malliaris and Salchenberger [1993b] and Hutchinson et al. [1994], more than one hundred papers in the academic literature concern the use of artificial neural networks (ANNs) for option pricing and hedging. This work provides a review of this literature. The motivation for this summary arose from our companion paper Ruf and Wang [2020b, 2021]. There we continue the discussions of this note; in particular, of potentially problematic data leakage when training ANNs to historical financial data.

A linear regression model can be thought of as an affine function that maps some input x to an output y. Similarly, an ANN can be thought of as a (possibly repeated) composition of linear and nonlinear functions, again mapping some input x to an output y. Training an ANN usually corresponds to choosing the linear components so that this mapping is optimal, in some sense, for (a subset of) a given dataset (the *training set*) $(x_i, y_i)_i$. Optimality is usually measured by means of a *loss function*, which measures the distance between the ANN output and the given data.

The Stone-Weierstrass theorem asserts that any continuous function on a compact set can be approximated by polynomials. Similarly, the *universal approximation* theorems ensure that ANNs approximate continuous functions in a suitable way. In particular, ANNs are able to capture nonlinear dependencies between input and output.

With this understanding, an ANN can be used for many applications related to option pricing and hedging. In the most common form, an ANN learns the price of an option as a function of the underlying price, strike price, and possibly other relevant option characteristics. Similarly, ANNs might also be trained to learn implied volatility surfaces or optimal hedging ratios. In the pricing task, the corresponding loss function is often chosen to be the squared distance of the observed (simulated) option prices and the ANN predicted prices. In the hedging task, one would compare observed (simulated) option prices and the values of the ANN hedging portfolios.

Let us provide a formal example in the context of the pricing task, namely a two-hidden layer ANN with linear output. Such an architecture maps an input x (usually a vector consisting of several features, such as moneyness, contract-specific implied volatility, etc.) to an output y (the option price) as follows:

$$y = w_2 \cdot \phi(w_1 \cdot x).$$

Here ϕ is a nonlinear function (the so called *activation function*), w_1, w_2 are weight vectors, and the dot denotes the scalar product. Training such an ANN corresponds to finding weight vectors $\hat{w}_1, \hat{w}_2$ such that the output $\hat{y}$ of the ANN is close to the option price y, for all samples in a subset of the data (the *training set*). As already mentioned, a widely used criterion to measure what ’close’ means is the mean squared error.

The papers discussed here mostly study how well such an approximation by an ANN works on either simulated or real datasets. Different performance measures are employed, and often the ANNs are compared to a variety of benchmarks, the simplest one being the Black-Scholes formula. We shall also summarize how the individual papers choose the training data.

The universal approximation theorems allow a ‘model-based’ usage of ANNs. Imagine a data-generating process, along with a computationally involved pricing algorithm, which relies, for example, on solving partial differential equations or Monte-Carlo simulations. When facing such a situation, ANNs can be used to learn directly the pricing formula. We review this literature in Section 2.4.

This chapter is organised in the following way. Section 2.2 features Table 2.1, a summary of the literature that concerns the use of ANNs for nonparametric pricing (and hedging) of options. Section 2.3 provides a list of recommended papers from Table 2.1. Section 2.4 provides an overview of related work where ANNs are applied in the context of option pricing and hedging, but not necessarily as nonparametric estimation tools. Section 2.5 briefly discusses various regularisation techniques used in the reviewed literature.

2.2 ANN based option pricing and hedging in the literature

Bennell and Sutcliffe [2004], Chen and Sutcliffe [2012], and Hahn [2013][1] provide extensive literature surveys on the application of ANNs to option pricing and hedging problems. Here we complement these surveys with additional and more recent papers.

Table 2.1 summarises a large part of the literature and compares six relevant characteristics. They are features (or so-called explanatory variables), outputs of the ANN, benchmark models, data partition between training and test sets, and the underlyings along with the time span of the data. In Table 2.1, we only list papers that study an ANN's performance for the option pricing and hedging problem with a somehow statistical perspective. Other papers have different approaches, e.g., a computational perspective, and hence do not fit naturally in the table. These papers are discussed separately in Section 2.4.

We have not included a comparison of methodologies for the parameter estimation or of ANN architectures, such as number of nodes and layers, activation functions, etc. These specifications vary strongly between the papers summarized here. As an overall trend let us only remark that more recent papers use more complex architectures, in line with improved availability of computational resources. We also do not include a paper-by-paper summary of specific conclusions been drawn. However, more than half of the paper abstracts explicitly emphasize the positive performance of ANNs in the option pricing and hedging task.

Let us explain how to read Table 2.1. It summarises six relevant characteristics that describe how each paper treats the pricing/hedging problem. The columns 'Features' and 'Outputs' show explanatory features given to the ANN as inputs and outputs, respectively. Table 2.2 explains notations and abbreviations used for these columns. The 'Benchmarks' column lists non ANN-based techniques with which an ANN is compared. Table 2.3 explains the corresponding abbreviations. Table 2.4 presents abbreviations and definitions for the 'Performance measures' column, which summarises how an ANN (and its benchmarks) are evaluated in each paper. The performance measures marked bold are related to evaluations along multiple periods. Table 2.5 explains abbreviations for the underlying assets used in each study and listed in the 'Underlyings' column.

Here an 'executive summary' of Table 2.1:

- There exist two ways of using the stock price and option strike as inputs to an ANN. Sometimes they are used as two separate features. Other times, only their ratio (the so-called moneyness) is used as an input. In the previous ten years, the second approach is used more often. See also Subsection 2.2.1 for a discussion of this point.

[1] Hahn [2013] also surveys the use of ANNs to predict realised volatility. Here we do not aim to do so.

- There are many different choices of volatility estimates concerning input features and benchmarks. The conclusions drawn often depend on this choice. Subsections 2.2.1 and 2.2.3 provide more details on this point.

- Most papers focus on estimating option prices, around fifteen papers (10% of all papers listed) on estimating implied volatilities, and very few deal with the hedging problem directly; see also Subsection 2.2.2.

- In some studies, data is partitioned into a training and a test set in a way that violates the underlying time series structure. This introduces information leakage and underestimates the generalization error of the ANN. This is further discussed in Subsection 2.2.4.

For the reader interested in a small selection of all these papers, we refer to Section 2.3.

After reading about 150 papers and creating Table 2.1, we would like to offer three pieces of (personal) advice when implementing ANNs as nonparametric estimation tool of option prices and hedges. First, stationary features should be used as input. Secondly, the ANN performance should be appropriately benchmarked. Third, the time series structure should not be violated when partitioning the data set into training and test sets.

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Malliaris and Salchenberger [1993a,b]	S, K, τ, σ_{IM}, r, lagged C and S	C	BS-IM	MAE, MAPE, MSE	Chronological	S&P100. 6M
Hutchinson et al. [1994]	$S/K, \tau$	C/K	BS-H, Linear	**MATE**, **PE**, R^2	Chronological	Simulation (BS); S&P500. 5Y
Kelly [1994]	$S, K, \tau, \sigma_{\text{H}}$	C	CRR	MAE, **MTE**, MSE, R^2	?	Individual stocks. 6M
Boek et al. [1995]	S/K, τ, σ_{H}, r	$(C - C_{\text{BS-H}})/K$	BS-H	MAPE, R^2	?	AOSPI. 2Y
Miranda and Burgess [1995]	?	$\Delta\sigma_{\text{I}}$	Linear	?	?	IBEX35. ?
Krause [1996]	$C_{\text{BS-H}}$, S, K, τ, σ_{H}	C	BS-H	R^2	Chronological	DAX. 3Y
Lachtermacher and Rodrigues Gaspar [1996]	S, K, τ, σ_{H}, r	C	BS-H	MAE, MAPE, MPE, MSE	Random	Individual stocks. 2M
Lajbcygier and Flitman [1996]	S/K, τ, σ_{IH}	$(C - C_{\text{BS-IH}})/K$	BS-IH, KR, Linear	MAE, R^2	Chronological	AOSPI. 3Y
Lajbcygier et al. [1996a][2]	?	?	BS-?, BW	?	?	AOSPI. ?
Lajbcygier et al. [1996b], Lajbcygier [2002]	S/K, τ, σ_{H}, r	C/K	BS-H, BW, Linear	MAPE/MAE, MSE, R^2	Random	AOSPI. 2Y
Liu [1996]	S	S[3]	BS-H	MAE, MAX, MSE	Chronological	S&P500. 5Y[4]
Malliaris and Salchenberger [1996]	τ, lagged σ_{IM}, and others	σ_{IM}	None	MAE, MSE	Chronological	S&P100. 1Y
Niranjan [1996]	$S/K, \tau$	C/K	BS-H	MSE	?	FTSE100. 11M
Qi and Maddala [1996][5]	S, K, τ, r, open interest	C	BS-H	MAE, MSE, R^2	Random	S&P500. 2M

[2] We were not able to obtain a copy of this paper.

[3] The network learns the dynamics of the underlying iteratively and then relies on Monte-Carlo to determine option prices.

[4] The network is trained on a five-year long stock price path, but uses only one day's option price data.

[5] This paper relies on the PhD thesis Qi [1996].

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Hanke [1997]	S/K, τ, σ_{G},[6] r	C/K, $(C - C_{\mathrm{BS-G}})/K$	None	MSE	Chronological	Simulation (SV)
Herrmann and Narr [1997]	S, K, τ, σ_{I}, σ_{V}, r	C	BS-V	MAE, ME, MSE, R^2	?	Simulation (BS); DAX. 1Y
Karaali et al. [1997]	S, K, σ_{H}	C	None	None	Chronological	DEM volatility. 5Y
Lajbcygier and Connor [1997a,b]	S/K, τ, σ_{IH}	$(C - C_{\mathrm{BS-IH}})/K$	BS-IH	MAE, SR	Chronological	AOSPI. 1Y
Lajbcygier et al. [1997][2]	S/K, ?	$(C - C_{\mathrm{BS-?}})/K$	?	?	?	AOSPI. ?
Ahmed and Swidler [1998]	S/K, τ, σ_{H}, volume	σ_{I}	None	MAE, MSE	Random	Individual stocks. 3Y
Anders et al. [1998]	S/K, S, τ, σ_{H}, σ_{V}, r	C/K, $(C - C_{\mathrm{BS-V}})/K$	BS-H, BS-V	MAE, MAPE, ME, MSE, R^2	?	DAX. 3Y
Avellaneda et al. [1998]	S/K, τ	σ_{I}	None	%E	?	USD-DEM. Several days
Garcia and Gençay [1998, 2000]	S/K, τ	C/K	BS-H, Linear	DM, **MATE**, MSE	Chronological	Simulation (BS); S&P500. 8Y
White [1998]	?	C	None	MAE, MSE	Random	Simulation (BS)
Chen and Lee [1999]	S, τ, σ_{H}, Γ, Δ, ρ, $\mathcal{V}$, volume	C	BS-H, CRR	MAE, MAPE, MSE	Chronological	Individual stocks. 1Y
Geigle and Aronson [1999][7]	S/K, τ, σ_{H}, r	C/K	BS-H	MAE, MAPE	Chronological	S&P500. 6Y
Hanke [1999a]	S/K	$(C - C_{\mathrm{BS-H}})/K$	BS-H	MSE	Chronological	DAX. 1Y
Hanke [1999b]	S/K, τ, σ_{Cal}	C/K, $(C - C_{\mathrm{BS-Cal}})/K$	BS-Cal	MSE	Chronological	DAX. 10M
Ormoneit [1999]	S/K	C/K	BS-H, BS-IH	**MATE**, MSE, R^2	?	DAX. 9M
Tsaih [1999]	S, K, τ, σ_{I}, r	C	BS-IH	Sensitivity analysis	Chronological	Simulation (BS)
Briegel and Tresp [2000]	S, τ	C	BS-?, lagged C	MSE	?	FTSE100. 10M

[6] Additional GARCH parameters are also added as features.

[7] This paper relies on the PhD thesis Geigle [1999].

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Carelli et al. [2000]	K, τ	σ_I	None	%E	?	USD-DEM. Several days
de Freitas et al. [2000a,b]	S/K, τ	C/K	BS-H	R^2	?	FTSE100. 11M
Galindo-Flores [2000]	S, K, τ	C	Decision tree, Linear, Nearest neighbour	MSE	?	Simulation (BS)
Ghaziri et al. [2000]	S, K, τ, σ_H, r, open interest	C	BS-H	MSE	?	S&P500. 2M
Raberto et al. [2000]	S/K, τ, $\lvert S-K\rvert/\tau$	C/K	None	None	?	BUND. ?
Saito and Jun [2000][2]	?	?	BS-?	?	?	S&P500. ?
White [2000]	S, K, τ, σ_H	C	BS-H	MAE, MSE	Random	Simulation (BS); Eurodollar. 7M
Yao et al. [2000]	S, K, τ	C	BS-H	R^2	Chronological	NIKKEI225. 1Y
Dugas et al. [2001, 2009]	$S/K, \tau$	C/K	None	MSE	Chronological	S&P500. 5Y
Gençay and Qi [2001]	$S/K, \tau$	C/K	BS-H	DM, **MATE**, MSE	Chronological	S&P500. 6Y
le Roux and du Toit [2001]	S, K, τ, σ_I, r	C	None	MSE	Chronological	Simulation (BS)
Meissner and Kawano [2001]	S/K, τ, σ_G	C/K	BS-G	MAE, MAPE, ME, MSE, R^2	?	Individual stocks. 8M
Schittenkopf and Dorffner [2001]	τ	Gaussian parameters[8]	BS-H, CS	MAE, **MATE**, ME, MSE	Chronological	FTSE100. 5Y
Andreou et al. [2002]	$S/K, \tau, \sigma_\mathrm{H}$, σ_V, r, and others	C/K, $(C-C_{\mathrm{BS-H}})/K$, $(C-C_{\mathrm{BS-V}})/K$	BS-H, BS-V	MdAE	Chronological	S&P500. 3Y
Billio et al. [2002]	S/K, τ, σ_I, r	C/K	BS-?	MSE	Chronological	FTSE100. 1Y
Ghosn and Bengio [2002]	S/K, τ	C/K	None	MSE	Chronological	S&P500. 6Y

[8] ANNs output parameters for a Gaussian mixture density as a model for the risk-neutral density.

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Healy et al. [2002]	$S, K, \tau, \sigma_{\mathrm{I}}, r$, spread, open interest, volume	C	None	MAE, ME, R^2	Random	FTSE100. 5Y
Zapart [2002, 2003b]	Lagged wavelet coefficients	Wavelet coefficients[9]	BS-?	MAE	Chronological	Individual stocks. 6M/1Y
Amilon [2003]	$S/K, \tau, \sigma_{\mathrm{H}}, r$, lagged S	C_{Ask}/K, C_{Bid}/K	BS-H, BS-IM	ME, **MTE**, MSE	Chronological	OMX. 2Y
Carverhill and Cheuk [2003]	$K/S, \tau, \sigma_{\mathrm{I}}, r$	C/K, HR	CRR	**?TE**	Chronological	S&P500. 11Y
Gençay and Salih [2003]	$S/K, \tau, \sigma_{\mathrm{H}}, r$	C/K	BS-H	DM, MSE	Chronological	S&P500. 6Y
Healy et al. [2003, 2004][10]	$S/K, \tau$	C/K	None	MSE, R^2	Random	FTSE100. 6Y
Lajbcygier [2003, 2004]	$S/K, \tau$	$(C - C_{\mathrm{BS-IH}})/K$	None	MAE, MSE, R^2	Chronological	AOSPI. 3Y
Montagna et al. [2003]	S, τ	C	None	None	?	Simulation (BS)
Zapart [2003a][11]	$S/K, \tau, \sigma_{\mathrm{H}}, r$	C/K	BS-?	MAE	Chronological	Individual stocks. ?
Bennell and Sutcliffe [2004]	$S, K, S/K, \tau$, σ_{IM}, open interest, volume	$C, C/K$	BS-IM	MAE, ME, MPE, MSE	Chronological	FTSE100. 1Y
Choi et al. [2004]	$S, K, \tau, \sigma_?$	C	BS-?	?	Random	KOSPI200. 1Y
Dindar and Marwala [2004]	$S, \tau, \sigma_{\mathrm{H}}, r$	K/C	None	?	Random	South Africa Foreign Exchange. 3Y
Morelli et al. [2004]	$S, K, \tau, \sigma_{\mathrm{I}}, r$	C	None	?	?	Simulation (BS)
Pires and Marwala [2004a,b]	$K, \tau, \sigma_{\mathrm{H}}$	C	SVM	MAX, ME	?	Johannesburg Stock Exchange. 3Y
Xu et al. [2004]	$S, K, \tau, \sigma_{\mathrm{I}}, r$	C	None	R^2	Random	FTSE100. 5Y
Charalambous and Martzoukos [2005]	$S, K, \sigma_{\mathrm{H}}, r$, correlations[12]	$C - C^{10}_{\mathrm{LA}}$	LA-10	MAE, MAX, MSE	Chronological	Simulation (BS)
Hamid and Habib [2005]	$S, \tau, \sigma_{\mathrm{H}}, r$	C	None	MAE, MSE	?	S&P500. 12Y

[9] An ANN is used to predict the future volatility of the underlying. The volatility is represented in terms of wavelets and the underlying modelled as a binomial tree.

[10] These papers also derive prediction intervals for ANN estimates of option prices.

[11] This paper also treats the setup of Zapart [2002].

[12] Correlations between underlyings.

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Kakati [2005][2]	?	?	?	?	?	Individual stocks. ?
Ko et al. [2005], Ko [2009]	S, K, τ, σ_{H}	Coefficients[13]	BS-H	**MATE**	?	TAIEX. 1Y/2Y
Lin and Yeh [2005]	$S, K, \tau, \sigma_{\mathrm{H}}, r$	C	BS-H	MAE, MSE	?	TAIEX. 2Y
Pires and Marwala [2005]	$K, \tau, \sigma_{\mathrm{H}}$	C	SVM	MAX, ME, MSE	?	ALSI. 3Y
Tung and Quek [2005]	$S - K$, τ, σ_{H}	C	None	MSE, Correlation[14]	Random	GBP-USD. 1Y
Andreou et al. [2006][15]	S/K, τ, σ_{Cal}, σ_{H}, σ_{V}, r	C/K, $(C{-}C_{\mathrm{BS-Cal}})/K$, $(C - C_{\mathrm{BS-H}})/K$, $(C - C_{\mathrm{BS-V}})/K$	BS-Cal, BS-H, BS-V	MAE, MSE	Chronological	S&P500. 3Y
Blynski and Faseruk [2006]	$S/K, \tau$, σ_{H}, σ_{IH}	C/K, $(C - C_{\mathrm{BS-H}})/K$, $(C - C_{\mathrm{BS-N}})/K$	BS-H, BS-IH	MAE, MAPE, ME, MSE, R^2	?	S&P100. 7Y
Huang and Wu [2006], Huang [2008]	S/K, τ, σ_{K}	$(C - C_{\mathrm{BS-K}})/K$	SVM	MAE, MAPE, MSE	Chronological	TAIEX. 9M
Jung et al. [2006]	S, K, τ, σ_{IH}	C	BS-IH	MSE	?	KOSPI200. 1Y
Kim et al. [2006]	K, τ	σ_{Cal}	SI	MSE	?	S&P500. 1M
Liang et al. [2006]	$\hat{C}$[16]	C	BS-?, CRR	MAE	Chronological	Individual stocks. 5M
Mitra [2006]	$S, K, \tau, \sigma_{\mathrm{H}}, r$	C	None	MAE, MSE	Chronological	NIFTY50. 1Y
Pande and Sahu [2006]	S/K, τ, σ_{PCA}, r	C or(?) C/K	None	ME, MSE, Correlation[17]	?	Individual stocks. 1Y
Teddy et al. [2006]	$S - K$, τ, σ_{H}	C	None	MSE, Correlation[14]	Random	GBP-USD. 1Y
Tzastoudis et al. [2006]	$S, K, \sigma_{\mathrm{H}}$	C	BS-H	MAE, R^2	Chronological	S&P500. Several days

[13] Coefficients for a linear regression that returns option prices.

[14] Pearson correlation coefficient, a statistical measure to verify the goodness-of-fit between the predicted and desired function.

[15] This paper relies on the PhD thesis Andreou [2008].

[16] Various price estimations from parametric option pricing models.

[17] Correlation between the actual and computed prices.

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Wang [2006]	S/K, σ_{IH}, $(S-K)^+$, $C-(S-K)^+$, $CS/\sqrt{K}$	σ_{I}	BS-IH	MAE, MSE, R^2	?	Individual stocks. 2M
Amornwattana et al. [2007]	S, K, τ, r	$C-C_{\mathrm{BS-N}}$, σ_{I}	BS-H, BS-N	MAE, MSE	Chronological	Individual stocks. 3M
Gençay and Gibson [2007]	S, K, τ, σ_{G}, r	C	BS-G, BS-H, SV, SVJ	MAE, MSE	?	S&P500. 3Y
Gregoriou et al. [2007]	S_{Ask}, S_{Bid}, S_{Mid}, K, τ, σ_{I}, r	C	None	None	Random	FTSE100. 5Y
Healy et al. [2007]	S, K, τ, σ_{I}, r	C	None	R^2	Chronological	FTSE100. ?
Thomaidis et al. [2007]	S, K, τ	C	BS-G, BS-H	MAE, MSE	Chronological	S&P500. Several days
Zhou et al. [2007]	S/K, S, K, τ, r	C/K	BS-?, CRR	MAE, MAPE, ME, MSE, R^2	Chronological	Convertible bonds. 2Y
Andreou et al. [2008][15]	S/K, σ_{Cal}, σ_{H}, σ_{V}, r, kurtosis, skewness	C/K, $(C-C_{\mathrm{BS-Cal}})/K$, $(C-C_{\mathrm{BS-H}})/K$, $(C-C_{\mathrm{BS-V}})/K$	BS-Cal, BS-H, BS-V, CS	MAE, **MATE**, MdAE, MSE, **MTE**	Chronological	S&P500. 4Y
Chiu and Lin [2008]	S, C_{BS}, volume, and others	C	None	MSE	Chronological	Individual stocks. 1Y
Kakati [2008]	S/K, τ, σ_{G}, σ_{H}, σ_{IH}, r	C/K	BS-G, BS-H, BS-IH	MSE	?	Individual stocks. Several days
Mostafa and Dillon [2008][18]	S/K, τ, σ_{H}	C/K, σ_{I}	BS-H, SV	MAPE, **MATE**, MPE	?	FTSE100. 2Y
Quek et al. [2008]	lagged C	C	None	None	?	GBP-USD, Gold, Oil. 2Y
Saxena [2008]	S/K, τ, σ_{H}, r	$(C-C_{\mathrm{BS-H}})/K$	BS-H	MAE, ME, MPE, MSE, R^2	?	NIFTY50. 1Y
Teddy et al. [2008]	$S-K$, τ, σ_{H}	C	BS-H	MSE, R^2	Random	GBP-USD. 9M

[18] This paper relies on the PhD thesis Mostafa [2011].

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Tseng et al. [2008]	$S, K, \tau, \sigma_{\mathrm{G}}, r$	C	None	MAE, MAPE, MSE	?	TAIEX. 2Y
Chen [2009]	$S, K, \tau, \sigma_{\mathrm{H}}, r$	C	BS-H, SVM	MAE, MSE	Chronological	S&P500. Several days
Gradojevic et al. [2009]	$S/K, \tau$	C/K	BS-H	DM, MSE, MSPE	Chronological	S&P500. 8Y
Leung et al. [2009]	$\sigma_{\mathrm{H}}, \sigma_{\mathrm{IH}}$, volume, open interest	σ_{I}	BS-IH, Linear, Polynomial	ME	Chronological	Several currencies. 17Y
Liang et al. [2009]	$\hat{C}$[16]	C	CRR, SVM	MAE, MAPE	Chronological	Individual stocks. 2Y
Martel et al. [2009]	$S/K, \tau, \sigma_{\mathrm{H}}, r$	C_{Bid}/K, C_{Ask}/K	BS-H	ME, MSE, **MTE**	Chronological	IBEX35. 2Y
Samur and Temur [2009]	$S, K, \tau, \sigma_{\mathrm{H}}, r$	C	None	MAE, MSE, R^2	?	S&P100. Several days
Wang [2009a]	$S/K, \tau, \sigma_{\mathrm{G}}, r$	C/K	None	MAE, MAPE, MSE	?	TAIEX. 2Y
Wang [2009b]	$S/K, \tau, \sigma_{\mathrm{G}}, \sigma_{\mathrm{H}}, \sigma_{\mathrm{IH}}, r$	C/K	None	MAE, MAPE, MSE	?	TAIEX. 2Y
Andreou et al. [2010][15]	$S/K, \tau$	σ_{I}	BS-Cal, CS, SV, SVJ	MAE, **MATE**, MdAE, MSE	Chronological	S&P500. 3Y
Barunikova and Barunik [2011]	S, K, τ	C	BS-H	MAE, MAPE, MSE	Random	S&P500. 3Y
Gradojevic and Kukolj [2011]	$S/K, \tau, \sigma_{\mathrm{IH}}, r$	C/K	BS-H	DM, MAPE, MSE	Chronological	S&P500. 7Y
Liu and Zhang [2011]	$S/K, \tau, \sigma_{\mathrm{H}}$,[19] r	C/K	BS-H	MAE, MSE	Chronological	Individual stocks. 2Y
Phani et al. [2011]	S, K, τ	C	BS-?, SVM	MAE	?	NIFTY50. 2Y
Tung and Quek [2011]	σ_{IH}	σ_{I}	None	MAPE, MSE, R^2	Chronological	HSI. 5Y
Wang [2011]	$S/K, S, \tau, \sigma_{\mathrm{Cal}}, r$	C	SV, SVJ, SVM	MAE, MAPE	Chronological	Several currencies. 7M
Ahn et al. [2012]	Lagged σ_{I}, Greeks	Sign$(\Delta\sigma_{\mathrm{I}})$	None	Accuracy	Chronological	KOSPI200. 2Y

[19] More precisely, a Markov regime switching model is used to estimate the volatility.

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Chen and Sutcliffe [2012]	$S/K, \tau$	C/K, $(C - C_{\text{BS-H}})/K$, HR	BS-H	MAE, ME, MSE	Random	Sterling futures. 2Y
Mitra [2012]	S, K, τ, σ_{H}, r	C	BS-H	ME, MSE	Chronological	NIFTY50. 3Y
Shin and Ryu [2012]	S, K, τ, r	HR	None	MPE	Chronological	KOSPI200. 10Y
Wang et al. [2012]	S, K, τ, σ_{Cal}, σ_{G}, σ_{H}, σ_{IH}	C	None	MAE, MAPE, MSE	Chronological	TAIEX. 2Y
Chang et al. [2013]	$S/K, \tau, \sigma_{\text{G}}, r$	C or(?) C/K	None	MAE, MAPE	?	TAIEX. 2Y
Hahn [2013]	S/K, τ, σ_{H}, r	C/K	SV	MAE, MAPE, MSE	Chronological	Individual stocks. 10Y
Can and Fadda [2014]	$S/K, S, \tau, r$	C/K	BS-H	MAE	Chronological	S&P100. Several days
Lai [2014]	$S/K, \tau, r$	σ_{I}	KR, SI	KS	?	Simulation (BS, SV, SVJ)
Park et al. [2014]	$S/K, \tau$	C/K	BS-H, SV	MSE	Chronological	KOSPI200. 10Y
von Spreckelsen et al. [2014]	S/K, K, τ	C/K	None	MSE, R^2	Chronological	EUR-USD. 1M
Ludwig [2015]	$S/K, \tau$	σ_{I}	Quadratic	MSE, R^2	?	S&P500. 12Y
Liu and Huang [2016]	$S/K, \tau$	$(C - C_{\text{BS-H}})/K$	BS-H	MAE, MAPE, ME, MSE	?	HSI. 6Y
Montesdeoca and Niranjan [2016]	S/K, τ, σ_{H}, volume	C/K	None	MSE	Chronological	FTSE100. ?; Individual stocks. ?
Culkin and Das [2017]	$S/K, \tau, \sigma_{\text{I}}, r$	C/K	None	MSE, R^2	Chronological	Simulation (BS)
Das and Padhy [2017]	S/K, τ, $\widehat{C}$[16]	C	BS-H, SVM	MAE, MSE	Chronological	NIFTY50. 2Y
Fang and George [2017]	σ_{H}	σ_{I}	None	MSE, R^2	Chronological	Simulation (BS); WTI. 1M
Palmer and Gorse [2017]	S, K, σ_{I}, r	C	None	MAE, MdAE, MAPE	Chronological	Simulation (BS)
Yang et al. [2017][20]	$K/S, \tau$	C/S	BS-?, Kou, VG	MAPE, MSE	?	S&P500. 10Y
Ferguson and Green [2018]	S, τ, σ_{I}, correlations[12]	C	None	MSE	Chronological	Simulation (BS)

[20] This paper relies on the PhD thesis Zheng [2017].

Authors & year	Features	Outputs	Benchmarks	Performance measures	Partition method	Underlyings
Ackerer et al. [2019]	$\log(K/S)$, τ, $\log(K/S)\tau^{-0.5}$, $\log(K/S)\tau^{-0.95}$	σ_{I}	None	MAPE, MSE	Random	S&P500. 1M
Buehler et al. [2019a,b]	$\log(S)$	HR	BS-I	**CVaR**	Chronological	Simulation (BS, SV); S&P500. 5Y
Cao et al. [2020]	S/K, τ, σ_{V}, underlying return	σ_{I}	HW	MSE	Random	S&P500. 8Y
Jang and Lee [2019]	?	C	BS-Cal, BW, KR, LSM, LV, SVJ, SVM	MAE, MAPE, MPE, MSE	?	S&P100. 9Y
Liu et al. [2019b]	$S/K, \tau$	σ_{I}	None	MAE, MAPE, MSE	Chronological	Simulation (BS)
Liu et al. [2019c]	S/K, τ, σ_{Cal}, r	$(C - C_{\mathrm{BS-H}})/K$	BS-Cal, SVJ	MAE, **MATE**, MPE, MSE	Chronological	DAX. 4Y
Karatas et al. [2019]	S/K, τ, r, ?	C/K	None	MSE, R^2	Chronological	Simulation (BS, SV, VG)
Palmer [2019]	S/K, $\sigma_{\mathrm{I}}\sqrt{\tau}$, r	C/K	BS-I, LSM	MAE, MAPE	Chronological	Simulation (BS)
Zheng et al. [2019]	$S/K, \tau$	σ_{I}	SSVI	MAPE	?	S&P500. 10Y
Ruf and Wang [2020b]	S/K, $\sigma_{\mathrm{I}}\sqrt{\tau}$, Δ, $\mathcal{V}$, Vanna	HR	BS-I, HW, Linear	MSE	Chronological	Simulation (BS, SV); S&P500. 8Y; STOXX50. 3Y

Table 2.1 This table summarises more than 150 papers that use ANNs as a nonparametric option pricing or hedging tool. These papers are compared in terms of features (or so-called explanatory variables), outputs of the ANN, benchmark models, data partition between training and test sets, and the underlyings along with the time span of the data. The performance measures marked bold are related to evaluations along multiple periods. We refer to Tables 2.2–2.5 for a dictionary of all abbreviations used here.

C	Option price
$C_{\mathrm{BS-X}}$	Option price given by the Black-Scholes formula; see Table 2.3 for the different meanings of X
C^{n}_{LA}	Option price given by n-step multi-dimensional lattice scheme
HR	Hedging ratio
K	Strike price
S	Stock price
r	Interest rate
Γ	Gamma: second-order sensitivity of option price with respect to underlying price
Δ	Delta: sensitivity of option price with respect to underlying price
$\mathcal{V}$	Vega: sensitivity of option price with respect to volatility
ρ	Rho: sensitivity of option price with respect to interest rate
σ_{Cal}	Volatility from calibration (e.g., constant across strikes and maturities)
σ_{G}	GARCH–generated volatility
σ_{H}	Historical volatility
σ_{I}	Implied volatility
σ_{IH}	Implied historical volatility
σ_{IM}	At-the-money implied volatility
σ_{K}	Volatility obtained from Kalman filter
σ_{PCA}	Macroeconomic variables that contribute the most to volatility, determined by principle component analysis
σ_{V}	Volatility index such as VIX and DVAX
τ	Time to maturity

Table 2.2 This table presents notations and abbreviations for features and outputs, used in Table 2.1.

BS-Cal	Black-Scholes formula with calibrated volatility
BS-G	Black-Scholes formula with GARCH-generated volatility
BS-H	Black-Scholes formula with historical volatility
BS-I	Black-Scholes formula with contract-specific implied volatility
BS-IH	Black-Scholes formula with historical implied volatility
BS-IM	Black-Scholes formula with at-the-money implied volatility
BS-K	Black-Scholes formula with volatility obtained from Kalman filter
BS-N	Black-Scholes formula with ANN-generated volatility
BS-V	Black-Scholes formula with volatility index, such as VIX or VDAX
BW	Barone-Adesi and Whaley [1987] pricing method
CRR	Cox et al. [1979] model
CS	Corrado and Su [1996] model
HW	Hull and White [2017] model
Kou	Kou [2002]'s jump diffusion model
KR	Kernel regression
LA-n	n-step multi-dimensional lattice scheme
Linear	Linear regression on features
LSM	Longstaff and Schwartz [2001] method
LV	Local volatility model
Quadratic	Quadratic regression on features
SI	Spline interpolation
SSVI	Surface stochastic volatility inspired model, see Gatheral and Jacquier [2014]
SV	Stochastic volatility models, such as Heston [1993] or GARCH
SVJ	Stochastic volatility with jumps model, see Bates [1996] or Carr et al. [2003]
SVM	Support vector machine
VG	Variance Gamma model, see Madan et al. [1998]

Table 2.3 This table presents abbreviations for various benchmarks, used in Table 2.1.

DM	Diebold and Mariano test	
KS	Kolmogorov and Smirnov two-sample test	
MAE	Mean absolute error	$\frac{1}{N}\sum \lvert\hat{y}_i - y_i\rvert$
MAPE	Mean absolute percentage error	$\frac{1}{N}\sum \frac{\lvert\hat{y}_i - y_i\rvert}{y_i}$
MAX	Maximum error	$\max_i \lvert\hat{y}_i - y_i\rvert$
MdAE	Median absolute error	$\sup_z \left\{\frac{1}{N}\sum \mathbf{1}_{\lvert\hat{y}_i - y_i\rvert < z} \leq 0.5\right\}$
ME	Mean error	$\frac{1}{N}\sum(\hat{y}_i - y_i)$
MPE	Mean percentage error	$\frac{1}{N}\sum \frac{\hat{y}_i - y_i}{y_i}$
MSE	Mean squared error	$\frac{1}{N}\sum(\hat{y}_i - y_i)^2$
R^2	Coefficient of determination	$1 - \frac{\sum(\hat{y}_i - y_i)^2}{\sum(\bar{y} - y_i)^2}$
SR	Sharpe ratio of a trading ratio	
%E	Sample-wise percentage error	$\frac{\hat{y}_i - y_i}{y_i}$
CVaR	Conditional value-at-risk	
MATE	Mean absolute tracking error	$\frac{1}{N}\sum e^{-rT_i}\lvert V(T_i)\rvert$

MTE	Mean tracking error	$\frac{1}{N}\sum e^{-rT_i}V(T_i)$
PE	Prediction error	$\sqrt{\mathrm{MTE}^2+\frac{1}{N}\sum\left(e^{-rT_i}V\left(T_i\right)-\mathrm{MTE}\right)^2}$

Table 2.4 This table presents abbreviations and definitions for performance measures, used in Table 2.1. Here, $\hat{y}_i$ is the estimated option price / implied volatility / portfolio value, y_i is the target value, $\bar{y}$ is the average of target values, and N denotes the number of samples. Moreover, $V(T)$, also called tracking error, denotes the terminal value at T of a hedged option portfolio starting with zero wealth. All performance measures marked bold are related to evaluations along multiple periods.

ALSI	South African All Share Index
AOSPI	Australian All Ordinaries Share Price Index
BUND	German treasury bond
DAX	German stock index
DEM	Deutsche Mark
FTSE100	UK Financial Times Stock Exchange 100 index
HSI	Hong Kong Heng Seng Index
IBEX35	Spanish stock index
KOSPI200	Korea Composite Stock Price Index
NIFTY50	Indian National Stock Exchange Fifty
NIKKEI225	Japanese stock index
OMX	Swedish stock index
S&P100	US Standard & Poor's 100
S&P500	US Standard & Poor's 500
STOXX50	Eurozone stock index
TAIEX	Taiwanese stock index
WTI	US Light Sweet Crude Oil Futures

Table 2.5 This table presents abbreviations for various stock market indices and other underlyings, used in Table 2.1. For the shortcuts used to describe simulation data, we refer to Table 2.3.

In the following, we compare and classify papers listed in Table 2.1 in terms of features, outputs, performance measures and benchmarks, data partition methods, underlying assets and time span.

2.2.1 Features

To estimate the option price, the underlying price and the strike price are two indispensable variables. Two ways of feeding these two variables into an ANN as input have been suggested. One way is to use the underlying price and strike price separately. An alternative is to use a ratio (i.e., moneyness) instead. Several arguments are formulated in the literature in favor of using moneyness:

- Using moneyness instead of the stock price and the strike price separately reduces the number of inputs and thus makes the training of the ANN easier; see Hutchinson et al. [1994].

- Many parametric models assume that the statistical distribution of the underlying asset's return is independent of the level of the underlying. Hence, the option pricing function is homogeneous of degree one with respect to the underlying stock price and the strike price, so that only moneyness is needed to learn the function. Incorporating this assumption into the ANN can potentially reduce overfitting; see Hutchinson et al. [1994], Lajbcygier and Connor [1997a,b], Anders et al. [1998], and Garcia and Gençay [1998, 2000].

- Moneyness is a stationary input feature in contrast to the stock price and the strike price. Using it helps generalisation and reduces overfitting; see Ghysels et al. [1998] and Garcia and Gençay [1998, 2000]. Our own experiments also confirm that the use of moneyness can significantly improve the generalisation, see Section 4.5.2.

Bennell and Sutcliffe [2004] undertake a systematic experiment on various choices of input features, including underlying price, strike price, moneyness, and on choices of outputs, including option price and option price divided by strike.

Apart from the underlying price and the strike price, volatilities are also widely used as input features. This can be done in several different ways. The most relevant ones are the following:

- Using historical volatility estimates of the underlying as features.
- Using volatility indices such as VIX as features.
- Using option-specific implied volatilities as features.
- Using GARCH forecasts of (realised or implied) volatility as features.

Table 2.2 lists further volatility features. The choices of features by the different papers are worked out in the 'Features' column of Table 2.1. There exist also several papers that do not use any volatility-type feature as input for their ANNs. They are mostly early papers. Due to computational constraint or their preference for simplicity, they assume that volatility stays constant in a short period of time. Thus the volatility is a constant rather than a variable. With the rolling window scheme that fits a regression on a short window, the ANN is expected to learn such an implicit constant from the data.

A few papers, e.g., Blynski and Faseruk [2006], Andreou et al. [2008], or Wang [2009b], compare different volatility features. Here we summarize their results. Blynski and Faseruk [2006] show an ANN outperforms the conventional Black-Scholes when using historical volatility as input, but underperforms when using implied volatility. Andreou et al. [2008] show that replacing historical by implied volatility improves the performance of ANNs. Wang [2009a] argue that an ANN with a GARCH volatility forecast outperforms that with historical and implied volatility as features.

Some papers investigate whether additional features can help the ANN with prediction. To name a few, Ghaziri et al. [2000] and Healy et al. [2002] incorporate option open interests.

Samur and Temur [2009] study whether the inclusion of variance improves the performance of the ANN. Montesdeoca and Niranjan [2016] explore the potential prediction power of trading volume, option interest, and other variables. Cao et al. [2020] investigate the benefit from using the underlying return.

2.2.2 Outputs

The papers of Table 2.1 can also be categorised in terms of their outputs:

- The most common output is the option price. Depending on whether moneyness is used, or underlying price and strike price are used separately, the output can be the option price or the option price divided by the strike price. Some papers also investigate the ANN's ability when it is trained to learn the so-called bias; i.e., the difference between market price and a price estimated by a parametric model. Such an ANN is called hybrid ANN; see, for example, Boek et al. [1995] or Lajbcygier and Connor [1997a,a]. While most of the early papers train their ANNs to fit prices, Garcia and Gençay [2000] train to prices, but validate to hedging errors in order to determine the network size that gives the lowest hedging error. Andreou et al. [2010] emphasize the relevance of choosing the right loss function when interested in the hedging task.

- Another type of output is the implied volatility. The obtained implied volatilities can be converted to option prices by the Black-Scholes formula. Mostafa and Dillon [2008] compare ANNs that output option prices to ANNs that output implied volatilities. More recently, Liu et al. [2019b] evaluate an ANN's ability to approximate the inverse of the Black-Scholes formula.

- The third kind of output (always denoted by HR in Table 2.1) is a sensitivity or a hedging ratio. Only a few papers discuss such an architecture for an ANN. The first papers are Carverhill and Cheuk [2003], Chen and Sutcliffe [2012], and Shin and Ryu [2012]. More recently, Buehler et al. [2019a,b] and Ruf and Wang [2020b] follow up on this line of research. Buehler et al. [2019b] consider also the hedging of exotic options such as barrier options.

We could have also added the so-called calibration papers to Table 2.1, which construct ANNs to map prices to specific model parameters or vice versa. Instead we decided to dedicate Section 2.4.1 below to these papers.

2.2.3 Performance measures and benchmarks

When evaluating the performance of ANNs, common statistical measures are mean absolute error (MAE), mean absolute percentage error (MAPE), and mean squared error (MSE).[21] These are related to evaluations over a single period, in terms of pricing or hedging. Some papers also propose to evaluate the ANN's performance over multiple periods. For instance, Hutchinson et al. [1994] introduce the mean absolute tracking error (MATE) and prediction error (PE), which appear also in many later papers. Buehler et al. [2019a] introduce the conditional value-at-risk (CVaR) for evaluating hedging strategies.

An ANN's performance should also be compared to a benchmark, for example, a parametric pricing model. The most widely used benchmark is the Black-Scholes formula, which requires a volatility as input. As Table 2.1 summarises a historic volatility estimate is used the most often. Also certain implied volatilities (e.g., historical or at-the-money) appear in the literature. Blynski and Faseruk [2006] compare historical realised and historical implied volatility for the Black-Scholes benchmark.

The Black-Scholes formula with contract-specific implied volatility is a valid benchmark for the hedging task. For the pricing task, however, such a benchmark would lead to zero error as by definition of implied volatility it prices options without errors. Thus, for the pricing task, the Black-Scholes formula with contract-specific implied volatility is not a suitable benchmark .

In addition to the Black-Scholes formula, other widely used parametric benchmarks are stochastic volatility pricing models; e.g., used in Gençay and Gibson [2007], Jang and Lee [2019], or Liu et al. [2019b]. Ruf and Wang [2020b] observe that if a benchmark is chosen that incorporates both delta and vega hedging then an ANN does not outperform even a simple two-factor regression model.

For American type options, benchmarks used are the Barone-Adesi and Whaley [1987] pricing method (e.g., Lajbcygier [2002]), and the Cox-Ross-Rubinstein model (e.g., Chen and Lee [1999]).

2.2.4 Data partition methods

An ANN needs to be trained on a training set (in-sample) and then tested on a test set (out-of-sample). There exist several ways to partition a data set into such a training and test set. The first way is chronologically. That is, the early data constitutes the training set, and the late data constitutes the test set. Table 2.1 indicates that most of the papers follow this approach.

[21] Several papers use equivalent versions of the measures in Table 4. For example, sometimes root mean squared error is used instead of mean squared error. For consistency, in Table 1, we have made the corresponding adjustments.

However, some studies violate this time structure in the data by choosing a different way to partition the data. Violations can be introduced by randomly partitioning the data into a training and a test set or by using a so-called 'odd-even split.'

As we mentioned in Chapter 1, a cross-section of options share the same underlying price. Random partitioning breaks the time structure and overlooks such cross-section dependence. It allows the underlying prices to show both in the training and test sets. When an ANN is trained on a training set constructed in such a way, the error on the test set underestimates the generalisation error of the ANN. Yao et al. [2000] and our companion paper Ruf and Wang [2020b] provide more discussion on this point.

Some papers only work with independent draws from various distributions, and therefore do not involve any time series structure. Although these papers randomly partition the whole data set into a training and test set, no time structure is violated. Hence, in Table 2.1, we classify this approach as chronological partition.

A related issue is the existence of time-inhomogeneity in financial data; in particular, volatility changes over time. When working with real data, some papers use a rolling window method to tackle this issue, especially when the time range is long and volatilities are not included as input features. Such papers include Hutchinson et al. [1994], Dugas et al. [2009], and others. However, it remains an open question how big window sizes need to be.

2.2.5 Underlying assets and time span

Both simulation data and real data can be used to train an ANN for a specific problem. Simulation data is much easier to work with, since it is free of noise and sometimes a close-to-optimal solution is available as a benchmark, such as for the Black-Scholes and Heston models. For instance, le Roux and du Toit [2001], Morelli et al. [2004], and Karatas et al. [2019] investigate an ANN's performance on simulation data. Most other papers use either both simulation and real data or only real data. Options on S&P500 have been studied by the largest number of papers, since they are the most liquidly traded options. Options on FTSE100 and S&P100 have also been studied in several papers. We refer to Table 2.5 for a more complete list of all the underlyings being used.

Some papers focus on American option pricing and hedging. Underlyings for American options are usually individual stocks. Papers involving American options include Kelly [1994], Chen and Lee [1999], Meissner and Kawano [2001], Pires and Marwala [2004a], Pires and Marwala [2005], and Amornwattana et al. [2007]. As elaborated in Subsection 2.4.3, American options can also be priced differently by ANNs, via learning the value function or optimal stopping rule in a dynamic programming setting; see Kohler et al. [2010] and Becker et al. [2019].

2.3 Recommended papers

Among the many papers of Table 2.1, we would like to highlight a few. Such a selection is clearly personal and subjective. Despite the subjective selection, we believe that this list might serve as a good starting point to get an overview of this field. We also provide a Google Scholar citation count.[22] As mentioned before, Table 2.1 focuses only on those papers that use ANNs to estimate option prices and related variables. Recently there have been many interesting and promising developments in the use of ANNs for calibration purposes or as computational tools. These papers are not included here, but Section 2.4 provides some pointers to this literature.

Among the following highlighted papers, some are the first to propose innovative solutions. Others investigate the problem in a systematic way.

- Hutchinson et al. [1994] (# citations: 864) is one of the first papers and the most highly cited one to use ANNs to estimate option prices. They introduce a methodology to evaluate the hedging performance over multiple periods, applied by many papers later on.

- Lajbcygier and Connor [1997a] (# citations:[23] 58) is one of the first papers that propose to learn the difference between model prices and observed market option prices.

- Anders et al. [1998] (# citations: 118) compare the performance of ANNs and of the Black-Scholes benchmark when using different volatility estimates.

- Garcia and Gençay [2000] (# citations:[24] 241) incorporate a homogeneity hint for the ANN. Hence, this is one of the first papers that embed financial domain knowledge into the construction of an ANN.

- Carverhill and Cheuk [2003] (# citations: 18) first propose an ANN that outputs hedging strategies directly, instead of option prices.

- Bennell and Sutcliffe [2004] (# citations: 101), Chen and Sutcliffe [2012] (# citations: 18), and Hahn [2013] (# citations: 13) provide three extensive literature surveys.

- Dugas et al. [2009] (# citations:[25] 307) first design an ANN architecture that enforces no-arbitrage conditions such as convexity of option prices.

[22] As of March 1, 2021.

[23] This count includes the number of citations for Lajbcygier and Connor [1997b].

[24] This count includes the number of citations for Garcia and Gençay [1998].

[25] This count includes the number of citations for Dugas et al. [2001].

- Andreou et al. [2010] (# citations: 24) combines an ANN with parametric models to learn functions that return implied model parameters. Such an ANN essentially calibrates parametric models.

- Buehler et al. [2019a] (# citations: 100) develop a novel framework for hedging a portfolio of derivatives in the presence of market frictions, and allow convex risk measures as loss functions. Their framework allows pricing and hedging without observing option prices.

As this is a subjective selection, we also would like to highlight our companion paper Ruf and Wang [2020b, 2021], which provides a new benchmark based on delta-vega hedging and discusses data leakage issues.

2.4 Related papers

In the last few years, many novel techniques have been developed to apply ANNs to tasks arising in option pricing beyond the nonparametric estimation of prices and hedging ratios. In this section we provide a few pointers to this rapidly developing literature.[26]

2.4.1 Calibration

As already mentioned in Section 2.3, Andreou et al. [2010] propose an ANN that returns implied model parameters. Hence, the ANN essentially calibrates parametric models. We observe a recent surge of the application of ANN to calibration. In this approach option prices are first mapped to a parametric model, which is then used to determine option prices. This approach can move the computationally heavy calibration off-line, thus significantly accelerating option pricing.

Abu-Mostafa [2001] use neural networks to calibrate the Vasicek model with a consistency hint to produce valid parameters. More recently, Hernandez [2017] uses an ANN to calibrate a single-factor Hull-White model. Dimitroff et al. [2018], McGhee [2018] and Liu et al. [2019a] calibrate stochastic volatility models, and Stone [2019] and Bayer et al. [2019][27] calibrate rough volatility models. Itkin [2019] highlights some pitfalls in the existing approaches and proposes resolutions that improve both performance and accuracy of calibration.

[26] At times it was not always clear cut to us whether a paper should be included in Table 2.1 or in this section. For example, the calibration papers of Section 2.4.1 could have been put into Table 2.1 as mentioned in Section 2.2.2. Similarly, Barucci et al. [1996, 1997], discussed in Section 2.4.2, learn the Black-Scholes model and hence could have been put into Table 2.1.

[27] For more details, see also Bayer and Stemper [2018] and Horvath et al. [2021].

Going the ‘indirect’ way via first calibrating a model and then using it to determine the hedging ratio has at least two advantages. First, it provides additional interpretability as only the calibration step is replaced by an ANN. This can be important for a financial entity subject to regulatory requirements. Second, it provides an arguably strong tailor-made regularisation effect as it replaces a nonparametric estimation task by the task of estimating a model with usually less than 5-10 parameters.

2.4.2 Solving partial differential equations

The option pricing problem sometimes involves solving a partial differential equation (PDE). Barucci et al. [1996, 1997] use the Galerkin method and ANNs for solving the Black-Scholes PDE. E et al. [2017], Han et al. [2018], and Beck et al. [2019] utilize ANNs to solve high-dimensional semilinear parabolic PDEs. They propose to reformulate the PDEs using backward stochastic differential equations, and the gradient of the unknown solutions is approximated by ANNs. Their numerical results suggest that the method is effective for a wide variety of (possibly high-dimensional) problems. One case study involves the pricing of European options on 100 defaultable underlying assets. There are several recent papers, such as Henry-Labordère [2017], Sirignano and Spiliopoulos [2018], Chan-Wai-Nam et al. [2019], Huré et al. [2020] , Jacquier and Oumgari [2019], and Vidales et al. [2019], who have developed this application of ANNs further.

2.4.3 Approximating value functions in optimal control problems

ANNs can be used to approximate value functions that appear in dynamic programming, for example arising in the American option pricing problem; see for example Ye and Zhang [2019]. Kohler et al. [2010] use ANNs to estimate continuation values for high-dimensional American option pricing. Becker et al. [2019] use ANNs for optimal stopping problems by learning the optimal stopping rule from Monte Carlo samples. ANNs have also been proposed to approximate the value function of a dynamic program for real option pricing, see Taudes et al. [1998].

In this context, we also mention Fecamp et al. [2019], who use an ANN as a computational tool to solve the pricing and hedging problem under market frictions such as transaction costs.

2.4.4 Further work

Albanese et al. [2021] use an ANN to compute the conditional value-at-risk and expected shortfall necessary for certain XVA computations, by solving a quantile regression.

We would like to also mention Halperin [2017] and Kolm and Ritter [2019] who suggest a reinforcement learning methodology to take market frictions into account for the option pricing task.

Finally, generative ANNs have been suggested recently as a non-parametric simulation tool for stock prices; see, for example, Henry-Labordère [2019], Kondratyev and Schwarz [2019], and Wiese et al. [2019b]. Such simulation engines could then be used for option pricing and hedging, a direction still to be explored systematically. Just after finishing this survey, Wiese et al. [2019a] proposed a generative ANN for option prices (instead of stock prices).

2.5 Digression: regularisation techniques

As the advance of hardware allows for bigger ANNs to be built, regularization techniques have become more important as part of the ANN training. Such techniques include L^2, dropout, early stopping, etc.; see Ormoneit [1999], Gençay and Qi [2001], Gençay and Salih [2003], and Liu et al. [2019b]. Complementing these universal regularisations, several papers embed financial domain knowledge into ANNs, either at the stage of architecture design or training. Let us here also mention the suggested feature design by Lu and Ohta [2003a,b], who consider the pricing of exotic options and suggest to use digital option prices as features.

For the architecture design the following has been suggested:

- Homogeneity hint. Garcia and Gençay [1998, 2000] incorporate a homogeneity hint by considering an ANN consisting of two parts, one controlled by moneyness and the other controlled by time-to-maturity.

- Shape-restricted outputs. Dugas et al. [2001, 2009], Lajbcygier [2004], Yang et al. [2017], Huh [2019], and Zheng et al. [2019] enforce certain no-arbitrage conditions such as monotonicity and convexity of the ANN pricing function by fixing an appropriate architecture.

At the training state the following techniques are being used:

- Data augmentation. Yang et al. [2017] and Zheng et al. [2019] create additional synthetic options to help with the training of ANNs.

- Loss penalty. Itkin [2019] and Ackerer et al. [2019] add various penalty terms to the loss function. Those terms present no-arbitrage conditions. For example, parameter configurations that allow for calendar arbitrage are being penalised.

In the context of ANN training, we would like also to mention Niranjan [1996], de Freitas et al. [2000a,b], and Palmer [2019]. These papers propose and examine novel training algorithms for ANNs and illustrate them in the context of option hedging; these algorithms include the extended Kalman filter, sequential Monte Carlo, and evolutionary algorithms.

Chapter 3

Hedging with Linear Regressions and Neural Networks

This chapter is based on joint work with Prof. Johannes Ruf, and contains the major content of the papers

Ruf, J and Wang, W, Hedging with linear regressions and neural networks, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3580132, accepted by the *Journal of Business and Economic Statistics* subject to minor revisions,

and

Ruf, J and Wang, W, Information leakage in backtesting, working paper.

We thank Matthias Büchner, Agostino Capponi, Aleš Černý, Jean-Pierre Fouque, Camilo Garcia, Lukas Gonon, Harald Oberhauser, Philipp Illeditsch, Antoine Jacquier, Johannes Muhle-Karbe, Peter Spoida, Josef Teichmann, and James Wolter for helpful discussions on the subject matter of this paper. We are grateful to Deutsche Börse, in particular, Peter Spoida, for providing us with Euro Stoxx 50 options and futures tick data. We are indebted to two anonymous referees and an associate editor for several very insightful comments that improved the paper. The code to reproduce results in this paper can be found at https://github.com/weiguanwang/Hedging_Neural_Networks.git.

3.1 Introduction

Beginning with Hutchinson et al. [1994] and Malliaris and Salchenberger [1993b], *artificial neural networks* (ANNs) are being proposed as a nonparametric tool for the risk management of options. Since then about 150 papers have been published that apply ANNs to price and hedge options; see Section 3.3 for several pointers to this literature. We show that for the estimation of the optimal hedging ratio ANNs do *not* outperform simple linear regressions that use only standard option sensitivities.

We study a specific and well defined risk management application, namely the reduction of variance of the hedging error in daily options' trading. We have in mind a financial entity (or 'operator') acting as a market maker; i.e., taking on (short) positions in options as 'inventory' to satisfy some market demand. Alternatively, such financial entity could be on the 'buy-side,' taking on short positions in options to collect the volatility risk premium. This entity would then try to hedge the exposure to the price movements in the underlying by trading it. Such entity might not be bound by regulatory requirements, as 'sell-side' institutions are. Hence it would not be required to provide a specific model as an interim step. The marking to market accounting convention requires a good control of the hedging error for short periods, even when considering long-dated options.

More precisely, we consider a one-period model and imagine an operator who is short an option (or a cross section of options). To reduce the variance of her portfolio she is allowed to buy or sell the underlying. Today, the operator sells the option, say at price C_0. She is now allowed to buy δ shares of the underlying at price S_0 and $C_0 - \delta S_0$ units of the risk-free asset. Then today's portfolio value equals $V_0 = 0$. Tomorrow, her portfolio has value

$$V_1^\delta = \delta S_1 + (1 + r_{\mathrm{onr}} \Delta t)(C_0 - \delta S_0) - C_1, \tag{3.1}$$

where S_1 and C_1 denote tomorrow's prices of the underlying and the option, respectively, r_{onr} is the over-night rate at which the operator can borrow / lend money, and $\Delta t = 1/253$. The operator' goal is to choose δ in such a way that the variance of tomorrow's wealth, $\mathrm{Var}[V_1^\delta]$ is minimised.

To make headway, since Δt is small, we are allowed to approximate the variance by the expected squared mean. Indeed, if the expected return on the risky asset happens to be equal to the risk-free return then the expected value $\mathbf{E}[V_1^\delta]$ does not depend on δ at all. Then the operator's objective is to minimise the *mean squared hedging error* (MSHE)

$$\mathbf{E}\left[\left(V_1^\delta\right)^2\right] = \mathbf{E}\left[(\delta S_1 + (1 + r_{\mathrm{onr}} \Delta t)(C_0 - \delta S_0) - C_1)^2\right]. \tag{3.2}$$

Let us assume for the moment that the option is a European call. Then a standard and simple choice is using the *practitioner's Black-Scholes Delta* (BS-Delta)

$$\delta_{\mathrm{BS}} = \mathbf{N}(d_1), \tag{3.3}$$

where **N** denotes the cumulative normal distribution function and

$$d_1 = \frac{1}{\sigma_{\text{impl}}\sqrt{\tau}}\left[\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\sigma^2_{\text{impl}}\right)\tau\right]. \tag{3.4}$$

Here, τ is the time-to-maturity in year fraction, σ_{impl} the annualised implied volatility of the option, K the strike price, and r the risk-free interest rate corresponding to the option's maturity. The operator would choose $\delta = \delta_{\text{BS}}$; if the option was a put then she would choose $\delta = \delta_{\text{BS}} - 1$ in line with put-call parity. Since the interest rate r is negligible, we assume for the moment that it is zero. Then the BS-Delta can be written as a function of two variables, namely the moneyness $M = S_0/K$ and the time-proportional implied volatility $\sigma_{\text{impl}}\sqrt{\tau}$. Thus, we get the functional representation

$$\delta_{\text{BS}} = f_{\text{BS}}\left(M, \sigma_{\text{impl}}\sqrt{\tau}\right).$$

It is now reasonable to study other functionals.We shall replace f_{BS} by an ANN f_{NN} with the two input features M and $\sigma_{\text{impl}}\sqrt{\tau}$, trained to minimise the expression in (3.2). That corresponds to a nonparametric estimation of the optimal hedging ratio that minimises the variance of the hedging error. We will provide more details on the implementation in Section 3.3. The motivation to study ANNs arises from the large amount of historical data available, the universal approximation ability of ANNs, and the sometimes unrealistic assumptions underlying parametric models.

To benchmark the hedging performance of the ANN, we introduce linear regression models that lead to hedging ratios that are linear in several option sensitivities. They are motivated by the *leverage effect*, credited to Black [1976]. The leverage effect describes the negative correlation between an underlying's price and its volatility. To illustrate how this matters, consider a call and assume it is hedged with the BS-Delta $\delta_{\text{BS}} > 0$. If now the underlying's price goes up so do the call price and the hedging position. Due to the leverage effect, the underlying (implied) volatility tends to go down simultaneously, thus having a negative effect on the option price. Indeed, everything else equal, both call and put prices go up as (implied) volatility increases – their 'Vega' is positive. The BS-Delta δ_{BS} does not take into consideration this additional effect. As we only allow hedging with the underlying the obvious change is to hedge only partially, i.e., use the hedging ratio $\delta_{\text{LR}} = a\delta_{\text{BS}}$, where a is estimated (in a training set). Here, LR stands for linear regression. For the moment it suffices to note that these arguments let us expect $a > 1$ for puts and $a < 1$ for calls. (It turns out that hedging with $a\delta_{\text{BS}}$, where $a = 0.9$ for calls and $a = 1.1$ for puts works extremely well on real datasets; see Subsection 3.5.5.) We shall discuss such simple modifications of the BS-Delta in Section 3.4, all based on statistical hedging models involving various option sensitivities.

The performance of the ANN and the benchmarks is tested on daily end-of-day mid-prices obtained from *OptionMetrics* and tick data provided by *Deutsche Börse*. These data are described in more detail in Section 3.2. We also vary the length Δt of the hedging period from 1 hour to 2 days. All in all, the ANN performs well in terms of MSHE relative to the BS-Delta, even when the latter is being used with contract-specific implied volatility. However, using the linear regression hedging ratios δ_{LR} performs roughly as well or at times better than δ_{NN}. They lead to roughly 15%-20% reduction in the MSHE. For a summary of the results, see Sections 3.5.1 and 3.5.2. In addition, Sections 3.5.3 and 3.5.4 contain extensive simulation experiments using data generated from the standard Black-Scholes model and from Heston's stochastic volatility model.

An interpretation of these observations is that the option sensitivities already encapsulate all relevant nonlinearities in the data necessary for the hedging task. Hence, the ANN seems to be able to learn the leverage effect, but cannot improve on a simple linear regression involving the relevant option sensitivities. What have we learned? Initially we were satisfied about the outperformance of the ANN relative to the BS-Delta on real datasets. When investigating what the ANN is learning, the linear regression models appeared as natural competitors. These statistical models are extremely simple – for the easiest such model one only replaces the BS-Delta by a multiple of it. Nevertheless, as far as we know, these models have not been used in the literature to benchmark more complicated models.

We proceed as follows. Section 3.2 describes the datasets and the experimental setup. Section 3.3 introduces the HedgeNet architecture and implementation. This section also discusses the advantage of outputting directly the hedging ratio instead of option prices and then using a sensitivity as hedging ratio. Section 3.4 describes how the leverage effect motivates various benchmark models to be compared with ANNs. Section 3.5 presents the experimental results. Section 3.6 reflects on the potential information leakage from two sources, either from disregarding the data's time series structure or from cleaning the datasets. Section 3.7 summarises the main findings. Several appendices provide further details on the various sections.

3.2 Datasets, data preparation, and setup of experiments

This section presents the data used. Section 3.2.1 explains the data-generating mechanism for the simulated data. Section 3.2.2 and Section 3.2.3 describe the two real datasets containing options on the S&P 500 and Euro Stoxx 50. Section 3.2.4 discusses the experimental setup. Section 3.2.5 contains additional details on the datasets. Section 3.2.6 concludes the section by providing some economic implications of reducing the MSHE.

3.2.1 Simulated data: Black-Scholes and Heston

For the simulation study two data-generating mechanisms are considered. In the first one, the underlying's price process is simulated from the Black-Scholes stochastic integral equation

$$S_t = 2000 + \mu \int_0^t S_u \mathrm{d}u + \sigma \int_0^t S_u \mathrm{d}W_u,$$

with annualised rate of return $\mu = 0.1$, and annualised volatility $\sigma = 0.2$. In the second example the underlying's price process is simulated from the Heston [1993] model given by

$$S_t = 2000 + \int_0^t \sqrt{Y_u} S_u \mathrm{d}W_u; \tag{3.5a}$$

$$Y_t = Y_0 + \kappa \int_0^t (\theta - Y_u)\, \mathrm{d}u + \sigma_Y \int_0^t \sqrt{Y_u} \mathrm{d}\widetilde{W}_u; \tag{3.5b}$$

$$\mathrm{Cov}(\mathrm{W_t}, \widetilde{\mathrm{W}}_\mathrm{t}) = \rho t, \tag{3.5c}$$

with initial and long-term variance $Y_0 = \theta = 0.04$, rate of mean reversion $\kappa = 5$, volatility of variance $\sigma_Y = 0.3$, and correlation $\rho = -0.6$. Here the volatility $\sqrt{Y_t}$ of the underlying is stochastic and modelled as the square root of a process mean-reverting to 0.04. Thanks to Feller's test of explosions, the volatility is always strictly positive. We intentionally omit the drift to focus on the role that stochastic volatility plays.

We first simulate 1.25 years of the underlying's price from the Black-Scholes and Heston model, respectively. For the Black-Scholes dataset, we use exact simulation. For the Heston dataset we use a standard Euler and Milstein scheme. The initial value of 2000 is relevant to get a realistic number of options as their generation depends on the underlying's absolute value, as we explain next.

Then, along the simulated spot path, options are created following the Chicago Board Options Exchange (CBOE) rules. Details of these rules are provided on http://www.cboe.com/products/stock-index-options-spx-rut-msci-ftse/s-p-500-index-options/s-p-500-options-with-a-m-settlement-spx/spx-options-specs. The idea is the following. The option expiration date is always the fourth Friday of its expiration month. The expiration months are the 12 immediate calendar months, plus some additional long-term months (we do not generate options for those long-term months). At each expiration date new options are created, so that the market still trades options with 12 expiration months. In general, the strike price step is set to 5 dollars. The two strike prices closest to the current underlying's price are initially listed. If the underlying's price is close to any one of the two strikes, a third strike will be included to cover the larger range. New series are generally added when the underlying's price trades through the highest or lowest available strike price for each expiration.

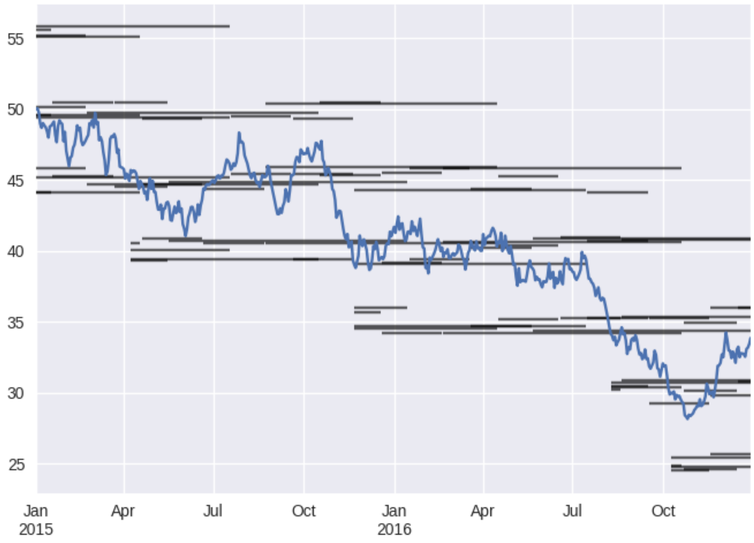


Fig. 3.1 A simulated path of underlying asset, along which options are created following the CBOE rules. The initial underlying price is chosen to be 50 for the sake of readability, since the number of options increases as underlying prices increases.

Next, we price the options on each trading day using the Black-Scholes formula and the standard pricing formulas available for Heston, respectively; see, for example, Albrecher et al. [2007]. Here, we set the dividend and interest rate to zero. Moreover, in the Heston case, we fix this pricing measure, under which $\widetilde{W}$ is also Brownian motion.

These 1.25 years of simulated data correspond to the in-sample data (training and validation), on which the benchmarks and ANNs are trained. To estimate the MSHE, more data are simulated; however those data are only used to estimate the out-of-sample performance of the different statistical models. Choosing a time length of 1.25 years is done for the following reason. As explained Section 3.2.4, when training the ANNs for the real datasets, we split the data up in training, validation, and out-of-sample (test) data using the ratio $4{:}1{:}1$. For the simulated datasets we keep this ratio and choose the training set to be one year long. This then yields 1.25 years of training and validation data. Simulating options according to the CBOE rules yields roughly the same magnitude of training data as available in each time window of the real datasets.

After computing the option prices and the sensitivities necessary for the statistical models, the data are organised in a table so that each row corresponds to exactly one observation, i.e., one option at one trading day (along with tomorrow's price for training). Finally, samples with option price less than 0.01 (the tick size) or moneyness M outside of the interval $[0.8, 1.5]$ are removed. This means that if an option has a time-to-maturity of 90 trading days, it might appear, for example, 85 times in the dataset. The option might have a moneyness outside of the interval or a too small price for the other four trading days.

3.2.2 S&P 500 end-of-day midprices

We obtained daily closing bid and ask prices on calls and puts written on the S&P 500 between January 2010 and June 2019 from *OptionMetrics*[1]. We interpreted the midprice as the true market price. Figure 3.2 displays a sample of the obtained options, namely those puts with price quotes in the first three months of 2010 or 2015. Sensitivities are provided for the majority of options and are filled in for missing values. The required interest rates were interpolated from the rates provided by *OptionMetrics*. For maturities less than one week (in which case *OptionMetrics* does not provide the corresponding rates), we used the *Overnight Libor Rates* from *Bloomberg*. The results presented below are robust to whether we use computed sensitivities for all options or the sensitivities provided by *OptionMetrics* where available.

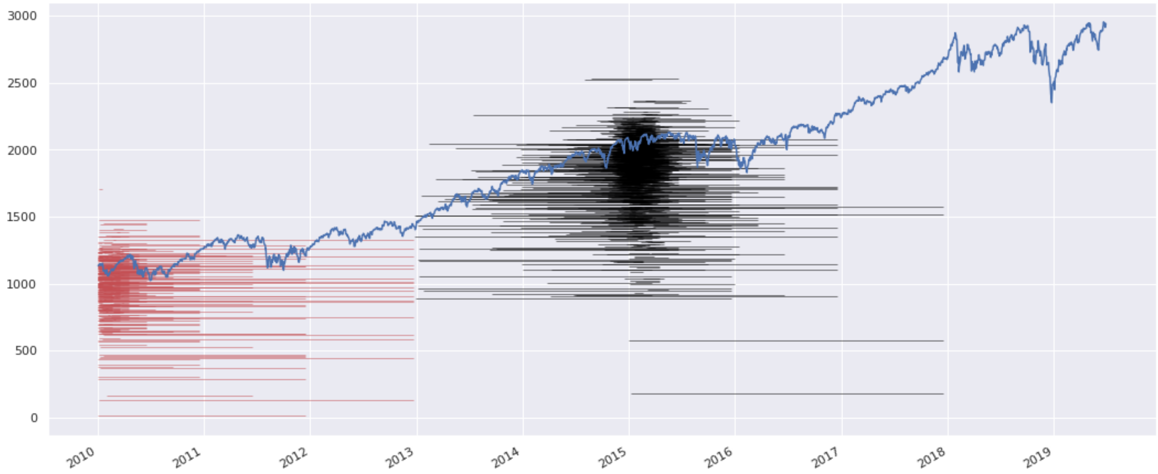


Fig. 3.2 A sample of the obtained put options along with the underlying's (S&P 500) price process in blue. Only options that have a trading volume of more than 1000 on some trading day are included. Each red (black) line segment represents a put option that had price quotes within the first quarter of 2010 (2015). The corresponding strike is indicated as the value on the y–axis. Small random vertical shifts are added to increase the visibility of the options.

We again arrange the data so that each row corresponds to exactly one option at one day. In the cleaning process, we remove the following samples:

- Samples with negative time value.
- Samples with time-to-maturity less than 1 day.
- Samples where the moneyness is outside the interval $[0.8, 1.5]$.
- Samples with an implied volatility higher than 100% or smaller than 1%.

[1] See https://optionmetrics.com/.

- Samples with zero trading volume.
- Samples where the ask is at least twice the bid.
- Samples with bid less than 0.05.
- Samples that do not have available next trade prices.

3.2.3 Euro Stoxx 50 tick data

We are grateful to *Deutsche Börse*, who provided us with tick data[2] of Euro Stoxx 50 index options and futures between January 2016 and July 2018.

We now briefly outline how we process these data. If several trades are executed at exactly the same time stamp we aggregate these orders and consider the volume-weighted average price. We match each option transaction with the most recent tick price of the future with the shortest maturity (again, volume-weighted if several trades happen simultaneously). These futures, which are the most liquid ones, shall be used to hedge the option position. The computation of the option sensitivities requires a risk-free rate. We use interpolated Euro LIBOR rates from *Thomson Reuters' DataStream*.

To train and measure the hedging performance we require the option price after Δt (1 hour, 1 day, 2 days, etc.). There might not be a trade exactly after this time period. Hence we allow a *matching tolerance window* of 6 minutes, equivalent to 0.1 hours. Hence, for example, if Δt is a business day and we have a trade on Monday, say at 2.12pm, then we match it with the first price observation of this option on Tuesday after 2.12pm. If there is no transaction before 2:18 pm, this sample gets discarded. (We refer to Subsection 3.6.2 for a discussion of potential information leakage introduced in this step.)

In the cleaning process, the following samples are removed:

- Samples with negative time value.
- Samples with time-to-maturity less than 1 day.
- Samples where the moneyness is outside the interval $[0.8, 1.5]$.
- Samples with an implied volatility higher than 100% or smaller than 1%.
- Samples on expiry dates of a future.
- Samples that cannot be matched to a next trade (within the mathing tolerance window of 6 minutes).
- Samples that are traded in the first or last half an hour of each trading day.

[2] See https://datashop.deutsche-boerse.com/samples-dbag/File_Description_Eurex_Tick.pdf for a description of the data.

3.2.4 Data preparation and experimental setup

As discussed in the introduction, our goal is to determine the hedging ratio δ as a function of observable quantities to minimise the variance over one period of the hedged portfolio

$$V_1^\delta = \delta S_1 + (1 + r_{\mathrm{onr}} \Delta t)(C_0 - \delta S_0) - C_1. \tag{3.6}$$

Here S_0 and S_1 denote the prices of the hedging instrument at the beginning and end of the period and C_0 and C_1 denote the prices of the call or put. We study how well an ANN performs in this task on simulated data (Black-Scholes and Heston – see Subsection 3.2.1), on end-of-day midprices (see Subsection 3.2.2), and on tick data (see Subsection 3.2.3). We benchmark these results with linear regression models for the hedging ratio δ.

Each of the datasets is split up into in-sample and out-of-sample ('test') data. Both the ANN and the benchmark models are trained to (estimated by) the in-sample dataset only. The variance of the hedged portfolio is approximated by the MSHE. The performance of each of the methods is measured on the out-of-sample dataset as follows:

$$\mathrm{Var}(\mathrm{V}_1^\delta) \approx \mathrm{MSHE} = \frac{1}{\mathrm{N_{test}}} \sum_{\mathrm{t,j}}^{\mathrm{N_{test}}} \left(100 \frac{\mathrm{V}_{\mathrm{t+1,j}}^\delta}{\mathrm{S_t}} \right)^2, \tag{3.7}$$

where δ is either modelled by an ANN or by a linear regression. Both the indexing and the normalisation by $S_t/100$ need explanation.

First of all, the indexing has changed from (3.6) to (3.7). Indeed, each traded option yields a series of samples, one for each trading period. Moreover, several options corresponding to different strikes (indexed by j) are being priced in any specific period (e.g., a day). To emphasise this point, the samples are double indexed in (3.7). Next, (3.7) normalises the value of the hedging portfolio by dividing it by $S_t/100$. This normalisation 'removes the units' and allows to compare errors across the different datasets, and arguably more importantly, across time. Equivalently, at any point of time t, instead of replicating a full option we replicate the fraction $100/S_t$ of this option.

One could have considered a different normalisation. For example, in (3.7), one could have divided by the time-t-option price C_t instead of S_t. This would induce a different weighting of the samples. However, a fixed Dollar position in a far out-of-the money option is riskier than in an at-the-money option. Indeed, a move in the underlying tends to have a larger effect on the far out-of-the money position. Hence from a risk perspective, the alternative normalisation would put too much weight on far out-of-the money options. For this reason we choose the normalisation of (3.7).

We now provide more details on how we prepare each dataset. First we store each dataset in a dataframe as in Table 3.2. We then remove all in-the-money samples. That is, if at one specific date an option was in the money, we discard this specific date for the corresponding option.

Index	Date	Features				Additional information					Target
		$\sigma_{\text{impl}}\sqrt{\tau}$	M	δ_{BS}	$\mathcal{V}_{\text{BS}}$	S_0	S_1	C_0	r_{onr}	CP flag	C_1
0	2018/07/02	0.047	1.003	0.531	9.357	100	98.223	2.002	1.0	0	1.130
⋮											

Table 3.2 This table presents a (simplified) preview of one of the four processed datasets. The 'Features' columns are used as inputs for the ANN and the linear regressions. The labels $\sigma_{\text{impl}}\sqrt{\tau}$ and M denote the time-proportional implied volatility and moneyness of the option. The labels δ_{BS} and $\mathcal{V}_{\text{BS}}$ are the BS-Delta and Vega. The CP flag indicates whether the corresponding option is a call or a put. Prices and sensitivities are all normalised.

We break up the S&P 500 dataset in 14 overlapping time windows of length 3 years in order to understand whether the comparisons between the ANNs and the linear regressions are consistent across time. In each time window, the first 900 days form the in-sample set, while the last 180 days are used for the out-of-sample set, yielding a ratio 5:1. For the training of the ANN, the 900 days are furthermore split into 720 days of training and and 180 days of validation yielding a ratio 4:1:1. We roll the time windows forward by 180 days, so that sample appears maximally once in the aggregated out-of-sample set. The Euro Stoxx 50 dataset is much shorter, and we do not break it up in different time windows. This leads to 750 (600+150) days in the in-sample set and 150 days in the out-of-sample set, yielding again a ratio 4:1:1.

The Black-Scholes and Heston datasets consist both of a single time window of 1.5 years. The first 450 days form the in-sample set. For the ANN, the 450 days are furthermore split into 360 (training) and 90 (validation) days. To get a more precise estimate of the MSHE, twenty out-of-sample sets of 90 days are simulated, as illustrated in Figure 3.6.

In practice, one would expect to retrain each statistical model weekly or daily instead of every 180/90 days. For computational limitations we are not able to do so. (Currently, training and running one ANN configuration for the 14 S&P 500 time windows takes about 10 hours on a GTX 1060 6GB GPU cluster.) However, we do treat the statistical benchmark models in the same way, also only retraining them every 180/90 days. Thus the comparisons below are valid.

3.2.5 Sizes of in-sample and out-of-sample sets

Recall that we only consider out-of-the-money and at-the-money options. Figure 3.3 shows the number of samples in each time window for the S&P 500 and Euro Stoxx 50 datasets. For the S&P 500 data (ranging from 2010 to 2019), the overall number of samples is 2.6 million. On average, there are 1144 samples per trading day. In each time window, the number of total samples grows continually. More puts than calls are traded, and the number of puts traded grows faster than that of calls traded. For the Euro Stoxx 50 data (ranging from 2016 to 2018), the number of samples overall is 0.62 million. On average, there are 988 samples per trading day. Roughly the same number of puts and calls are traded.

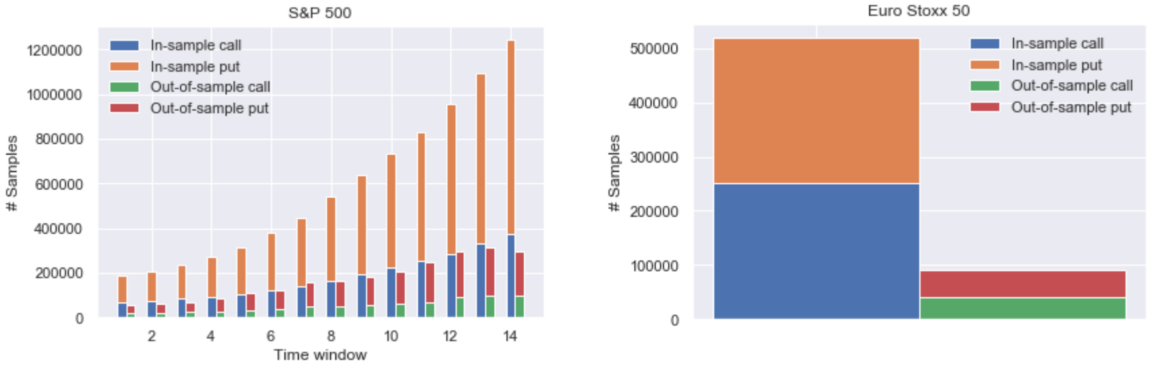


Fig. 3.3 Sample size of out-of-the-money and at-the-money calls and puts in in-sample and out-of-sample sets. The left panel corresponds to the S&P 500 dataset, the right panel to the Euro Stoxx 50 dataset.

Figure 3.4 shows the distribution of moneyness in the S&P 500 and Euro Stoxx 50 datasets. As we only consider out-of-the-money and at-the-money options each sample with moneyness less than 1 corresponds to a call, and similarly, each sample with moneyness greater than 1 corresponds to a put. The distribution of moneyness for Euro Stoxx 50 data is more concentrated around a moneyness of 1. This difference is explained by the fact that the S&P 500 dataset consists of end-of-day quotations of all listed options, while the Euro Stoxx 50 dataset consists of tick prices of all traded options. Since close-to-the money options are more frequently traded, the Euro Stoxx 50 dataset hence has relatively more such samples.

Figure 3.5 shows the distribution of time-to-maturity for both datasets. The S&P 500 dataset has many more long-dated options than the Euro Stoxx 50 dataset.

We conclude by summarising that the in-sample dataset in the Black-Scholes dataset is 0.36 million and in the Heston dataset 0.26 million. As explained in Subsection 3.2.1, for the simulated datasets we created options according to the CBOE rules and then removed all in-the-money samples. Since the underlying tends to move upwards in the Black-Scholes dataset (the drift rate was set to 10%) we expect to have more out-of-the money put samples

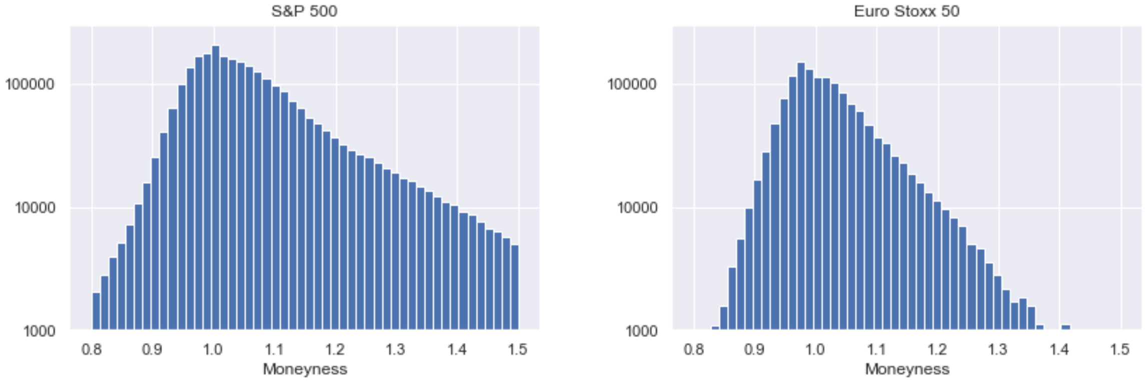


Fig. 3.4 Histogram of moneyness in the S&P 500 (left panel) and the Euro Stoxx 50 (right panel) datasets. Samples with moneyness less than 1 correspond to calls, and samples with moneyness greater than 1 to puts.

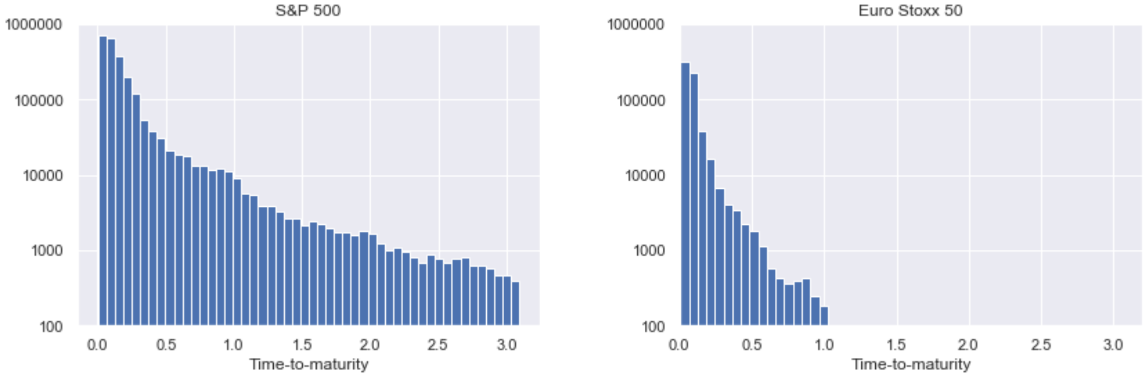


Fig. 3.5 Histogram of time-to-maturity in the S&P 500 (left panel) and Euro Stoxx 50 (right panel) datasets.

than call samples. Indeed, an investigation of the Black-Scholes dataset yields that we have roughly 91k call samples and 277k put samples in the in-sample set. It turns out that the Heston in-sample dataset, just by chance (the simulated underlying's path process moves from 2000 to about 2600) also has more put samples (192k) than call samples (69k).

3.2.6 Digression: economic interpretation of the mean squared hedging error

We now briefly comment on the economic gains when using hedging strategies that lead to reduced MSHEs. To this end, we consider a financial entity (the 'operator') acting as a market maker. This operator sells a cross section of delta-hedged puts or calls. In the classical one-period framework of Stoll [1978] (see also Chapter 2.2 in O'hara [1997]), the operator charges a premium (e.g., through a bid-ask spread) to take on the additional inventory (i.e.,

Algorithm 1 This algorithm describes the rolling window scheme used for the S&P 500 and Euro Stoxx 50 datasets. We use the training and evaluation of ANN on the S&P 500 dataset as an example. For the linear regression models, step 13–15 should be removed, and the training and validation dataset are merged to one in-sample dataset.

```
 1: procedure ROLLING($t_0, T$)                                  ▷ The time of the data is from $t_0$ to $T$.
 2:     Define a function $\Gamma(a, b)$ to return the daily samples between time $a$ and time $b$.

 3:     Initialize $\Delta \leftarrow 180$ days.
 4:     Initialize $t_a \leftarrow t_0$.
 5:     Initialize $t_b \leftarrow t_a + 720$ days.
 6:     Initialize $t_c \leftarrow t_b + 180$ days.
 7:     Initialize $t_d \leftarrow t_c + \Delta$.

 8:     Initialize $i \leftarrow 0$.

 9:     while $t_d < T$ do
10:         Initialize $\mathcal{D}_A \leftarrow \Gamma(t_a, t_b)$.                         ▷ training dataset
11:         Initialize $\mathcal{D}_B \leftarrow \Gamma(t_b, t_c)$.                       ▷ validation dataset
12:         Initialize $\mathcal{D}_C \leftarrow \Gamma(t_c, t_d)$.                             ▷ test dataset
13:         Standardize $\mathcal{D}_A, \mathcal{D}_B$, and $\mathcal{D}_C$.

14:         Train the network on $\mathcal{D}_A$, while simultaneously evaluating MSHE on $\mathcal{D}_B$.

15:         Retrieve the ANN weight $\hat{\mathbf{w}}$ such that MSHE is the lowest on $\mathcal{D}_B$.
16:         Compute MSHE on $\mathcal{D}_C$.

17:         $t_a \leftarrow t_a + \Delta$; $t_b \leftarrow t_b + \Delta$; $t_c \leftarrow t_c + \Delta$; $t_d \leftarrow t_d + \Delta$; $i \leftarrow i + 1$.
18:     end while
19:     return MSHE on $\mathcal{D}_C$.
20: end procedure
```

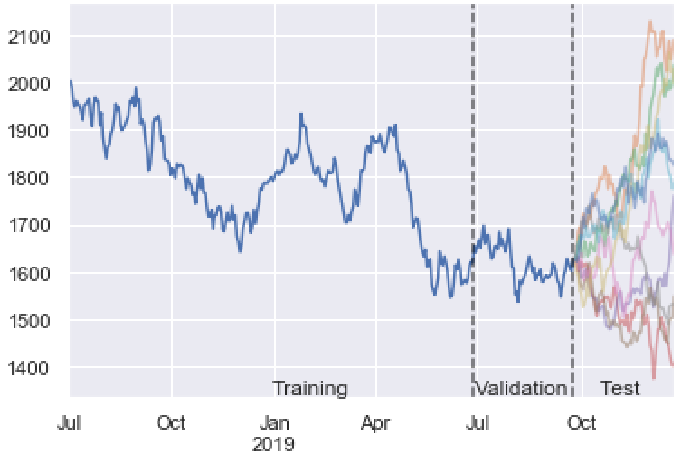


Fig. 3.6 The single simulated price path on which options are created for the in-sample set, and the multiple paths on which options are created for the out-of-sample sets. To reduce the estimate of the generalisation error, we compare the different methods below for each out-of-sample set and average them.

the short position of delta-hedged options). Reducing the MSHE allows the operator to charge a lower premium as we outline next.

Formally, we equip the operator with quadratic utility $x \mapsto x - \gamma x^2/2$, where $\gamma > 0$ denotes her coefficient of risk aversion. We suppose that the delta-hedged short-position is uncorrelated with the operator's optimal wealth. Furthermore, we assume that the expected return of a delta-hedged option position does not depend on the hedging strategy (e.g., if the expected return of the risky asset equals the risk-free return) and set it to zero for simplicity. Under Bertrand competition, the operator charges $\gamma/2$ times the MSHE as a premium. Hence, if the MSHE can be reduced by a certain percentage, the premium reduces by the same percentage times $\gamma/2$. For example, if the MSHE error is reduced by 15% and $\gamma = 2$ then the premium decreases by 15%.

A similar argument applies if the operator acts as a speculator, interested in maximising the Sharpe ratio of her option position. If the expected return of a delta-hedged option position does not depend on the hedging strategy and the MSHE is reduced by 15% then the new Sharpe ratio is $1/\sqrt{0.85} \approx 1.085$ times the old one.

3.3 HedgeNet

There exists a long line of research on the use of ANNs in the context of option pricing and hedging. Ruf and Wang [2020a] provide an overview of this literature. Here we only give a few pointers to papers that we found especially insightful. Early on, Hutchinson et al. [1994] suggest ANNs as nonparametric alternative for the pricing of options. They show that already quite small ANNs with only a few nodes perform well for the pricing task. Garcia and Gençay [2000] are among the first to introduce financial domain knowledge (a so called 'homogeneity hint') in the design of ANNs. This type of regularisation improves the pricing performance of ANNs further. Carverhill and Cheuk [2003] propose an ANN that directly outputs hedging strategies, instead of first outputting option prices and then deriving hedging strategies as sensitivities. Dugas et al. [2009] suggest an ANN architecture that guarantees that the outputted prices satisfy a set of no-arbitrage conditions. Buehler et al. [2019a] bring several innovations forward. In order to train their ANN, additional artificial data are drawn from an appropriately fitted econometric model. Their framework for hedging options includes the presence of transaction costs and other market frictions, allowing general convex risk measures as loss functions. All these references discussed here consider the pricing / hedging task over the lifespan of an option.

We now introduce the ANN used in this study. As discussed in the introduction, we focus on the one-period setup, and benchmark the hedging performance of the ANN with appropriate linear regressions based on the options' sensitivities, as described in the next section. The ANN maps an option's relevant features (e.g, moneyness and time-proportional implied volatility) to a hedging ratio δ_{NN}. In Subsection 3.3.1 we provide details about the architecture, implementation, and training of such an ANN. Subsection 3.3.2 provides some additional motivation why the ANN is designed to output directly the hedging ratio instead of the option price.

3.3.1 Architecture of HedgeNet, its implementation and training

An ANN is a composition of simple elements called neurons, which maps input features to outputs. Such an ANN then forms a directed, weighted graph.

As we shall discuss below in Subsection 3.3.2 it is not satisfactory to compute or estimate option prices and then use their sensitivities as hedging ratios. It is better to obtain the hedging ratio, our quantity of interest, directly. Hence, we desire that the ANN returns a hedging ratio and not a price. However, when training such an ANN what should it be trained to? Optimal hedging ratios are not provided in the data. For this reason, we design an ANN, named HedgeNet, to have two parts, as illustrated in Figure 3.7.

The first part, a multilayer fully-connected feed-forward neural network (FCNN), transforms features into a hedging position, which is then turned by the second part into the replication value $\widehat{C}_1 = V_1 + C_1$. This output of HedgeNet can then be trained to the observed option prices C_1 at the end of each period by minimising the sum of squared differences.

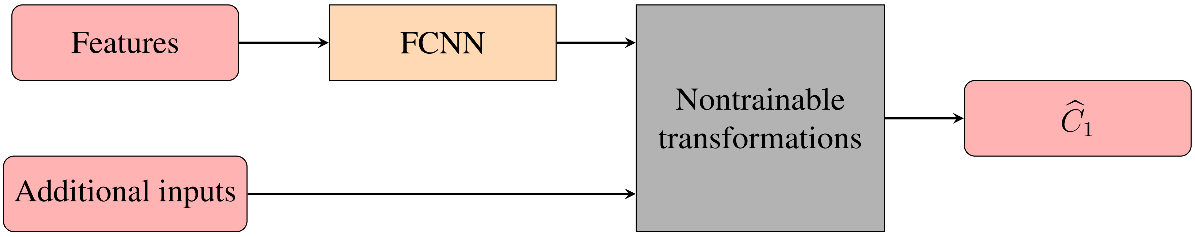


Fig. 3.7 A schematic graph of HedgeNet. The features are transformed into a hedging position by a fully-connected feed-forward neural network (FCNN). The additional input is used to compute the value $\widehat{C}_1$ of the hedging position.

The FCNN has two hidden layers with 30 nodes each, connected by ReLU activation. The output of the FCNN is provided by a linear node (with truncation at zero and one) and corresponds to the the hedging ratio δ_{NN}. We tried different architectures, for example 100 nodes in each hidden layer, or three (instead of two) hidden layers with 30 nodes each. Motivated by the representation of the BS-Delta in (3.3), we also tried the cumulative distribution function N of a standard normally distributed random variable as output function instead of the linear output function. None of these modifications changed the overall conclusions below. We also tried a modification, where we interpret the output not as the hedging ratio but as the 'bias' term $\delta - \delta_{\mathrm{BS}}$, which corrects the BS-Delta. Such change did not help the performance of the ANN – a similar observation as in Chen and Sutcliffe [2012].

As illustrated in Figure 3.8, the non-trainable transformation module turns the hedging ratio δ_{NN} into the replication value $\widehat{C}_1$ by following (3.1). As the data includes both puts and calls, this module also requires an option type flag, which is set to 1 in the case of a put and to 0 in the case of a call. If the sample is a put, the module replaces δ_{NN} by $\delta_{\mathrm{NN}} - 1$ in line with put-call parity. The non-trainable transformation module consists of a series of affine transformations, and hence does not affect the universal approximation property, discussed for example in Yarotsky [2017].

All numerical experiments are run on a standard desktop with GPU accelerated computation. We use *Python* as programming language. The ANN is implemented with the deep learning framework *Tensorflow* along with *Keras*. The inputs to the trainable part of HedgeNet are standardised[3]. The weights of the ANN are initialised via the 'Xavier' initialiser (Glorot and Bengio [2010]) and the 'Adam' optimiser (Kingma and Ba [2015]) is applied for training the ANN. Appendix 4.1 contains details on the choice of additional hyperparameters.

For each dataset we consider three different feature sets for the trainable part of HedgeNet:

- ANN$(M;\, \sigma_{\mathrm{impl}}\sqrt{\tau})$: The first one is already indicated in Figure 3.8. It uses moneyness M, time-proportional implied volatility $\sigma_{\mathrm{impl}}\sqrt{\tau}$, and a flag to indicate whether the

[3] Standardizing features works by removing the mean of each dimension of the features and scaling their variance to one. This simple technique helps speed up the training of ANN.

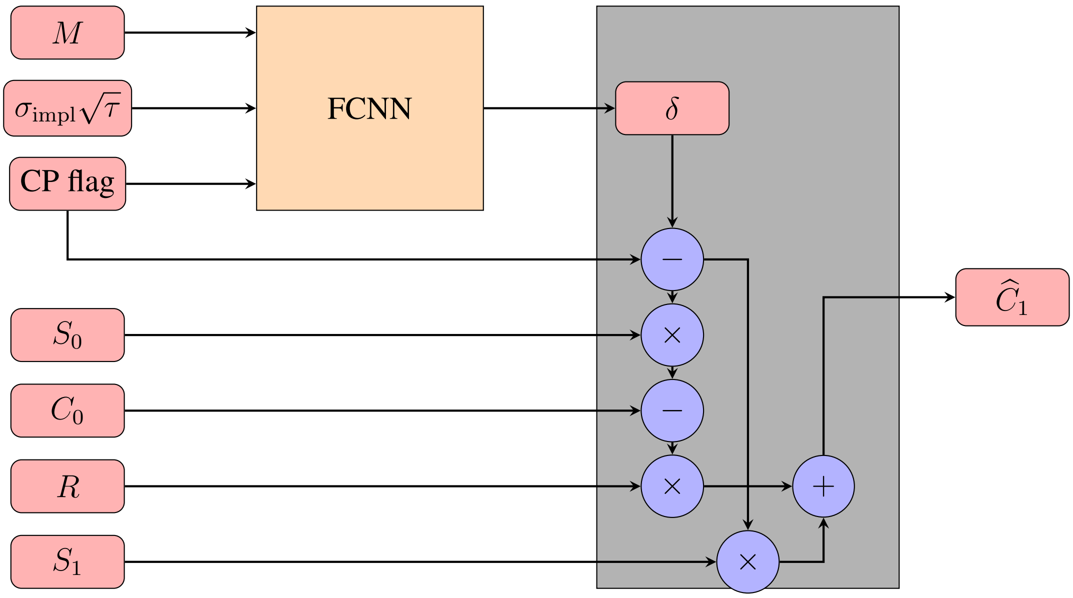


Fig. 3.8 A detailed schematic presentation of HedgeNet. Recall that $M = S_0/K$ and $\sigma_{\text{impl}}\sqrt{\tau}$ are moneyness and time-proportional implied volatility. 'CP flag' is a Boolean flag for the option type; it equals 1 for puts and 0 for calls. Next, S_0 and S_1 are the underlying's market prices at the beginning and end of the hedging period, C_0 denotes the option price at the beginning of the period, and $\widehat{C}_1$ denotes the replication value. Finally, $R = 1 + r_{\text{onr}}\Delta t$ is the risk-free overnight return.

option is a call or a put. It is worth pointing out that using moneyness instead of the underlying's price and the strike price separately offers a better generalisation performance. The most important reason for its better performance is that moneyness resembles more a stationary feature compared to the underlying's price and strike price separately. Indeed, options are created and traded only for a certain range of moneyness values. Ghysels et al. [1998], Garcia and Gençay [2000], and Ruf and Wang [2020a] provide more comments on the advantage of using moneyness. The choice of time-proportional implied volatility is motivated by the fact that volatility squares with the square root of time; see also the expression for δ_{BS} in (3.3)&(3.4).

- ANN(Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ): Motivated by the leverage effect discussed in Section 3.4 below, we also consider a second set of features consisting of δ_{BS}, $\mathcal{V}_{\text{BS}}$, $1/\sqrt{\tau}$, and the put-call flag. Here $\mathcal{V}_{\text{BS}}$ denotes *Vega*, the sensitivity of the option price with respect to the implied volatility.

- ANN(Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ): Since we shall use Vanna, the sensitivity of Delta with respect to volatility, as a feature for linear regression benchmarks in Section 3.4, we also consider using a third feature set consisting of the three sensitivities, $1/\sqrt{\tau}$, and the put-call flag.

3.3.2 Digression: Why outputting the hedging ratio instead of computing price sensitivities?

Most ANNs constructed in the literature for the risk management of options first learn the pricing function. Then in a second step hedging strategy is computed as the sensitivity of the option price with respect to the underlying price; see Ruf and Wang [2020a] for an overview of the literature. In contrast, HedgeNet allows to predict the underlying hedging position directly. In this way, the hedging strategy is no longer interpreted as a sensitivity.

From a risk-management point of view the hedging ratio is the main quantity of interest. It has been recommended, see for example Bengio [1997] or Claeskens and Hjort [2003], to estimate relevant quantities directly. This is in line with the important observation made in Lyons [1995] that different models might yield similar option prices but completely different hedging strategies. Obtaining directly the hedging ratio also avoids the otherwise necessary step to differentiate, possibly numerically, the trained option prices.

There are further important advantages of outputting directly the hedging ratio. Computing sensitivities usually does not take into consideration that other model parameters also might change, in line with the underlying. Hence, such sensitivities tend to be not optimal for reducing the MSHE. Theoretical results supporting this observation are ample; see for example Denkl et al. [2013]. This discussion is continued in Subsection 3.4.2 below. Moreover, as Buehler et al. [2019a] show, training to hedging ratios allows to incorporate market frictions conveniently.

3.4 Linear regression models as benchmarks

We now discuss how we benchmark the hedging performance of the ANN. Although not very reasonable, one benchmark could be not hedging at all, i.e., $\delta = 0$. In this case the variance of the hedging error is just the variance of the change in the option price. More reasonable is to use the BS-Delta, obtained from the Black-Scholes formula, as discussed in Subsection 3.4.1. Subsections 3.4.2 and 3.4.3 introduce some further simple statistical hedging models.

3.4.1 Black-Scholes benchmark

Hedging via the BS-Delta is a standard benchmark. That is, for each option and for each date the corresponding implied volatility is used to obtain the hedge in (3.3), namely the partial derivative of the Black-Scholes option price with respect to the price of the underlying. Black-Scholes performs best if implied volatility is plugged in. In the literature, other volatilities, such as historical volatility estimates or GARCH predicted volatilities have been used. We refer to Ruf and Wang [2020a] for an overview.

Since here we hedge only discretely, using the BS-Delta leads to an error even if the data are simulated from the Black-Scholes model. The performance of discrete-time hedging has been extensively studied; some pointers to the literature include Boyle and Emanuel [1980], Bertsimas et al. [2000], and Tankov and Voltchkova [2009], who provide an asymptotic analysis of hedging errors.

3.4.2 Delta hedging other sensitivities

The leverage effect, first discussed in Black [1976], describes the negative correlation of observed returns and their volatilities in equity markets. This effect has been confirmed in many follow-up studies which also consider implied volatilities. For example, Cont and Da Fonseca [2002] claim that the leverage effect is due to the general level of the implied volatility surface and not due to relative movements, that is, changes in the shape of the implied volatility surface. The non-zero correlation of returns and the implied option volatilities indicates that the BS-Delta can usually be outperformed by some relatively simple adjustments.

In this spirit, Vähämaa [2004] and Crépey [2004] use the observed smile in option implied volatilities to improve on the hedging performance of the BS-Delta. These ideas are developed further in several papers; see for example, Alexander et al. [2012].

The central idea is to note that a first-order Taylor series expansion of option prices yields

$$\mathrm{d}C \approx \delta_{\mathrm{BS}}\,\mathrm{d}S + \mathcal{V}_{\mathrm{BS}}\,\mathrm{d}\sigma_{\mathrm{impl}} = \delta_{\mathrm{BS}}\,\mathrm{d}S + \mathcal{V}_{\mathrm{BS}}\,\frac{\mathrm{d}\sigma_{\mathrm{impl}}}{\mathrm{d}S}\,\mathrm{d}S + \mathcal{V}_{\mathrm{BS}}\,\mathrm{d}S^{\perp},$$

where $S^{\perp}$ is orthogonal to S. In words, the change in the option price is approximately the BS-Delta times the change in the underlying's price plus Vega times the change in the implied volatility. The second term can be written in terms of changes in the underlying's price and changes in the implied volatility that are uncorrelated with the changes in the underlying's price. These observations lead us to consider a statistical model of the form:

$$\delta = a\,\delta_{\mathrm{BS}} + b\,\mathcal{V}_{\mathrm{BS}}.$$

This statistical model replaces the BS-Delta by a multiple a of it plus a multiple b of Vega $\mathcal{V}_{\mathrm{BS}}$. Here, a and b are estimated in the in-sample set, separately for puts and calls. More precisely, estimating a and b is equivalent to running a linear regression with two independent variables and no intercept on the in-sample set. Indeed, we minimise the expression in (3.7), where each summand can be written as the square of

$$a\,(\delta_{\mathrm{BS},t,j}\,x_t) + b\,(\mathcal{V}_{\mathrm{BS},t,j}\,x_t) - y_{t,j},$$

where $x_t = 100(S_{t+1}/S_t - (1 + r_{\mathrm{onr}}\Delta t))$ and $y_{t,j} = 100/S_t(C_{t+1,j} - (1 + r_{\mathrm{onr}}\Delta t)C_{t,j})$.

Next, a Taylor series expansion of the BS-Delta yields

$$\mathrm{d}\delta \approx \Gamma_{\mathrm{BS}}\,\mathrm{d}S + \mathrm{Va}_{\mathrm{BS}}\,\mathrm{d}\sigma_{\mathrm{impl}}.$$

Here, Γ_{BS} denotes *Gamma*, namely the sensitivity of the BS-Delta to changes in the underlying's price; $\mathrm{Va}_{\mathrm{BS}}$ denotes *Vanna*, namely the sensitivity of the BS-Delta to changes in the implied volatility.

Combining these two expansions we obtain the linear regression model

$$\delta_{\mathrm{LR}} = a\,\delta_{\mathrm{BS}} + b\,\mathcal{V}_{\mathrm{BS}} + c\,\mathrm{Va}_{\mathrm{BS}} + d\,\Gamma_{\mathrm{BS}}. \tag{3.8}$$

Again, a, b, c, d are estimated for puts and calls separately on each in-sample set. We also consider nested models; in this case, we force either a to be one or one (or more) of the other coefficients to be zero and estimate the remaining coefficients. The Vega and Gamma sensitivities are large for options when the strike is close to the underlying's current price. Thus, including these sensitivities allow the statistical model to make adjustments to the hedging ratio depending on whether an option is at-the-money or out-of-the money. Using both two sensitivities helps, moreover, to make additional adjustments depending on the option's time-to-maturity. Finally, Vanna for an out-of-the money option is largest when the option is somehow out-of-the-money but not too much. This allows the model to make the corresponding additional adjustments. We have also experimented with an additional intercept term in (3.8). Including it does not change the conclusions below; we hence only report the results without this additional term.

One can find several arguments why the Gamma needs to appear in hedging. For instance, Bakshi et al. [1997] show that changing the rebalancing frequency affects the BS-Delta dramatically but not the stochastic volatility models, and after adding the second hedging instrument to the BS-Delta to make it 'delta-plus-vega-neutral', their is little difference between models in the longer hedging period. They then argue that adding an extra hedging instrument reduces the Gamma exposure that matters in long period. Björk [2004] advocates the "Delta-Gamma hedging" to reduce the impact of transaction costs, although we do not consider transaction costs here; see also Kallsen and Muhle-Karbe [2015]. In additon, Alexander and Nogueira [2007] state that the Delta and Gamma from minimum variance hedging account for the total effect of a change in the underlying on option prices, and this hedging is applied to the S&P 500 data, in which Vanna appears. A more careful treatment instead of the Taylor expansion may be needed to fully justify the incorporation the Gamma and Vanna into the regression features, and this remains as future research.

Furthermore, we include below the proposed hedging ratio of Hull and White [2017], given by

$$\delta_{\mathrm{HW}} = \delta_{\mathrm{BS}} + \frac{\mathcal{V}_{\mathrm{BS}}}{\sqrt{\tau} S}(a + b\delta_{\mathrm{BS}} + c\delta_{\mathrm{BS}}^2). \tag{3.9}$$

Here, τ is the time-to-maturity and a, b, c are again estimated for puts and calls separately on each in-sample set. Hull and White [2017] obtain this model from a careful analysis of S&P 500 options and observe its excellent hedging performance on options written on the S&P 500 and other indices. We furthermore include a 'Relaxed Hull-White' model, where the coefficient in front of δ is not restricted to one.

The models in (3.8) and (3.9) should be considered 'statistical' in contrast to 'model-driven' as the hedging ratio is derived purely from statistical considerations instead of being derived from stochastic models. In the language of Carr and Wu [2020], these models are 'local' and 'decentralised,' as only one period is considered instead of the option's whole time horizon, and as each option contract is treated separately instead of finding an overall consistent valuation model. To the best of our knowledge, the model in (3.8) not been suggested in the literature before, despite its simplicity. (In the context credit risk, Cont and Kan [2011] also provide a careful study of regression-based hedging. While here the hedging ratio is regressed on option sensitivities, they regress changes in the option price on changes in the underlying.)

3.4.3 Possible other benchmarks

One could consider hedging ratios derived from parametric models such as stochastic volatility models. Bakshi et al. [1997] observe that such models outperform the BS-Delta in the case of hedging out-of-the money options, but not necessarily in-the-money options. Vähämaa [2004] provides additional references that test the hedging performance of stochastic volatility models and concludes with the observation that "such models do not necessarily provide better hedging performance." Hull and White [2017] note that the hedging ratio δ_{HW} of (3.9) leads to a better performance than stochastic volatility models.

We initially also investigated the following two (semi-)linear benchmarks:

$$\overline{\delta}_1 = aM + b\sigma_{\mathrm{impl}}\sqrt{\tau} + c; \qquad \overline{\delta}_2 = \mathbf{N}\left(aM + b\sigma_{\mathrm{impl}}\sqrt{\tau} + c\right),$$

where M denotes moneyness, $\sigma_{\text{impl}}\sqrt{\tau}$ time-proportional implied volatility, and N the cumulative normal distribution function. Here, the parameters a, b, c were estimated again in each in-sample set. It turns out that these two linear regressions perform far worse than the BS-Delta δ_{BS}; hence we will not present results on these two benchmarks. The underperformance of these two linear regressions also shows that the performance of the ANN is not entirely due to the hand-crafted features.

3.5 Results

We now present the results on the performance of the various statistical hedging models in terms of MSHE reduction. As a quick summary, the hedging ratios of the ANNs do not outperform the linear regression models. On the S&P 500 dataset, the Hull-White and Delta-Vega-Vanna regressions tend to perform the best, with Hull-White better on the one-day hedging period, and the Delta-Vega-Vanna regression better on the two-day period. On the Euro Stoxx 50 dataset, the Delta-Vega-Gamma-Vanna regression tends to perform the best. However, the differences between these linear regressions with three or four coefficents are neither statistically nor economically significant, as we shall discuss.

In the next four subsections we discuss each of the datasets. We start with the real datasets (Subsections 3.5.1 and 3.5.2) and and then briefly summarise the results on the simulated data (Subsections 3.5.3 and 3.5.4). In Subsection 3.5.5, we conclude with some general observations on these experiments.

Recall from Subsection 3.2.4 that each data sample is normalised so that the underlying's price S_0 at time 0 is 100. This allows to compare the absolute hedging errors across different datasets. Recall also that we only consider out-of-the (and at-the)-money puts and calls.

3.5.1 S&P 500 end-of-day midprices

Table 3.3 gives an overview of the MSHEs across different hedging periods. The first two rows give the MSHEs for the zero hedge and the BS-Delta. The remaining rows give the relative improvement over the BS-Delta, i.e.,

$$\frac{\text{MSHE}(\delta_*) - \text{MSHE}(\delta_{\text{BS}})}{\text{MSHE}(\delta_{\text{BS}})}, \tag{3.10}$$

All competing methods outperform the BS-Delta. Among them, the Delta-Vega-Vanna and (relaxed) Hull-White regressions perform the best, with Hull-White doing slightly better on the one-day hedging period while Delta-Vega-Vanna performing better on two-day hedging period. Indeed, Hull and White [2017] study the same dataset to create the Hull-White regression, so it is surprising how close the other regressions get. The major improvement in the regressions (apart from the Hull-White regression) comes from allowing the coefficient in front of Delta to be estimated, rather than equal to one. Regressions with the second-order sensitivities on its own (i.e., with the Delta coefficient fixed to one as in Hull-White) are not performing as well, and we have omitted them from Table 3.3. The two ANNs perform similarly to the regressions in case of the one-day period, but underperform for the two-day period.

		1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	4.01	4.78	4.54	8.31	9.73	9.29
	BS-Delta	0.687	0.655	0.665	1.58	1.54	1.55
Regressions	Delta-only	-21.3	-14.8	-16.9	-16.3	-12.8	-13.9
	Vega-only	-13.7	-11.7	-12.3	-10.4	-10.1	-10.2
	Gamma-only	-15.5	-10.1	-11.8	-14.5	-11.2	-12.2
	Vanna-only	-12.4	-12.6	-12.5	-10.6	-13.0	-12.2
	Delta-Gamma	-21.6	-14.8	-17.0	-17.1	-13.1	-14.4
	Delta-Vega	-21.4	-14.9	-17.0	-16.4	-12.8	-13.9
	Delta-Vanna	-22.6	**-16.6**	**-18.5**	**-17.7**	**-15.4**	**-16.1**
	Delta-Vega-Gamma	-21.5	-14.8	-17.0	-16.8	-13.5	-14.5
	Delta-Vega-Vanna	**-23.0**	**-16.6**	**-18.7**	**-18.1**	**-15.4**	**-16.2**
	Delta-Gamma-Vanna	-22.6	**-16.6**	**-18.5**	**-17.7**	**-15.2**	**-16.0**
	Delta-Vega-Gamma-Vanna	**-22.9**	**-16.4**	**-18.5**	-17.4	**-14.9**	**-15.7**
	Hull-White	**-23.1**	**-16.9**	**-18.9**	**-17.8**	**-14.5**	**-15.5**
	Relaxed Hull-White	**-23.2**	**-16.9**	**-18.9**	**-18.3**	**-14.6**	**-15.8**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	-22.3	-15.6	-17.7	-17.1	-10.9	-12.8
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	**-23.4**	**-16.9**	**-18.9**	**-18.6**	-12.9	-14.7
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-21.9	-14.4	-16.8	-12.5	-12.9	-12.8

Table 3.3 Performance of the linear regressions and ANNs on the S&P 500 dataset. The hedging periods Δt are here either one day or two days. The columns 'Both' are the weighted average of the 'Puts' and 'Calls' columns. The row 'Zero hedge' corresponds to the MSHE when $\delta = 0$ is chosen; i.e., the mean squared changes in the option prices. The values in the top two rows are multiplied by 100 to improve readability. The regression and ANN rows correspond to the various statistical models including HedgeNet with two different feature sets. For these two sets of rows, the numbers are reported as relative improvements in MSHE over using the BS-Delta, i.e., (3.10). Numbers in bold represent the largest outperformance (in each column the best one is chosen along with the ones that are within 1% of the best).

Table 3.3 indicates that it is easier to outperform the BS-Delta when hedging out-of-the money calls than out-of-the money puts. However, note that the BS-Delta itself reduces the MSHE more for puts than for calls when using the zero hedge as baseline. To see this, let us have a closer look at the one-day period. For calls, hedging with the BS-Delta reduces the MSHE by $1 - 0.687/4.01 \approx 83\%$, while for puts, it reduces the MSHE by $1 - 0.655/4.78 \approx 88\%$. Using the Hull-White Delta reduces the MSHE for calls only by $1 - (1 - 0.231) \times 0.687/4.01 \approx 87\%$, but for puts by $1 - (1 - 0.169) \times 0.655/4.78 \approx 89\%$. Hence, the relative outperformance of the linear regressions and ANNs over the BS-Delta is higher exactly when the BS-Delta has a worse performance. These observations are not due to the asymmetric choice of moneyness (recall that we only consider out-of-the money options with moneyness $M = S_0/K$ between 0.8 and 1 for calls and between 1 and 1.5 for puts). Indeed the same results as outlined in this paragraph hold true when we allow moneyness to be between 0.6 and 1 for calls and restrict it to be between 1 and 1.2 for puts.[4]

Recall from Section 3.2 that the S&P 500 dataset is been split in rolling windows, each time shifted by 180 days. This yields 14 out-of-sample sets. The samples in each out-of-sample set are evaluated with the model parameters estimated on its corresponding in-sample set. Figure 3.9 compares the MSHEs of different statistical models by time window. Consistent with Table 3.3, the blue dots corresponding to the BS-Delta are usually the largest. However sometimes, for example in the first time window, the competing models underperform relative to BS-Delta. Both Table 3.3 and Figure 3.9 show that for two-day hedging period, the MSHEs are about twice those for the one-day period. The only exceptions are the 7th and the 13th time window, when the errors are about 4 times and 3 times larger in the two-day period.

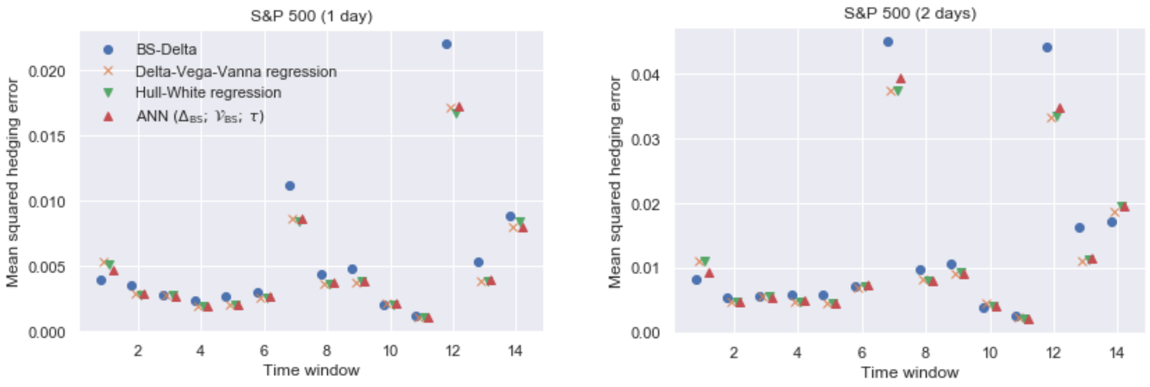


Fig. 3.9 MSHEs of four different statistical models for the hedging ratio across all 14 time windows in the S&P 500 dataset, for the one-day (left) and two-day (right) hedging period. Note that in the first time window the models lead to a higher MSHE than the BS-Delta. We try to give an explanation for this effect in Subsection 3.6.1.

[4] When one does not remove samples with very small or very large moneyness in the cleaning process then the median moneyness in the S&P 500 dataset for out-of-the-money and at-the money calls (puts) is 0.97 (1.09). In this case, 95% of the out-of-the-money and at-the money calls (puts) satisfy $0.89 \leq M$ ($M \leq 1.51$).

Figure 3.10 provides the coefficients (plus their standard errors) for the Delta-Vega-Vanna regression in the one-day period setting.[5] The intervals are getting smaller for later time windows due to the fact that later time windows contain more samples as illustrated in Section 3.2.5. Especially the Vanna coefficients for calls are very stable across time windows. Figure 3.9 shows that both the 7th and the 12th time window, whose out-of-sample data are the second half of 2015 and the first half of 2018, respectively, lead to an overall large MSHE. The corresponding samples are then part of the in-sample set for the following periods. And indeed, Figure 3.10 indicates a jump in some of the coefficients in the 8th and 13th time window.

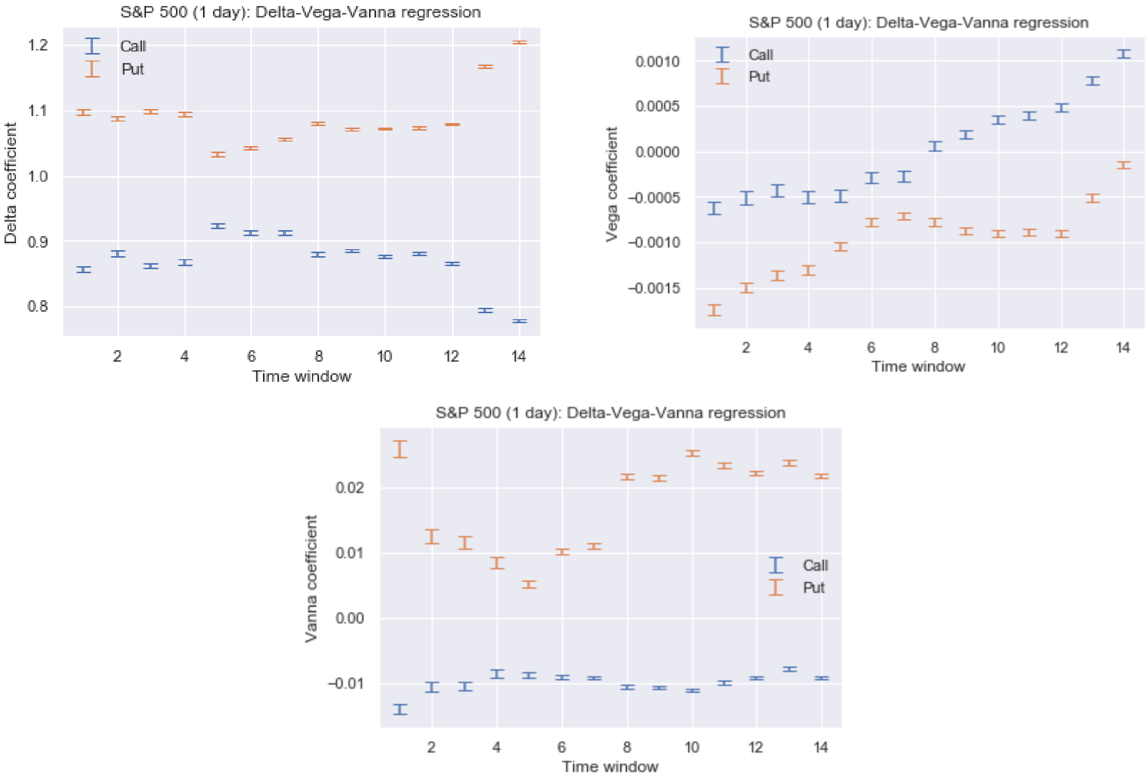


Fig. 3.10 The coefficients in the Delta-Vega-Vanna regression for each of the 14 time windows in the S&P 500 dataset. The top and bottom of each line segment are the point estimate plus/minus two standard errors. These numbers correspond to the one-day hedging period.

The Delta coefficients of calls being smaller than one implies that hedging a short position on a call, one would usually buy less of the underlying than implied by the BS-Delta. On the other hand, for hedging a short position on a put, one needs to short more of the underlying. This phenomenon is consistent with the leverage effect, discussed in Subsection 3.4.2. Note

[5] The coefficient plots for the two-day hedging periods (not displayed here) look very similar; in particular the Vanna coefficients for calls are again stable. However the Vanna coefficients for puts and the Vega coefficients for calls and puts are slightly more fluctuating.

that Vanna is positive (negative) for out-of-the money calls (puts). Hence the Vanna term in the regression further contributes to holding an even smaller number of the underlying than only implied by the Delta term. Since Vanna is largest in absolute value for slightly out-of-the money options, this correction term is largest for such options. The Vega coefficients are negative for puts and most time windows also for calls, adding yet a third correction, most effective for long-dated at-the-money options.

Additional diagnostics are available in Sections 4.2 and 4.3.1.

We run three extra experiments to see whether the above conclusions depend on the chosen setup.

1. In the first modified experiment we remove all options that have a time-to-maturity of 14 calendar days or less from both the in-sample and out-of-sample sets. This yields an additional relative improvement of about 2% in the one-day experiment and about 3% in the two-day experiment for all methods presented in Table 3.3. We omit presenting the precise numbers here.

2. In the second modified experiment we abstain from splitting the dataset in 14 time windows. Instead of 14 experiments we hence only have one, but with a much larger number of samples. We keep the ratio 4:1:1, now across the whole dataset, leading to an in-sample set of length 2850 (2280 + 570) days and a test set of length 570 days (instead of 14 test in-sample sets of length 900 (720 + 180) days and an out-of-sample set of length 180 days; see Subsection 3.2.4). We omit the detailed results of this experiment. The regression models and ANNs improve their relative performance by about 3% to 4% when using only one time window instead of 14 time windows. Again the ANNs do not outperform the linear regression models.

3. We put the options in two roughly equally sized buckets: at-the-money/close-to-the money options and out-of-the money options. We run the linear regressions and (appropriately tuned) ANNs on both buckets separately. The bucketing tends to help the linear regressions using a single sensitivity slightly, does not change the linear regressions using several sensitivities, and leads to a worse performance of the ANNs.

More details on these three experiments can be found in Section 4.5. Section 3.6 yields a fourth experiment to check whether the cleaning process of the raw data introduced any information leakage.

3.5.2 Euro Stoxx 50 tick data

Table 3.4 shows the performance of all competing methods on the Euro Stoxx 50 dataset. Again we can conclude that the ANNs in general do not outperform the linear regressions.

Now the Delta-Vega-Gamma-Vanna regression performs best, closely followed by the linear regressions using three sensitivities, which perform better than the Hull-White regressions.

		1 hour			1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	0.431	1.02	0.756	4.28	10.2	7.47	8.20	24.26	17.4
	BS-Delta	0.109	0.214	0.167	1.19	1.99	1.62	2.97	4.20	3.67
Regressions	Delta-only	-18.9	-11.4	-13.6	-21.7	-12.2	-15.4	**-36.0**	-10.7	-19.5
	Vega-only	**-25.3**	-13.8	**-17.2**	**-23.4**	-16.0	**-18.5**	-35.2	-16.2	-22.8
	Gamma-only	-0.62	-1.29	-1.10	-15.7	-4.64	-8.37	-32.7	-4.62	-14.4
	Vanna-only	-16.1	-5.18	-8.35	-17.1	-12.6	-14.1	-26.9	-16.9	-20.4
	Delta-Gamma	-18.0	-14.5	-15.5	-20.5	-12.7	-15.4	-33.5	-6.89	-16.1
	Delta-Vega	-23.9	-13.7	-16.7	-22.7	-15.4	**-17.9**	**-36.9**	-15.3	-22.8
	Delta-Vanna	-20.8	-11.4	-14.1	-19.2	-14.8	-16.3	-34.9	-17.2	**-23.4**
	Delta-Vega-Gamma	-21.6	**-15.2**	**-17.0**	-20.7	-15.4	-17.2	-34.4	-13.5	-20.8
	Delta-Vega-Vanna	-23.6	-13.7	-16.6	-19.6	-16.7	**-17.7**	-35.1	-18.5	**-24.2**
	Delta-Gamma-Vanna	-23.1	**-15.5**	**-17.7**	-20.2	**-17.9**	**-18.7**	-33.8	**-17.7**	**-23.3**
	Delta-Vega-Gamma-Vanna	-23.3	**-15.6**	**-17.8**	-20.1	**-18.0**	**-18.7**	-34.4	**-18.2**	**-23.9**
	Hull-White	-20.0	-12.5	-14.7	-20.7	-14.3	-16.4	**-36.1**	-13.3	-21.2
	Hull-White-relaxed	-20.3	-12.6	-14.8	-20.6	-14.2	-16.4	**-36.1**	-12.7	-20.8
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	-17.6	**-15.7**	-16.3	-8.96	-3.3	-5.21	-27.4	11.3	-2.12
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-16.1	-6.08	-9.01	-19.0	-6.83	-10.9	-25.6	-3.6	-11.2
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	**-25.0**	-13.3	-16.7	-18.8	-10.1	-13.1	-29.2	-6.96	-14.7

Table 3.4 Performance of the linear regressions and ANNs on the Euro Stoxx 50 data set, when the in-sample and out-of-sample are split into one time window. We refer to the caption of Table 3.3 for an explanation.

Just using the BS-Delta reduces the overall MSHE by about 78%-79%. This percentage is very stable across the three different hedging periods and smaller than in the S&P 500 dataset. Again, the BS-Delta reduces the MSHE more for puts than for calls, and the relative outperformance of the regression models is larger when the BS-Delta is worse.

We list the coefficients of the Delta-Vega-Gamma-Vanna regression (plus their standard errors) in Table 3.5. Again, the Delta coefficients for calls (puts) are smaller (larger) than one, consistent with the leverage effect. Additional diagnostics are available in Online Appendices 4.2 and 4.3.2.

Similarly to the S&P 500 dataset we run two additional experiments.

1. In the first one, we only consider options with a time-to-maturity of 14 calendar days or more. This yields an additional relative improvement of about 4% to 8%, in comparison with Table 3.4. The improvement tends to be larger for the regressions using a smaller

	1 hour		1 day		2 days	
	Calls	Puts	Calls	Puts	Calls	Puts
Delta	0.944 ± 0.002	1.134 ± 0.002	0.755 ± 0.003	1.056 ± 0.003	0.821 ± 0.004	1.021 ± 0.003
Vega	-0.002 ± 0.000	0.000 ± 0.000	-0.001 ± 0.000	-0.002 ± 0.000	-0.001 ± 0.000	-0.002 ± 0.000
Gamma	-0.021 ± 0.004	0.213 ± 0.003	0.226 ± 0.008	0.393 ± 0.006	0.109 ± 0.010	0.417 ± 0.008
Vanna	-0.010 ± 0.000	0.014 ± 0.000	0.004 ± 0.000	0.029 ± 0.000	0.003 ± 0.000	0.025 ± 0.000

Table 3.5 Coefficients of Delta-Vega-Gamma-Vanna regression for each sensitivity on the Euro Stoxx 50 dataset. Coefficients are presented for calls and puts separately. Each cell shows the coefficient and its standard error.

number of sensitivities. In particular, the Delta-Vega-Vanna regression now seems to dominate the Delta-Vega-Gamma-Vanna regression, especially for the two-day hedging period. We again omit the precise numbers here as the overall conclusions do not change.

2. We again put the options in two roughly equally sized buckets: at-the-money/close-to-the money options and out-of-the money options. Running the statistical models on both buckets separately seems to help slightly the linear regressions with only one sensitivity but does not change or worsens the performance of the other linear regressions and ANNs.

We also refer to Section 3.6 for another experiment to check how the cleaning of the data might influence the results of this subsection.

3.5.3 Simulated data from Black-Scholes

As reported in Table 3.6, in the one-day hedging period, the BS-Delta performs best (with the exception of the Vanna-only and Vega-only regressions). For the two-day hedging period, all regressions outperform the BS-Delta. Relative to the BS-Delta the regressions are about 2% to 3% better. At first glance, this seems surprising since the BS-Delta should be close to optimal for data generated from the Black-Scholes model. Indeed, in both hedging periods, using the BS-Delta instead of not hedging at all reduces the MSHE by about 99%.

What is happening? Recall that we do not hedge continuously but only once in each hedging period. During the hedging period, the underlying's price changes, and thus, the BS-Delta chosen at the beginning of the hedging period is not optimal at other times during the hedging period. Gamma measures how fast the option's Delta changes as the underlying moves. Since the underlying's price path has been simulated with an annualised drift rate of 10% (see Subsection 3.2.1), in average the option's Delta increases over the hedging period.[6] The linear regressions are able to capture this effect. For example, in the Delta-only regression, the Delta coefficient is larger than one for out-of-the money calls and smaller than one for out-of-the money puts (in which case the BS-Delta is negative). This is in line with the observation that the option's Delta increases over the hedging period in average.

For the one-day hedging period this drift effect is not strong enough for the linear regression models to outperform; they tend to slightly overfit to the in-sample data. For the two-day hedging period, however, this drift effect is captured by the linear regressions, as can be seen in Table 3.6. The ANNs are not able to capture this effect, due to overfitting.

We have run another experiment, where we set the drift rate of the underlying's price path to zero and leave all others parameters the same. In this case, the linear regressions underperform (overperform) relative to the BS-Delta by about 0.5% for the one-day (two-day) hedging period. Again, ANNs have the lowest performance among all considered models.

		1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	27.0	12.3	16.0	54.9	23.4	31.2
	BS-Delta	0.164	**0.094**	0.111	0.719	0.341	0.437
Regressions	Delta-only	**-1.38**	1.05	0.11	-4.74	-0.82	-2.12
	Gamma-only	**-1.25**	0.97	0.12	**-6.27**	**-1.39**	**-2.76**
	Vega-only	**-1.22**	0.81	**-0.02**	-3.85	-0.56	-1.68
	Vanna-only	**-1.64**	0.32	**-0.46**	-5.61	-0.60	-1.99
	Delta-Gamma	**-1.38**	0.96	0.07	**-6.26**	**-1.42**	**-2.79**
	Delta-Vega	-1.09	1.1	0.35	-4.97	-0.89	**-2.27**
	Delta-Vanna	**-1.30**	1.01	0.11	**-6.03**	-0.83	**-2.47**
	Delta-Vega-Gamma	**-1.16**	0.99	0.21	**-6.37**	**-1.28**	**-2.78**
	Delta-Vega-Vanna	**-1.30**	1.08	0.24	**-6.49**	**-1.06**	**-2.68**
	Delta-Gamma-Vanna	-0.85	0.99	0.31	**-6.6**	**-1.2**	**-2.85**
	Delta-Vega-Gamma-Vanna	-1.03	0.98	0.26	**-6.62**	**-1.2**	**-2.86**
	Hull-White	**-1.44**	1.02	0.07	**-6.26**	-0.77	**-2.46**
	Relaxed Hull-White	**-1.43**	1.02	0.07	**-6.24**	-0.77	**-2.45**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	8.9	2.55	5.65	-3.21	0.55	0.08
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	2.11	2.81	2.16	-5.37	5.45	2.63
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-0.16	1.07	1.37	-5.83	2.44	-0.21

Table 3.6 Performance of the linear regressions and ANNs on the Black-Scholes simulated dataset. See the caption of Table 3.3 for further explanations.

We conclude this subsection with a remark. The experiments above are done with a realistic amount of samples in the in-sample set, namely obtained by following the CBOE

[6] Even with the drift being zero, such an effect would exist due to the convexity of option prices in the underlying.

rules on generating options as outlined in the previous subsection. If the in-sample set was to be augmented by additional data then eventually the overfitting of the statistical models in the one-day hedging period would disappear. Moreover, the more complex models will then outperform the simpler ones in the horse race of Table 3.6.

3.5.4 Simulated data from Heston

For the Heston dataset, we report the numbers in Table 3.7. Again the ANNs do not lead to a better performance than the regression models. Using the BS-Delta reduces the variance by more than 97% (96%) for both calls and puts, for the one-day (two-day) hedging period. This is a larger improvement than for the real datasets. Note that we have roughly 3 times more put samples than call samples in the in-sample test as Section 3.2.5 explains. The coefficients for the Delta-only regression for the one-day (two-day) hedging period are for calls 0.97 (0.99) and for puts 1.03 (1.03), all with standard deviation ± 0.001 or less.

Note the consistently worse relative performance for hedging calls than for hedging puts in the two-day hedging period in Table 3.7. The BS-Delta itself already performs better for calls than for puts; hence it is more difficult to improve on it in the case of calls than for puts. Indeed, there are two effects in play. They cancel each other for calls but reinforce themselves for puts. (a) For out-of-the money puts and calls convexity together with time-discrete hedging suggests a larger hedging ratio (in absolute terms). (b) The leverage effect suggests a lower hedging ratio for calls but a larger hedging ratio (in absolute terms) for puts. Since these two effects for calls go in opposite directions, but not for puts, the BS-Delta performs better for calls than for puts.

The same remark as at the end of the previous subsection also applies here. In additional experiments, we have augmented the data with additionally simulated samples. Eventually, the more complex models always outperform the simpler ones. The results as displayed in Table 3.7 show that with a limited amount of data sometimes simpler models outperform more complex ones.

As a sanity check, we consider two model-implied hedging strategies on the one-day period. The first one relies on δ_{HS}, the sensitivity of the option price with respect to the underlying price, computed under the Heston model with the correct parameters. This sensitivity is then adjusted by a multiple of ν_{HS}, the sensitivity of the option price with respect to the underlying variance Y_0. More precisely, the hedging strategy is given by

$$\delta_{\mathrm{HS}} + \nu_{\mathrm{HS}} \frac{\rho \sigma_Y}{S_0};$$

see, for example, Alexander and Nogueira [2007] for a derivation via a Taylor series expansion. Using this strategy leads to a reduction of 6.18% (calls only: 4.99%; puts only: 6.73%) of the

			1 day			2 days	
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	21.7	14.7	16.5	45.6	32.0	34.3
	BS-Delta	0.637	0.505	0.526	1.61	1.36	1.35
Regressions	Delta-only	**-3.73**	-4.80	-4.50	**-1.09**	-4.86	**-2.59**
	Gamma-only	-3.44	-4.55	-4.39	-0.74	-4.97	-2.33
	Vega-only	-3.20	-3.77	-3.77	**-1.21**	-3.97	-2.34
	Vanna-only	-3.38	-2.97	-3.62	**-1.46**	-3.54	-2.30
	Delta-Gamma	**-3.98**	-5.02	**-4.82**	-0.92	**-5.04**	-2.47
	Delta-Vega	-3.51	-4.84	-4.39	-0.97	-3.89	-2.03
	Delta-Vanna	**-4.04**	-5.14	**-4.92**	**-1.53**	**-5.42**	**-3.03**
	Delta-Vega-Gamma	**-3.64**	-4.97	-4.67	**-1.06**	-4.37	-2.25
	Delta-Vega-Vanna	**-4.07**	**-5.36**	**-5.03**	**-1.23**	-4.74	-2.46
	Delta-Gamma-Vanna	**-3.97**	-4.92	**-4.77**	**-1.43**	-4.62	-2.56
	Delta-Vega-Gamma-Vanna	**-4.13**	-5.22	**-4.96**	**-1.26**	-4.62	-2.43
	Hull-White	**-4.12**	-5.02	**-4.92**	**-1.23**	**-5.15**	**-2.75**
	Relaxed Hull-White	**-4.11**	**-5.02**	**-4.92**	**-1.21**	-5.16	**-2.74**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	4.49	**-5.49**	1.36	6.04	-5.01	2.96
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-3.01	-5.08	**-4.13**	0.74	-3.46	0.19
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-2.46	**-5.68**	-3.77	-0.27	-2.05	0.01

Table 3.7 Performance of the linear regressions and ANNs on the Heston dataset. See the caption of Table 3.3 for further explanations.

MSHE relative to using the BS-Delta. We note that the Delta-Vega-Vanna regression, which does not use any model information and leads to a reduction of 5.03%, performs almost as well as this model-specific hedging strategy.

The second model-implied hedging strategy, suggested by Bakshi et al. [1997], is often called ‘Delta-neutral strategy.’ It differs from all other hedging strategies used in this chapter, in so far that it uses a second hedging instrument; here an at-the-money (ATM) call with maturity equal to one month. The number η of at-the-money options held is chosen to satisfy

$$\eta\nu_{\text{HS}}^{\text{ATM}} - \nu_{\text{HS}} = 0,$$

i.e., to cancel out the ‘Vega’ risk in the hedged portfolio. The number of stocks held is then set equal to $\delta_{\text{HS}} - \eta\delta_{\text{HS}}^{\text{ATM}}$. Relative to using the BS-Delta, this hedging strategy leads to a reduction of the MSHE by 63.8% (calls only: 62.6%; puts only: 69.2%). None of the hedging strategies discussed above gets close to this one.

It seems that the BS-Delta works much on the Heston data than on the S&P 500, thus there are some features in the historical data that are not captured by the Heston model. [Ghysels et al., 1996, Section 2.2] list nine stylized facts of market data that a good model of volatility needs to consider. They are :

- fat tails,
- volatility clustering,
- leverage effects,
- information arrivals,
- long memory and persistence,
- volatility co-movements,
- implied volatility correlations,
- term structure of implied volatilities,
- volatility smiles.

The Heston model is used to simulate the leverage effect that is important in explaining the advantage of regression methods. Obviously it may fail in capturing other features. For example, the volatility predicted by the Heston is too flat compared to reality at the long end of maturity. The research trying to improve this are ample. Bakshi et al. [1997] observe that adding jump to the stochastic volatility model does not help hedging, since they expect the chance of jumps is relatively small in a one- or five-day internal, while it matters more for the pricing in long-term simulation. A later research by Huang and Wu [2004] suggest that a high frequency jump component based on time-changed Lévy process is needed to extend the model of Bakshi et al. [1997] for pricing. However, they do not discuss the hedging performance. More recently, Christoffersen et al. [2010] suggest that volatility should be specified linear rather than square root as in Heston by empirical tests. Giglio and Kelly [2018] suggest that different treatments are needed for short- and long-maturity options. It seems that the deviations of stochastic models from the markets in terms of hedging are much less understood than for pricing.

3.5.5 Guidelines on statistical hedging

We now develop some guidelines based on the results of the last two subsections.

In none of the datasets do ANNs outperform the linear regression models. We conclude that the option sensitivities suffice to capture the nonlinearities in the data that are relevant for the hedging task. Additional drawbacks of ANNs are their computational demands and the necessary effort to tune their hyperparameters (see Section 4.1).

Next, we have a closer look at the MSHEs of the linear regression models. To this end, in the spirit of (3.7), let us define the time-t MSHE by

$$\mathrm{MSHE}_{\mathrm{t}}^{\delta} = \frac{1}{\mathrm{N}_{\mathrm{t}}} \sum_{\mathrm{j}}^{\mathrm{N}_{\mathrm{t}}} \left(100 \frac{\mathrm{V}_{\mathrm{t+1,j}}^{\delta}}{\mathrm{S}_{\mathrm{t}}} \right)^{2},$$

where N_t denotes the number of samples at time t. Here, t ranges over days in the test set and δ denotes one of the hedging methods. Hence $\mathrm{MSHE}_{\mathrm{t}}^{\delta}$ denotes the average of a cross-section of hedging errors, namely those corresponding to the options traded at some time t. Next, for each pair of hedging methods (e.g., the Delta-only and the Delta-Vega-Vanna regressions), we compute an approximate confidence interval for the difference of the MSHEs by adding and subtracting twice the standard error to the mean of the differenced time-t MSHEs. Due to their possible statistical dependence in time, these confidence intervals need to be interpreted with caution. They allow us to make the following observations.

- For both hedging periods in the S&P 500 dataset, the confidence intervals for time-t MSHEs of BS-Delta hedging paired with any of the statistical regressions (except for Gamma-only and Vanna-only regressions) do not contain zero, strongly suggesting that their relative outperformance is not due to noise only. The same observation also holds for the one-hour and two-day hedging periods in the Euro Stoxx 50 dataset. For the one-day hedging period in the Euro Stoxx 50 dataset the statistical methods reduce the BS-Delta hedging error by up to 18.7%, but the corresponding confidence intervals include zero. This gives an instance where the outperformance seems to be economically significant but fails to be statistically significant.

- There is statistical evidence for the underperformance of the Gamma-only and Vanna-only regressions. Pairing them with any of the linear regression models usually leads to confidence intervals that do not include zero. However, among any pairs of the remaining linear regression models the evidence is not clear cut. Sometimes the corresponding confidence intervals contain zero, sometimes they do not.

We recommend to choose one of the linear regression models, for example, the Delta-Vega-Vanna or the Delta-Vega-Gamma-Vanna regressions, which perform best in the above experiments. Let us also note that the choice between the two probably does not matter much from an economic perspective. Indeed, let us consider the one-day hedging period in Euro Stoxx 50, where the two regressions yield a relative reduction of 17.7% and 18.7% (see Table 3.4). If we now consider the Sharpe ratio of a delta-hedged option as in Subsection 3.2.6, these relative reductions increase the Sharpe ratio by a factor of $1/\sqrt{0.823} \approx 1.10$ and

$1/\sqrt{0.813} \approx 1.11$, respectively. While either one leads to an economically significant increase in Sharpe ratio, their relative difference seems to be very minor.

We conclude this section with a further observation. Motivated by the reported results we try another 'fixed' hedging strategy that does not require any historical data. All calls are hedged by $0.9 * \delta_{\mathrm{BS}}$ and puts are hedged by $1.1 * \delta_{\mathrm{BS}}$. We have not run other such 'fixed' hedging strategies (hence, we have not optimised this 10% relative correction term). Table 3.8 shows the relative performance of this 'fixed' strategy with respect to BS-Delta on the S&P 500 and Euro Stoxx 50 datasets. The out-of-sample tests are the same ones that were used for Tables 3.3 and 3.4. This simple strategy does very well but underperforms the linear regression models.

	1 hour			1 day			2 days		
	Calls	Puts	Both	Calls	Puts	Both	Calls	Puts	Both
S&P 500	-	-	-	-18.6	-13.1	-14.8	-15.0	-11.4	-12.6
Euro Stoxx 50	-15.4	-10.3	-11.8	-15.4	-12.7	-13.6	-23.7	-16.6	-19.0

Table 3.8 Performance of the 'fixed' hedging strategy on the S&P 500 and Euro Stoxx 50 datasets. In the 'fixed' hedges strategy, calls (puts) are hedged by $0.9 * \delta_{\mathrm{BS}}$ ($1.1 * \delta_{\mathrm{BS}}$). See the caption of Table 3.3 for further explanations.

3.6 Information leakage

Information leakage occurs when trained model parameters (such as in a linear regression or in an ANN) are unintentionally allowed to depend on certain information that would not be available when using the model in real time. Hence the backtesting and comparison of different statistical models, as in this work, requires extra care.

In this section we provide some examples for information leakage in the context of the hedging problem and discuss its implications. More precisely, in Subsection 3.6.1 we illustrate how important it is to keep the time series structure of the data in mind. In Subsection 3.6.2 we illustrate how the data cleaning process can introduce information leakage and we argue that it is very difficult to avoid any information leakage due to missing observations.

3.6.1 Potential information leakage for time series

In this chapter the one-period hedging problem is studied. Hence, when preparing the data as described in Subsection 3.2.4, the intrinsic time series structure of the data is not automatically preserved as each time series is broken up in many one-period samples.

As discussed in Subsection 3.2.4, here the data in each time window are separated chronologically into an in-sample and an out-of-sample set, as shown in the left panel of Figure 3.11. In each time window roughly the first 83% (=5/6) of days are assigned to the in-sample set (again chronologically split in a training and a validation set for ANNs) and the last 17% (=1/6) of days are assigned to the out-of-sample set.[7] [8]

Alternatively, we could have split the data randomly into an in-sample and out-of-sample dataset, as shown in the right panel of Figure 3.11. (This approach has been taken in several research papers; see Ruf and Wang [2020a] for a review.) In this approach, the in-sample and out-of-sample sets are also disjoint. However, we now argue that such an approach introduces significant information leakage. Indeed, as on each day several options are traded and hence we have several samples, the same day might show up both in in-sample and out-of-sample sets, with different options.

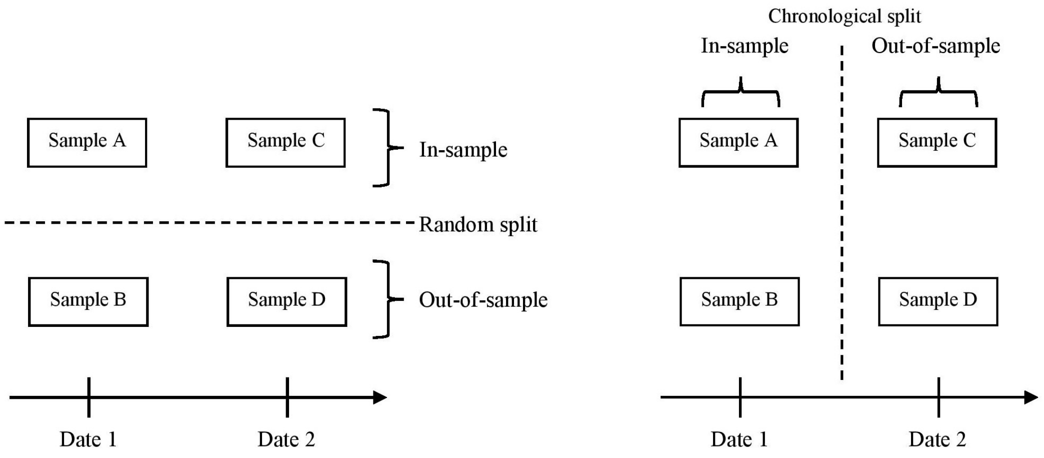


Fig. 3.11 Illustration of chronological (left) and random (right) split. Each day has two one-period samples. Sample A and B (as well as sample B and D) are from two different options traded on the same day. Sample A and C (as well as sample B and D) are from the same option traded in two different days. Chronological split implies that all samples belonging to an early day constitute the in-sample set and those belonging to a late day constitute the out-of-sample set. Random split implies that samples belonging to the same day could (but not always) appear in both in-sample and out-of-sample sets.

[7] Due to the growth of traded options (see Figure 3.3 in Appendix 3.2.5), this actually corresponds to about 23% of samples in each time window being in the out-of-sample set.

[8] For the two-day hedging period, we additionally make sure that the samples on the day separating the in-sample and out-of-sample sets are taken out. This avoids that the last day in the in-sample set overlaps with the first day in the out-of-sample set.

We illustrate with a series of experiments how such a wrong split in in-sample and out-of-sample sets may lead to wrong conclusions. We run these experiments both on the Black-Scholes simulated data and the S&P 500 data, both for the one-day hedging period. For each of these two datasets we simulate a 'fake VIX'; i.e., we simulate daily samples from an Ornstein-Uhlenbeck process[9] completely independent from either dataset. Clearly, adding this 'fake VIX' value as a feature should not help at all in reducing the MSHE, as the corresponding Ornstein-Uhlenbeck process is independently simulated.

The four experiments are the following.

1. The '*Baseline*' experiment corresponds to the standard setup of Section 3.2.4. The dataset is separated chronologically in in-sample and out-of-sample sets. We consider $\mathrm{ANN}(\Delta_{\mathrm{BS}};\ \mathcal{V}_{\mathrm{BS}};\ \tau)$ and the Delta-Vega-Vanna linear regression.

2. The '*VIX*' experiment takes the baseline setup, but adds the simulated 'fake VIX' variable as an additional feature to the linear regression and the ANN.

3. The '*Permute*' experiment is done as follows. We compute the number of training, validation, and test samples. Then within each time window we permute the samples by randomly reassigning training, validation, and test labels to them. We do this in such a way that the numbers of training, validation, and test samples do not change. For the linear regression, the permuted training and validation sets are merged to be the in-sample set. Then, $\mathrm{ANN}(\Delta_{\mathrm{BS}};\ \mathcal{V}_{\mathrm{BS}};\ \tau)$ and the Delta-Vega-Vanna regression are trained again on this permuted dataset. After each permutation, the Black-Scholes benchmark is recomputed since each permutation changes the constituents of the out-of-sample set.

4. The '*Permute + VIX*' experiment is executed exactly as the '*Permute*' experiment, but now with the 'fake VIX' variable as an additional feature.

The simulated and real data need slightly different treatments. Recall that the S&P 500 dataset is split into 14 time windows. We keep these 14 time windows, and run all four experiments for each of them. More precisely, for each time window, we run the third and fourth experiments five times as different permutations might lead to different results. For the Black-Scholes data, we run each experiment twenty times, on different out-of-sample sets but the same in-sample set, so that the 'Baseline' and 'VIX' experiments yield exactly the same trained ANN and regression coefficients.

Figures 3.12 and 3.13 summarise the results on the Black-Scholes and the S&P 500 datasets, respectively. The left panels show the the relative improvement over the BS-Delta,

[9] As parameters we use 1 for the rate of mean reversion, 25 for the volatility coefficient, 13 for the starting value, and 15 for the long-term mean.

as given in (3.10), for each of the four experiments, averaged over time windows and the permutation sets, respectively. The right panels in Figures 3.12 and 3.13 show these results broken down by permutation set (Black-Scholes data) or time window (S&P 500 data). The time windows for the S&P 500 data are chronologically ordered; the permutation sets for the Black-Scholes data are ordered by performance of the ANN in the baseline experiment. Each of the presented numbers in the right panel corresponds to the additional relative improvement over the BS-Delta due to the permutation and 'fake VIX' feature. For example, a value of -20% for the 'ANN (Permute + VIX)' setup means that the 'Permute + VIX' experiment adds an extra 20% to the relative improvement of the ANN in the 'Baseline' experiment.

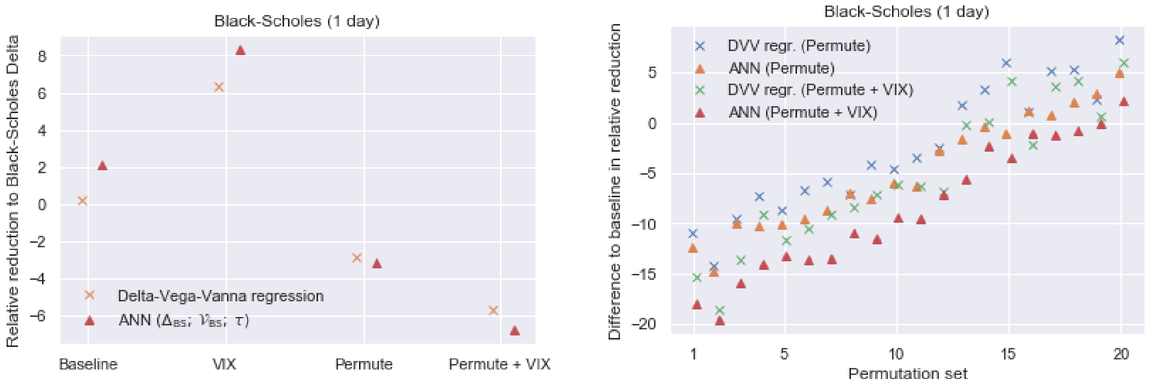


Fig. 3.12 Illustration of information leakage when failing to take into account the time series structure of a simulated dataset. The left panel displays the relative reduction in MSHE over using the BS-Delta for each of the four experiments described in the main text. In the 'Baseline' experiment, neither the ANN nor the linear regression improve the MSHE relative to the BS-Delta. Adding the 'fake VIX' feature reduces furthermore their performance since this feature is simulated independently of the data, and thus, pure noise. However, when the in-sample and out-of-sample sets are randomly permuted, both the ANN and the linear regression outperform the BS-Delta. Moreover, now the 'fake VIX' feature reduces the error further, illustrating the information leakage induced by the random permutations.
The right panel displays by how much the relative reduction improves by permuting in-sample and out-of-sample sets. The relative reduction is improved more in the case of the ANN than in the case of the linear regression, and the 'fake VIX' helps both the linear regression and the ANN. The permutation sets are ordered from left to right by the relative reduction of the ANN in the baseline case; the permutation set on which the ANN performs the worst in the 'Baseline' experiment is on the left. The increasing trend hence illustrates that the relative improvement is the largest when the ANN has the least relative reduction.

Let us summarise now our observations.

- In the 'VIX' experiment, both the linear regression and the ANN perform worse than in the 'Baseline' experiment. This effect is stronger for the ANN than for the linear regression, at least in the S&P 500 dataset. An explanation is easy. The additional

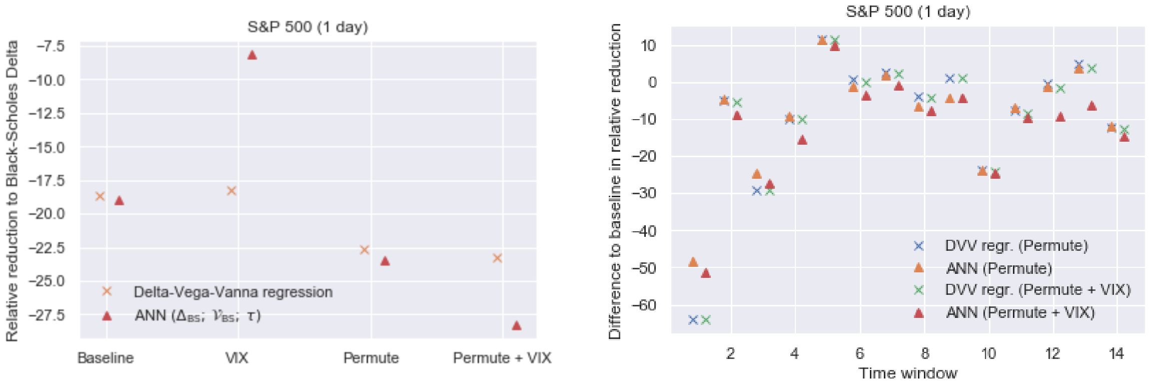


Fig. 3.13 Illustration of information leakage when failing to take into account the time series structure of the S&P 500 dataset. See the caption of Figure 3.12 for explanations. The 'Baseline' numbers of the left panel are -18.7% and -18.9%, as presented in Table 3.3. Apart from the first time window, the relative reduction is improved more in the case of the ANN than in the case of the linear regression, and in each time window the 'fake VIX' helps both the linear regression and the ANN.

feature is simulated completely independently from the data. Hence, it has no predictive power for the hedging ratio at all. Its inclusion adds additional noise, and the lower performance is due to an overfit of the training procedure, being more dramatic for the nonparametric ANN than for the three-parameter linear regression.

- Even without using the 'fake VIX' as feature, the permuted datasets lead to a better performance relative to the BS-Delta benchmark. This holds even in the case that the samples are generated by a time-homogeneous Black-Scholes model. There, instead of underperforming by about 0.2% for the linear regression and 2% for the ANN (see Table 3.6), the linear regression and ANN reduce the BS-Delta in the Black-Scholes model by about 3% after data permutation, with a larger relative improvement for the ANN.

- More striking are the results if the 'fake VIX' is included as an additional feature. Both statistical models improve, but most dramatically the ANN, which now outperforms the BS-Delta by about 7% in the Black-Scholes simulated data and by about 29% in the S&P 500 data. What is going on? By construction, different samples have the same 'fake VIX' value. Indeed, each day has several options (corresponding to different strikes) but only one 'fake VIX' value. The random permutation now allows samples from the same day to appear both in the training and in the test set. It is now possible for the ANN (and partially also for the linear regression models) to learn whether on one specific day the underlying's price goes up or down (or, in case of the S&P 500 data, there is a shift in the implied volatility surface). Hence, the 'fake VIX' tags the different days and the models are able to pick up on it.

- The right panel of Figure 3.12 breaks the average value of the left panel up into the twenty repetitions of the experiment. The differences between the repetitions are different out-of-sample sets for the 'Baseline' setup, and different random permutations. A relative improvement of up to 20% can be observed. As the outperformance of the ANN in the 'Baseline' setup increases, the improvement through the permutations becomes less significant.
- As indicated by Figure 3.9, in the first time window of the S&P 500 data, the statistical models underperform the BS-Delta. This is most likely due to the in-sample and out-of-sample sets being very different. Figure 3.13 seems to support this – it shows that in the first time window shuffling the in-sample and out-of-sample sets (the 'Permutation' experiment) leads to the largest benefit.
- As mentioned above for each of the 14 time windows in the S&P 500 dataset we did five repetitions of the experiment, their only difference being different random permutations. The five repetitions lead to very similar results. Figure 3.13 reports the average.

To conclude this subsection, let us summarise these observations. We show experimentally how random permutations of the in-sample and out-of-sample sets lead to a remarkable overestimation of the relative performance. This effect is especially strong for the ANN, but is also present for the linear regression. When adding an independent feature to the data, random permutations make this feature informative, leading to a further seemingly important improvement. For example, when using Black-Scholes data, such a random permutation leads to an outperformance of the ANN relative to the BS-Delta by about 7%. Of course, this additional feature by construction has nothing to do with finding a good hedging ratio. Thus, we have illustrated that a wrong split in in-sample and out-of-sample sets leads to significant information leakage, along with a wrong conclusion on the benefits of using an ANN over parametric models.

3.6.2 Potential information leakage through data cleaning

In this subsection we briefly discuss information leakage issues connected to the data cleaning process. One obvious mistake would be removing samples with *wrong-way option price changes*. An example is the removal of call option samples, whenever the underlying's price increases but the call price decreases. Although a first thought might be that this is a data issue such samples are very well possible due to changes in the bid-ask spread or due to the leverage effect; see also Bakshi et al. [2000] and Pérignon [2006] for empirical evidence.

The availability of end-of-period prices is a more difficult issue to be resolved. Here, in our opinion, information leakage cannot be completely avoided since it is not clear at the beginning of a period whether prices can be observed at its end. If those prices were missing at random, it would be fine to remove those samples during backtesting. However, for financial price data, such an assumption cannot be easily justified. Indeed, missing observations tend to be caused by missing market liquidity. Market liquidity and the implied volatility surface might very well depend on each other. Hence, removing missing observations could potentially lead to biased parameter estimations.

To understand whether information leakage through missing price observations appears in our experiments we ran robustness checks for both the S&P 500 and the Euro Stoxx 50 datasets.

We begin with the S&P 500 dataset. For these data, we have quoted prices for all options, along with trading volumes. For the results in Subsection 3.5.1, we remove all samples whose trading volume at the beginning of its period are zero. We keep those samples whose volume at the beginning is positive, but zero at the end of the period. As a robustness check we rerun the complete analysis with those samples removed whose trading volume is zero at the end of the period. This reduces the overall dataset by about 22% and increases the MSHE of the zero-hedge for puts (by more than 10%). An explanation for this increase is that this modified cleaning procedure removes especially deep out-of-the-money puts, thus increasing the average squared prices changes. However, the relative performance improvement of the models with respect to the BS-Delta does not change much; in particular, the conclusions of Subsection 3.5.1 seem to be robust with respect to this cleaning procedure.

Next, let us discuss the Euro Stoxx 50 dataset consisting of tick data. Using such tick data leads to several difficulties concerning missing price observations. First, the underlying's prices (we use short-term futures on the Euro Stoxx 50) and option prices are not observed synchronously. This issue is relatively mild since futures are extremely liquid. For an option observation at some time t we thus use the future's price at the last transaction before t.

However, a major issue in the data cleaning process is to determine the price of the option at the end of a period. To illustrate, consider the one-hour period setup. If an option transaction in the dataset is observed at some time t, then we would like to know the option price at time $t{+}1$ hour to backtest the hedging performance of the different methods. It is very unlikely to find a trade at exactly this time. To handle this issue we introduced a *matching tolerance window* of 6 mins (see Subsection 3.2.3). That is, if at some time t a transaction occurs then the sample's end-of-period price is the first price observation after time $t{+}1$ hour, and the sample is discarded if this end-of-period transaction occurs later than $t{+}66$ minutes.

As discussed above, we have clearly introduced some information leakage by removing illiquid samples for which no end-of-period price is observed. Let us now do again a robustness check. To this end, we increase the matching tolerance window from 6 minutes to 30 minutes. In the one-day period situation, this increases the overall number of samples from 0.6 million to 1.4 million, a 133% increase. This modified set contains now many more illiquid options, reflected also in a smaller MSHE of the zero-hedge.

Let us first summarise how the Delta-Vega-Gamma-Vanna regression performs on this modified and enlarged dataset. For the two-day hedging period, the performance improves on calls but worsens on puts, reducing the overall performance from about -23.9% to -23.0%. For the one-day period, the longer matching tolerance window improves the Delta-Vega-Gamma-Vanna regression by 0.59% with respect to BS-Delta, from -18.7% to -19.3%, benefiting both calls and puts. For the one-hour hedging period, the overall performance worsens by 0.1% with respect to BS-Delta, from -17.8% to -17.7%, and the longer matching tolerance window benefits calls and not puts. All in all, for the regression models, the conclusions of Subsection 3.5.2 are still valid. However, the longer matching tolerance window has a significantly negative effect for the ANNs. Now ANN (Δ_{BS}; $\mathcal{V}_{\mathrm{BS}}$; $\mathrm{Va}_{\mathrm{BS}}$; τ) always produce worse performance for the three hedging periods, up to even a 6% loss in outperformance. Overall, doubling the dataset by increasing the matching tolerance window does not change the regression results much, but significantly handicaps the training of the ANNs. A further test with a matching tolerance window of 60 minutes leads to the same conclusions.

3.7 Conclusion and discussion

In this chapter, we consider the problem of hedging an option over one period. We consider statistical, regression-type hedging ratios (in contrast to model-implied hedging ratios). To study whether the option sensitivities already capture the relevant nonlinearities we develop an ANN. Experiments involving both quoted prices (S&P 500 options) and high-frequency tick data (Euro Stoxx 50 options) show that the ANNs perform roughly as well (but not better) as the sensitivity-based linear regression models. However, the ANNs are not able to find additional non-linear features. Hence option sensitivities by themselves (in particular, Delta, Vega, and Vanna) in combination with a linear regression are sufficient for a good hedging performance.

The linear regression models improve the hedging performance (in terms of MSHE) of the BS-Delta by about 15-20%. An explanation is the leverage effect that allows the partial hedging of changes in the implied volatility by using the underlying. As a rule of thumb, historical data seem to imply that calls should be hedged with about $0.9\delta_{\mathrm{BS}}$ and puts with about $1.1\delta_{\mathrm{BS}}$.

We also show how information leakage in backtesting can lead to the wrong conclusions. Splitting data in in-sample and out-of-sample data without paying attention to their time series structure can mislead researchers to conclude that ANNs (or, in general, complex statistical models) outperform. Moreover, even for linear regression models with few parameters, such a wrong split may lead to strongly overconfident estimates of their performance.

We have not performed a cross-sectional study where the hedging ratio is estimated not only from options written on the same underlying. It would be interesting to see whether the hedging ratios of the linear regression models can be further improved by using options written on different underlyings, e.g., the constituents of an index.

3.8 Appendix

3.8.1 Simulation and pricing under the Heston model

To simulate the Heston model (3.5), the Euler and Milsten discretization schemes are used for the underlying stock process and the variance process, respectively. The variance process can take negative value if it is not discretized properly. The Milsten scheme can substantially alleviates the negative variance problem, and we follow the formula given in [Gatheral, 2011, chapter 2].

Discretize the time interval $[0, t]$ equally by setting $0 < t_1 < ... < t_n < t$, and $t_i - t_{i-1} = \Delta t$ constant. The stock process (3.5a) can be simulated by following

$$S_{t_{i+1}} = S_{t_i} + \sqrt{Y_{t_i}} S_i \Delta t Z_1.$$

The variance process (3.5b) can then be simulated by following

$$Y_{t_{i+1}} = Y_{t_i} + \kappa(\theta - Y_{t_i})\Delta t + \sigma_Y \sqrt{Y_{t_i} \Delta t} Z_2 + \frac{1}{4}\sigma_Y \Delta t(Z_2^2 - 1).$$

Here, Z_1 and Z_2 are correlated normally distributed random variables with a correlation ρ. The two correlated random variables can be generated by

$$\begin{aligned} Z_1 &= \widetilde{Z}_1, \\ Z_2 &= \rho \widetilde{Z}_1 + \sqrt{1 - \rho^2} \widetilde{Z}_2, \end{aligned}$$

where $\widetilde{Z}_1$ and $\widetilde{Z}_2$ are two independent and identically distributed random variables.

The price of European call option is given by

$$\begin{aligned}\mathbb{E}\left(S_T-\mathrm{e}^k\right)_+ =& S_0\left(\frac{1}{2}+\frac{1}{\pi}\int_0^\infty \Re\left(\frac{\Phi_T(\xi-\mathrm{i})}{\mathrm{i}\xi\Phi_T(-\mathrm{i})}\mathrm{e}^{-\mathrm{i}k\xi}\right)\mathrm{d}\xi\right)\\ &-K\mathrm{e}^{-rT}\left(\frac{1}{2}+\frac{1}{\pi}\int_0^\infty \Re\left(\frac{\Phi_T(\xi)}{\mathrm{i}\xi}\mathrm{e}^{-\mathrm{i}k\xi}\right)\mathrm{d}\xi\right)\\ =:& S_0\Pi_1-K\mathrm{e}^{-rT}\Pi_2,\end{aligned}$$

where $e^k=K$, and $\Phi_T(\xi):=\mathbb{E}\left(S_T^{\mathrm{i}\xi}\right)$ is the characteristic function of the log-stock price at time T. We have $\Phi_T(\xi)=\exp\Big(C_T(\xi)+D_T(\xi)V_0+\mathrm{i}\xi\log(S_0)\Big)$, where

$$\begin{aligned}C_T(\xi)&:=\mathrm{i}r\xi T+\frac{\kappa\theta}{\sigma_V^2}\left\{(\kappa-\mathrm{i}\rho\sigma_V\xi-d_T(\xi))\,T-2\log\left(\frac{1-\gamma_T(\xi)\mathrm{e}^{-d_T(\xi)T}}{1-\gamma_T(\xi)}\right)\right\},\\ D_T(\xi)&:=\frac{\kappa-\mathrm{i}\rho\sigma_V\xi-d_T(\xi)}{\sigma_V^2}\left(\frac{1-\mathrm{e}^{-d_T(\xi)T}}{1-\gamma_T(\xi)\mathrm{e}^{-d_T(\xi)T}}\right),\\ \gamma_T(\xi)&:=\frac{\kappa-\mathrm{i}\rho\sigma_V\xi-d_T(\xi)}{\kappa-\mathrm{i}\rho\sigma_V\xi+d_T(\xi)},\qquad d_T(\xi):=\sqrt{(\kappa-\mathrm{i}\rho\sigma_V\xi)^2+\sigma_V^2(\mathrm{i}\xi+\xi^2)}.\end{aligned}$$

3.8.2 Preliminary Euro Stoxx 50 data cleaning

In this appendix, we describe the Euro Stoxx 50 raw data, and how they are cleaned before the steps in Section 3.2.3 are taken. Two pairs of files are provided to us. The first pair is option tick data and its reference, and the second pair is futures tick data and its reference.

Option tick data and its reference

The reference file contains contract specifications for each option traded. They include SecurityID, SecurityType, Expiry, and StrikePrice. These specifications are matched to the option tick data for each option by SecurityID. Then we introduce two processing steps:

- SecurityType indicates the type of options, and it takes value OC, OP, OPT, and MLEG. Options with type OPT or MLEG are removed, since specifications such as strike and expiry are not provided.
- Expiry indicates the expiry of options. Options with this field empty are removed.

The data covers from January 4th 2016 to July 31st 2018, and it has more than 4.3 million trades. Each trade includes the following fields: date, SecurityID, MDEntryTime, MsgSeqNum, SenderCompID, MDUpdateAction, MDEntryPx, MDEntrySize, TrdType, TradeCondition, AggressorTimestamp, RequestTime, AggressorSide, NumberOfBuyOrders, NumberOfSellOrders, RestingCxlQty, and MDEntryID.

We choose to keep the following fields only and remove the rest: RestingCxlQty, TrdType, MDUpdateAction, MDEntryPx, MDEntrySize, AggressorSide, ExecuteTime, and SecurityID. We now explain on this choice.

RestingCxlQty indicates the quantity of trades that are deleted due to some events. These deleted trades are removed. TrdType indicates when the trade happens, such as opening auction, intraday trading and etc. There exist three trade types in the data, and they are '1', '1100', and empty. We remove trades with TrdType '1', since this type is not documented.

MDEntryPx and MDEntrySize indicate the price and quantity of each trade. Multiple trades can be executed at the same time, but with different prices. This happens because a large order eats into the order book across several levels. As we mentioned in Section 3.2.3, we use volume-weighted average price for trades that are executed at exactly the same time. The averages are calculated on trades for which SecurityID, ExecuteTime, AggressorSide, TradeCondition, and TrdType are the same.

Futures tick data and its reference

The reference file has exactly the same fields as the options file. SecurityType and Expiry are appended to the futures tick data file for each order by SecurityID. We also introduce two preprocessing:

- SecurityType indicates the type of the futures, and it takes values FUT or MLEG. We remove all MLEG, since expiry is not provided.

- Expiry indicates the expiry of futures. We remove all futures that do not have expiry provided.

Futures are used as the hedging instruments for options. Same procedures are taken to clean the futures tick data as used for option tick data. The future that has its expiry closest to an option trade time is used as the hedging instrument of that option. The number of futures traded is much fewer than that of options, but each has much higher trading volume. Futures that have short maturities are traded much more often than the rest, accounting for more than 90%.

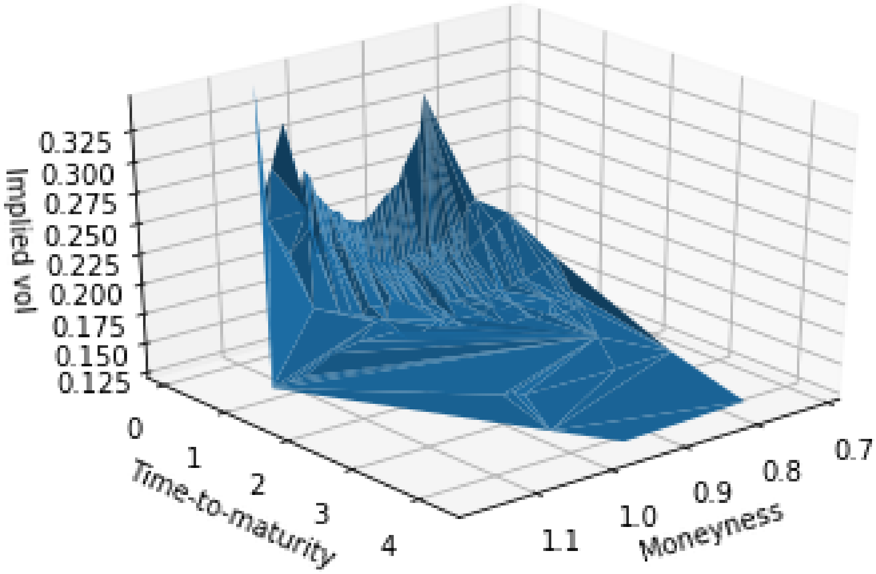


Fig. 3.14 The implied volatility surface at around 12 o'clock on January 5th, 2016. The future ID is 1292203, which has the shortest maturity among all futures on that date.

CHAPTER 4

DIAGNOSTICS AND ROBUSTNESS

This chapter is based on joint work with Prof. Johannes Ruf, and contains the appendix of the paper

Ruf, J and Wang, W, Hedging with linear regressions and neural networks, available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3580132. Accepted by the *Journal of Business and Economic Statistics* subject to minor revisions.

4.1 Additional hyperparameters of HedgeNet

We now add details on the implementation and training of HedgeNet (see Subsection 3.3.1).

Based on preliminary experiments on simulated data we set the learning rate to 10^{-4} and the batch size to 64. Usually we train each ANN for 300 epochs.[1] Using a validation set, we apply *early stopping* by choosing the ANN with the smallest validation error.

The optimisation criterion is a Tikhonov regularised version of squared loss. We use an L^2 penalty term for the ANN weights. We also experimented with other regularisations, such as an L^1 penalty, a combined L^1-L^2 penalty, and dropout. They all lead to similar results and the same conclusions. The regularisation strength α is tuned for each dataset and hedging period. The larger α is the more the weights are pushed to zero. In case of the simulated data, α is tuned by using an independent dataset that is simulated from the same model but with a different random seed. Hence, the actual training and test datasets are different from the ones used for tuning. For the S&P 500 dataset, we tune only using the first four time windows. For the Euro Stoxx 50 dataset, due to its single window experimental setup, we choose the tuning parameter α on the validation set. Such in-sample tuning favours the performance of the ANNs. With a proper out-of-sample tuning, the ANNs would perform worse.

[1] We also apply visual inspections of the training / validation loss to confirm that the ANN is indeed trained.

For each dataset and value α on a logarithmic grid, we run five iterations of the ANN training, each with a different (random) weight initialisation. For each dataset we then pick an α after inspecting the average and standard deviations of the test errors (on the independent dataset when using simulated data, and on the first few time windows when using real data). Table 4.1 summarises the chosen L^2 regularisation parameters.

		S&P 500	Euro Stoxx 50	Black-Scholes	Heston
M; $\sigma_{\text{impl}}\sqrt{\tau}$	1H	-	10^{-5}	-	-
	1D	10^{-7}	10^{-2}	10^{-4}	10^{-4}
	2D	10^{-3}	10^{-2}	10^{-4}	10^{-4}
Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	1H	-	10^{-3}	-	-
	1D	10^{-4}	10^{-2}	10^{-4}	10^{-3}
	2D	10^{-3}	10^{-1}	10^{-3}	10^{-3}
Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	1H	-	10^{-3}	-	-
	1D	10^{-4}	10^{-3}	10^{-4}	10^{-3}
	2D	10^{-2}	10^{-3}	10^{-3}	10^{-3}

Table 4.1 Regularisation parameters used for the training of HedgeNet in the different experiments.

4.2 Some heuristics on the leverage effect

To understand the leverage effect and its interaction with the coefficients of the linear regressions a bit better we do the following empirical study. For each option type (put or call) and for different time-to-maturities (namely τ smaller than 1 month, τ between 1 and 6 months, and τ greater than 6 months) we regress $\Delta\sigma_{\text{impl}}$ on ΔS, without intercept. This yields a slope b. We then compute

$$\mathrm{LC} = \mathrm{b}\,\frac{1}{\mathrm{N_{train}}}\sum_{\mathrm{t,j}}^{\mathrm{N_{train}}}\frac{\mathcal{V}_{\mathrm{BS,t,j}}}{\delta_{\mathrm{BS,t,j}}}, \tag{4.1}$$

which we call *leverage coefficient*. These heuristics are motivated by how much we should adjust a hedge due to the leverage effect. Indeed, a change of $\Delta\sigma_{\text{impl}}$ leads roughly to a change of $\mathcal{V}_{\text{BS}}\Delta\sigma_{\text{impl}}$ in the option price. A part $\mathcal{V}_{\text{BS}}b\Delta S$ of this change can be explained by the change in the underlying's price due to the correlation of implied volatilities and returns. Considering a multiplicative effect on the BS-Delta, we need to divide this number by δ_{BS}.

Figure 4.1 shows the leverage coefficients for the different option categories for the one-day hedging period. The plots for the other hedging periods (for which $\Delta\sigma_{\text{impl}}$ and

ΔS are different, yielding slightly different estimates for b in (4.1)) look similar. The fact that the leverage coefficient tends to be negative for calls (positive for puts) reflects how the regression models replace the BS-Delta by a number smaller (larger) than one. Note the jumps of the leverage coefficient in the S&P 500 plot from period 4 to 5, 7 to 8, and 12 to 13. This is consistent with the change of the Delta coefficient in Delta-Vega-Vanna regression of Figure 3.10.

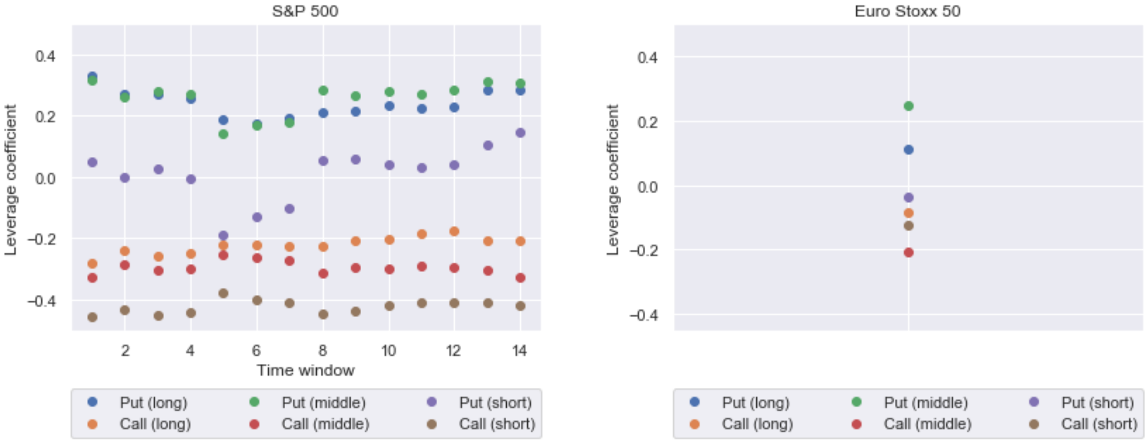


Fig. 4.1 Leverage coefficients as given in (4.1) on the three categories of time-to-maturity in the S&P 500 (left) and Euro Stoxx 50 (right) dataset for the one-day hedging period. 'Short' means a time-to-maturity of less than 1 month, 'middle' means between 1 month and 6 months, and 'long' means more than 6 months.

4.3 Additional diagnostics

4.3.1 Additional diagnostics for the S&P 500 dataset

We use this subsection to provide some additional figures concerning the performance of the various statistical models on the S&P 500 dataset.

Figure 4.2 extends Figure 3.9 by including the MSHE of the zero hedge strategy. As we can see, the MSHE for any of the methods is large exactly when the MSHE of the unhedged portfolio is large. Figure 4.3 shows the ratio of the MSHEs of the same four statistical models to the zero hedge MSHE. The hedging performance gets worse in later periods. The MSHE corresponding to the BS-Delta minus the MSHE of one of the statistical models divided by the zero hedge MSHE is about 2%.

Figure 4.4 shows the average logarithmic return and its standard deviation of the S&P 500 dataset in each time window. We see that the standard deviations in the out-of-sample sets tend to be large when the zero hedge MSHEs in Figure 4.2 are large.

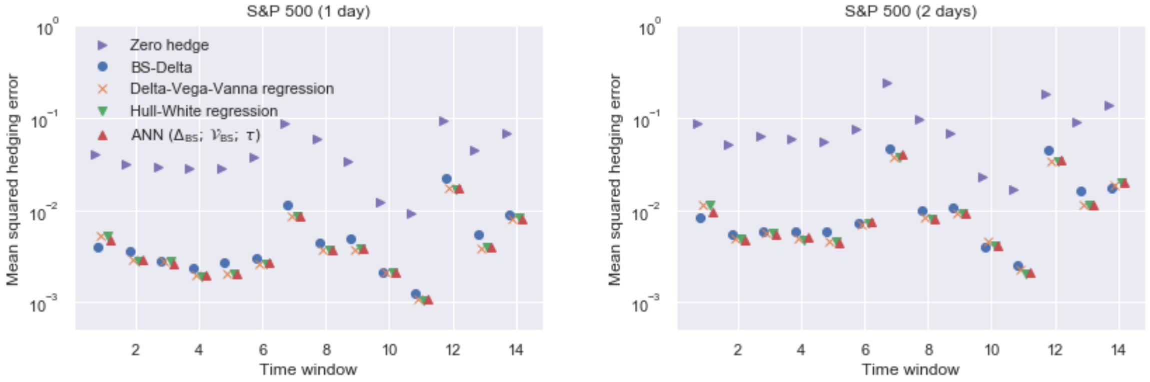


Fig. 4.2 MSHEs for four different statistical models of the hedging ratio and the zero hedge across all 14 time windows in the S&P 500 dataset for the one-day (left) and two-day (right) hedging periods. The numbers of the statistical models correspond to the numbers in Figure 3.9, but are now presented on a logarithmic scale.

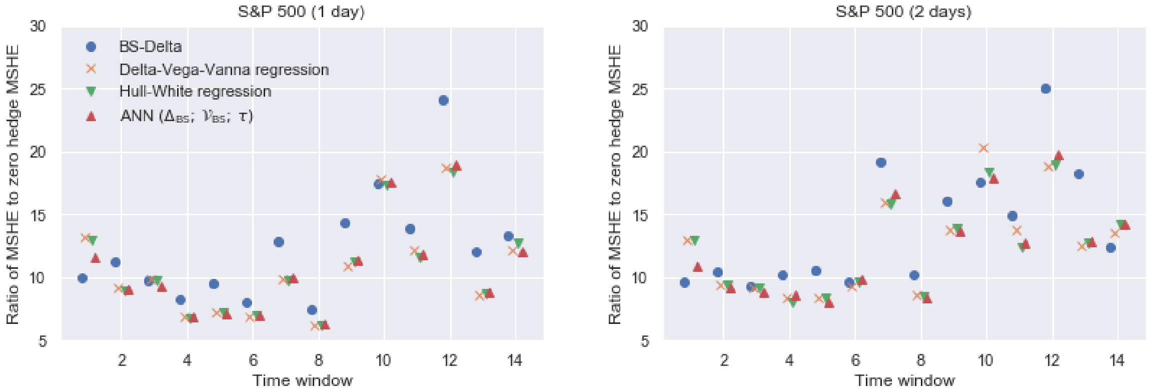


Fig. 4.3 The ratio of the MSHEs of four statistical models to the hedging ratio and the zero hedge MSHE in the S&P 500 dataset for the one-day (left) and two-day (right) hedging.

Figure 4.5 scatterplots the hedging ratios corresponding to the different statistical models. Here, we provide only one such plot, namely comparing the Delta-Vega-Vanna hedging ratio with the hedging ratio of the ANN(Δ_{BS}; $\mathcal{V}_{\mathrm{BS}}$; τ) for the one-day hedging period. Each point is a sample in the test set. We do not directly plot the hedging ratios but $\mathrm{N}^{-1}(\delta_{\mathrm{NN}})$ against $\mathrm{N}^{-1}(\delta_{\mathrm{LR}})$, where N denotes again the cumulative standard normal distribution. The ratios are very similar but different in the tails, where the ANN seems to overfit. We only provide the plots for two representative time windows. In window 1, the BS-Delta outperforms all regression models, while window 12 represents a more typical situation where the BS-Delta underperforms the regression model and the ANN.

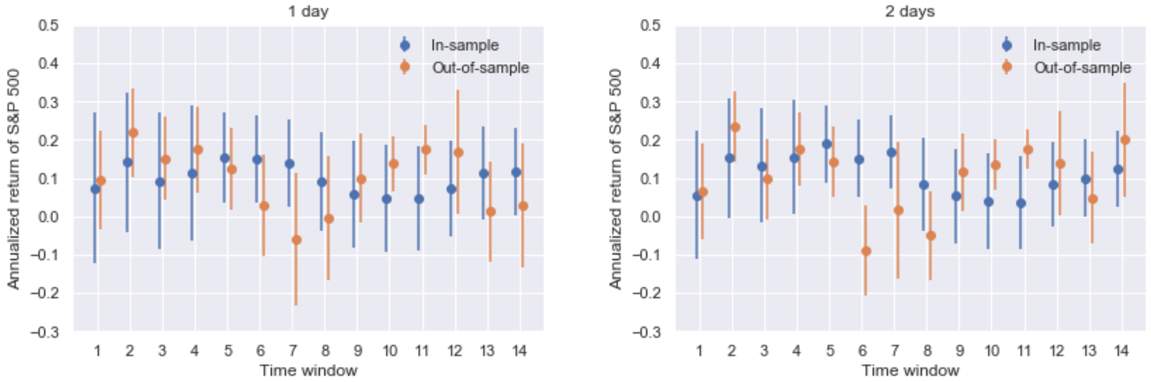


Fig. 4.4 The average annualised logarithmic one-day (left) and two-day (right) return of the S&P 500 in each of the 14 time windows. Each line segment shows the average logarithmic return plus/minus one standard error of the logarithmic returns for each time window. The lines tend to be longer, meaning a higher standard deviation, when the returns are smaller, illustrating the leverage effect.

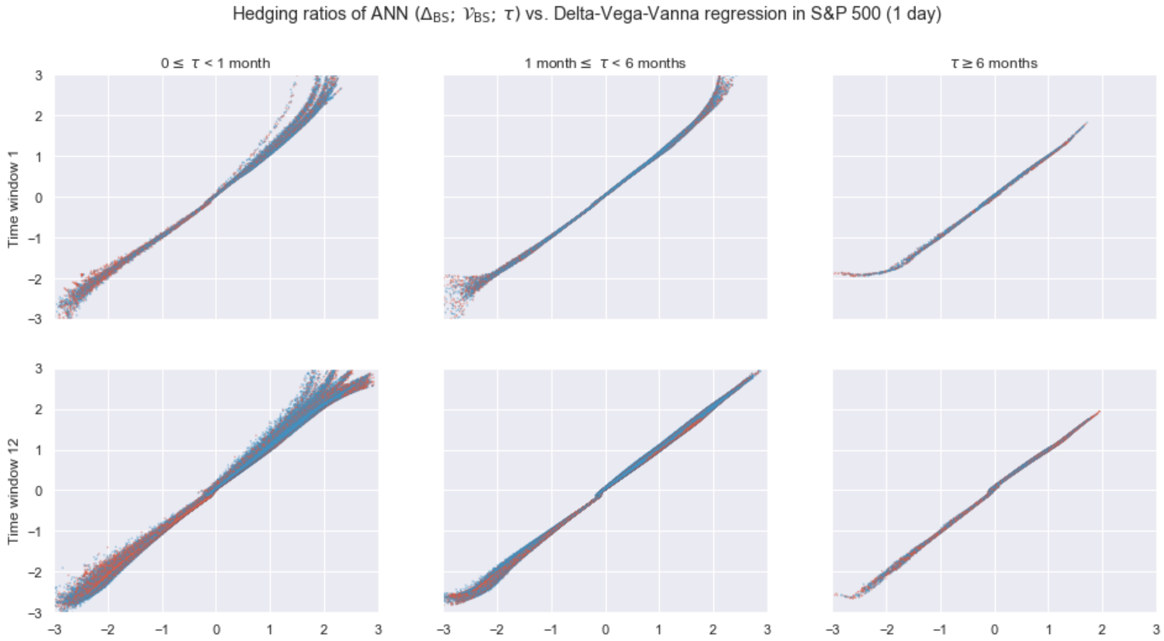


Fig. 4.5 ANN$(\Delta_{\mathrm{BS}};\, \mathcal{V}_{\mathrm{BS}};\, \tau)$ versus Delta-Vega-Vanna regression hedging ratios in S&P 500 dataset. Each point represents a sample. We use transformed scales so that the x-value of each sample corresponds to $\mathbf{N}^{-1}(\delta_{\mathrm{LR}})$ and the y-value to $\mathbf{N}^{-1}(\delta_{\mathrm{NN}})$, where $\mathbf{N}$ denotes the cumulative standard normal distribution. If the point is blue the MSHE corresponding to the ANN is smaller than the one corresponding to the linear regression. On the other hand, if the point is red the linear regression outperforms. Each row shows a time window; the one on the top is a window when the BS-Delta outperforms the statistical models; the one on the bottom is a more typical one, when the linear regressions and ANNs outperform the BS-Delta. Each column corresponds to a different set of maturities; namely less than one month (left); 1 month to 6 months (middle), and more than 6 months (right).

Figure 4.6 plots the mean squared relative hedging error, i.e., the average of the hedging errors divided by the option prices, of the Delta-Vega-Vanna regression against time-to-maturity, Vega, moneyness, Delta, and Gamma for S&P 500 data. The first left panel shows an exponential decrease (due to the logarithmic scale) of the relative hedging error with respect to time to maturity. The first right panel shows that the relative hedging errors decrease super-exponentially as Vega increases, i.e., as the options have a longer time-to-maturity and are less out-of-the money. The second left panel shows that the relative hedging error is larger for at-the-money options. The second right panel shows the relative hedging error is large when Delta is close to zero, i.e. as the options have short time-to-maturity and are out-of-money. The bottom panel shows the relative hedging error increases as Gamma increases, i.e., as the options are at-the-money and short time-to-maturity.

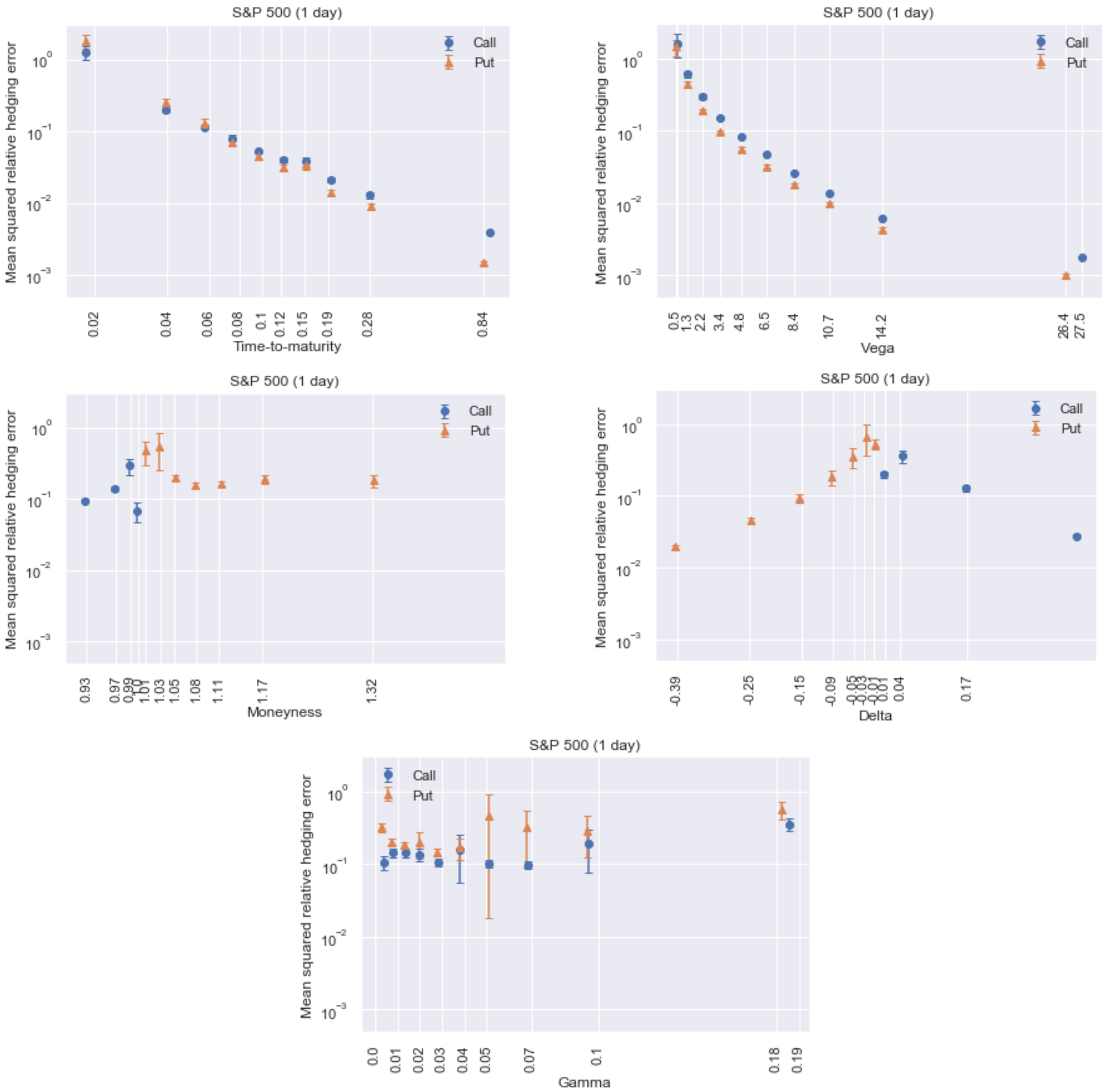


Fig. 4.6 Mean squared relative hedging error of the Delta-Vega-Vanna regression on a logarithmic scale against time-to-maturity (first left), Vega (first right), moneyness (second left), Delta (second right), and Gamma (bottom) for the one-day hedging period with S&P 500 dataset. Each line segment provides a point estimate plus/minus one standard error. Each interval has 10% of the overall samples, and the tick on the x-axis shows the average of the feature of the samples falling into the corresponding interval. Calls and puts may have different averages in each interval.

4.3.2 Additional diagnostics for the Euro Stoxx 50 dataset

Figure 4.7 scatterplots the hedging ratios corresponding to the different statistical models. We refer to the caption of Figure 4.5 for explanations. Different to Figure 4.5 with the S&P 500 dataset, the hedging ratios of the ANN now look quite different from the linear regression model. Consistently with the prevalence of red points, for the one-day hedging period, the ANNs display a relatively bad performance (recall Table 3.4).

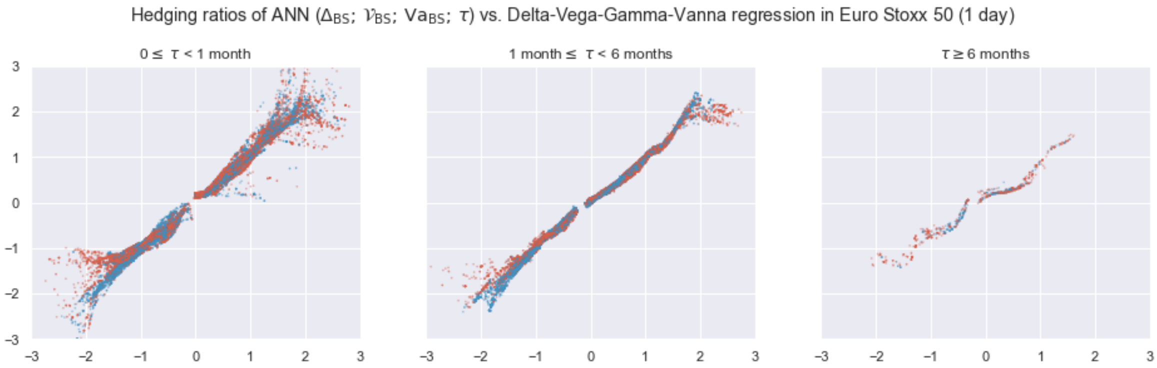


Fig. 4.7 ANN(Δ_{BS}; $\mathcal{V}_{\mathrm{BS}}$; $\mathrm{Va}_{\mathrm{BS}}$; τ) versus Delta-Vega-Gamma-Vanna regression hedging ratios in the Euro Stoxx 50 dataset. See Figure 4.5 for additional explanations.

Figure 4.8 plots the mean squared relative hedging error, i.e., the average of the hedging errors divided by the option prices, of the Delta-Vega-Gamma-Vanna regression against time-to-maturity, Vega, moneyness, Delta, and Gamma for Euro Stoxx 50 data. In comparison to the S&P 500 dataset (see Figure 4.6), the decrease seems to be a little bit smaller as time-to-maturity and Vega increase, respectively. The hedging error is also larger when Delta is close to zero. However, the plots of hedging error against Delta or Gamma are different to those in S&P 500 dataset; hedging error no longer increases as options move to at-the-money or Gamma is large.

4.3.3 Additional diagnostic for the Black-Scholes dataset

Figure 4.9 shows the MSHE of the four statistical hedging models and the unhedged portfolio. Same as the other two datasets, the MSHE of any models is large when the unhedged MSHE is large.

Figure 4.10 shows the mean squared relative hedging error, i.e., the average of the hedging errors divided by the option prices, of the Delta-Vega-Vanna regression against time-to-maturity, Vega, moneyness, Delta, and Gamma for Black-Scholes data. The overall shapes of the mean squared relative hedging error in these figures resemble those in S&P 500 and Euro Stoxx 50 datasets, but with smaller value.

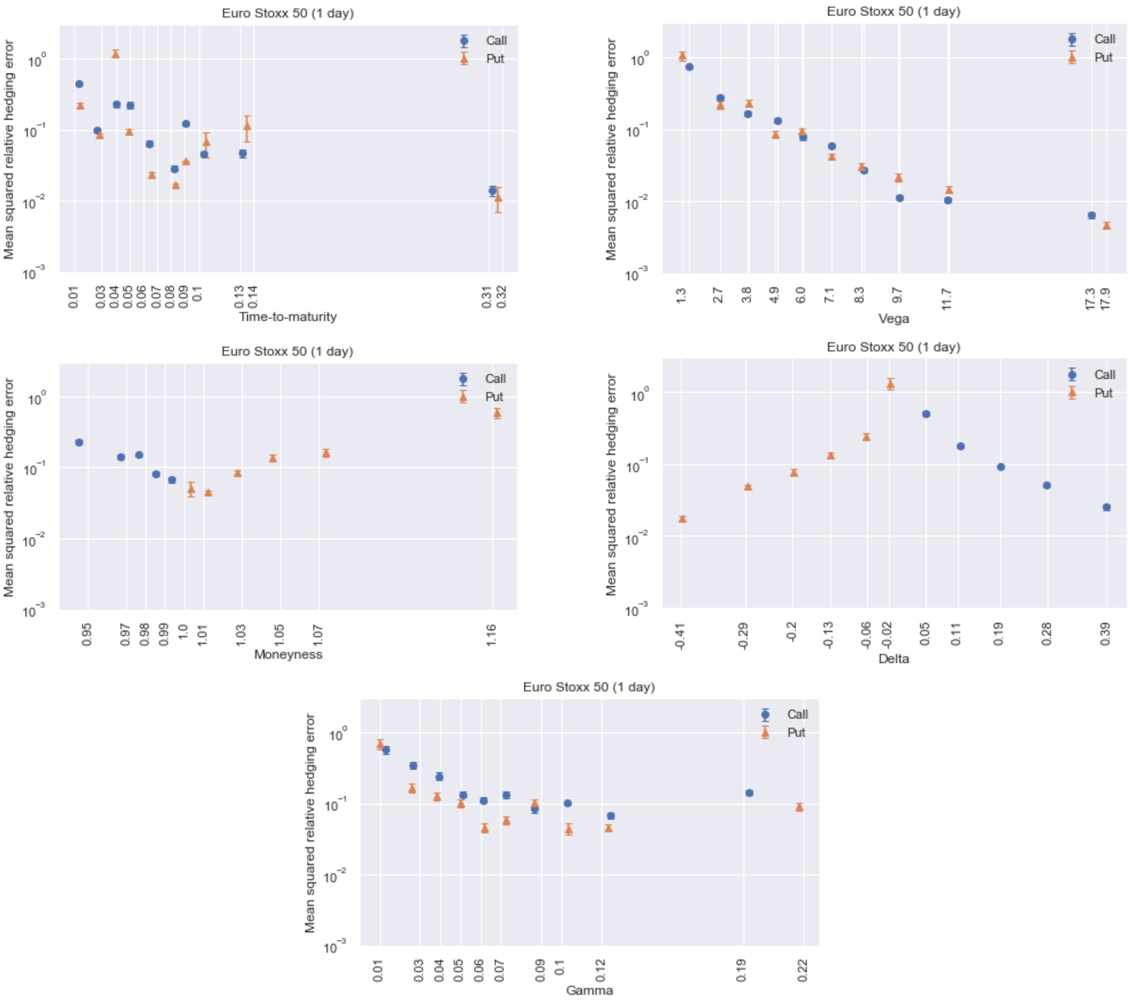


Fig. 4.8 Mean squared relative hedging error of the Delta-Vega-Gamma-Vanna regression on a logarithmic scale against time-to-maturity (first left), Vega (first right), moneyness (second left), Delta (second right), and Gamma (bottom) for the one-day hedging period with Euro Stoxx 50 data. See Figure 4.6 for additional explanations.

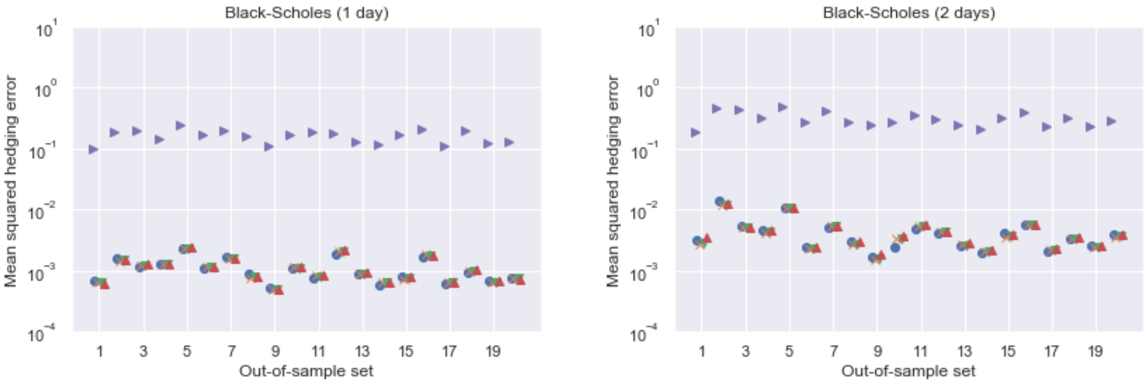


Fig. 4.9 MSHEs for four different statistical models of the hedging ratio and the zero hedge across the 20 out-of-sample sets in the Heston dataset for the one-day (left) and two-day (right) hedging periods.

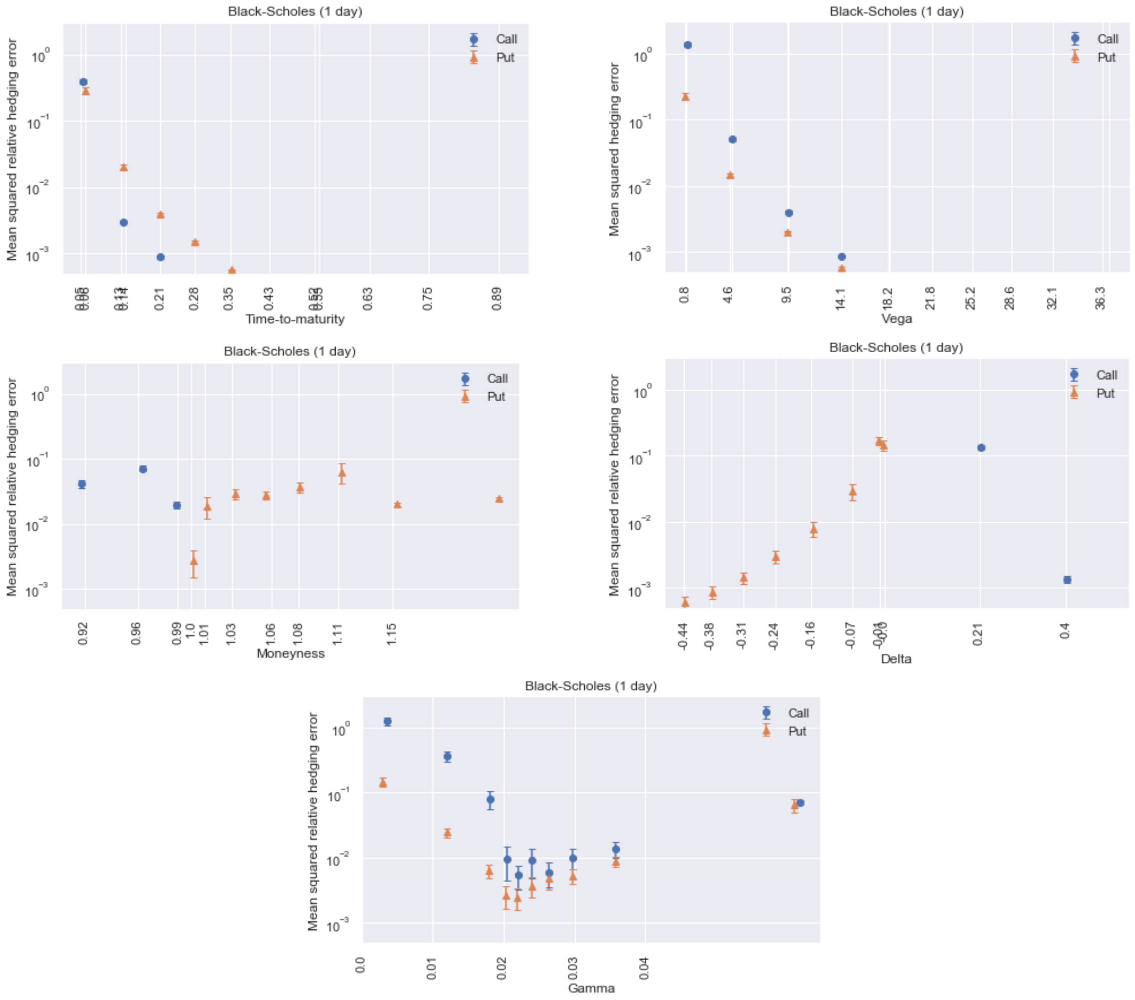


Fig. 4.10 Mean squared relative hedging error of the Gamma-only regression on a logarithmic scale against time-to-maturity (first left), Vega (first right), moneyness (second left), Delta (second right), and Gamma (bottom) for the one-day hedging period with the Black-Scholes dataset. Each line segment provides a point estimate plus/minus one standard error. See Figure 4.6 for additional explanations.

4.3.4 Additional diagnostics for the Heston dataset

Figure 4.11 shows the MSHE of the four statistical hedging models and the unhedged portfolio. Same as on the other three datasets, the MSHE of any model is large exactly when the unhedged MSHE is large. It also implies all of the four model reduces the MSHE by more than 90% with respect to the unhedged portfolio. The MSHE corresponding to the BS-Delta minus the MSHE of one of the statistical models divided by the unhedged MSHE is about 1%. This is smaller than that in S&P 500 dataset.

Figure 4.12 plots the mean squared relative hedging error, i.e., the average of the hedging errors divided by the option prices, of the Delta-Vega-Vanna regression against time-to-maturity, Vega, moneyness, Delta, and Gamma. The first four panels are very similar to those in S&P 500. They are also close to those in Euro Stoxx 50, except the one plotted against time-to-maturity. The Gamma plot seems to indicate options with either small or large Gamma have large mean squared relative hedging error.

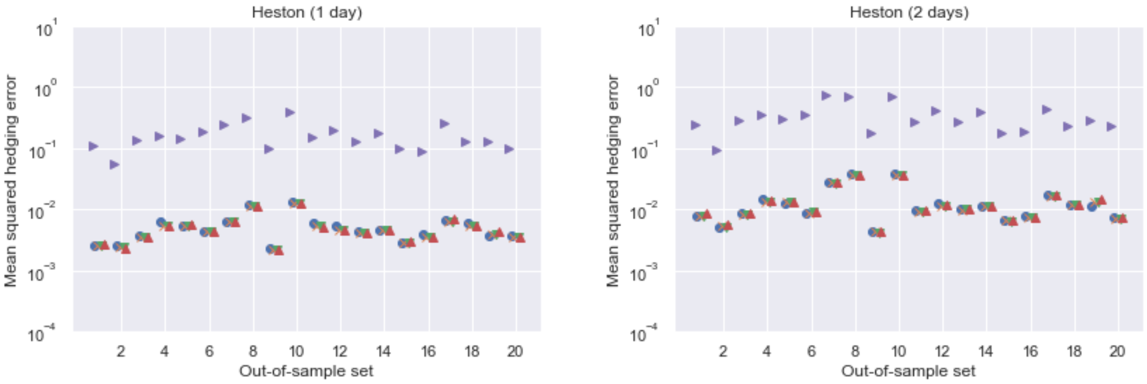


Fig. 4.11 MSHEs for four different statistical models of the hedging ratio and the zero hedge across the 10 out-of-sample sets in the Heston dataset for the one-day (left) and two-day (right) hedging periods.

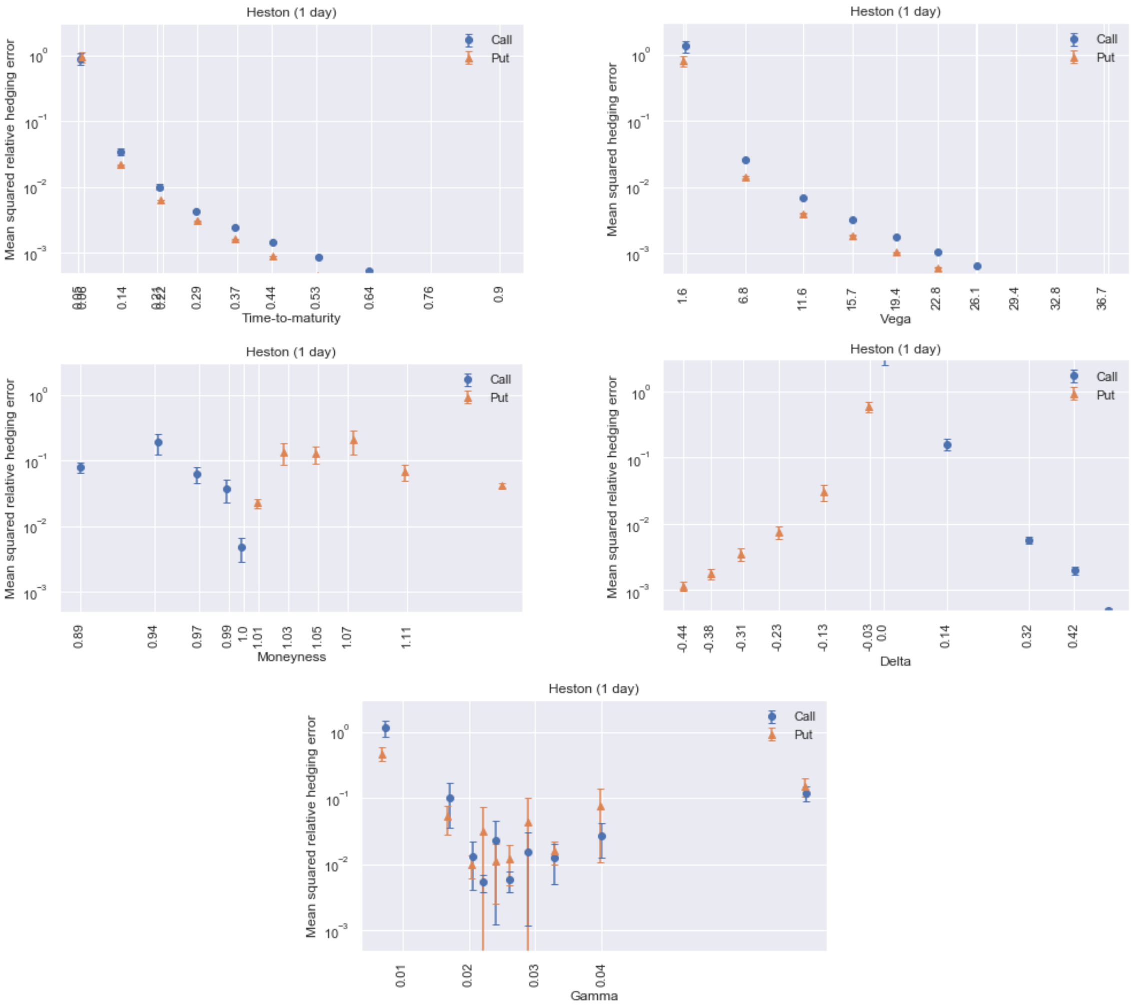


Fig. 4.12 Mean squared relative hedging error of the Delta-Vega-Vanna regression on a logarithmic scale against time-to-maturity (first left), Vega (first right), moneyness (second left), Delta (second right), and Gamma (bottom) in the Heston dataset for the one-day hedging period. Each line segment provides a point estimate plus/minus one standard error. See Figure 4.6 for additional explanations.

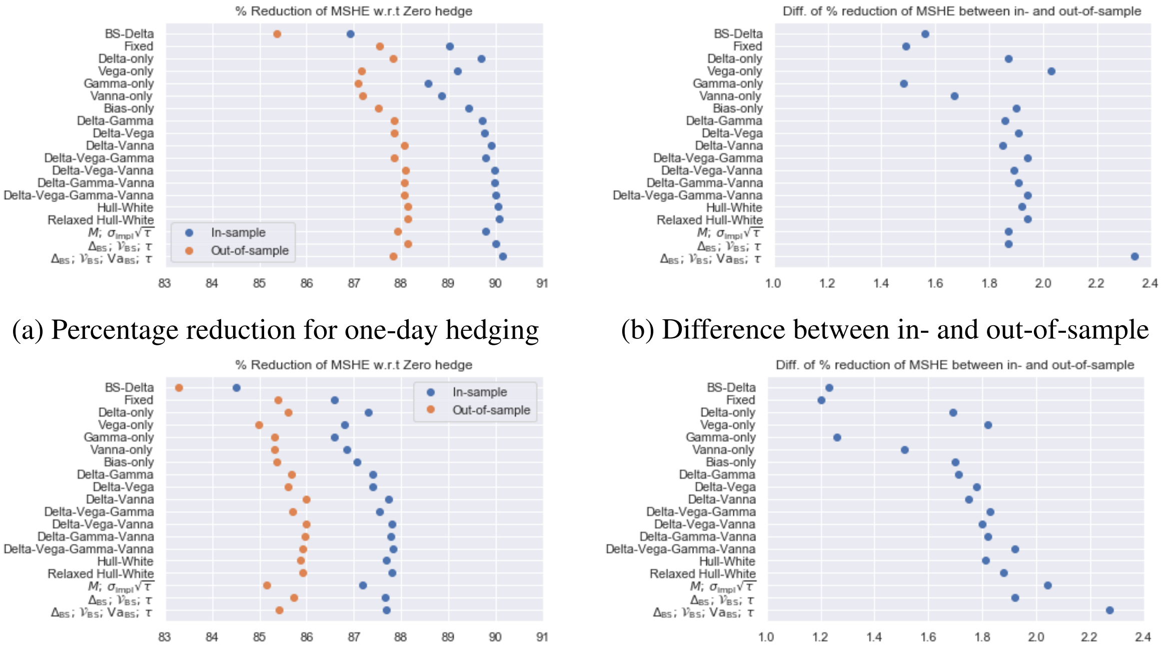
4.4 In-sample performance and overfitting

Until now, we have presented and analysed results based on out-of-sample data only. In this subsection, we present the performance of BS-Delta and statistical hedging methods on in-sample datasets, and discuss the existence of overfitting.

In the following, we evaluate the percentages of reduced MSHEs of statistical hedging methods with respect to zero-hedge MSHE, instead of with respect to BS-Delta as shown in Table 3.3 and alike. Since we optimise the loss (3.7), overfitting should be investigated based on it, instead of BS-Delta which itself is an strategy to be investigated.

Since we roll forward the window by the number of test days on the S&P 500 dataset, the in-sample data are overlapped between several consecutive time windows. To deal with this, we keep all in-sample hedging errors of the same sample but from different windows, instead of overwriting each in-sample hedging error by the last time it is used for fitting. Then for each window, we calculate the in-sample MSHEs and the number of samples, in order to calculate the weighted MSHE. There is no such overlapping issue for the Euro Stoxx 50 dataset, and we simply take each unique in-sample hedging error to evaluate the in-sample MSHE.

Figure 4.13 shows the percentages (on the left) of reduced MSHE by BS-Delta, ‘fixed’, and statistical hedging with respect to zero-hedge on in-sample and out-of-sample data and their difference (on the right) for both hedging periods on S&P 500 data. The two panels on the left show in-sample performance is always better than out-of-sample performance, even for the BS-Delta and ’fixed’ strategy which do not involve any fitting. All methods reduce more MSHE in the one-day than in the two-day hedging. The right panels show the differences of percentages of MSHEs reduced by each method between the in-sample and out-of-sample. The larger this difference is, the more the model overfits. They show BS-Delta and ’fixed’ strategies stay the two least overfitting models, since they do not involve fitting. ANN(Δ_{BS}; $\mathcal{V}_{\mathrm{BS}}$; $\mathrm{Va}_{\mathrm{BS}}$; τ) overfits the most, corresponding to the bad performance in Table 3.3.



(a) Percentage reduction for one-day hedging

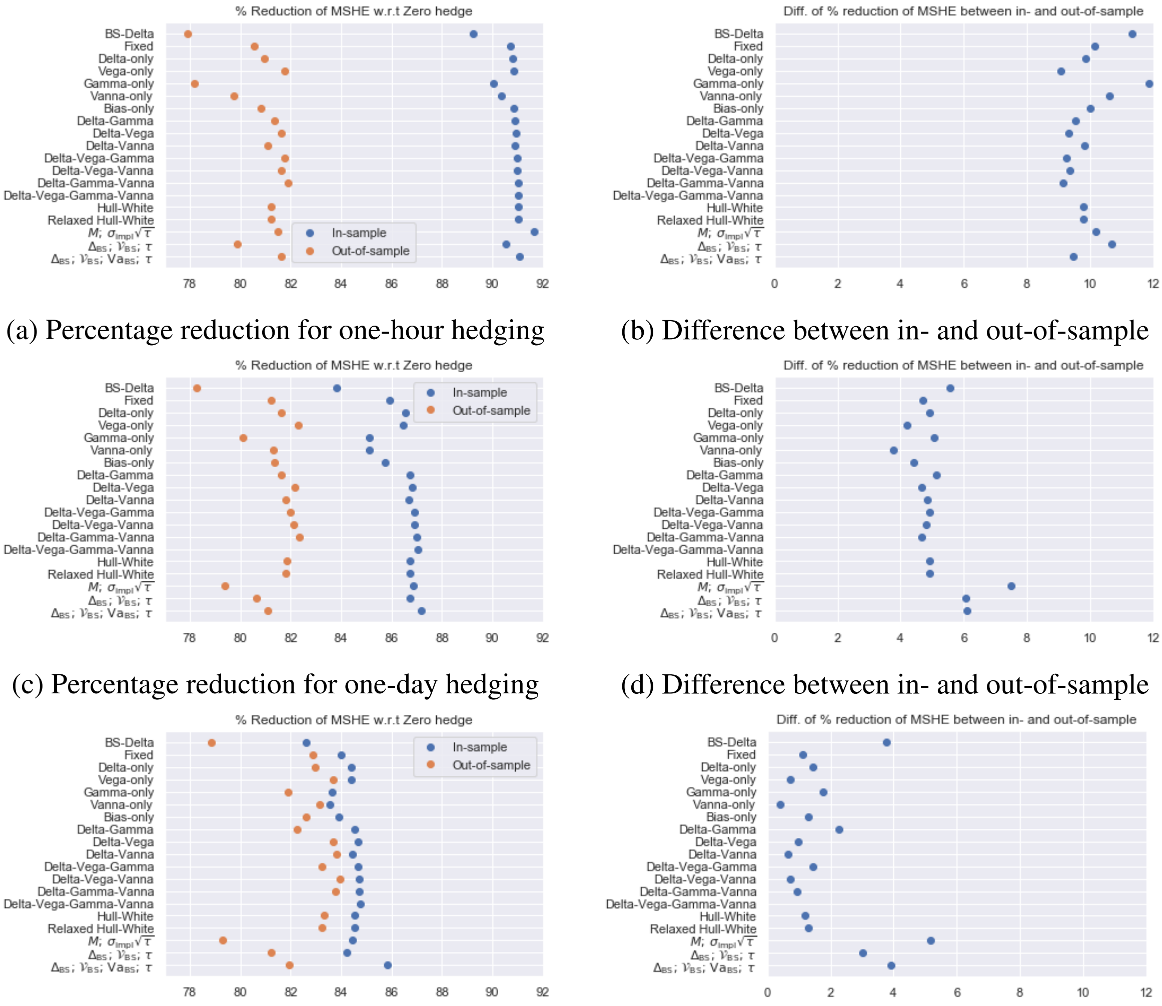
(b) Difference between in- and out-of-sample

(c) Percentage reduction for two-day hedging

(d) Difference between in- and out-of-sample

Fig. 4.13 The percentages of reduced MSHE by BS-Delta and statistical regressions with respect to zero-hedge on in-sample and out-of-sample data (left) and their differences (right) for both hedging periods on the S&P 500 data. The top two panels correspond to one-day hedging, and the bottom two-day hedging.

Figure 4.14 shows the corresponding results on the Euro Stoxx 50 data. The in-sample performances decrease as hedging periods increase, but the out-of-sample performances remain stable, giving the smaller differences between in-sample and out-of-sample in the right panels as hedging periods increase. It implies that the statistical hedging methods are less likely to overfit on large hedging periods. Figure 4.14d and Figure 4.14f show that the three ANNs have the largest differences in reduction of MSHE between in- and out-of-sample. This explains the significant underperformance of ANNs compared to other regressions for one-day and two-day hedging, shown in Table 3.4.



(a) Percentage reduction for one-hour hedging (b) Difference between in- and out-of-sample

(c) Percentage reduction for one-day hedging (d) Difference between in- and out-of-sample

(e) Percentage reduction for two-day hedging (f) Difference between in- and out-of-sample

Fig. 4.14 The percentages of reduced MSHE by BS-Delta and statistical regressions with respect to zero-hedge on in-sample and out-of-sample data (left) and their difference (right) for the three hedging periods on the Euro Stoxx 50 dataset. The top two panels correspond to one-hour hedging, the middle one-day hedging, and the bottom two-day hedging.

We note from Figures 4.13 and 4.14 that the BS-Delta is consistently worse for out-of-sample than in-sample data. This seems odd, since there is no fitting involved. Hence, one should not see a gap between in- and out-of-sample performance consistently. Figure 4.15 shows the reduced MSHE by BS-Delta as a percentage of zero-hedge, shown window-by-window for both in- and out-of-sample data of the S&P 500 data. There are more windows in which the BS-Delta performs better on the out-of-sample set than on corresponding in-sample set. Overall, we don't have an explanation for the consistent underperformance of BS-Delta on in-sample datasets yet.

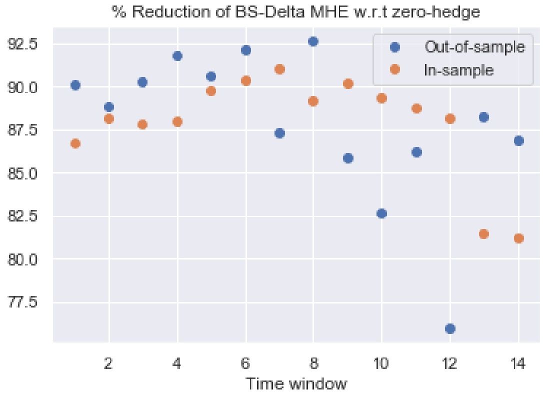


Fig. 4.15 The reduced MSHE by BS-Delta as a percentage of zero-hedge MSHE, shown window-by-window for both in- and out-of-sample sets for S&P 500 dataset.

4.5 Robustness experiments

In this section, we investigate the sensitivity of the performance in Chapter 3 with respect to various cleaning and experimental configurations. We only work on selective alternative experiments that have been undertaken in the literature, or that matter to show the information leakage issue that we discovered.

This section is organized as follows. Section 4.5.1 discusses the effect of removing samples that have a time-to-maturities less than 14 calender days. Section 4.5.2 discusses the performance of ANNs trained with non-stationary features. Section 4.5.3 shows the performance of ANNs with the cumulative distribution function of standard normal distribution as the output activation function. The above three subsections concern the S&P 500 and Euro Stoxx 50 datasets. Section 4.5.4 discusses the information leakage concerning removing options with zero trading volume. Section 4.5.5 discusses the single window setup rather than using rolling window. The above two subsections concern the S&P 500 dataset only. Section 4.5.6 discusses the matching tolerance for the Euro Stoxx 50 data. Section 4.5.7 discusses the effect of drift on the regression methods based in the two simulated datasets.

In this section, we refer to the cleaning, experimental setup, and clean data used in Section 3.2 as 'Baseline' cleaning, setup, and clean data, for each corresponding raw dataset.

4.5.1 Removing options with short time-to-maturity

It is common to restrict the range of moneyness for options when cleaning the raw data. In the literature, filters different to ours are applied on the raw data, and they tend to remove more data. For instance, Anders et al. [1998] remove options for which time-to-maturity is less than 15 days, and moneyness is between 0.85 and 1.15. Hull and White [2017] instead remove options for which time-to-maturity is less than 14 days, calls for which the practitioner Black-Scholes delta is less than 0.05 or greater than 0.95, and puts for which delta is less than -0.95 or greater than -0.05. Vähämaa [2004] only keep options for which time-to-maturity is between 5 and 120 days, and moneyness is between 0.9 and 1.1. Other kinds of cleaning can be found in Bakshi et al. [1997], Malliaris and Salchenberger [1993a,b], and White [1998].

The reasons for removing these kinds of data are various, mostly arguing against the quality of such data. For instance, Bakshi et al. [1997] argue that short time-to-maturity options have liquidity-related issues. Anders et al. [1998] argue that these options have small time value, and that the integer pricing behaviour[2] leads to severe pricing deviation. In addition, deep out-of-money options have very low trading volume.

In this subsection, we apply stricter cleaning than we have used in Sections 3.2.2 and 3.2.3, and investigate how the results change. Two changes are made: 1) The range of moneyness included is changed from $[0.8, 1.5]$ to $[0.8, 1.2]$ so that calls and puts are treated equally, and 2) Options for which time-to-maturity is less than 14 days (equivalent to 10 business days) are removed.

S&P 500

If we only apply the first change and keep only options for which moneyness is between 0.8 and 1.2, there are 2.3 million samples left, and the reduction of number of samples is 12% of the 'Baseline' data. We applied the same analysis that is used to make Table 3.3. To summarize, the zero-hedge MSHE increases by 11.8%. This increase is because deep out-of-money puts are removed, for which the hedging error is less than average. Relatively, the reductions of MSHE by regressions changes very little. To save space, we do not report the exact results here.

After applying the second change, the size of the new data set is 1.8 million, a reduction of 31% compared to 'Baseline' dataset. Table 4.2 shows the performance of the statistical hedging methods on the S&P 500 for which moneyness is restricted to be between 0.8 and 1.2, and time-to-maturity greater than 14 calender days. Compare this table with Table 3.3, the zero-hedge MSHE for calls does not change much, but increases a lot for puts. The BS-Delta MSHE decreases for calls but increases for puts. All the regressions gain an extra 2% reduction in terms of MSHE against the BS-Delta, when moneyness and time-to-maturity are restricted. Given that the regressions work much better on calls than puts, the roughly same amount of gain implies restricting moneyness and time-to-maturity helps more for puts and calls. Indeed, focusing on the Delta-Vega-Vanna regression, restricting the data changes the performance from -23.0% to -25.5% for calls, and from -16.6% to -18.8% for

[2] Their preliminary analysis shows that there is a tendency for options to be traded at integer values.

puts. ANNs also benefit from this change. Taking the MSHE for both calls and puts by the ANN (Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ) for instance, it changes from -18.9% to -19.8%, which is a 0.9% improvement compared to 2.3% for Delta-Vega-Vanna regression.

		1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	4.11	6.28	5.48	8.40	12.7	11.2
	BS-Delta	0.621	0.765	0.712	1.40	1.76	1.63
Regressions	Delta-only	-23.4	-16.9	-19.0	-18.8	-15.4	-16.5
	Vega-only	-16.3	-13.8	-14.6	-12.6	-12.5	-12.6
	Gamma-only	-20.0	-14.6	-16.3	-19.0	-16.0	-16.9
	Vanna-only	-14.3	-14.3	-14.3	-12.7	-15.5	-14.6
	Delta-Gamma	-24.1	-16.7	-19.1	**-20.4**	-16.4	-17.7
	Delta-Vega	-23.5	-16.9	-19.0	-18.9	-15.4	-16.5
	Delta-Vanna	**-25.1**	**-18.8**	**-20.8**	**-20.6**	**-18.5**	**-19.1**
	Delta-Vega-Gamma	-24.0	-16.7	-19.0	-20.1	-16.8	-17.8
	Delta-Vega-Vanna	**-25.5**	**-18.8**	**-21.0**	**-21.1**	**-18.4**	**-19.3**
	Delta-Gamma-Vanna	**-25.0**	**-18.5**	**-20.6**	**-20.9**	**-18.3**	**-19.1**
	Delta-Vega-Gamma-Vanna	**-25.2**	-17.9	-20.3	-20.1	**-17.5**	**-18.3**
	Hull-White	**-25.5**	**-19.3**	**-21.3**	**-20.6**	**-17.6**	**-18.5**
	Relaxed Hull-White	**-26.0**	**-19.4**	**-21.5**	**-21.2**	**-17.7**	**-18.8**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	**-25.4**	-16.6	-19.3	-19.9	-15.5	-16.9
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-24.9	-17.6	-19.8	-18.9	-14.9	-16.0
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	**-25.5**	-17.7	-20.2	-19.4	-16.0	-17.1

Table 4.2 Performance of the linear regressions and ANNs on the S&P 500 dataset for which moneyness is between 0.8 and 1.2 and time-to-maturity is greater than 14 calender days. We refer to the caption of Table 3.3 for an explanation.

Euro Stoxx 50

If we only apply the first change, there are 0.60 million samples remaining, and the reduction of number of samples is 3.2% of the 'Baseline' dataset. We apply the same analysis as before. To summarize, the MSHE of zero hedge increases by 1.79%, due to the same reason mentioned before. There is very little change in relative performance of regressions. To save space, we do not report the exact results here either.

After applying both changes, the size of the new data set is 0.41 million , 64% of the 'Baseline' dataset. Table 4.3 shows the performance on the restricted Euro Stoxx 50 data for which moneyness is between 0.8 and 1.2 and time-to-maturity is greater than 14 calender days. Comparing it with Table 3.4, the MSHEs of zero hedge increase for both option types and all hedging periods, except calls in the one-hour hedging. Most regressions and ANNs gain about 4% improvement in relative performance against the BS-Delta on the three different hedging periods. Focusing on the Delta-Vega-Gamma-Vanna regression in the one-day hedging, restricting the data changes the performance from -20.1% to -27.9% for calls, and from -18.0% to -19.6% for puts. The improvement is more significant for one-hour hedging than for other longer hedging periods.

		1 hour			1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	0.361	1.05	0.739	4.33	10.4	7.63	8.86	29.5	20.6
	BS-Delta	0.116	0.208	0.167	1.21	1.85	1.56	3.23	4.92	4.19
Regressions	Delta-only	-27.2	-15.7	-19.3	**-31.0**	-17.3	-22.2	-44.7	-17.9	-26.8
	Vega-only	**-29.2**	-16.0	-20.2	-29.6	-19.7	**-23.3**	-40.7	-19.4	-26.5
	Gamma-only	-22.1	-7.23	-11.9	-28.4	-13.1	-18.5	**-49.6**	-19.8	-29.7
	Vanna-only	-20.1	-6.20	-10.6	-22.8	-15.4	-18.1	-33.5	**-24.0**	-27.1
	Delta-Gamma	-27.1	**-17.9**	-20.8	-29.3	-17.3	-21.6	-41.0	-9.75	-20.1
	Delta-Vega	**-29.5**	-16.6	-20.7	**-30.5**	-19.7	**-23.6**	-44.9	-19.0	-27.6
	Delta-Vanna	-28.5	-15.7	-19.7	**-30.0**	-19.5	**-23.2**	-44.6	**-24.0**	**-30.8**
	Delta-Vega-Gamma	**-29.5**	**-18.0**	**-21.6**	-28.7	**-19.9**	-23.0	-41.2	-14.2	-23.1
	Delta-Vega-Vanna	**-29.7**	-16.6	-20.8	**-30.1**	**-20.8**	**-24.1**	-44.3	-22.2	-29.5
	Delta-Gamma-Vanna	**-28.8**	**-18.6**	**-21.9**	-29.3	-19.7	**-23.1**	-40.0	-14.2	-22.7
	Delta-Vega-Gamma-Vanna	**-28.8**	**-18.6**	**-21.8**	-27.9	-19.6	-22.6	-38.6	-14.6	-22.6
	Hull-White	-27.5	-16.1	-19.7	**-31.0**	-18.9	**-23.2**	-44.6	-20.9	-28.8
	Relaxed Hull-White	-27.7	-16.0	-19.7	**-30.8**	-18.9	**-23.1**	-44.7	-19.3	-27.7
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	-22.8	-17.3	-19.1	-8.91	-3.92	-5.69	-30.8	-10.3	-17.1
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-22.9	-11.0	-14.8	-22.9	-15.9	-18.4	-32.6	-5.48	-14.5
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-26.3	**-18.4**	**-20.9**	-25.6	-9.22	-15.0	-26.3	-13.0	-17.4

Table 4.3 Performance of the linear regressions and ANNs on the Euro Stoxx 50 dataset, for which the moneyness is between 0.8 and 1.2 and time-to-maturity is greater than 14 calender days. We refer to the caption of Table 3.3 for an explanation.

4.5.2 Non-stationary features

As we have discussed in Section 2.2.1, there are two ways regarding how to use underlying price and strike price; one either use them separately, or use the ratio (so-called moneyness). Many papers mention using the ratio can improve the generalisation performance of ANNs, see Section 2.2.1 for a pointer of the literature.

In this subsection, we investigate the performance of ANNs trained on the two non-stationary features, i.e. underlying and strike price. Table 4.4 shows the performance of ANN trained on a feature set consisting of underlying price, strike, and time-proportional implied volatility. The regularisation parameters are shown in Table 4.5, which are tuned with the same approach explained in Section 4.1. Comparing the first row with the ANN (M; $\sigma_{\text{impl}}\sqrt{\tau}$) in Table 3.3, the performances on both hedging periods worsen. In particular on the one-day hedging, the percentage reduction of MSHE against the BS-Delta changes from -17.7% to 24.6% for calls and puts together. The performance in the two-day hedging also worsens, but only by 6% against BS-Delta. For the Euro Stoxx 50, we compare the second row with the ANN (M; $\sigma_{\text{impl}}\sqrt{\tau}$) in Table 3.4. The performances of the ANN changes the most in the one-hour, from -16.3% to -1.25% against the BS-Delta for calls and puts together. It also worsens a bit on the two-day hedging. However, training on non-stationary features even helps on the one-day hedging, by about 2.5% against the BS-Delta. Overall, we conclude that using non-stationary features tends to handicap the ANN training.

	1 hour			1 day			2 days		
	Calls	Puts	Both	Calls	Puts	Both	Calls	Puts	Both
S&P 500	-	-	-	11.5	30.7	24.6	-5.8	-6.52	-6.30
Euro Stoxx 50	-5.98	0.7	-1.25	3.82	-14.0	-7.95	-16.7	5.7	-2.09

Table 4.4 The performance of ANNs trained with a feature set consisting of underlying price, strike price and time-proportional implied volatility. The regularisation parameters used to generate these results are now in Table 4.5.

	1 hour	1 day	2 days
S&P 500	-	1×10^{-6}	1×10^{-3}
Euro Stoxx 50	1×10^{-3}	1×10^{-1}	1×10^{-2}

Table 4.5 The regularisation parameters used to obtain the results in Table 4.4.

Here we mention another trivial way to ‘stabilise’ features instead of using moneyness. That is using the normalised underlying price and normalised strike price (normalised by S_t as in (3.7)). By doing so, the normalised underlying prices stay constant, and the normalised strikes serve the role played by moneyness.

4.5.3 Using the CDF of a standard normal distribution

Inspired by the Black-Scholes formula, the cumulative distribution function (CDF) N of the standard normal distribution can be used for the network's output activation function, instead of a ReLU. This change has two potential benefits. First, it restricts the output to be between 0 and 1. Secondly, the ANN then only needs to learn a function similar to d_1 in (3.4) instead of $\mathrm{N}(d_1)$, potentially making the learning easier.

In this subsection, we investigate if using N as the output activation function can improve the performance of ANN. The 'Baseline' experimental setups are used as in Table 3.3 for S&P 500 dataset and in Table 3.4 for Euro Stoxx 50 dataset. We first train the ANNs with feature set $(M;\ \sigma_{\text{impl}}\sqrt{\tau})$. The CDF activation is aimed to replace the CDF in (3.3), so that the ANN only needs to learn d_1. For the sake of completeness, we also train ANN on the other two feature sets. We use the same regularisation parameters in Table 4.1, since the feature sets and the number of ANN weights do not change.

		1 hour			1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both	Calls	Puts	Both
S&P 500	$M;\ \sigma_{\text{impl}}\sqrt{\tau}$	-	-	-	**-22.7**	-15.3	-17.7	-15.1	-10.4	-11.9
	$\Delta_{\text{BS}};\ \mathcal{V}_{\text{BS}};\ \tau$	-	-	-	**-22.7**	**-16.7**	**-18.6**	-15.1	-10.6	-12.0
	$\Delta_{\text{BS}};\ \mathcal{V}_{\text{BS}};\ \text{Va}_{\text{BS}};\ \tau$	-	-	-	-20.7	-14.2	-16.2	-12.5	-10.7	-11.3
Euro Stoxx 50	$M;\ \sigma_{\text{impl}}\sqrt{\tau}$	-17.1	**-15.7**	-16.1	-2.20	-2.04	-2.09	-25.6	1.62	-7.84
	$\Delta_{\text{BS}};\ \mathcal{V}_{\text{BS}};\ \tau$	-13.4	-4.63	-7.18	-16.3	-3.21	-7.65	-25.4	-0.02	-8.82
	$\Delta_{\text{BS}};\ \mathcal{V}_{\text{BS}};\ \text{Va}_{\text{BS}};\ \tau$	-22.2	**-16.2**	**-18.0**	-14.2	-8.04	-10.1	-35.3	-13.2	-20.9

Table 4.6 The performance of ANNs of which the output activation is the CDF of the standard normal distribution. The bold numbers follows the same rule used in Table 3.3 for S&P 500 dataset and Table 3.4 for Euro Stoxx 50 dataset. The regularisation parameters can be found in Table 4.1.

Table 4.6 shows the performance of ANNs using the CDF of the standard normal distribution on the S&P 500 and Euro Stoxx 50 datasets. We compare the first three rows with Table 3.3. Looking at the call and put together, the ANNs trained with N as the output activation have close to the 'Baseline' performance in the one-day hedging, and slightly worse in the two-day hedging for all the three feature sets. As for Euro Stoxx 50 dataset, using the CDF helps the ANN $(M;\ \sigma_{\text{impl}}\sqrt{\tau})$ and ANN $(\Delta_{\text{BS}};\ \mathcal{V}_{\text{BS}};\ \text{Va}_{\text{BS}};\ \tau)$ a lot for the one-hour hedging. Especially, the ANN $(\Delta_{\text{BS}};\ \mathcal{V}_{\text{BS}};\ \text{Va}_{\text{BS}};\ \tau)$ now outperforms by -18.0% against the BS-Delta, increased from -16.7%. This beats all other regressions. However, the normal CDF does not help for any other scenarios.

4.5.4 Zero volume

Potential information leakage can be introduced during cleaning. One such source is by improperly removing samples for which trading volume is zero. The dataset given to us contains daily close price of options, and there are two different ways to remove samples with zero volume. The first way is what we have followed previously in the 'Baseline'; it first matches trades and then only removes samples for which current volume is zero regardless of the next period. The alternative is to first remove all samples for which the volume is zero, and then match trades to the next period. If the dataset is cleaned in the first way, there exist some samples for which the next period volumes are zero. With the second way, only samples for which volumes of current and next period are both non-zero remain. Since we hedge every option (if today's volume is not zero), we would not know at the time of hedging whether the volume of the corresponding trade tomorrow is zero or not. Hence, taking the second way implicitly remove an extra amount of data that should have been preserved.

By taking the alternative approach, the size of dataset changes from 2.6 million to 2.1 in the one-day hedging, nearly a 19% reduction of the 'Baseline' dataset. Table 4.7 shows the performance of the linear regressions and ANNs on the S&P 500 dataset when the alternative approach is taken to remove zero volume data. Comparing this table to Table 3.3, the MSHE of zero hedge and BS-Delta both increases, and it is more significant for puts than for calls. The MSHE of BS-Delta increases by 6.4% for calls and 18% for puts. Most of the regressions gain a small improvement at around 0.2% against the BS-Delta, and puts benefit more than calls. Taking the Delta-Vega-Vanna regression for instance, the performance changes from -23.0% to -23.2% for calls, and from -16.6% to -16.9% for puts, against the BS-Delta, with overall changing from -18.7% to -18.9%.

The ANN (Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ) improves from -18.9% to -19.3% against the BS-Delta. This improvement is from the enhanced performance on puts, and it is more than that of the Delta-Vega-Vanna regression.

4.5.5 Single window

As we have pointed out in Section 2.2.4, many studies adopt a rolling window approach when working with a large real dataset across many years. The reason is the existence of potential time-inhomogeneity, as mentioned by Hutchinson et al. [1994], Garcia and Gençay [2000], and many others. However, not all research follows this approach. If a rolling window scheme is taken, the numbers of samples in each training and test set are much smaller. This may lead to overfit for complex models such as the ANNs. However, this scheme has an advantage, in that it allows the comparison between time windows, see Garcia and Gençay [2000]. In the above two papers, authors are particularly interested in the difference between 1987 October crash and other years. Intuitively, whether using the rolling window scheme depends on the time length of the data available; it is more likely to use rolling window on a dataset spanning tens of years than on a dataset spanning days. We refer to Table 2.1 for a summary of the time length of datasets investigated in the literature.

In the following, we undertake the single window experiment only on the S&P 500 dataset. In the 'Baseline' setup, the training, validation and test set consist of 720, 180, and 180 calender days, and the window size is 180 days, giving 14 windows. To obtain a single

			1 day			2 days	
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	4.00	5.43	4.97	8.14	11.1	10.1
	BS-Delta	0.731	0.774	0.76	1.72	1.86	1.81
Regressions	Delta-only	-21.7	-15.0	-17.1	-17.6	-12.6	-14.1
	Vega-only	-14.9	-12.6	-13.3	-11.7	-10.6	-10.9
	Gamma-only	-15.6	-10.1	-11.9	-14.8	-11.1	-12.2
	Vanna-only	-12.2	-12.8	-12.6	-10.3	-13.3	-12.4
	Delta-Gamma	-21.9	-14.9	-17.1	-17.9	-12.9	-14.5
	Delta-Vega	-21.8	-15.2	-17.3	-17.6	-12.8	-14.3
	Delta-Vanna	**-22.9**	**-16.8**	**-18.7**	**-18.5**	**-15.5**	**-16.4**
	Delta-Vega-Gamma	-21.8	-15.1	-17.2	-17.6	-13.4	-14.7
	Delta-Vega-Vanna	**-23.2**	**-16.9**	**-18.9**	**-18.8**	**-15.5**	**-16.5**
	Delta-Gamma-Vanna	**-22.8**	**-16.8**	**-18.7**	**-18.4**	**-15.3**	**-16.3**
	Delta-Vega-Gamma-Vanna	**-23.0**	**-16.7**	**-18.7**	-18.1	**-15.0**	**-16.0**
	Hull-White	**-23.4**	**-17.2**	**-19.2**	**-18.7**	-14.5	**-15.8**
	Relaxed Hull-White	**-23.4**	**-17.2**	**-19.2**	**-19.3**	**-14.7**	**-16.1**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	**-23.2**	-15.3	-17.7	-15.0	-8.56	-10.5
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	**-23.2**	**-17.6**	**-19.3**	-16.5	-11.1	-12.8
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-22.0	-14.6	-17.0	-15.4	-11.4	-12.6

Table 4.7 Performance of the linear regressions and ANNs on the S&P 500 dataset on which samples with zero volume are removed before matching, i.e. the alternative approach. We refer to the caption of Table 3.3 for an explanation.

window, the window size is set to 570 days[3], and the ratio remains 4:1:1 for the training, validation and test sets. Hence, they consist of 2280, 570, and 570 days.

Table 4.8 shows the performance of the linear regressions and ANNs on the S&P 500 dataset using single window. It is compared to Table 3.3. Note the out-of-sample set is almost completely different to that in the 'Baseline' setup. It is now the last one-sixth of the entire dataset instead of the last 78%[4]. Hence, there is no need to compare the zero hedge MSHE with that in the 'Baseline'. Using a single window improves the performance for all regressions and hedging periods. The increments are around 2% on the one-day hedging period. Specifically, the Delta-Vega-Vanna regression reduces the MSHE against the BS-Delta by -25.8% for calls and by -18.8% for puts, increased from -23.0% and -16.6% respectively on the one-day hedging. This means that puts benefit more than calls from the single window setup.

[3] There are 9.5 years data available. We take 360 days per year, and round to a window size of 570 days.

[4] Only the first 720 days out of the 9.5 years are not in the out-of-sample dataset in the 'Baseline' setup.

		1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	5.28	6.70	6.25	11.0	13.2	12.5
	BS-Delta	1.10	1.11	1.10	2.37	2.36	2.37
Regressions	Delta-only	-25.2	-17.8	-20.2	-21.0	-15.7	-17.4
	Vega-only	-15.3	-12.9	-13.6	-12.7	-11.2	-11.7
	Gamma-only	-18.9	-12.5	-14.5	-18.8	-14.9	-16.2
	Vanna-only	-13.8	-14.9	-14.5	-12.6	-16.5	-15.2
	Delta-Gamma	-25.4	-17.5	-20.0	-21.9	-15.5	-17.5
	Delta-Vega	-24.3	-17.1	-19.3	-20.3	-14.9	-16.6
	Delta-Vanna	**-26.5**	**-19.7**	**-21.9**	**-22.5**	**-19.1**	**-20.2**
	Delta-Vega-Gamma	-24.7	-17.3	-19.7	-21.5	-16.2	-17.9
	Delta-Vega-Vanna	-25.8	-18.8	-21.0	-22.1	-18.0	**-19.3**
	Delta-Gamma-Vanna	-26.0	-19.4	-21.4	-22.2	-17.8	**-19.2**
	Delta-Vega-Gamma-Vanna	-25.8	-18.8	-21.0	-22.0	-17.7	-19.1
	Hull-White	**-27.0**	**-20.5**	**-22.5**	**-22.7**	-17.8	**-19.4**
	Relaxed Hull-White	**-27.1**	**-20.5**	**-22.6**	**-23.4**	**-18.2**	**-19.8**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	-25.0	-17.7	-20.0	-21.0	-12.0	-14.8
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-23.2	-15.1	-17.6	-16.2	-12.3	-13.5
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-22.8	**-20.3**	-21.1	-16.5	-15.4	-15.7

Table 4.8 Performance of the linear regressions and ANNs on the S&P 500 dataset, which is partitioned into single window. We refer to the caption of Table 3.3 for an explanation.

As a comparison to Table 3.4, we discuss the performance of rolling window scheme on Euro Stoxx 50. The window size is now 90 days instead of 150. Almost all the regressions significantly worsens by about 8% in their relative performance against the BS-Delta. The difference is larger as the hedging period increases. The Vega-only regression becomes the best-performing model instead of the Delta-Vega-Gamma-Vanna regression, because the four-factor regression overfits on the small-size dataset.

Given the fact that using single window improves the performance rather than harming it on both datasets, we conclude that the time-inhomogeneity does not matter in our setup. Instead, statistical hedging methods benefit from including more data in the training set, thus reducing potential over-fitting. One of the reasons that require using rolling window in papers such as Hutchinson et al. [1994], Garcia and Gençay [2000] and others, we conjecture, is that their ANNs do not have any kind of volatility in the feature set. Therefore, training and testing are restricted to data that are relatively close in time to help tackle the changing volatility in the market. Our feature set includes implied volatility, hence making the rolling window scheme less necessary.

4.5.6 Matching tolerance

Contrary to the S&P 500 data, Euro Stoxx 50 is a high frequency tick data set. Therefore, two sources of asynchronism exist. First, futures and options are not traded at the same instant. Secondly, options do not have an exact matching trade one period after. The former issue is relatively mild, since futures, especially short time-to-maturity futures, are extremely liquid; Chen and Sutcliffe [2012] have a similar observation on a different high frequency tick dataset. Hence we ignore this issue. For the latter, a smaller matching tolerance restricts data to liquid options for which data quality is assumed to be better. However, it also reduces the size of dataset, and presumably handicaps the ANN's generalisation.

In the following, we present results on the dataset when the matching tolerance is increased to 30 minutes, and then investigate its implications. After applying this change, the number of samples changes from 0.6 million to 1.4, a 133% increase of the 'Baseline' dataset. Table 4.9 shows the results based on this enlarged dataset. Let us focus on the Delta-Vega-Gamma-Vanna regression. Only one-day hedging benefits from the greater matching tolerance. For other heding periods, this change seems always improve the performance on calls, but less likely on puts, leaving the overall performance relatively unchanged.

However, the ANNs significantly worsen, despite the size of dataset doubles. In terms of the performance on both calls and puts, all ANNs, except the ANN $(M; \sigma_{\text{impl}}\sqrt{\tau})$ in the one-hour hedging, perform worse than the 'Baseline'.

Overall, we observe that doubling the dataset by increasing the matching tolerance does not help the linear regressions much, but clearly handicaps the performance of ANNs.

4.5.7 Drift effect

In this subsection, we investigate the effect of drift parameters on the statistical hedging methods based on the Black-Scholes and Heston datasets. We show the convexity property of option price with respect to underlying, the drift, and the leverage effect together affect the performance of such regression methods.

Black-Scholes dataset

In the 'Baseline' simulation setup, the drift of the geometric Brownian motion is set to 10% annually. The linear regression methods underperform the BS-Delta in the one-day period, but outperform in the two-day period, shown in Table 3.6. We argued that the drift matters for long periods, and regression methods are able to learn the effect that option's Delta increases over the hedging period. To support this argument, we run two extra experiments; in one experiment, we set the drift to 0, and in the other to 50%.

		1 hour			1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	0.433	0.888	0.70	4.07	9.34	7.14	7.55	20.4	15.2
	BS-Delta	0.116	0.20	0.165	1.16	1.74	1.50	2.73	3.64	3.27
Regressions	Delta-only	-22.8	-10.7	-14.2	-21.4	-14.3	-16.6	**-37.1**	-12.8	-21.0
	Vega-only	**-25.8**	-12.6	-16.5	**-22.5**	-16.7	**-18.5**	-36.4	**-17.2**	**-23.7**
	Gamma-only	-7.37	-1.49	-3.20	-15.7	-5.35	-8.68	-34.8	-4.48	-14.7
	Vanna-only	-16.2	-4.49	-7.88	-16.0	-10.9	-12.6	-30.7	-15.8	-20.8
	Delta-Gamma	-22.5	**-13.5**	-16.1	-20.6	-15.2	-16.9	-33.5	-8.46	-16.9
	Delta-Vega	**-26.2**	-12.8	**-16.7**	-22.6	-16.5	**-18.5**	**-38.1**	-16.1	**-23.5**
	Delta-Vanna	-23.7	-10.6	-14.4	-21.0	-15.6	-17.3	-36.7	**-16.6**	**-23.4**
	Delta-Vega-Gamma	-24.7	**-14.1**	**-17.2**	**-21.5**	-16.6	-18.2	-34.4	-13.1	-20.3
	Delta-Vega-Vanna	**-25.8**	-12.8	-16.5	**-21.6**	-17.0	**-18.5**	-36.7	**-17.1**	**-23.7**
	Delta-Gamma-Vanna	**-25.8**	**-14.0**	**-17.5**	**-21.7**	**-18.1**	**-19.3**	-35.0	**-16.9**	**-23.0**
	Delta-Vega-Gamma-Vanna	**-25.9**	**-14.3**	**-17.7**	**-21.8**	**-18.1**	**-19.3**	-34.7	**-17.1**	**-23.0**
	Hull-White	-22.4	-11.9	-14.9	-21.1	-15.4	-17.2	-36.9	-14.4	-22.0
	Hull-White-relaxed	-22.5	-11.9	-15.0	-21.1	-15.4	-17.2	-36.9	-13.9	-21.7
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	-23.7	**-14.0**	**-16.8**	-3.78	-1.36	-2.14	-23.9	14.8	1.76
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-16.7	-0.88	-5.49	-16.3	-5.15	-8.74	-22.9	-1.05	-8.45
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va$_{\text{BS}}$; τ	-22.1	-12.2	-15.1	-14.8	-10.5	-11.9	-16.1	-4.11	-8.17

Table 4.9 Performance of the linear regressions and ANNs on the Euro Stoxx 50 data set, using a matching tolerance of 30 minutes. We refer to the caption of Table 3.3 for an explanation.

Table 4.10 shows the performance on the Black-Scholes dataset simulated with zero drift. The linear regression hedging methods underperform the BS-Delta in the one-day period, but outperform in the two-day period. Why do the statistical regression methods still outperform when there is no drift to be learned? Remember we only use out-of-money options. The Delta for out-of-money calls (puts) is an increasing and convex (concave) function of the underlying price. For the calls (puts), the Delta increases more (less) as the underlying increases than it decreases as the underlying decreases. This implies the Delta coefficient needs to be larger than one for calls and puts (puts have negative Delta) for the Delta-only regression to reduce MSHE. Indeed, the Delta coefficient is still larger than one for calls in the Delta-only regression. This effect is easier to capture in the two-day hedging period. Hence, the outperformance of linear regression is greater in the two-day than in the one-day hedging. However, we find the Delta coefficient for puts is less than one. This also explains why the regression methods underperforms on the puts.

Table 4.11 shows the performance on the Black-Scholes dataset simulated with a drift of 50% annually. All statistical hedging methods outperform the BS-Delta. The outperformance is more significant in the two-day hedging than in the one-day. Among all statistical hedging methods, the Gamma-only regression stays the best for both hedging periods and both option types. Gamma captures the sensitivity of Delta with respect to underlying price change. Hence, it is more useful than other sensitivities in capturing the increasing Delta effect, caused by the large drift. The performance is much better compared to Table 4.10 and Table 3.6, where the drifts are 0% and 10%, respectively. Taking Gamma-only regression in the one-day hedging for example, the performance are 0.32%, 0.12%, and -5.92% when the drifts equal to 0%, 10%, and 50% respectively. Compared to the other two configurations, the Delta coefficient is even greater than one for calls and less than one for puts. The effect caused by the large drift overwhelms the the convexity (concavity) of out-of-money call (put) options.

		1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	22.9	14.9	17.4	45.6	28.9	34.0
	BS-Delta	0.145	**0.116**	**0.121**	0.602	0.414	0.468
Regressions	Delta-only	**-0.43**	1.55	0.59	-2.03	0.13	**-0.53**
	Gamma-only	**-0.32**	1.84	0.81	**-2.36**	0.16	**-0.43**
	Vega-only	**-0.51**	1.12	0.32	-1.95	0.04	**-0.54**
	Vanna-only	**-0.82**	0.56	0	**-2.6**	**-0.02**	**-0.24**
	Delta-Gamma	**-0.37**	1.82	0.77	**-2.36**	0.15	**-0.47**
	Delta-Vega	**-0.07**	1.74	0.88	-1.86	0.4	**-0.51**
	Delta-Vanna	**-0.39**	1.53	0.59	**-2.73**	0.26	**-0.44**
	Delta-Vega-Gamma	**-0.16**	1.86	0.88	**-2.53**	0.38	**-0.45**
	Delta-Vega-Vanna	**-0.22**	1.77	0.84	**-2.44**	0.41	**-0.27**
	Delta-Gamma-Vanna	**-0.05**	1.91	0.93	**-2.61**	0.58	**-0.43**
	Delta-Vega-Gamma-Vanna	**-0.09**	1.9	0.92	**-2.56**	0.59	**-0.4**
	Hull-White	**-0.47**	1.54	0.58	**-2.7**	0.3	**-0.35**
	Relaxed Hull-White	**-0.46**	1.54	0.58	**-2.68**	0.3	**-0.34**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	4.22	2.73	3.94	1.49	1.98	3.06
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	3.61	2.71	3.15	-1.84	8.6	5.66
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	4.60	3.22	3.72	-0.69	6.96	5.32

Table 4.10 Performance of the linear regressions and ANNs on the Black-Scholes dataset simulated with a drift of zero. See the caption of Table 3.3 for further explanations.

When investigating the performance on datasets simulated with various drift, we are aware that the number of samples changes as the drift changes due to the CBOE rules. As pointed out in Section 3.2.5, the number of samples in the in-sample set is 360k. It decreases to 315k when the drift is zero, and increases to 611k when the drift is 50%. The significantly larger in-sample size might also help the regression methods when the drift is 50%, and we need to rule out this possibility. Hence, we conduct another experiment, in which the in-sample dataset includes 900 days, twice the ‘Baseline’ number, and the drift is 10%. This gives

		1 day			2 days		
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	36.2	8.23	10.8	78.9	15.6	21.1
	BS-Delta	0.202	0.062	0.076	0.992	0.243	0.308
Regressions	Delta-only	-6.18	-3.84	-4.26	-14.1	-10.6	-11.0
	Gamma-only	**-7.42**	**-5.74**	**-5.92**	**-20.0**	**-14.4**	**-14.7**
	Vega-only	-4.3	-2.84	-3.14	-9.64	-8.01	-8.17
	Vanna-only	-4.8	-2.6	-3.21	-12.5	-6.67	-7.91
	Delta-Gamma	**-7.51**	**-5.74**	**-5.94**	**-19.7**	**-14.4**	**-14.6**
	Delta-Vega	**-7.45**	-4.23	-4.73	-17.5	-11.5	-12.3
	Delta-Vanna	-6.48	-4.17	-4.64	-15.1	-11.3	-11.9
	Delta-Vega-Gamma	**-7.62**	**-5.74**	**-5.95**	**-19.5**	**-14.4**	**-14.5**
	Delta-Vega-Vanna	**-7.88**	-5.02	**-5.51**	-18.8	-13.3	-13.9
	Delta-Gamma-Vanna	**-7.44**	**-5.74**	**-5.91**	**-19.7**	**-14.4**	**-14.6**
	Delta-Vega-Gamma-Vanna	**-7.59**	**-5.74**	**-5.95**	**-19.7**	**-14.4**	**-14.6**
	Hull-White	-6.8	-4.07	-4.66	-16.2	-11.1	-12.0
	Relaxed Hull-White	-6.8	-4.08	-4.66	-16.2	-11.2	-12.0
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	2.67	-2.43	-1.28	-10.0	-11.5	-9.58
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-6.74	**-5.58**	**-5.74**	-10.1	-0.62	-1.91
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-2.73	-3.74	-2.79	-3.48	-12.1	-7.87

Table 4.11 Performance of the linear regressions and ANNs on the Black-Scholes dataset simulated with a drift of 50% annually. See the caption of Table 3.3 for further explanations.

a similar number of samples in the in-sample set as the one simulated with a drift of 50%. The performance of the statistical regression methods is close to Table 3.6 (corresponding to the drift equal to 10%). Hence, we rule out the possibility that the significantly improved performance in Table 4.11 is caused by the increased in-sample size.

Heston dataset

Parameters used for simulation of the Heston model as in Section 3.2.1 are chosen after some investigation on the scale of MSHE of BS-Delta. In principle, we set initial and long-term variance to the volatility used in the Black-Scholes simulation for comparison. Small mean-reverting rate κ or larger volatility of variance σ increases the MSHE. A more negative correlation makes the Heston data different from the Black-Scholes data and close to real data, showing more leverage effect.

In this subsection, we present the performance of statistical hedging models on an extra configuration of Heston dataset, with the drift of the underlying stock being 10%. The performance of the linear regressions and ANNs is shown in Table 4.12. The observations given in Section 3.5.4 still hold in this configuration, although most of the statistical regressions, except the ANNs, worsen by roughly 1% compared to the corresponding BS-Delta. This can be explained by the effect of non-zero drift. As we mentioned before, a non-zero drift increases the option Delta; for calls (puts), it increases (decreases) the coefficient of Delta in Delta-only regression from 1. This effect cancels part of the leverage effect, and makes the statistical regression methods less outperforming. Indeed, the Delta coefficients for call and put are 0.97 and 1.02 in the one-day hedging, compared to 0.97 and 1.03 in the 'Baseline' experiment.

			1 day			2 days	
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	27.1	12.2	15.7	58.5	26.0	32.6
	BS-Delta	0.693	0.423	0.481	1.86	1.14	1.27
Regressions	Delta-only	**-2.34**	-3.86	**-3.66**	**-0.22**	-3.16	**-1.76**
	Gamma-only	-1.79	-3.62	**-3.41**	-0.01	-3.08	-1.32
	Vega-only	-1.87	-3.1	-2.99	**-0.31**	-2.68	**-1.66**
	Vanna-only	-1.60	-2.60	-2.65	**-0.34**	-2.54	-1.47
	Delta-Gamma	**-2.39**	-3.94	**-3.80**	**-0.22**	-3.02	-1.47
	Delta-Vega	**-2.12**	-3.78	**-3.48**	**-0.18**	-2.16	-1.01
	Delta-Vanna	**-2.14**	**-4.19**	**-3.81**	**-0.34**	-3.60	**-1.99**
	Delta-Vega-Gamma	-1.87	-3.87	**-3.56**	**-0.19**	-2.44	-1.10
	Delta-Vega-Vanna	**-2.09**	**-4.32**	**-3.83**	**-0.19**	-2.82	-1.28
	Delta-Gamma-Vanna	**-2.07**	-3.89	**-3.57**	**-0.52**	-2.74	-1.42
	Delta-Vega-Gamma-Vanna	**-2.21**	**-4.18**	**-3.77**	**-0.34**	-2.74	-1.29
	Hull-White	**-2.23**	**-4.11**	**-3.83**	**-0.16**	-3.48	**-1.82**
	Relaxed Hull-White	**-2.20**	**-4.11**	**-3.81**	**-0.13**	-3.47	**-1.81**
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	4.96	**-4.54**	1.10	3.85	**-4.88**	1.44
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	-1.0	-3.32	-2.55	0.89	-2.0	0.86
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-0.16	-3.66	-2.46	2.51	3.2	2.86

Table 4.12 Performance of the linear regressions and ANNs on the Heston dataset with the drift being 10%. See the caption of Table 3.3 for further explanations.

4.5.8 Correlation

This subsection investigates the sensitivity of hedging performance with respect to the correlation ρ between the two driving Brownian motions in the Heston model. We set $\rho = -1$, the maximum negative correlation.

Table 4.13 shows the performance of statistical hedging methods on the Heston dataset simulated with $\rho = -1$. Using BS-Delta instead of not hedging at all now reduces the MSHE

			1 day			2 days	
		Calls	Puts	Both	Calls	Puts	Both
	Zero hedge	21.4	18.7	17.9	44.3	42.5	38.0
	BS-Delta	0.249	0.303	0.246	0.872	1.22	0.91
Regressions	Delta-only	-21.7	-29.4	-25.9	**-8.57**	-20.9	-13.9
	Gamma-only	-18.7	-27.0	-23.8	-6.97	-20.1	-12.7
	Vega-only	-17.9	-23.4	-21.2	-6.98	-17.1	-11.5
	Vanna-only	-15.6	-17.7	-18.0	-4.85	-13.3	-8.79
	Delta-Gamma	-22.6	-30.7	-27.2	**-8.49**	-21.9	**-14.2**
	Delta-Vega	-22.4	-30.0	-26.4	**-8.76**	-19.8	-13.7
	Delta-Vanna	-22.7	-31.6	-27.8	-7.40	-22.3	-13.9
	Delta-Vega-Gamma	-22.7	-30.6	-27.1	**-7.98**	-20.6	-13.4
	Delta-Vega-Vanna	-24.6	-33.1	**-29.6**	**-8.06**	-22.1	**-14.3**
	Delta-Gamma-Vanna	-22.9	-31.2	-28.0	-7.46	-21.2	-13.5
	Delta-Vega-Gamma-Vanna	**-25.3**	-33.1	**-29.8**	**-8.62**	-22.1	**-14.5**
	Hull-White	-23.0	-31.3	-27.8	-7.4	-22.1	-13.8
	Relaxed Hull-White	-23.0	-31.4	-27.8	-7.42	-22.1	-13.8
ANNs	M; $\sigma_{\text{impl}}\sqrt{\tau}$	-18.3	**-34.9**	-25.7	1.95	**-25.8**	-8.81
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; τ	**-24.5**	**-34.6**	**-29.9**	-6.18	-24.6	-12.9
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}; τ	-19.2	**-35.1**	-27.8	-5.01	-9.35	-7.50

Table 4.13 Performance of the linear regressions and ANNs on the Heston dataset with the correlation being -1. See the caption of Table 3.3 for further explanations.

by almost $1 - 0.246/17.9 \approx 99\%$ (98%) for one-day (two-day) hedging, as opposed to 97% (96%) before. These two percentages are almost same as what the BS-Delta can achieve in the Black-Scholes dataset, i.e. 99%. It indicates that an option simulated in the Heston model with $\rho = -1$ can almost be hedged by trading in the underlying only, as if it were in the Black-Scholes world. As expected, the regressions significantly improve against the BS-Delta. The Delta-Vega-Vanna regression now reduces the MSHE against the BS-Delta by 29.6% and 14.3%, compared to 5.03% and 2.46% in Table 3.7. The higher (absolute) correlation implies that more volatility risk can be hedged with the underlying asset. Indeed, the coefficients of Delta-only regression in the one-day hedging are now 0.96 and 1.06 for calls and puts respectively, as opposed to 0.97 and 1.03. The ANNs do not have a clear advantage over the linear models.

We notice that the regression hedging in two-day performs much worse against the BS-Delta than in one-day. It changes from -29.6% to -14.3%, more than 50% decrease, for Delta-Vega-Vanna regression. This is in contrast to the observation for the Black-Scholes dataset; in all of Tables 3.6, 4.10 and 4.11, the regressions outperform the BS-Delta more in the two-day than one-day hedging. Why is so? Our conjecture is that the mean-reverting property of the Heston model pulling the volatility to its mean is stronger in the long period than in the short period. Hence, the drift cancels part of the leverage, albeit the strong negative correlation between the two Brownian motions. To verify such a conjecture, we take yet another experiment, where the mean-reverting rate is set to $\kappa = 1.5$. With this configuration, the Delta-only regression reduces the MSHE of BS-Delta by 69.2% and 56.6% for one- and two-day hedging, when considering calls and puts together. Within our expectation, the difference between these two numbers are smaller. Moreover, these two numbers increase by a factor of two as opposed to those when $\kappa = 5$. This further supports our argument that a strong κ cancels the leverage to some extent.

Chapter 5

Hedging in Multiple Periods

5.1 Introduction

Following Chapter 3, we extend the minimum variance hedging in daily options' trading from the single period setup to multiple periods. We consider an investor who shorts a European option. Given the price of the option each day, she wants to dynamically rebalance her replication portfolio so that the portfolio value matches the payoff of the option.

Assume there are three assets in the market; they are options, the underlying, and the risk-free asset. Today, the investor sells the option at the market price p_0. To hedge, she buys δ_0 shares of the underlying at price S_0, and puts the rest $p_0 - \delta_0 S_0$ into risk-free asset earning an overnight rate r_{onr}. We denote the value of the portfolio at time t by u_t, and it holds that $p_0 = u_0$. At the next day, her portfolio value u_{t+1} is given by

$$u_{t+1} = (1 + r_{\text{onr}}\Delta t)u_t + \delta_t \left[S_{t+1} - (1 + r_{\text{onr}}\Delta t)S_t\right], \tag{5.1}$$

for $t = 0$, and $\Delta t = 1/253$, since the underlying price changes. The investor adjusts δ_t for $t = 1, ..., T/\Delta t$ until the option expires at maturity T. Denote the payoff function by Φ, her goal is to choose δ_t for $t = 1, ..., T/\Delta t$ in such a way that $\mathbb{E}\left[u_T - \Phi(S_T)\right]^2$ is minimised.

The quantity $|u_T - \Phi(S_T)|$ is called the *tracking error* by Hutchinson et al. [1994]. It has been used as a performance measure to evaluate the hedging of ANNs trained to learn the pricing function in literature; we refer to Chapter 2 where performance measures are discussed. This measure differs to the *hedging error* in that it evaluates hedging error over the lifetime of the option.

In the multiple period hedging, a recurrent neural network (RNN) is used to learn the strategy $\{\delta_t\}$, by minimising the mean squared tracking error (MSTE). Put it simply, a set of fully-connected layers are used to determine the hedging strategy $\{\delta_t : t = 0, ..\}$ at each time step, and then a recurrent layer passes the strategy through time.

We investigate the performance of the RNNs on several feature sets in the multiple period hedging. As in Chapter 3, we also introduce the recurrent linear regression on sensitivities and compare them with the RNNs. Again, no significant difference between the linear regressions and RNNs is found. As an additional check, we first fit the statistical hedging methods as in the single period hedging situation, and then apply to the multiple periods over the lifetime of the option. This approach again yields similar performance for the statistical hedging methods.

This chapter is organised as follows. Section 5.2 explains the RNN architecture in details. Section 5.3 explains the data simulation process, feature engineering, and the experimental setup. Section 5.4 presents the results. Section 5.5 investigate the sensitivity of RNN performance on the number of simulation paths and size of time steps. Section 5.6 concludes.

5.2 Network architecture and model ensemble

A short description of the RNN is given in Section 1.2. Unlike a simple fully-connected neural network that is suitable for learning a stationary function, the RNN is specialized for learning the behaviour from time sequences.

We show our designed network architecture in Figure 5.1. The initial state u_0 of the RNN is given as the theoretical option price p_0 at $t = 0$, depending on the model of simulation being used. The RNN needs to estimate the hedging strategy δ_t for all t based on information available up to time t. As mentioned in Section 1.2, the RNN consists of a fully-connected part with weights $\widehat{w}$ and a recurrent part with weights $\widetilde{w}$. Here, both $\widehat{w}$ and $\widetilde{w}$ denotes the collections of weights instead of individual weights. In the figure, we denote by H the fully-connected part of the network. The weights $\widehat{w}$ of H are trainable, and the output $H(x_t|\widehat{w})$ is the hedging strategy δ_t. The recurrent layer O simply implements (5.1) to output u_{t+1}, and its weights $\widetilde{w}$ are fixed.[1] The output u_t, i.e. hidden state, is given to the next step to rebalance the portfolio.

We summarise the architecture of Figure 5.1 by the following equations,

$$\begin{aligned} u_0 &= p_0, \\ \delta_t &= H(x_t|\widehat{w}), && \text{(5.2a)} \\ u_{t+1} &= (1 + r_{\text{onr}}\Delta t)u_t + \delta_t \left[S_{t+1} - (1 + r_{\text{onr}}\Delta t)S_t\right], && \text{(5.2b)} \end{aligned}$$

for $t \in \{0, 1, ..., T-1\}$.

[1] To calculate u_{t+1}, the recurrent layer needs also S_{t+1}, apart from δ_t and u_t. However, they are provided at each step only to calculate u_{t+1}, but are not involved in calculating δ_t. We emphasize this point by separating the two input nodes in Figure 5.1.

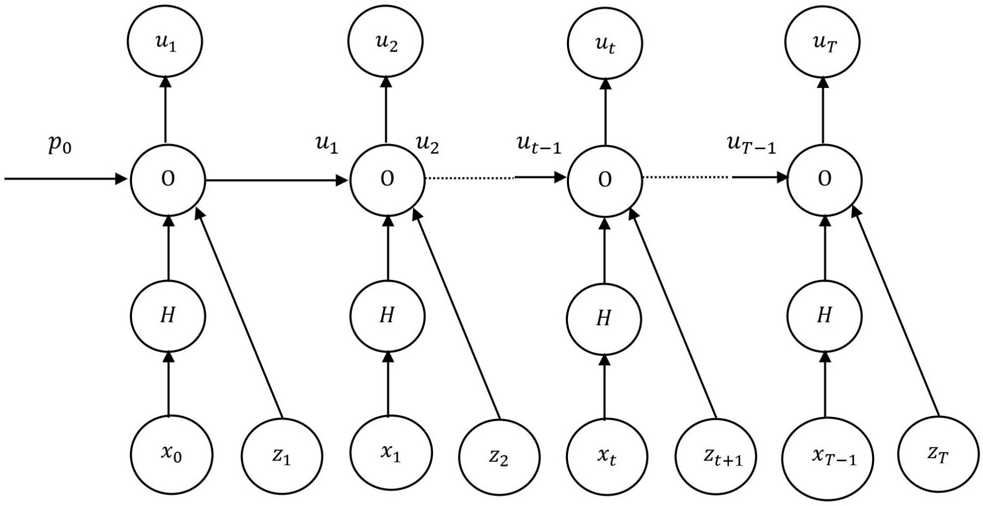


Fig. 5.1 A schematic structure of the RNN model used for hedging in the multiple periods. Here, the notation x_i denote the input for the fully-connected part H outputting the hedging strategy δ_t for $t \in \{0, 1, ..., T-1\}$. The recurrent layer O rebalances the portfolio, and it needs extra input z_t to evaluate the portfolio value at the next time step. The input z_t includes S_t and S_{t+1}.

To train the RNN, we minimise the mean squared difference between the final output u_T and the payoff $\Phi(S_T)$ of the option, i.e. minimising the mean squared tracking error. Hence, the optimisation is given by

$$\arg\min_{\widehat{w}} L(\widehat{w}) = \arg\min_{\widehat{w}} \frac{1}{N} \sum_{i=1}^{N} \left(u_{T,i} - \Phi(S_{T,i})\right)^2, \tag{5.3}$$

where N is the number of simulated paths, $u_{T,i}$ and $\Phi(S_{T,i})$ are the terminal output of the RNN and the payoff of the option on path i, respectively.

The RNN we will use for our experiments has two fully-connected layers, and each layer has 100 hidden nodes. The hidden activation is again the ReLU. The output node represents the hedging strategy, and it needs to be between 0 and 1 for calls. Hence, we choose the output activation to be a *sigmoid* (so-called logistic) function, given by $\frac{e^x}{e^x+1}$. The weights of the RNN are initialised via the 'Xavier' initialiser, and the 'Adam' optimiser is applied for training, the same as in Section 3.3.1. Hyperparameters are all set to default values for this optimizer, with the learning rate being 0.001. We point out here that there are three differences for the RNN compared to the network used in the single period hedging; they are the number of nodes, the output activation function, and the learning rate. We now justify our choices. The loss curve of the RNN during training behaves very differently to the feed-forward network used in the single period hedging. We observe that the loss curve decreases more slowly than that in the single period hedging experiment. Hence, it allows us to use a larger network and higher learning rate without overfitting on the dataset. The use of sigmoid activation function improves the generalisation performance in the case that the training terminates before reaching the minimum of the loss curve. For the same reasons, we increase the number of training epochs to 1000.

In practice, one reliable approach to improve the generalisation performance of ANNs is to train the ANN multiple times independently with random initialisation of weights, and at test time average these predictions. As the number of training increases, the performance usually improves. This technique is called ‘model ensemble’. In the following, we shall use this ensemble approach for all the following experiments to improve the performance, and show its benefits as a sensitivity check in Section 5.5.

5.3 Data, features, and experimental setup

Two thousand underlying paths are simulated under the Black-Scholes model, with the drift being 10%, the volatility 20%, and the initial stock price 2000. We investigate the hedging of an at-the-money option with time-to-maturity equal to 3 months. The hedging periods are one day and two days. Additionally sensitivity analysis on the number of paths and hedging periods are provided in Section 5.5. We train the RNN ten times and average these predictions. The tracking error is then calculated with this average prediction for the evaluation of performance.

In Chapter 3, we have shown that the linear regressions have comparable performance to ANNs if they are trained with option Greeks as features. Here, we also investigate these linear competing models on the same feature sets.

An additional question to ask is: How does the strategy learned in the single period work in the multiple periods? Consider the Delta-only regression. We first fit this regression as we have done in Chapter 3. To do this, we need not only the payoffs at expiry, but also option prices at all intermediate steps calculated by the Black-Scholes formula. After we have fitted the Delta-only regression and obtained the coefficient, this regression is applied iteratively on the time series to produce δ_t for all t with the fixed coefficient. We denote this strategy by Strat-S, and the one learned in the multiple periods by Strat-M. We will compare these two strategies.

We use an ANN of two hidden layers to learn Strat-S with 30 nodes per layer, same as used in Chapter 3. As for Strat-M, we use an RNN of two hidden layers, but 100 nodes per layer. The reason for this increased network size has been given in Section 5.2.

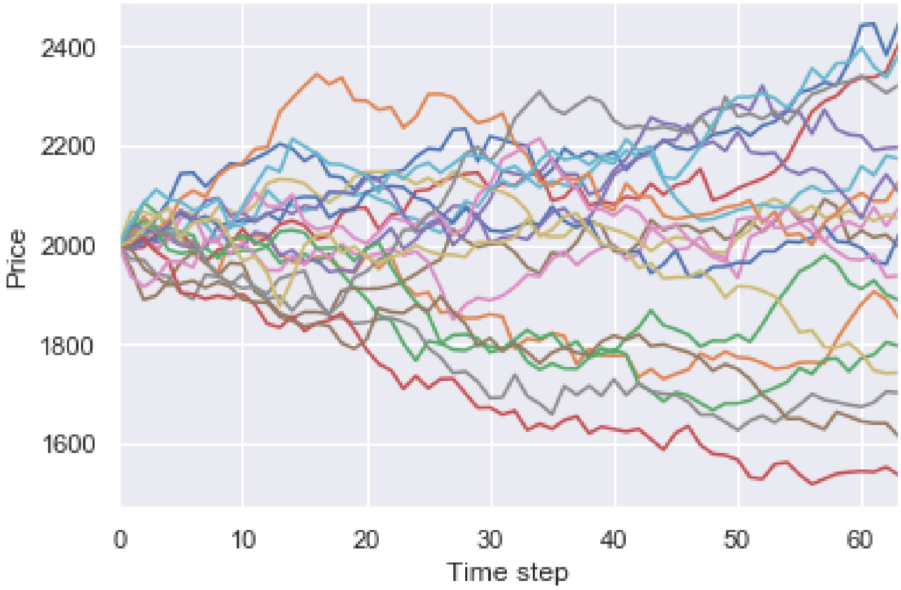


Fig. 5.2 Twenty underlying paths simulated under the Black-Scholes model, with the drift being 10%, the volatility 20%, and the initial stock price 2000. Each unit of the time steps represents one day.

5.4 Results

Figure 5.3 shows the hedging strategies given by the trained RNN on feature $(M;\ \sigma_{\text{impl}}\sqrt{\tau})$ and the Black-Scholes, and Figure 5.4 shows their difference. We observe little difference between the two strategies. The RNN tends to overhedge when $M < 1$, and underhedge when $M > 1$. The most significant difference arises when M is around 1.15.

When evaluating the performance of statistical hedging methods, we use the mean absolute tracking error (MATE) instead of squared error. This is because the absolute tracking error is more widely used in the literature as a performance measure when considering hedging in the multiple periods, indicated by Table 2.1, whereas the mean squared error is a better choice for the loss function during the training.

The relative performance of strategies are defined as the reduction of MATE against the BS-Delta, given by

$$\frac{\text{MATE}(\delta_*) - \text{MATE}(\delta_{\text{BS}})}{\text{MATE}(\delta_{\text{BS}})}. \tag{5.4}$$

Here, δ_* denotes various statistical hedging strategies fitted in either the single- or multiple-periods.

Table 5.1 shows the performance of various hedging methods. The price of the at-the-money call is 79.8. The zero hedge MATEs of one-day and two-day are 110, about 1.25

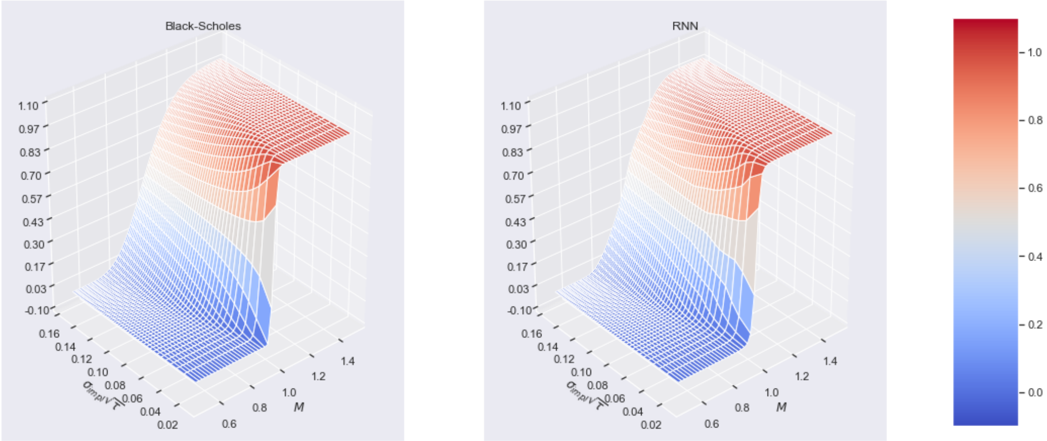


Fig. 5.3 The hedging strategies given by the Black-Scholes benchmark and the RNN trained on the feature set $(M;\ \sigma_{\text{impl}}\sqrt{\tau})$. The RNN strategy is the average of ten iterations.

times the option price.[2] We first compare the strategies fitted in the multiple periods. The BS-Delta reduces the MATE of zero hedge by more than 94% for one-day hedging, and 92% for two-day hedging. Most of the regression methods reduce the MATE compared to the BS-Delta, with the RNN $(M;\ \sigma_{\text{impl}}\sqrt{\tau})$ being the best for one day and Delta-only regression for two days. Regression methods outperform the BS-Delta more on the two-day hedging than on the one-day overall. This observation coincides with that in the single period hedging; see Table 3.6 for the results on the simulated Black-Scholes data. However, the advantage of regression methods is less significant here than in the single period hedging. The coefficient of Delta in the Delta-only is 1.002 and 1.004 on the one-day and two-day hedging periods respectively. The greater deviation of the Delta-only coefficient from one also explains why the Delta-only regression works better on the two-day hedging.

In the following, we investigate the performance of the family ‘Strat-S’. Remember that they are fitted in the single period, and then evaluated in the multiple periods. Their results are shown in the columns ‘Strat-S’ of Table 5.1. Two overall observations still hold; most regressions outperform the BS-Delta, and their outperformance are more significant on the two-day than on the one-day period. Among the Strat-S, the Delta-only regression performs the best on both the one-day and two-day period. Comparing the family of Strat-S and Strat-M, we see no dominance of one family over the other.

[2] The MATEs of zero hedging in one-day and two-day are almost the same. The investor simply holds the option up to the maturity, hence the MATE depends only on the prices of the underlying at maturity, regardless of hedging frequencies. However, a minor difference exist since the last day of two-day hedging does not always coincides with the maturity date.

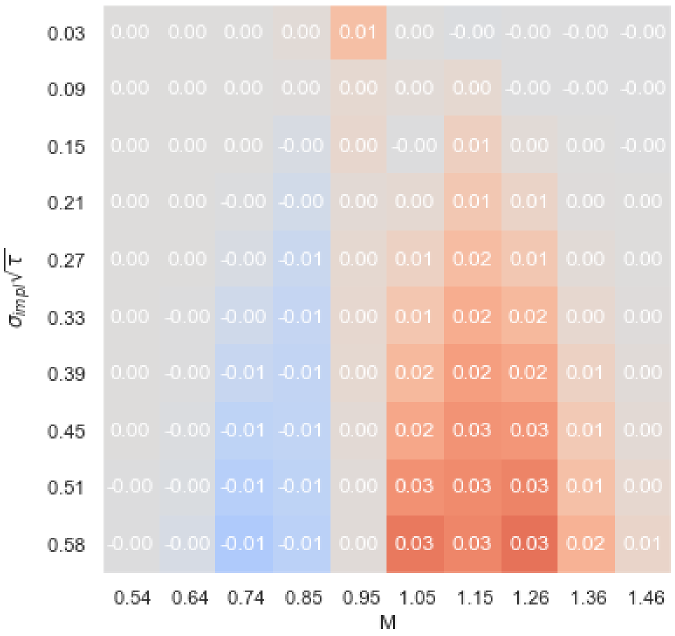


Fig. 5.4 The difference of strategies between the Black-Scholes and the RNN$(M;\ \sigma_{\text{impl}}\sqrt{\tau})$ shown in Figure 5.3. The original difference was plotted on a 50×50 grid like Figure 5.3. We then average each non-overlapping 5×5 box to obtain each tile in this figure. The numbers on the axes show the averages of moneyness and time-to-maturity over 5 consecutive intervals, respectively.

5.5 Sensitivity on simulation parameters

The number of paths and the size of time steps need to be chosen when simulating underlying paths for the at-the-money option to be hedged. Ideally, the tracking error decreases as the size of time step decreases, since a finer time step allows the hedging to approximate to continuous hedging that leads to zero tracking error. Here we investigate the sensitivity of RNN performance under different number of paths and time steps.

As before, the strike and the time-to-maturity of the at-the-money option are 2000 and 0.25. We compare three different numbers of paths, $N \in \{1000, 2000, 5000\}$, and three different sizes of time steps, $\Delta t \in \{0.5/253, 1/253, 2/253\}$ equivalent to half-day, one-day, and two-day hedging. We notice that the number of weight updates depends on the size of time step, since the training of the RNN depends on the length of sequences. To enable a fair comparison, we adjust the number of training epochs according to hedging periods. Specifically, we set the number of epochs to 500 for half-day, 1000 for one-day, and 2000 for two-day hedging. However, there are still minor differences for the training of RNN. For instance, we fix the batch size for the 'Adam' optimiser, hence the number of paths affects the number of batches as well as the number of weight update. We recognize this

		1 day		2 days	
		Strat-M	Strat-S	Strat-M	Strat-S
	Zero hedge	110		110	
	BS-Delta	6.52		9.22	
Regressions	Delta-only	**-0.12**	**-0.15**	**-0.36**	**-0.42**
	Vega-only	0.0	**-0.04**	0.08	**-0.26**
	Gamma-only	**-0.03**	0.04	**-0.25**	**-0.27**
	Vanna-only	0.04	0.06	0.14	0.13
	Delta-Gamma	0.01	0.10	**-0.26**	**-0.16**
	Delta-Vega	**-0.09**	**-0.08**	**-0.30**	**-0.31**
	Delta-Vanna	**-0.06**	**-0.12**	**-0.12**	**-0.17**
	Delta-Vega-Gamma	**-0.10**	0.08	**-0.31**	**-0.17**
	Delta-Vega-Vanna	**-0.07**	**-0.10**	0.17	**-0.19**
	Delta-Gamma-Vanna	**-0.07**	0.08	**-0.21**	**-0.15**
	Delta-Vega-Gamma-Vanna	**-0.08**	0.05	0.14	**-0.15**
	Hull-White	-0.01	-0.02	**-0.16**	-0.24
Networks	M; $\sigma_{\text{impl}}\sqrt{\tau}$	**-0.42**	1.56	**-0.09**	4.24
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$	1.30	2.94	0.05	1.59
	Δ_{BS}; $\mathcal{V}_{\text{BS}}$; Va_{BS}	3.90	0.59	1.51	0.74

Table 5.1 The mean absolute tracking errors given by linear regression methods and RNNs on different feature sets. The top two rows show the absolute values of MATE for zero-hedge and BS-Delta, and those below show the percentage change relative to 'BS-Delta'. Numbers in bold represent the best along with the ones within 0.5% of the best, if it outperforms.

defect. However, the intention is to investigate the effect of discretisation and simulation paths on the performance, rather than finding the setting that produces the best results given the same computation resources. We halve the number of ensemble to 5 due to computational constraint.

Figure 5.5 shows the mean and standard deviation of absolute tracking errors of the BS-Delta and the RNN (M; $\sigma_{\text{impl}}\sqrt{\tau}$) under different simulation setups. Fixing the number of paths and changing the size of time step, it holds for both the BS-Delta and RNN that the mean and standard deviation of the absolute tracking error decrease when the size of time step decreases. Fixing the size of time step, the increased number of paths helps the RNN significantly but not much for the BS-Delta. It is not a surprise, since the BS-Delta is not affected by the number of paths while the RNN depends on the amount of available training data. When the number of paths is small ($N = 1000$), the RNN significantly underperforms the BS-Delta, especially when the hedging period is short, i.e. half-day hedging. The RNN improves, when hedging over long period (on the two-day period). However, when $N \geq 2000$, the relative performance of BS-Delta and RNN are not affected by the size of hedging periods. The RNN (M; $\sigma_{\text{impl}}\sqrt{\tau}$) outperforms the BS-Delta for the following combinations: a) $N = 2000$, $\Delta t = 2/253$, b) $N = 4000$, $\Delta t = 1/253$, and c) $N = 4000$, $\Delta t = 2/253$. This is a clear indication that more underlying paths and longer hedging periods can improve the performance of RNN.

Figure 5.6 shows the difference of strategies between the RNN (M; $\sigma_{\text{impl}}\sqrt{\tau}$) and BS-Delta under the nine different simulation setups. Overall, the absolute differences between the two strategies are smaller when $N = 2000$ and $\Delta t = 1/253$, indicated by the light colour; that is, when the RNN performs close to the BS-Delta. We can see great difference of the two strategies if we look at the upper left panel and the bottom right panel of Figure 5.6. They correspond to the simulation setups when the RNN underperforms and outperforms the BS-Delta the most. Although the RNN produces a similar strategy as the BS-Delta when the option is away from money and close to maturity, it is actually not so much trained on these regions, see the distribution of the simulated paths in Figure 5.2. The success of RNNs on these regions is more credited to the *sigmoid* activation function as the output node.

5.6 Conclusion

In this chapter, we apply the recurrent neural network to hedge an at-the-money call under the Black-Scholes model. We show that the RNN can successfully learn to hedge the option in the multiple periods, by minimising the difference between the option payoff and the value of replicating portfolio. However, we find that simple linear regressions on sensitivities can give comparable results. This observation coincides with that in the single period. As a second comparison, we investigate the performance of strategies that are learned in the single period and then applied to the multiple periods. These two families of hedging strategies also give similar results. Hence, we find neither a clear advantage of using RNN in the multiple periods hedging over using linear regression, nor a clear advantage of doing multi-period hedging over single-period. However, we need to point out the observations here are restricted to the simplified setup where the data are simulated in the Black-Scholes model without market frictions. Future research to include market frictions or use other data-generating models could potentially give an advantage to either neural networks or multiple-period hedging.

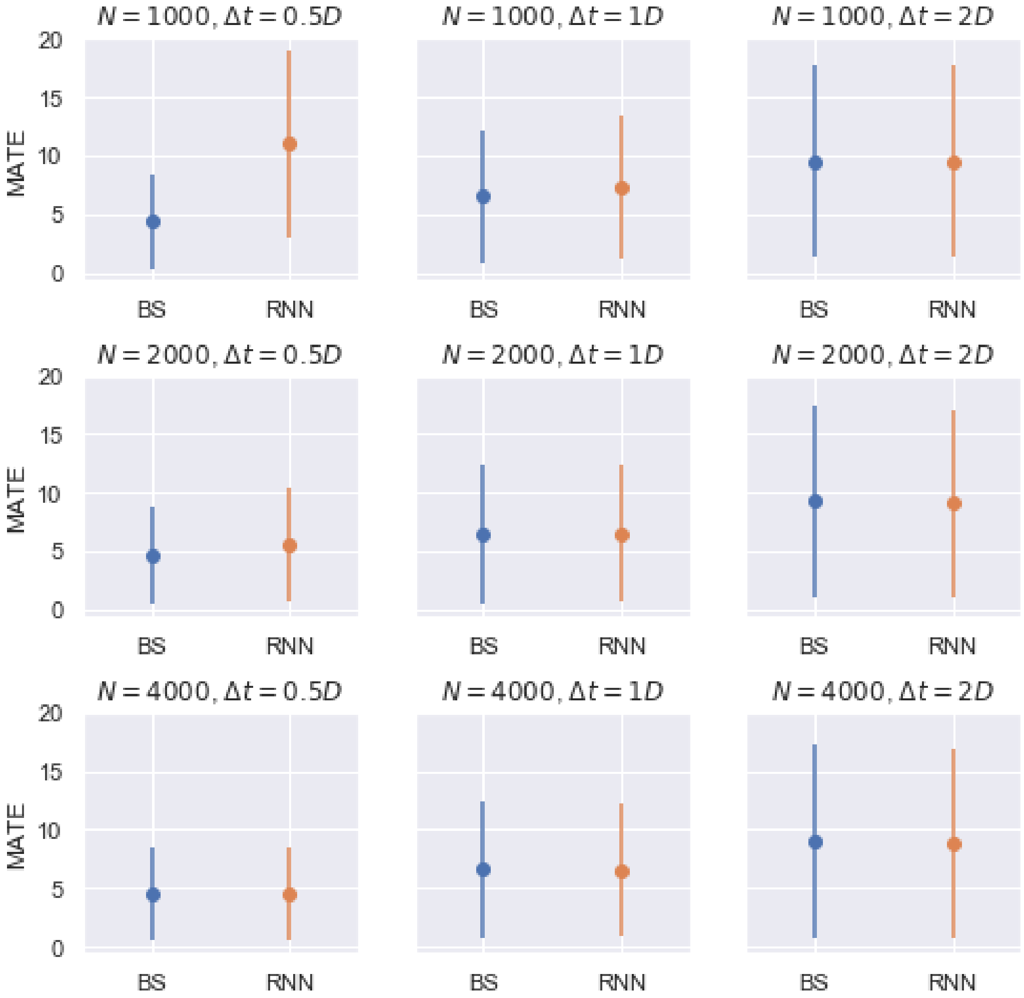


Fig. 5.5 The mean and the standard deviation of MATEs for the Black-Scholes and the RNN under different simulation setups. The mean is indicated by each dot, and the standard deviation by the line segment on both sides of the dot. The mean and the standard deviations are calculated over the underlying paths.

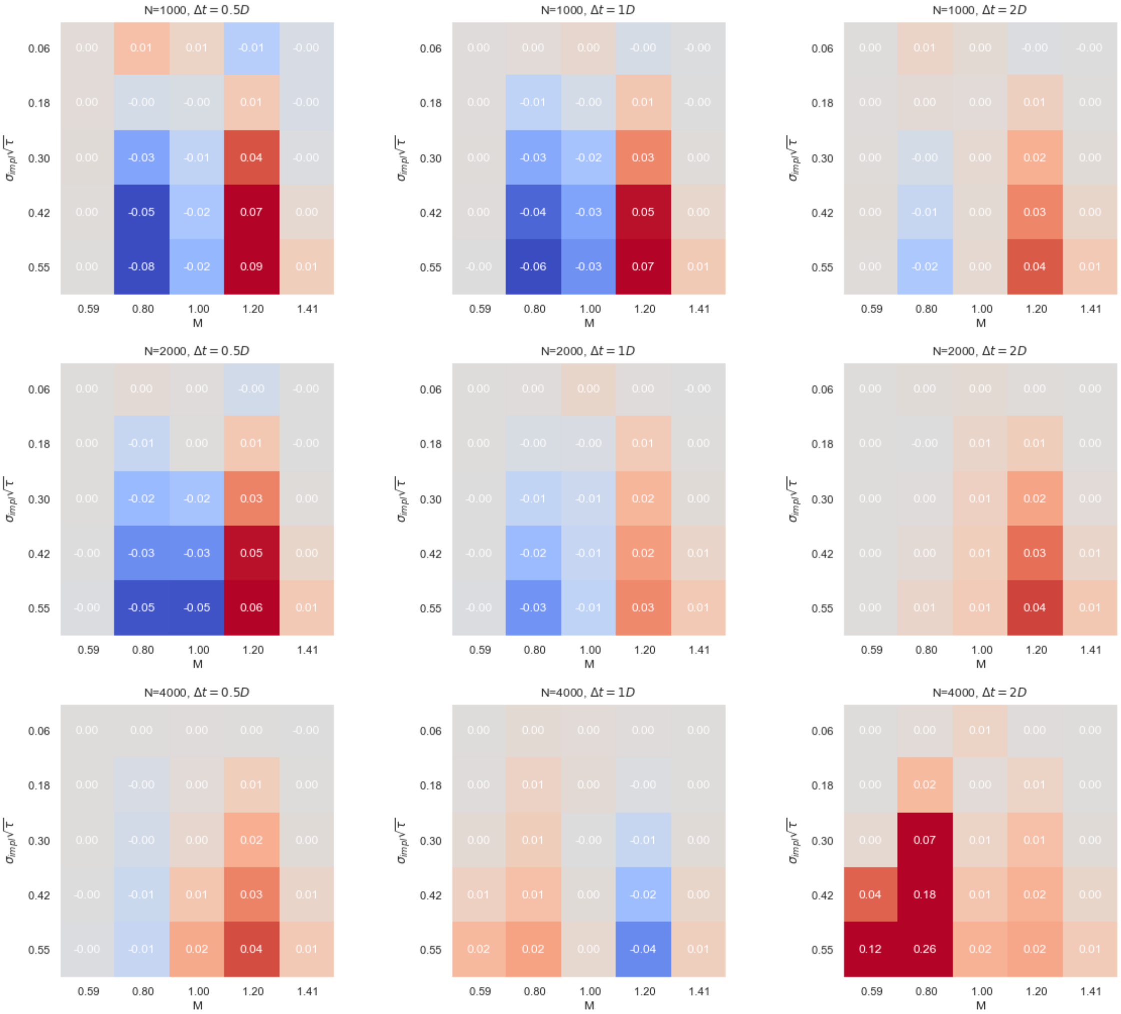


Fig. 5.6 The difference of strategies between the RNN ($M;\ \sigma_{\text{impl}}\sqrt{\tau}$) and the BS-Delta under the nine different simulation setups. We refer to the caption of Figure 5.4 for the explanation of the figure. For the sake of readability, each tile is the average of a 10×10 box.

Chapter 6

Discussion

6.1 Estimation v.s. calibration

The ANNs have been used for option pricing or hedging in various ways. Here we discuss two major competing approaches; they are using ANNs as non-parametric tools to estimate option prices, and as computational tools for calibration.

The first approach stems from Hutchinson et al. [1994], with numerous followers primarily from finance or computer science community. This approach does not start with modelling the underlying asset. Instead, it learns the mapping from observable features (as well as some estimations) to options prices by means of ANNs. Hence it avoids mis-specifying models for the underlying asset. This reduces the model risk to some extent. In addition, it allows for a flexible design of the mapping one wish to learn and investigate. This thesis is developed in this spirit. However, since the parameters in non-parametric models do not have any economic meaning, models lack interpretations when evaluating. This feature brings in other kinds of model risks; see Cohen et al. [2021]. Such models can further be over-parametrised. One can always change to models that are more complex, hence proper regularisations are needed especially for ANNs. Indeed the risk management of 'sell-side institutions' is subject to regulatory purposes. Lacking interpretations is one of the reasons that stop regulators and hence sell-sides from trusting non-parametric models despite sometime better performance.

The other approach is more recent, and mostly followed by financial mathematicians. It aims to move the computational heavy calibration of stochastic models offline, by means of ANNs. Since Bachelier [1900] first use the Brownian motion in his Ph.D. thesis, numerous extensions have been proposed as alternatives for modelling the underlying, see Bakshi et al. [1997], with the most widely used being the Black-Scholes despite many well-known defects. To make a model popular, it needs not only to capture the stylized characteristics of the asset, but often more importantly to be efficient in computation. For instance, the Heston model gains its popularity only after its fast Fourier pricing is derived. Many stochastic models have limited use in the industry due to their slow pricing and calibration, i.e. the lack of tractability of stochastic models. ANNs have provided a way to bypass the computational bottleneck as approximators for pricing functionals from model parameters to option prices. Therefore, the online model calibration procedure can use the off-line trained approximation; see Horvath et al. [2021] for example on the rough Bergomi model. This approach considerably speeds up the calibration procedure that banks perform regularly, hence undoubtedly enhancing the popularity of those more complex models. Valuing options are still built on the underlying models, and hence one can interpret the price and hedging strategy under the model. Thus, it does not revolutionise the risk-management practice as it is now, and is gaining popularity. However, the disadvantage is that the ANNs as computational tools do not offer any insight in the historical data as our statistical hedging does.

6.2 Limitations

There are several drawbacks in the present statistical hedging. The major one is that it does not provide an internal consistency across options with respect to the underlying reference model. Much of the research on valuing options starts with choosing stochastic models for the underlying assets. The most important property is that the underlying dynamics serves as a reference, through which cross-sectional consistency is maintained. Put it simply, one can calibrate the reference dynamics to the market prevailing prices of the most liquidly traded options, and then use the calibrated reference model to price other illiquid options. Theoretically, this bottom-up approach ensures that all options, including the exotics to be priced, are consistent with the reference and other options. In practice the calibration may not be perfect, in which case option prices are not necessarily consistent with the liquid ones.

Although a complete incorporation of the internal consistency into our statistical hedging method may be too much beyond this thesis and requires a careful restructuring, our approach could offer different viewpoints. Early in Bakshi et al. [1997], they mention that the need for a minimum variance hedging arises because of un-traded risk or model specifications and transaction cost when a perfect delta-hedging is unfeasible. More recently, Carr and Wu [2020] propose a top-down framework to attribute the profit and loss of one particular option to various risk sensitivities. Hence, they shfit the focus from terminal payoff to the behaviour of short-term return; see also plenty of references therein. This approach is hence‘localized’. They also offer two reasons to argue against sticking to the sometimes ‘foolish’ consistency. First, being consistent does not solve the model selection problem. In particular, when pricing long-dated options, the model needs to simulate the underlying price and instantaneous volatility far into the future, which deviate from the reality. Giglio and Kelly [2018] show that the internal consistency conditions imposed by autoregressive models between short-

and long-maturity options are strongly rejected in a wide range of options; see Carr and Wu [2003] and Bakshi et al. [1997] for more examples. Secondly, when one only holds the option for a short period, there is no need to make long-term predictions on the underlying. As in our setup, the trader's marking to the market practice makes it important to decompose the short-term risk sources, instead of predicting the terminal payoff of the options.

Our approach is localised in terms of the hedging period it considers, and this allows to investigate the effects of various parameters of simulation models. For instance, we confirm that with the Black-Scholes simulation data, the asset return in physical measure plays a role in optimal discrete delta hedging, a fact also known in quadratic hedging literature. In addition, given the internal inconsistency between short- and long-dated options observed in the literature, our framework readily allows for an investigation into the difference of hedging on these two kinds of options in future research.

Another related yet less severe drawback of our method is that it relies on the availability of historical prices of options to be hedged. While this is not an issue for liquid exchange-traded options such as European options on index or American options on individual stocks of large companies, it limits the extensions to exotics that are traded over-the-counter. Although it is difficult to obtain such data from public sources, large market makers of exotics may have access to their own trading data with higher quality, and hence have a better chance in implementing our statistical hedging methods.

6.3 Extensions

In this thesis, we have focused on the single-period hedging of European options. Below we discuss the possibility and challenge of extending our approach to other types of options. The following sketch of extensions does not apply to the multiple period hedging presented in Chapter 5.

6.3.1 Barrier and Asian options

Consider a single-barrier up-and-out call with no knock-out rebate. This option is path-dependent, and has a discontinuity in its payoff function. These two features bring difficulties to the hedging of barrier options. The first issue can be solved in two ways. One can add appropriate state variables to the feature space for the ANNs; the running maximum of the underlying price is such a state for the option being considered. Alternatively, one can use a type of ANNs that can generate internal state to pass along the time series, such as the Long-Short Term Memory network. The second difficulty means that the Delta of the option can be very negative when the underlying is near the barrier. Gamma is also large near the barrier, and this puts pressure on delta-hedging, due to the fast changing Delta. Although there is no problem in applying our approach to hedge such options, one needs to analyse the performance on such regions on the out-of-sample data to understand the implications of the barrier. Moreover, one may consider using liquid vanilla options instead as the hedging instrument to deal with the large Gamma.

The same method of including state variables can also be applied to Asian options. Consider an arithmetic Asian option. This kind of options is difficult to price and hedge due to the lack of analytic solutions. Milevsky and Posner [1998] deduce an approximately analytic solution for such options. Inspired by their functional form, one can add the average observed underlying prices as the extra state variable to the feature set of ANNs for a fixed-strike Asian option.

6.3.2 American options

American options on large-capitalisation companies are quite liquid and exchange-traded. Hence, the data of such option prices have relatively good quality. Such prices have already included the early exercise premiums. Therefore, one can simply arrange the data and then apply our statistical hedging in the same way as before.

6.3.3 Transaction costs

In the present thesis, the trader sell and buy the option both at mid-prices. However, traders are incurred to transaction costs such as commission and bid-ask spread. These two costs can be added to (3.1) rather easily. If one considers proportional transaction costs that charge ϵ_1 and ϵ_2 for trading in stocks and options respectively, then the loss function is replaced by

$$\mathbf{E}\left[\left(V_1^{\delta}\right)^2\right] = \mathbf{E}\left[\left(\delta S_1 + (1 + r_{\text{onr}}\Delta t)(C_0^{\text{ask}} - \delta S_0) - C_1^{\text{bid}} - \varphi_{\text{tc}}\right)^2\right],$$

where $\varphi_{\text{tc}} = \epsilon_1 \delta(S_0 + S_1) + \epsilon_2(C_0^{\text{ask}} + C_1^{\text{bid}})$.

While the above loss function considers market frictions in the single period, transaction costs matter much more in the multiple period where dynamic rebalancing is needed. To this end, one would instead consider a utility maximization framework; see Kallsen and Muhle-Karbe [2015].

Bibliography

Y. S. Abu-Mostafa. Financial model calibration using consistency hints. *IEEE transactions on neural networks*, 12(4):791–808, 2001.

D. Ackerer, N. Tagasovska, and T. Vatter. Deep smoothing of the implied volatility surface. SSRN 3402942, 2019.

P. Ahmed and S. Swidler. Forecasting properties of neural network generated volatility estimates. In *Decision Technologies for Computational Finance*, pages 247–258. Springer, 1998.

J. J. Ahn, D. H. Kim, K. J. Oh, and T. Y. Kim. Applying option Greeks to directional forecasting of implied volatility in the options market: an intelligent approach. *Expert Systems with Applications*, 39(10):9315–9322, 2012.

C. Albanese, S. Crépey, R. Hoskinson, and B. Saadeddine. XVA analysis from the balance sheet. *Quantitative Finance*, 21(1):99–123, 2021.

H. Albrecher, P. Mayer, W. Schoutens, and J. Tistaert. The little Heston trap. *Wilmott*, 1: 83–92, 2007.

C. Alexander and L. M. Nogueira. Model-free hedge ratios and scale-invariant models. *Journal of Banking & Finance*, 31(6):1839–1861, 2007.

C. Alexander, A. Rubinov, M. Kalepky, and S. Leontsinis. Regime-dependent smile-adjusted delta hedging. *Journal of Futures Markets*, 32(3):203–229, 2012.

H. Amilon. A neural network versus Black–Scholes: a comparison of pricing and hedging performances. *Journal of Forecasting*, 22:317–335, 2003.

S. Amornwattana, D. Enke, and C. H. Dagli. A hybrid option pricing model using a neural network for estimating volatility. *International Journal of General Systems*, 36(5):558–573, 2007.

U. Anders, O. Korn, and C. Schmitt. Improving the pricing of options: a neural network approach. *Journal of Forecasting*, 17(5-6):369–388, 1998.

P. C. Andreou. *Parametric and Nonparametric Functional Estimation for Options Pricing with Applications in Hedging and Trading*. PhD thesis, University of Cyprus, 2008.

P. C. Andreou, C. Charalambous, and S. H. Martzoukos. Critical assessment of option pricing methods using artificial neural networks. In *International Conference on Artificial Neural Networks*, pages 1131–1136. Springer, 2002.

P. C. Andreou, C. Charalambous, and S. H. Martzoukos. Robust artificial neural networks for pricing of European options. *Computational Economics*, 27(2-3):329–351, 2006.

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

P. C. Andreou, C. Charalambous, and S. H. Martzoukos. Generalized parameter functions for option pricing. *Journal of Banking & Finance*, 34(3):633–646, 2010.

M. Avellaneda, A. Carelli, and F. Stella. Following the Bayes path to option pricing. *Journal of Computational Intelligence in Finance*, 1998.

L. Bachelier. Théorie de la spéculation. In *Annales scientifiques de l'École normale supérieure*, volume 17, pages 21–86, 1900.

G. Bakshi, C. Cao, and Z. Chen. Empirical performance of alternative option pricing models. *The Journal of Finance*, 52(5):2003–2049, 1997.

G. Bakshi, C. Cao, and Z. Chen. Do call prices and the underlying stock always move in the same direction? *The Review of Financial Studies*, 13(3):549–584, 2000.

G. Barone-Adesi and R. E. Whaley. Efficient analytic approximation of American option values. *The Journal of Finance*, 42(2):301–320, 1987.

E. Barucci, U. Cherubini, and L. Landi. No-arbitrage asset pricing with neural networks under stochastic volatility. In *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, pages 3–16. World Scientific, 1996.

E. Barucci, U. Cherubini, and L. Landi. Neural networks for contingent claim pricing via the Galerkin method. In *Computational Approaches to Economic Problems*, pages 127–141. Springer, 1997.

M. Barunikova and J. Barunik. Neural networks as semiparametric option pricing tool. *Bulletin of the Czech Econometric Society*, 18(28), 2011.

D. S. Bates. Jumps and stochastic volatility: exchange rate processes implicit in Deutsche mark options. *The Review of Financial Studies*, 9(1):69–107, 1996.

C. Bayer and B. Stemper. Deep calibration of rough stochastic volatility models. arXiv:1810.03399, 2018.

C. Bayer, B. Horvath, A. Muguruza, B. Stemper, and M. Tomas. On deep calibration of (rough) stochastic volatility models. arXiv:1908.08806, 2019.

C. Beck, S. Becker, P. Cheridito, A. Jentzen, and A. Neufeld. Deep splitting method for parabolic PDEs. arXiv:1907.03452, 2019.

S. Becker, P. Cheridito, and A. Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20(74):1–25, 2019.

Y. Bengio. Using a financial training criterion rather than a prediction criterion. *International Journal of Neural Systems*, 8(4):433–443, 1997.

Y. Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade*, pages 437–478. Springer, 2012.

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

D. Bertsimas, L. Kogan, and A. W. Lo. When is time continuous? *Journal of Financial Economics*, 55(2):173–204, 2000.

M. Billio, M. Corazza, and M. Gobbo. Option pricing via regime switching models and multilayer perceptrons: a comparative approach. *Rendiconti per gli Studi Economici Quantitativi*, 2002:39–59, 2002.

T. Björk. *Arbitrage theory in continuous time*. Oxford university press, 2004. ISBN 019957474X.

F. Black. Studies of stock market volatility changes. *Proceedings of the American Statistical Association Business and Economic Statistics Section*, 1976.

L. Blynski and A. Faseruk. Comparison of the effectiveness of option price forecasting: Black–Scholes vs. simple and hybrid neural networks. *Journal of Financial Management & Analysis*, 19(2):46–58, 2006.

C. Boek, P. Lajbcygier, M. Palaniswami, and A. Flitman. A hybrid neural network approach to the pricing of options. In *Proceedings of ICNN'95–International Conference on Neural Networks*, volume 2, pages 813–817. IEEE, 1995.

P. P. Boyle and D. Emanuel. Discretely adjusted option hedges. *Journal of Financial Economics*, 8(3):259–282, 1980.

T. Briegel and V. Tresp. Dynamic neural regression models. Retrieved on August 29, 2019 from https://epub.ub.uni-muenchen.de/1571/, 2000.

H. Buehler, L. Gonon, J. Teichmann, and B. Wood. Deep hedging. *Quantitative Finance*, 19 (8):1271–1291, 2019a.

H. Buehler, L. Gonon, J. Teichmann, B. Wood, B. Mohan, and J. Kochems. Deep hedging: hedging derivatives under generic market frictions using reinforcement learning. SSRN 3355706, 2019b.

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

J. Cao, J. Chen, and J. C. Hull. A neural network approach to understanding implied volatility movements. *Quantitative Finance*, 20, 2020.

A. Carelli, S. Silani, and F. Stella. Profiling neural networks for option pricing. *International Journal of Theoretical and Applied Finance*, 3(02):183–204, 2000.

P. Carr and L. Wu. The finite moment log stable process and option pricing. *The Journal of Finance*, 58(2):753–777, 2003.

P. Carr and L. Wu. Option profit and loss attribution and pricing: A new framework. *The Journal of Finance*, 75:2271–2316, 2020.

P. Carr, H. Geman, D. B. Madan, and M. Yor. Stochastic volatility for Lévy processes. *Mathematical Finance*, 13(3):345–382, 2003.

A. P. Carverhill and T. H. Cheuk. Alternative neural network approach for option pricing and hedging. SSRN 480562, 2003.

Q. Chan-Wai-Nam, J. Mikael, and X. Warin. Machine learning for semi linear PDEs. *Journal of Scientific Computing*, 79(3):1667–1712, 2019.

T.-Y. Chang, Y.-H. Wang, and H.-Y. Yeh. Forecasting of option prices using a neural network model. *Journal of Accounting, Finance & Management Strategy*, 8(1):123–136, 2013.

C. Charalambous and S. H. Martzoukos. Hybrid artificial neural networks for efficient valuation of real options and financial derivatives. *Computational Management Science*, 2 (2):155–161, 2005.

F. Chen and C. Sutcliffe. Pricing and hedging short sterling options using neural networks. *Intelligent Systems in Accounting, Finance and Management*, 19(2):128–149, 2012.

J. Chen. Learning the Black–Scholes formula via support vector machines. In *Recent Advances in Statistics Application and Related Areas, 2nd Conference of the International Institute of Applied Statistics Studies*, volume 1&2, pages 756–760, 2009.

S.-H. Chen and W.-C. Lee. Pricing call warrants with artificial neural networks: the case of the Taiwan derivative market. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 6, pages 3877–3882. IEEE, 1999.

D.-Y. Chiu and C.-C. Lin. Exploring internal mechanism of warrant in financial market with a hybrid approach. *Expert Systems with Applications*, 35(3):1237–1245, 2008.

H.-J. Choi, H.-S. Lee, G.-S. Han, and J. Lee. Efficient option pricing via a globally regularized neural network. In *International Symposium on Neural Networks*, pages 988–993. Springer, 2004.

P. Christoffersen, K. Jacobs, and K. Mimouni. Volatility dynamics for the S&P500: Evidence from realized volatility, daily returns, and option prices. *The Review of Financial Studies*, 23(8):3141–3189, 2010.

G. Claeskens and N. L. Hjort. The focused information criterion. *Journal of the American Statistical Association*, 98(464):900–916, 2003.

S. N. Cohen, D. Snow, and L. Szpruch. Black-box model risk in finance. arXiv:2102.04757, 2021.

R. Cont and J. Da Fonseca. Dynamics of implied volatility surfaces. *Quantitative Finance*, 2 (1):45–60, 2002.

R. Cont and Y. H. Kan. Dynamic hedging of portfolio credit derivatives. *SIAM Journal on Financial Mathematics*, 2(1):112–140, 2011.

C. J. Corrado and T. Su. Skewness and kurtosis in S&P 500 index returns implied by option prices. *Journal of Financial Research*, 19(2):175–192, 1996.

J. C. Cox, S. A. Ross, and M. Rubinstein. Option pricing: a simplified approach. *Journal of Financial Economics*, 7(3):229–263, 1979.

S. Crépey. Delta-hedging vega risk? *Quantitative Finance*, 4(5):559–579, 2004.

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

S. P. Das and S. Padhy. A new hybrid parametric and machine learning model with homogeneity hint for European-style index option pricing. *Neural Computing and Applications*, 28(12):4061–4077, 2017.

J. F. G. de Freitas, M. Niranjan, and A. H. Gee. Hierarchical Bayesian models for regularization in sequential learning. *Neural Computation*, 12(4):933–953, 2000a.

J. F. G. de Freitas, M. Niranjan, A. H. Gee, and A. Doucet. Sequential Monte Carlo methods to train neural network models. *Neural Computation*, 12(4):955–993, 2000b.

S. Denkl, M. Goy, J. Kallsen, J. Muhle-Karbe, and A. Pauwels. On the performance of delta hedging strategies in exponential Lévy models. *Quantitative Finance*, 13(8):1173–1184, 2013.

G. Dimitroff, D. Röder, and C. Fries. Volatility model calibration with convolutional neural networks. SSRN 3252432, 2018.

Z. A. Dindar and T. Marwala. Option pricing using a committee of neural networks and optimized networks. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 1, pages 434–438. IEEE, 2004.

C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating second-order functional knowledge for better option pricing. In *Advances in Neural Information Processing Systems*, pages 472–478. MIT Press, 2001.

C. Dugas, Y. Bengio, F. Bélisle, C. Nadeau, and R. Garcia. Incorporating functional knowledge in neural networks. *Journal of Machine Learning Research*, 10(Jun):1239–1262, 2009.

W. E, J. Han, and A. Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

Z. Fang and K. George. Application of machine learning: an analysis of Asian options pricing using neural network. In *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, pages 142–149. IEEE, 2017.

S. Fecamp, J. Mikael, and X. Warin. Risk management with machine-learning-based algorithms. arXiv:1902.05287, 2019.

R. Ferguson and A. Green. Deeply learning derivatives. SSRN 3244821, 2018.

J. Galindo-Flores. A framework for comparative analysis of statistical and machine learning methods: an application to the Black–Scholes option pricing model. In *Computational Finance 1999*, pages 635–660. MIT Press, 2000.

R. Garcia and R. Gençay. Option pricing with neural networks and a homogeneity hint. In *Decision Technologies for Computational Finance*, pages 195–205. Springer, 1998.

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

J. Gatheral. *The Volatility Surface: A Practitioner's Guide*, volume 357. John Wiley & Sons, 2011.

J. Gatheral and A. Jacquier. Arbitrage-free SVI volatility surfaces. *Quantitative Finance*, 14 (1):59–71, 2014.

D. S. Geigle. *An Artificial Neural Network Approach to the Valuation of Options and Forecasting of Volatility*. PhD thesis, Nova Southeastern University, 1999.

D. S. Geigle and J. E. Aronson. An artificial neural network approach to the valuation of options and forecasting of volatility. *Journal of Computational Intelligence in Finance*, 7 (6):19–25, 1999.

R. Gençay and R. Gibson. Model risk for European-style stock index options. *IEEE Transactions on Neural Networks*, 18(1):193–202, 2007.

R. Gençay and M. Qi. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *IEEE Transactions on Neural Networks*, 12(4):726–734, 2001.

R. Gençay and A. Salih. Degree of mispricing with the Black–Scholes model and nonparametric cures. *Annals of Economics and Finance*, 4:73–101, 2003.

H. Ghaziri, S. Elfakhani, and J. Assi. Neural networks approach to pricing options. *Neural Network World*, 10(1):271–277, 2000.

J. Ghosn and Y. Bengio. Multi-task learning for option pricing. Retrieved on October 29, 2019 from https://cirano.qc.ca/files/publications/2002s-53.pdf, 2002.

E. Ghysels, A. C. Harvey, and E. Renault. 5 stochastic volatility. *Handbook of statistics*, 14: 119–191, 1996.

E. Ghysels, V. Patilea, É. Renault, and O. Torrès. Nonparametric methods and option pricing. In D. Hand and S. Jacka, editors, *Statistics in Finance*, chapter 13, pages 261–282. John Wiley & Sons, 1998.

S. Giglio and B. Kelly. Excess volatility: Beyond discount rates. *The Quarterly Journal of Economics*, 133(1):71–127, 2018.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

N. Gradojevic and D. Kukolj. Parametric option pricing: a divide-and-conquer approach. *Physica D: Nonlinear Phenomena*, 240(19):1528–1535, 2011.

N. Gradojevic, R. Gençay, and D. Kukolj. Option pricing with modular neural networks. *IEEE Transactions on Neural Networks*, 20(4):626–637, 2009.

A. Gregoriou, J. Healy, and C. Ioannidis. Hedging under the influence of transaction costs: an empirical investigation on FTSE 100 index options. *Journal of Futures Markets*, 27(5): 471–494, 2007.

J. T. Hahn. *Option Pricing Using Artificial Neural Networks: An Australian Perspective*. PhD thesis, Bond University, 2013.

I. Halperin. QLBS: Q-learner in the Black-Scholes (-Merton) worlds. arXiv:1712.04609, 2017.

S. A. Hamid and A. Habib. Can neural networks learn the Black-Scholes model?: A simplified approach. Retrieved on September 9, 2019 from https://academicarchive.snhu.edu/bitstream/handle/10474/1662/cfs2005-01.pdf, 2005.

J. Han, A. Jentzen, and W. E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

M. Hanke. Neural network approximation of option pricing formulas for analytically intractable option pricing models. *Journal of Computational Intelligence in Finance*, 5(5): 20–27, 1997.

M. Hanke. Adaptive hybrid neural network option pricing. *Journal of Computational Intelligence in Finance*, 7(5):33–39, 1999a.

M. Hanke. Neural networks versus Black-Scholes: an empirical comparison of the pricing accuracy of two fundamentally different option pricing methods. *Journal of Computational Intelligence in Finance*, 5:26–34, 1999b.

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

J. V. Healy, M. Dixon, B. J. Read, and F. F. Cai. Confidence in data mining model predictions: a financial engineering application. In *IECON'03. 29th Annual Conference of the IEEE Industrial Electronics Society (IEEE Cat. No. 03CH37468)*, volume 2, pages 1926–1931. IEEE, 2003.

J. V. Healy, M. Dixon, B. J. Read, and F. F. Cai. Confidence limits for data mining models of options prices. *Physica A: Statistical Mechanics and its Applications*, 344(1-2):162–167, 2004.

J. V. Healy, M. Dixon, B. J. Read, and F. F. Cai. Non-parametric extraction of implied asset price distributions. *Physica A: Statistical Mechanics and its Applications*, 382(1):121–128, 2007.

P. Henry-Labordère. Deep primal-dual algorithm for BSDEs: applications of machine learning to CVA and IM. SSRN 3071506, 2017.

P. Henry-Labordère. Generative models for financial data. SSRN 3408007, 2019.

A. Hernandez. Model calibration with neural networks. *Risk Magazine*, pages 1–5, June 2017.

R. Herrmann and A. Narr. Neural networks and the evaluation of derivatives: some insights into the implied pricing mechanism of german stock index options. Retrieved on August 29, 2019 from http://finance.fbv.kit.edu/download/dp202.pdf, 1997.

S. L. Heston. 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, 1993.

B. Horvath, A. Muguruza, and M. Tomas. Deep learning volatility: a deep neural network perspective on pricing and calibration in (rough) volatility models. *Quantitative Finance*, 21(1):11–27, 2021. ISSN 1469-7688.

J. Huang and L. Wu. Specification analysis of option pricing models based on time-changed levy processes. *The Journal of Finance*, 59(3):1405–1439, 2004.

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

S.-C. Huang and T.-K. Wu. A hybrid unscented Kalman filter and support vector machine model in option price forecasting. In *International Conference on Natural Computation*, pages 303–312. Springer, 2006.

J. Huh. Pricing options with exponential Lévy neural network. *Expert Systems with Applications*, 127:128–140, 2019.

J. Hull and A. White. Optimal delta hedging for options. *Journal of Banking & Finance*, 82: 180–190, 2017.

C. Huré, H. Pham, and X. Warin. Deep backward schemes for high-dimensional nonlinear pdes. 2020.

J. M. Hutchinson, A. W. Lo, and T. Poggio. A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance*, 49(3):851–889, 1994.

A. Itkin. Deep learning calibration of option pricing models: some pitfalls and solutions. arXiv:1906.03507, 2019.

A. Jacquier and M. Oumgari. Deep PPDEs for rough local stochastic volatility. arXiv:1906.02551, 2019.

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

K.-H. Jung, H.-C. Kim, and J. Lee. A novel learning network for option pricing with confidence interval information. In *International Symposium on Neural Networks*, pages 491–497. Springer, 2006.

M. Kakati. Pricing and hedging performances of artificial neural net in Indian stock option market. *The ICFAI Journal of Applied Finance*, 11(1):62–73, 2005.

M. Kakati. Option pricing using adaptive neuro-fuzzy system (ANFIS). *ICFAI Journal of Derivatives Markets*, 5(2):53–62, 2008.

J. Kallsen and J. Muhle-Karbe. Option pricing and hedging with small transaction costs. *Mathematical Finance*, 25(4):702–723, 2015.

O. Karaali, W. Edelberg, and J. Higgins. Modelling volatility derivatives using neural networks. In *Proceedings of the IEEE/IAFE 1997 Computational Intelligence for Financial Engineering*, pages 280–286. IEEE, 1997.

T. Karatas, A. Oskoui, and A. Hirsa. Supervised deep neural networks (DNNs) for pricing/calibration of vanilla/exotic options under various different processes. arXiv:1902.05810, 2019.

D. L. Kelly. Valuing and hedging American put options using neural networks. Retrieved on August 29, 2019 from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.721.8497&rep=rep1&type=pdf, 1994.

B.-H. Kim, D. Lee, and J. Lee. Local volatility function approximation using reconstructed radial basis function networks. In *International Symposium on Neural Networks*, pages 524–530. Springer, 2006.

D. P. Kingma and J. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

P. Ko, P. Lin, W. Chien, and Y. Cheng. Hedging derivative securities based on the neural network coefficient model. In *Proceedings of the Eighth Joint Conference on Information Sciences*, pages 1163–1166. Curran Associates Inc, 2005.

P.-C. Ko. Option valuation based on the neural regression model. *Expert Systems with Applications*, 36(1):464–471, 2009.

M. Kohler, A. Krzyżak, and N. Todorovic. Pricing of high-dimensional American options by neural networks. *Mathematical Finance*, 20(3):383–410, 2010.

P. N. Kolm and G. Ritter. Dynamic replication and hedging: A reinforcement learning approach. *The Journal of Financial Data Science*, 1(1):159–171, 2019.

A. Kondratyev and C. Schwarz. The market generator. SSRN 3384948, 2019.

S. G. Kou. A jump-diffusion model for option pricing. *Management Science*, 48(8):1086–1101, 2002.

J. Krause. Option pricing with neural networks. In *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing*, volume 3, pages 2206–2210. Aachen Mainz, 1996.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

G. Lachtermacher and L. Rodrigues Gaspar. Neural networks in derivative securities pricing forecasting in Brazilian capital markets. In *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, pages 92–97. World Scientific, 1996.

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

P. R. Lajbcygier. Comparing conventional and artificial neural network models for the pricing of options. In *Neural Networks in Business: Techniques and Applications*, pages 220–235. IGI Global, 2002.

P. R. Lajbcygier. Improving option pricing with the product constrained hybrid neural network. In *Artificial Neural Networks and Neural Information Processing*, pages 615–621. Springer, 2003.

P. R. Lajbcygier. Improving option pricing with the product constrained hybrid neural network. *IEEE Transactions on Neural Networks*, 15(2):465–476, 2004.

P. R. Lajbcygier and J. T. Connor. Improved option pricing using artificial neural networks and bootstrap methods. *International Journal of Neural Systems*, 8(04):457–471, 1997a.

P. R. Lajbcygier and J. T. Connor. Improved option pricing using bootstrap methods. In *Proceedings of International Conference on Neural Networks*, volume 4, pages 2193–2197. IEEE, 1997b.

P. R. Lajbcygier and A. Flitman. A comparison of non-parametric regression techniques for the pricing of options using an optimal implied volatility. In *Decision Technologies for Financial Engineering: Proceedings of the Fourth International Conference on Neural Networks in Capital Markets*, pages 201–213. World Scientific, 1996.

P. R. Lajbcygier, C. Boek, A. Flitman, and M. Palaniswami. Comparing conventional and artificial neural network models for the pricing of options on futures. *NeuroVe$t Journal*, 4 (5):16–24, 1996a.

P. R. Lajbcygier, C. Boek, M. Palaniswami, and A. Flitman. Neural network pricing of all ordinaries SPI options on futures. In *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, pages 64–77, 1996b.

P. R. Lajbcygier, A. Flitman, A. Swan, and R. J. Hyndman. The pricing and trading of options using a hybrid neural network model with historical volatility. *NeuroVe$t Journal*, 5(1): 27–41, 1997.

L. J. le Roux and G. S. du Toit. Emulating the Black & Scholes model with a neural network. *Southern African Business Review*, 5(1):54–57, 2001.

M. T. Leung, A.-S. Chen, and R. Mancha. Making trading decisions for financial-engineered derivatives: a novel ensemble of neural networks using information content. *Intelligent Systems in Accounting, Finance & Management*, 16(4):257–277, 2009.

X. Liang, H. Zhang, and J. Yang. Pricing options in Hong Kong market based on neural networks. In *International Conference on Neural Information Processing*, pages 410–419. Springer, 2006.

X. Liang, H. Zhang, J. Xiao, and Y. Chen. Improving option price forecasts with neural networks and support vector regressions. *Neurocomputing*, 72(13-15):3055–3065, 2009.

C.-T. Lin and H.-Y. Yeh. The valuation of Taiwan stock index option price—comparison of performances between Black-Scholes and neural network model. *Journal of Statistics and Management Systems*, 8(2):355–367, 2005.

D. Liu and S. Huang. The performance of hybrid artificial neural network models for option pricing during financial crises. *Journal of Data Science*, 14(1):1–18, 2016.

D. Liu and L. Zhang. Pricing Chinese warrants using artificial neural networks coupled with Markov regime switching model. *International Journal of Financial Markets and Derivatives*, 2(4):314–330, 2011.

M. Liu. Option pricing with neural networks. In *Progress in Neural Information Processing*, volume 2, pages 760–765. Springer, 1996.

S. Liu, A. Borovykh, L. A. Grzelak, and C. W. Oosterlee. A neural network-based framework for financial model calibration. *Journal of Mathematics in Industry*, 9(1), 2019a.

S. Liu, C. W. Oosterlee, and S. M. Bohte. Pricing options and computing implied volatilities using neural networks. *Risks*, 7(1):1–22, 2019b.

X. Liu, Y. Cao, C. Ma, and L. Shen. Wavelet-based option pricing: an empirical study. *European Journal of Operational Research*, 272(3):1132–1142, 2019c.

F. A. Longstaff and E. S. Schwartz. Valuing American options by simulation: a simple least-squares approach. *The Review of Financial Studies*, 14(1):113–147, 2001.

J. Lu and H. Ohta. A data and digital-contracts driven method for pricing complex derivatives. *Quantitative Finance*, 3(3):212–219, 2003a.

J. Lu and H. Ohta. Digital contracts-driven method for pricing complex derivatives. *Journal of the Operational Research Society*, 54(9):1002–1010, 2003b.

M. Ludwig. Robust estimation of shape-constrained state price density surfaces. *The Journal of Derivatives*, 22(3):56–72, 2015.

T. J. Lyons. Uncertain volatility and the risk-free synthesis of derivatives. *Applied Mathematical Finance*, 2(2):117–133, 1995.

D. B. Madan, P. P. Carr, and E. C. Chang. The variance Gamma process and option pricing. *Review of Finance*, 2(1):79–105, 1998.

M. Malliaris and L. Salchenberger. Beating the best: a neural network challenges the Black-Scholes formula. In *Proceedings of 9th IEEE Conference on Artificial Intelligence for Applications*, pages 445–449. IEEE, 1993a.

M. Malliaris and L. Salchenberger. A neural network model for estimating option prices. *Journal of Applied Intelligence*, 3(3):193–206, 1993b.

M. Malliaris and L. Salchenberger. Using neural networks to forecast the S&P100 implied volatility. *Neurocomputing*, 10(2):183–195, 1996.

C. G. Martel, M. D. G. Artiles, and F. F. Rodriguez. A financial option pricing model based on learning algorithms. In *Proceedings of the World Multiconference on Applied Economics, Business and Development*, pages 153–157. WSEAS, 2009.

W. A. McGhee. An artificial neural network representation of the SABR stochastic volatility model. SSRN 3288882, 2018.

G. Meissner and N. Kawano. Capturing the volatility smile of options on high-tech stocks—a combined GARCH-neural network approach. *Journal of Economics and Finance*, 25(3): 276–292, 2001.

M. A. Milevsky and S. E. Posner. Asian options, the sum of lognormals, and the reciprocal gamma distribution. *Journal of Financial and Quantitative Analysis*, 33(3):409–422, 1998.

F. G. Miranda and N. Burgess. Intraday volatility forecasting for option pricing using a neural network approach. In *Proceedings of 1995 Conference on Computational Intelligence for Financial Engineering*, page 31. IEEE, 1995.

S. K. Mitra. Improving accuracy of option price estimation using artificial neural networks. SSRN 876881, 2006.

S. K. Mitra. An option pricing model that combines neural network approach and Black Scholes formula. *Global Journal of Computer Science and Technology*, 12(4):7–15, 2012.

G. Montagna, M. Morelli, O. Nicrosini, P. Amato, and M. Farina. Pricing derivatives by path integral and neural networks. *Physica A: Statistical Mechanics and its Applications*, 324 (1-2):189–195, 2003.

L. Montesdeoca and M. Niranjan. Extending the feature set of a data-driven artificial neural network model of pricing financial options. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6. IEEE, 2016.

M. J. Morelli, G. Montagna, O. Nicrosini, M. Treccani, M. Farina, and P. Amato. Pricing financial derivatives with neural networks. *Physica A: Statistical Mechanics and its Applications*, 338(1-2):160–165, 2004.

F. Mostafa. *Applications of Neural Networks in Market Risk*. PhD thesis, Curtin University, 2011.

F. Mostafa and T. Dillon. A neural network approach to option pricing. *WIT Transactions on Information and Communication Technologies*, 41:71–85, 2008.

M. Niranjan. Sequential tracking in pricing financial options using model based and neural network approaches. In *Advances in Neural Information Processing Systems*, pages 960–966. MIT Press, 1996.

M. O'hara. *Market Microstructure Theory*. Wiley, 1997.

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

S. Palmer. *Evolutionary Algorithms and Computational Methods for Derivatives Pricing*. PhD thesis, University College London, 2019.

S. Palmer and D. Gorse. Pseudo-analytical solutions for stochastic options pricing using Monte Carlo simulation and breeding PSO-trained neural networks. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 365–370, 2017.

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

H. Park, N. Kim, and J. Lee. Parametric models and non-parametric machine learning models for predicting option prices: empirical comparison study over KOSPI 200 index options. *Expert Systems with Applications*, 41(11):5227–5237, 2014.

C. Pérignon. Testing the monotonicity property of option prices. *The Journal of Derivatives*, 14(2):61–76, 2006.

B. Phani, B. Chandra, and V. Raghav. Quest for efficient option pricing prediction model using machine learning techniques. In *The 2011 International Joint Conference on Neural Networks*, pages 654–657. IEEE, 2011.

M. M. Pires and T. Marwala. American option pricing using multi-layer perceptron and support vector machine. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 2, pages 1279–1285. IEEE, 2004a.

M. M. Pires and T. Marwala. Option pricing using Bayesian neural networks. In *Fifteenth Annual Symposium of the Pattern Recognition Association of South Africa*, pages 161–166, 2004b.

M. M. Pires and T. Marwala. American option pricing using Bayesian multi-layer perceptrons and Bayesian support vector machines. In *IEEE 3rd International Conference on Computational Cybernetics*, pages 219–224. IEEE, 2005.

M. Qi. *Financial Applications of Generalized Nonlinear Nonparametric Econometric Methods (Artificial Neural Networks)*. PhD thesis, Ohio State University, 1996.

M. Qi and G. Maddala. Option pricing using artificial neural networks: the case of S&P 500 index call options. In *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, pages 78–91. World Scientific, 1996.

C. Quek, M. Pasquier, and N. Kumar. A novel recurrent neural network-based prediction system for option trading and hedging. *Applied Intelligence*, 29(2):138–151, 2008.

M. Raberto, G. Cuniberti, M. Riani, E. Scales, F. Mainardi, and G. Servizi. Learning short-option valuation in the presence of rare events. *International Journal of Theoretical and Applied Finance*, 3(03):563–564, 2000.

J. Ruf and W. Wang. Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 24(1):1–46, 2020a.

J. Ruf and W. Wang. Hedging with linear regressions and neural networks. SSRN 3580132, 2020b.

J. Ruf and W. Wang. Information leakage in backtesting. Working paper, 2021.

S. Saito and L. Jun. Neural network option pricing in connection with the Black and Scholes model. In *Proceedings of the Fifth Conference of the Asian Pacific Operations Research Society*, 2000.

Z. I. Samur and G. T. Temur. The use of artificial neural network in option pricing: the case of S&P 100 index options. *International Journal of Social, Behavioral, Educational, Economic, Business and Industrial Engineering*, 3(6):644–649, 2009.

A. Saxena. Valuation of S&P CNX Nifty options: comparison of Black-Scholes and hybrid ANN model. In *Proceedings SAS Global Forum*, 2008.

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

H. J. Shin and J. Ryu. A dynamic hedging strategy for option transaction using artificial neural networks. *International Journal of Software Engineering and its Applications*, 6(4): 111–116, 2012.

J. Sirignano and K. Spiliopoulos. DGM: a deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

H. R. Stoll. The supply of dealer services in securities markets. *The Journal of Finance*, 33 (4):1133–1151, 1978.

H. Stone. Calibrating rough volatility models: a convolutional neural network approach. *Quantitative Finance*, 20(3):1–14, 2019.

P. Tankov and E. Voltchkova. Asymptotic analysis of hedging errors in models with jumps. *Stochastic Processes and Their Applications*, 119(6):2004–2027, 2009.

A. Taudes, M. Natter, and M. Trcka. Real option valuation with neural networks. *Intelligent Systems in Accounting, Finance & Management*, 7(1):43–52, 1998.

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

S. D. Teddy, E.-K. Lai, and C. Quek. A cerebellar associative memory approach to option pricing and arbitrage trading. *Neurocomputing*, 71(16-18):3303–3315, 2008.

N. S. Thomaidis, V. S. Tzastoudis, and G. Dounias. A comparison of neural network model selection strategies for the pricing of S&P500 stock index options. *International Journal on Artificial Intelligence Tools*, 16(06):1093–1113, 2007.

A. N. Tikhonov, A. Goncharsky, V. Stepanov, and A. G. Yagola. *Numerical Methods for the Solution of Ill-posed Problems*, volume 328. Springer Science & Business Media, 1995.

R. Tsaih. Sensitivity analysis, neural networks, and the finance. In *IJCNN'99. International Joint Conference on Neural Networks. Proceedings (Cat. No. 99CH36339)*, volume 6, pages 3830–3835. IEEE, 1999.

C.-H. Tseng, S.-T. Cheng, Y.-H. Wang, and J.-T. Peng. Artificial neural network model of the hybrid EGARCH volatility of the Taiwan stock index option prices. *Physica A: Statistical Mechanics and its Applications*, 387(13):3192–3200, 2008.

W. L. Tung and C. Quek. GenSo-OPATS: a brain-inspired dynamically evolving option pricing model and arbitrage trading system. In *2005 IEEE Congress on Evolutionary Computation*, volume 3, pages 2429–2436. IEEE, 2005.

W. L. Tung and C. Quek. Financial volatility trading using a self-organising neural-fuzzy semantic network and option straddle-based approach. *Expert Systems with Applications*, 38(5):4668–4688, 2011.

V. S. Tzastoudis, N. S. Thomaidis, and G. D. Dounias. Improving neural network based option price forecasting. In *Hellenic Conference on Artificial Intelligence*, pages 378–388, 2006.

S. Vähämaa. Delta hedging with the smile. *Financial Markets and Portfolio Management*, 18 (3):241–255, 2004.

M. S. Vidales, D. Siska, and L. Szpruch. Unbiased deep solvers for parametric PDEs. arXiv:1810.05094, 2019.

C. von Spreckelsen, H.-J. von Mettenheim, and M. H. Breitner. Steps towards a high-frequency financial decision support system to pricing options on currency futures with neural networks. *International Journal of Applied Decision Sciences*, 7(3):223–238, 2014.

C.-P. Wang, S.-H. Lin, H.-H. Huang, and P.-C. Wu. Using neural network for forecasting TXO price under different volatility models. *Expert Systems with Applications*, 39(5): 5025–5032, 2012.

H.-W. Wang. Dual derivatives spreading and hedging with evolutionary data mining. *Journal of American Academy of Business*, 9:45–52, 2006.

P. Wang. Pricing currency options with support vector regression and stochastic volatility model with jumps. *Expert Systems with Applications*, 38(1):1–7, 2011.

Y.-H. Wang. Nonlinear neural network forecasting model for stock index option price: hybrid GJR–GARCH approach. *Expert Systems with Applications*, 36(1):564–570, 2009a.

Y.-H. Wang. Using neural network to forecast stock index option price: a new hybrid GARCH approach. *Quality & Quantity*, 43(5):833–843, 2009b.

A. White. *Pricing Options with Futures-Style Margining: a Genetic Adaptive Neural Network Approach.* Garland Publishing, 2000.

A. J. White. A genetic adaptive neural network approach to pricing options: a simulation analysis. *Journal of Computational Intelligence in Finance*, 6(2):13–23, 1998.

M. Wiese, L. Bai, B. Wood, and H. Buehler. Deep hedging: learning to simulate equity option markets. arXiv:1911.01700, 2019a.

M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer. Quant GANs: deep generation of financial time series. arXiv:1907.06673, 2019b.

L. Xu, M. Dixon, B. A. Eales, F. F. Cai, B. J. Read, and J. V. Healy. Barrier option pricing: modelling with neural nets. *Physica A: Statistical Mechanics and its Applications*, 344 (1-2):289–293, 2004.

Y. Yang, Y. Zheng, and T. M. Hospedales. Gated neural networks for option pricing: rationality by design. In *Association for the Advancement of Artificial Intelligence*, pages 52–58. AAAI Press, 2017.

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

D. Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

T. Ye and L. Zhang. Derivatives pricing via machine learning. SSRN 3352688, 2019.

C. Zapart. Stochastic volatility options pricing with wavelets and artificial neural networks. *Quantitative Finance*, 2(6):487–495, 2002.

C. Zapart. Beyond Black–Scholes: a neural networks-based approach to options pricing. *International Journal of Theoretical and Applied Finance*, 6(05):469–489, 2003a.

C. Zapart. Statistical arbitrage trading with wavelets and artificial neural networks. In *2003 IEEE International Conference on Computational Intelligence for Financial Engineering*, pages 429–435. IEEE, 2003b.

Y. Zheng. *Machine Learning and Option Implied Information*. PhD thesis, Imperial College London, 2017.

Y. Zheng, Y. Yang, and B. Chen. Gated deep neural networks for implied volatility surfaces. arXiv:1904.12834, 2019.

W. Zhou, M. Yang, and L. Han. A nonparametric approach to pricing convertible bond via neural network. In *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, volume 2, pages 564–569. IEEE, 2007.