Parametric and Nonparametric Functional Estimation for Options Pricing with Applications in Hedging and Trading

Panayiotis C. Andreou ©

A dissertation submitted to the Department of Business and Public Administration of the University of Cyprus in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Finance.

Lefkosia
March, 2008

UNIVERSITY OF CYPRUS
THESIS ACCEPTANCE CERTIFICATE

The undersigned, appointed by the Council of the Department of Public and Business Administration of the University of Cyprus (UCY), certify that:

1. We have examined the dissertation submitted by Panayiotis Andreou, entitled "Parametric and Nonparametric Functional Estimation for Options Pricing with Applications in Hedging and Trading".

2. The candidate successfully delivered a public defence of his dissertation in Nicosia, on March 4, 2008 and was subjected to an examination by the committee.

3. In our opinion this thesis is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

We hereby recommend to the Senate of the University of Cyprus that the aforementioned dissertation be accepted in partial fulfilment of the requirements for the degree of Doctor of Philosophy (PhD).

The examination committee:

Hercules Vladimirou (Chairman, UCY)

Elena Andreou (Econ. Dept., UCY)

Chris Charalambous (Co-supervisor, UCY)

Spiros H. Martzoukos (Co-supervisor, UCY)

Stylianos Perrakis (J. M. School of Bus., Concordia U.)

Date: March 5, 2008

Περίληψη Διδακτορικής Μελέτης

Η παρούσα διδακτορική διατριβή απαρτίζεται από τέσσαρα δοκίμια. Στο πρώτο δοκίμιο συγκρίνουμε τα παραμετρικά μοντέλα τιμολόγησης Ευρωπαϊκών παράγωγων προϊόντων προαιρετικής εξάσκησης (call options) των Black και Scholes (1973) και Corrado και Su (1996) με τα μη-παραμετρικά μοντέλα Τεχνητών Νευρωνικών Δικτύων. Για τις συγκρίσεις χρησιμοποιούμε διάφορους συνδυασμούς μεταβλητών εισαγωγής/εξαγωγής συμπεριλαμβανομένου και υβριδικών μοντέλων (όπου η εξαρτημένη μεταβλητή αντιπροσωπεύει την διαφορά μεταξύ της αγοραίας τιμής και της εκτίμησης ενός παραμετρικού μοντέλου). Ένας από τους επιμέρους στόχους της συγκεκριμένης μελέτης είναι επίσης η διερεύνηση δυναμικών στρατηγικών αντιστάθμισης κίνδυνων καθώς και στρατηγικών εμπορίας κινητών αξιών και παραγώγων κάτω από ρεαλιστικές συνθήκες διαπραγμάτευσης (συμπεριλαμβανομένου και κόστους συναλλαγής).

Το δεύτερο δοκίμιο εξετάζει εκ' νέου τα σημαντικότερα αποτελέσματα του πρώτου δοκιμίου χρησιμοποιώντας Τεχνητά Νευρωνικά Δίκτυα τα οποία εκτιμώνται με την συνάρτηση που προτάθηκε από τον Huber το 1981. Βάση αυτή της μεθοδολογίας, η επίδραση απόμακρων καθώς και άλλων παρατηρήσεων που μπορεί να δημιουργούν ανωμαλίες στα χρηματοοικονομικά δεδομένα ελαχιστοποιείται. Το βασικό συμπέρασμα από αυτό το δοκίμιο είναι ότι παρατηρείται σημαντική βελτίωση στα μέτρα άκριβειας για νέα δεδομένα εκτός του δείγματος εκτίμησης για τα μη-παραμετρικά μοντέλα που εκτιμώνται με την συνάρτηση του Huber σε σχέση με εκείνα που εκτιμώνται με συναρτήσεις που ελαχιστοποιούν το άθροισμα των τετραγωνικών αποκλίσεων.

Στο τρίτο δοκίμιο δίνεται σημαντική προσοχή στην ανάπτυξη ενός νέου ημι-παραμετρικού μοντέλου το οποίο ουσιαστικά συμβάλει στον εμπλουτισμό του περιεχομένου και της ποιότητας των παραμέτρων που χρησιμοποιούνται ως

δεδομένα εισαγωγής στα παραμετρικά μοντέλα. Η προτεινόμενη ημι-παραμετρική μεθοδολογία αποτελεί ουσιαστικά την επέκταση των Ντετερμινιστικών Συναρτήσεων Εκτίμησης της Μεταβλητότητας (δεύτερη ροπή) των Dumas et al. (1998). Η προτεινόμενη ημι-παραμετρική μεθοδολογία μπορεί να χρησιμοποιηθεί για την εκτίμηση Γενικευμένων Συναρτήσεων Εκτίμησης Παραμέτρων όχι κατ' ανάγκη μόνο για την μεταβλητότητα (δεύτερη ροπή). Συγκεκριμένα, σε αυτό το δοκίμιο, δείχνουμε τον τρόπο με τον οποίο μπορούν να αξιοποιηθούν στην περίπτωση του μοντέλου των Corrado και Su (1996) αναφορικά με την εκτίμηση της τρίτης και τέταρτης ροπής (skewness και kurtosis). Σε αυτό το δοκίμιο γίνεται ενδελεχής σύγκριση της προτεινόμενης μεθοδολογίας με πιο εξελιγμένα παραμετρικά υποδείγματα τιμολόγησης προαιρετικών δικαιωμάτων. Συγκεκριμένα χρησιμοποιούμε το παραμετρικό μοντέλο που προτάθηκε το 1996 από τον Bates το οποίο επιτρέπει στην μεταβλητότητα του υποκείμενου δείκτη να είναι στοχαστική και επιπλέον επιτρέπει ασυνέχειες (jumps) στις διαδοχικές τιμές των αποδόσεων του δείκτη. Το γενικό συμπέρασμα είναι ότι η προτεινόμενη ημι-παραμετρική μεθοδολογία συγκρίνεται πολύ ευνοϊκά σε σχέση με τα πιο εξελιγμένα παραμετρικά μοντέλα σε περιπτώσεις νέων δεδομένων που δεν χρησιμοποιηθήκαν κατά την εκτίμηση των μοντέλων. Επιπλέον τα διάφορα μοντέλα χρησιμοποιήθηκαν για υλοποίηση στρατηγικών αντιστάθμισης κινδύνων. Εδώ ακολουθήθηκαν δύο εναλλακτικές στρατηγικές για τα ημι-παραμετρικά μοντέλα: i) μια κατά την οποία τα ημι-παραμετρικά μοντέλα εκτιμώνται ώστε να ελαχιστοποιούν ένα κριτήριο αποτελεσματικότητας συνδεδεμένο με την ακρίβεια τιμολόγησης των παράγωγων προϊόντων, ii) μια άλλη κατά την οποία τα ημι-παραμετρικά μοντέλα εκτιμώνται ώστε να ελαχιστοποιούν ένα κριτήριο αποτελεσματικότητας συνδεδεμένο με την ακρίβεια αντιστάθμισης κινδύνου. Το γενικό συμπέρασμα είναι ότι η δεύτερη στρατηγική δουλεύει πολύ καλύτερα και παράγει αποτελέσματα που διαφέρουν από την προηγούμενη βιβλιογραφία.

Τέλος, το επίκεντρο του τέταρτου δοκιμίου είναι να εξερευνήσει τις δυνατότητες εφαρμογής στο αντικείμενο της τιμολόγησης μίας νέας μεθοδολογίας γνωστή ως Support Vector Machines. Αυτή η μεθοδολογία έχει αναπτυχθεί στο πλαίσιο της στατιστικής θεωρίας μάθησης (statistical learning theory) και μέχρι στιγμής δεν έχει τύχει ευρείας εφαρμογής στις χρηματοπιστωτικές οικονομετρικές περιπτώσεις. Σε αυτό το δοκίμιο δοκιμάζουμε την αρχική μεθοδολογία όπως αυτή προτάθηκε από τον Vapnik το 1995 και η οποία θεωρείται ιδανική για εφαρμογές όπου το στατιστικό σφάλμα δεν ακολουθεί την κανονική κατανομή. Επιπλέον, θεωρούμε μια νεότερη παραλλαγή αυτής της μεθοδολογίας η οποία βασίζεται σε κριτήριο ελάχιστων τετραγώνων και η οποία θεωρείται πιο ιδανική για περιπτώσεις όπου το στατιστικό σφάλμα ακολουθεί κανονική κατανομή. Αυτές οι νέες μέθοδοι συγκρίνονται με τα Τεχνητά Νευρωνικά Δίκτυα όπως αυτά αναπτύχθηκαν στα δύο πρώτα δοκίμια. Η εμπειρική ανάλυση αυτού του δοκιμιού καταδεικνύει ελπιδοφόρα αποτελέσματα για τις νέες μεθοδολογίες.

Exordium

The current thesis is composed by four essays. In the first essay we compare the options pricing performance of the parametric Black and Scholes (1973) and Corrado and Su (1996) models with the nonparametric feedforward Artificial Neural Networks. We do this by using a battery of historical and implied parameter measures as well as market and hybrid target functions (desired output) resulting to a significant number of input-outputs combinations. In this essay we investigate the dynamic performance of the models by using hedging strategies and their economic significance by using trading strategies.

The second essay re-examines the most important key results from the first essay by using Robust Artificial Neural Networks. The Huber (1981) loss function is used in this case in order to estimate the nonparametric models. The main conclusion from this essay is that the out of sample pricing accuracy of the nonparametric models can be improved under the robust estimation scheme considered.

In the third essay the major contribution regards the development of a novel semi-parametric approach were an enhancement of the implied parameter values is used in the parametric option pricing models. The proposed semi-parametric methodology is basically extending the Deterministic Volatility Functions approach of Dumas et al. (1998) that perform a smoothing in the Black and Scholes implied volatilities across strike prices and maturities. Our semi-parametric methodology is much more generic though since it can be utilized to estimate Generalized Parameter Functions with other parametric models. Specifically, in this essay we show how it can be utilized in the case of Corrado and Su model and how it can enhance the implied volatility, skewness and kurtosis. We also extend the Deterministic Volatility Functions approach for the Corrado and Su model. The proposed semi-parametric methodology is also compared with more sophisticated options pricing models like the Stochastic Volatility and Stochastic Volatility and Jump models of Bates (1996). The overall result from the out of sample pricing tests is that our methodology is robust and performs exceptionally well. Furthermore we test the hedging performance of all models developed and we distinguish between models selected based on a pricing criterion and models selected based on a hedging criterion. Our results are different from previous literature since we find that better out of sample hedging performance can be obtained when optimization is based on a hedging criterion.

Finally, the focus of the fourth essay is to explore the pricing performance of Support Vector Machines in options pricing. This is a novel nonparametric methodology that has been developed in the context of statistical learning theory and until now is has been practically neglected in financial econometric applications. In this essay we consider the original methodology as proposed by Vapnik (1995) which is relying on a robust loss function. In addition, we consider a later variant of this methodology that relies on a least squares loss function (called the Least Squares Support Vector Machines). The new methods are compared with feedforward Artificial Neural Networks and also with parametric options pricing models using standard implied parameters and parameters derived via Deterministic Volatility Functions. The empirical analysis of this essay reveals promising results for the new methodologies.

Preface

Black and Scholes introduced in 1973 a milestone options pricing model for pricing European options. This model was a breakthrough in the pricing of options and still has a tremendous influence on the way that academics and practitioners evaluate and trade alternative derivatives products. Nevertheless, empirical research has shown that the formula suffers from systematic biases when compared to options market prices giving birth to the well known volatility smile anomaly (also known as volatility smirk or sneer). This comes from the fact that the model has been developed using a set the simplifying assumptions resulting to a lognormal distribution with constant variance for the underlying asset price. This assumption is not flexible enough to approximate the (unknown) market options' pricing function since it is empirically true that the implicit stock returns distributions are negatively skewed with higher kurtosis than allowable in the Black and Scholes lognormal distribution. For this reason the use of this model with historical or overall average implied parameters (one per day) results in biased prices that translate into poor pricing performance. Nevertheless in this thesis it is shown that simple methodologies that mitigate the anomaly by allowing the application of this model with maturity or contract specific implied parameters can improve significantly its performance making it a tougher benchmark for more complex and sophisticated alternative models. In the quest for the best performing parametric model other parametric models are considered as well. Attention is also given to the Corrado and Su (1996) model, which is an extension of the Black and Scholes formula that can easily handle nonnormal skewness and kurtosis. In addition, other more advent parametric models are considered, like for instance the Stochastic Volatility and the Stochastic Volatility & Jump (Bates, 1996), which are probably the most widely referenced models from the parametric family. All these parametric models are based on a set of assumptions like continuous time trading and completeness of the markets (which can hold in the presence of many trading options contacts – in addition we note also the recent addition of trading contracts on the volatility). The most significant novelty of this thesis regards the investigation of nonparametric methods that can offer noticeable pricing improvements compared to the benchmark parametric models. Specifically the gist of our attention in the first three essays of this thesis is concentrated in the application of feedforward Artificial Neural Networks. This nonparametric technique has gained considerable popularity in financial and economic applications for (at least) three reasons. First, there are theoretical foundations showing that Artificial Neural Networks can be used for multidimensional nonlinear regression since they are universal approximators able to approximate any nonlinear function and its derivatives arbitrarily well. Second, they learn the empirical input/output relationships inductively using historical or implied input variables and transactions data. Third, they can become more accurate and computationally more efficient alternatives when the underlying asset's price dynamics are unknown, a property very important for the problem we investigate.

In the first essay we compare the ability of the parametric Black and Scholes, Corrado and Su models, and feedforward Artificial Neural Networks to price European call options on the S&P 500 index. We use several historical and implied parameter measures. Beyond the standard neural networks employed to directly approximate the unknown empirical options pricing function, in our analysis we include hybrid networks that incorporate information from the parametric models. Specifically in the hybrid models the target function is the residual between the actual call market price and the parametric option price estimate. In this essay our results are significant and differ from previous literature. We show that the Black and Scholes based hybrid artificial neural network models outperform both the standard neural networks estimated on the market target function and the parametric ones. We also investigate the economic significance of the best models using trading strategies (extended with the Chen and Johnson, 1985, modified hedging approach). We find that there exist profitable opportunities even in the presence of transaction costs.

In the second essay the significant difference compared to the first essay is that we develop Robust Artificial Neural Networks optimized with the Huber (1981) function. In the first essay Artificial Neural Networks have been optimized based on the least squares norm. This norm though is susceptible to the influence of large errors since some abnormal datapoints (or few outlier observations) can deliver non-reliable networks. On the contrary, robust optimization methods that exploit the least absolute norm are unaffected by large (or catastrophic) errors but are doomed to fail when dealing with small variation errors. The Huber function is an ideal candidate to be used since it utilizes the robustness of least absolute and the unbiasedness of least squares norm and has proved to be an efficient tool for robust optimization problems for various tasks. The analysis here is augmented again with the use of several historical and implied volatility measures. It is shown that the Artificial Neural Network models with the use of the Huber function outperform the ones optimized with least squares.

In the third essay we extend the Deterministic Volatility Functions of Dumas et al. (1998) to derive a semi-parametric approach that provides an enhancement of the implied parameter values that are used with the parametric option pricing models. With this new semi-parametric methodology we are able to enhance not only volatility but also skewness and kurtosis. Overall this methodology is proposed as a way to alleviate deficiencies of the modern parametric options models and standard nonparametric approaches. In addition, it utilizes information from the parametric models and preserves some very important properties which do not hold for the nonparametric models employed in the first two essays. The results obtained in this essay strongly support the proposed approach which compares very favorably to the more sophisticated parametric options pricing models considered, like the Stochastic Volatility and Jump model of Bates (1996). The out of sample results are shown to be robust under alternative dataset choices and model complexity. In addition, the economic significance of the approach is tested in terms of hedging where the evaluation and estimation loss functions are aligned: hedging results when enhancing skewness and kurtosis parameters are significantly improved.

Finally in the fourth essay we explore the pricing performance of Support Vector Machines for pricing S&P 500 index call options. This is a novel nonparametric (function approximation) methodology that has been developed in the context of statistical learning theory and until now its applications on financial econometric purposes are limited. Support Vector Machines employ the so called VC theory (developed by Vapnik and Chervonenkis in 1974), which is defined in a strictly statistical framework, that controls in specific ways the model's estimation and parameterization to preclude overfitting so as ensure good out of sample (generalization) results; this is a crucial property of paramount importance. Another significant characteristic of this methodology is that the estimation of its free parameters results from the solution of a convex optimization problem with a unique global (and sparse) solution. Compared to feedforward Artificial Neural Networks this methodology can be considered as an improvement due to its well defined regularization and optimization properties. In the fourth essay we consider both the traditional support vector machines originally developed by Vapnik (1995) as well as the Least Squares Support Vector Machines which are a subsequent variant of the original methodology. These new methods are compared with feedforward Artificial Neural Networks and also with parametric options pricing models considered in the third essay. The empirical results using three years of data indicate that this new methodology is promising enough since it can produce pricing results that are comparable to the benchmark models.

The reader of this thesis should be aware of three things. The first remark is that it was never attempted to downgrade the importance of existing parametric models and to exaggerate about the attractive characteristics and applicability of the nonparametric models. A vivid message from reading this thesis is that every approach has its own merits and limitations and that the best result can be obtained when they are handled them as complementary methodologies. Thus, most of the times the best performing models combine the two methodologies resulting in this way in semi-parametric models. Furthermore, a considerable effort has been spent in developing the methodologies in order to be implementable for real world applications. For example, beyond pricing performance tests this thesis also includes hedging results and economic significance tests.

The second remark is that each essay is almost practically independent in the sense that the reader can read it without having to know exactly the content of the rest essays since each essay it's aiming to an independent publication. In addition, the effort was to use the same nomenclature and symbolization in all essays yet in some cases minor differences might be observed.

The third remark is that the performance statistics (e.g. out of sample pricing, hedging and trading strategies, robustness, etc.) reported in each essay may be different. This occurs because the scope of each essay is different. For example in the first essay the effort is to provide a comprehensive comparison between the parametric models considered and the feedforward Artificial Neural Networks. For this reason extensive hedging simulations and trading strategies results are also reported. A similar extensive set of results is reported in the third essay in which a novel semi-parametric options pricing methodology is proposed and tested. On the contrary in the second essay where the focus is on the use of a robust loss function with Artificial Neural Networks, the battery of results is confined to those statistics needed to show the difference in performance between the alternative models. In the same spirit we create and report the statistics for the fourth essay.

Acknowledgements

I am most indebted to my PhD thesis supervisors, Chris Charalambous and Spiros Martzoukos for all the guidance and the excellent knowledge they bequeath me in both, the field of applied nonparametric methods and the area of financial derivative securities. They stood by me every single instant, always offering me their multi-fold knowledge and their critical reasoning in a variety of subjects. They have always been motivating me to aim at tougher and tougher goals and they have taught me how to successfully ponder any issue related with my fields of interest. Working with them I had the fortune to learn by heart the following motto: "*in the middle of every difficulty lies an opportunity* (by Albert Einstein).

Thanks also for financial support provided by the University of Cyprus during the period 2004-2007 from the research program "Artificial Neural Networks for Contingent Claims Valuation". I would also like to acknowledge partial funding by the Center for Banking and Financial Research at the University of Cyprus.

Most of my PhD work has been presented in various conferences where I got valuable feedback. In addition, for the first essay I am also thankful to Paul Lajbcygier for comments and discussions.

Finally, and most importantly, I am very grateful and I would like to thank my wife, Adamantia, and my parents, Pambies and Maro, for encouraging and supporting me during the completion of this thesis. I am very grateful to them because their unwavering love, presence and stimulation were crucial and invaluable to the completion of this thesis.

Contents

List of Figures

List of Tables

List of Abbreviations

ANNs: Feedforward Artificial Neural Networks (ANN in singular)

AggTs: Aggregate testing dataset (joint set of testing datasets)

BS: Black and Scholes (1973) options pricing model

CS: Corrado and Su (1996) options pricing model

DVF: Deterministic Valatility Functions

ePOPMs: Enhanced Parametric Option Pricing Models (ePOPM in singular)

GPFs: Generalized Parameter Functions (GPF in singular)

LM: Modified Levenberg-Marquardt algorithm

LS-SVMs: Least Squares Support Vector Machines (LS-SVM in singular)

NLS: Nonlinear Least Squares

OLS: Ordinary Least Squares

POPMs: Parametric Option Pricing Models (POPM in singular)

SV: Stochastic Volatility options pricing model of Bates (1996)

SVJ: Stochastic Volatility and Jumps options pricing model of Bates (1996)

SVMs: Support Vector Machines (SVM in singular)

Tr: Training dataset

Ts: Testing dataset

Vd: Validation dataset

I. Introduction – Overview of the Thesis

I.1. General Discussion

Black and Scholes introduced in 1973 their milestone Parametric Options Pricing Model (POPM) that is nowadays known as the Black-Scholes Formula (BS). This model was a breakthrough in the pricing of options and still has a tremendous influence on the way that practitioners price various derivative securities. Despite the fact that in the last three decades the BS model and its later variants/extensions (i.e. Bakshi et al., 1997) are considered as the most prominent achievements in financial theory, empirical research has shown that the formula suffers from *systematic biases* when compared to options market prices (see Rubinstein, 1985 and 1994, Black and Scholes, 1975, MacBeth and Merville, 1980, Gultekin et al., 1982, Bakshi et al., 1997, Cont and Fonseca, 2002, and Andersen et al., 2002). The BS bias steams from the fact that the model has been developed using a set of unrealistic simplified assumptions.

The post-BS financial engineering research came up with a variety of POPMs that made it possible to mitigate the bias associated with the original model. Nevertheless, none of the modern models has managed to generalize all BS assumptions, and provide results fully consistent with the observed market data.

After including in the analysis more realistic POPMs like the Corrado and Su (1996) formula and models that allow for stochastic volatility and jump discontinuities to the diffusion process (see Bates 1996 and 2000), it is found that the BS is still relevant either with the use of a contract specific implied volatility or with the use of statistical smoothing techniques that produce one volatility per contract (see Dumas et al., 1998).

In 1995, Fisher Black declared in his article “The holes in the Black and Scholes” that “it is rare that the value of an option comes out exactly equal to the price at which it trades on the exchange”. This evidence forces us to accept the hypothesis that the market is pricing the options correctly and that the models are incorrect due to their mis-specifications. So, in answering the previous posed question, researchers can address their attention to market-data driven models which can be promising alternatives, in respect to unbiasedness and pricing accuracy, relative to the existing POPMs.

Nonparametric techniques and especially feedforward Artificial Neural Networks (ANNs) that is the main focus of the current thesis, comprise an empirically practical option-pricing approach since they involve no financial theory whatsoever since the option's price is estimated inductively using historical or implied input variables and option transactions data. In addition, option pricing functions are multivariate and highly nonlinear. ANNs are appropriate tools for approximating the *unknown empirical option pricing function* since they can be used for non-linear regression[1].

Empirical research (see Bakshi, et al., 1997) reports that modern parametric models are sometimes characterized by poor-out-of sample performance and by overwhelming complexity. Research on the ANN option pricing capabilities (see Hutchison et al., 1994, Qi and Maddala, 1996, Lajbcygier et al., 1997, Lajbcygier and Connor, 1997, Hanke, 1999a, Hanke, 1999b, Yao et al., 2000, Lajbcygier, 2001 and Anders et al., 1998) has reported excellent out of sample performance whilst in many cases, ANNs can outperform the conventional parametric models.

The scope of this thesis is in developing option pricing models by combining the use of feedforward Artificial Neural Networks with information provided by POPMs (the BS and the CS model). For the empirical tests we use European call options on the S&P 500 Index. In the first chapter of the thesis, we develop simple ANNs (with input supplemented by historical or implied parameters specific either to BS or the CS model), and hybrid ANNs that in addition use pricing information derived by any of the two parametric models. These specifications are compared with BS and CS with various historical and implied parameters (most of them are considered for the first time). In order to check the robustness of the results, in addition to a full dataset we repeat the analysis using a reduced dataset (following Hutchison et al., 1994). The economic significance of the models is investigated through hedging and trading strategies.

---

[1] Even in the case where option market prices evolve based on a known diffusion process that can be parametrized, then the ANNs are expected to be as accurate and robust as the parametric models. Research by Hutchison et al. (1994) and Hanke (1997) has revealed that feedforward Neural Networks can approximate arbitrarily well the traditional Black and Scholes Formula and other analytically intractable models like the GARCH (1,1).

Instead of naive trading strategies we implement improved (dynamic and cost-effective) ones. Furthermore, we also refine these strategies with the Chen and Johnson (1985) modified hedging approach.

In the second chapter we compare models from the first chapter of the thesis which are estimated based on the least squares loss function with robust ANNs that use the Huber loss function. In this chapter the gist of the attention is to develop ANNs based on the Huber loss function (Huber, 1981) so that the estimation of the standard and hybrid ANNs is robust in the presence of data-point peculiarities.

In the third chapter, we propose and examine nonparametric options pricing models which are dedicated to the pricing and hedging of European options. Specifically, we extend the Deterministic Volatility Functions (DVF) of Dumas[2] et al. (1998) to provide a nonparametric enhancement of the implied parameter values to be used in the parametric option pricing models. We estimate not only volatility but also skewness and kurtosis. The resulting enhanced structure is compared to parametric models with both standard implied parameters and parameters derived via DVF. The models developed are compared to the benchmark Stochastic Volatility (SV) and Stochastic Volatility and Jump (SVJ) models (Bates, 1996). The economic significance of the approach is also considered in terms of hedging retaining the intuition in Christoffersen and Jacobs (2004) that the estimation loss function should be aligned with the evaluation loss function.

Finally, in the last chapter of the thesis, we are examining the application of Support Vector Machines (SVM) to the pricing of European options extending in this way the results of the first two chapters (for details see Vapnik, 1995). This methodology is used for robust nonlinear regression problems based on a well defined statistical framework that predicts better out of sample generalization ability compared to other alternative methodologies.

[2] The DVF approach relaxes the BS assumption of having a single volatility per day.

I.2. Parametric models and the deterministic volatility functions

The parametric models used in this thesis are explained very briefly below. Specifically the following models as proposed in the literature are considered: the Black and Scholes (1973) model (which is used in all chapters), the Corrado and Su (1996) model (which is used in all chapters except the second one) and also the Stochastic Volatility and Stochastic Volatility and Jump models of Bates (1996) (used for the needs of the third and fourth chapters). Moreover, it is explained the application of the DVF for the case of BS, as originally proposed by Dumas et al. (1998), as well as the extensions made in this thesis concerning the use of DVF with the CS and SVJ models. The DVF models are used for the needs of the third and fourth chapters. The parametric models are used as benchmark models but in addition they are also used as part of the non parametric models that are proposed and developed.

I.2.1. Parametric models used

I.2.1.1. Black and Scholes model

The first model examined is the Black and Scholes (1973) since it is a benchmark and widely referenced model. The BS formula for European call options modified for a dividend-paying (see also Merton, 1973) underlying asset is:

$$c^{BS} = Se^{-d_y T} N(d) - Xe^{-rT} N(d - \sigma\sqrt{T}) \qquad \text{(I.1)}$$

$$d = \frac{\ln(S/X) + (r - d_y)T + (\sigma\sqrt{T})^2/2}{\sigma\sqrt{T}} \qquad \text{(I.1.1)}$$

where c^{BS} is premium paid for the European call option, S is the spot price of the underlying asset, X is the exercise price of the call option, r is the continuously compounded risk free interest rate, d_y is the continuous dividend yield paid by the underlying asset, T is the time left until the option expiration date, σ^2 is the yearly variance rate of return for the underlying asset and $N(.)$

stands for the standard normal cumulative distribution. The BS model is based on the following assumptions:

The stock (underlying asset) price follows a Geometric Brownian Motion in continuous time with a constant drift and volatility rate. Thus the distribution of possible stock prices at the end of any finite interval is lognormal. In addition, short-term risk free interest rate is known and is constant through time.

The stock pays no dividends[3] or other cash distributions.

The option is "European," that is, it can only be exercised at maturity.

There are no transaction costs in buying or selling the stock or the options.

It is possible to borrow any fraction of the price of a security to buy it or to hold it, at the short-term risk-free interest rate.

There are no penalties for short selling. A seller who does not own a security will simply accept the price of the security from a buyer, and will agree to settle with the buyer on some future date by paying an amount equal to the price of the security at that date.

In addition, the BS formula has been derived under conditions that allow a continuous and costless hedging of a risk less portfolio that is short one call option against a long position in the associated stock (underlying asset). In this word the markets are complete and the options are redundant securities. To this, the underlying and the derivatives markets are assumed to be efficient and that it would be impossible to make sure profits by creating such perfectly hedged risk less portfolios which can only earn the risk-free rate. An implicit assumption of the model is that its option estimates are independent of the characteristics of other securities and the preferences of investors (Rubinstein, 1976).

Most of the above assumptions are violated in the financial markets (Constantinides et al, 2008). For instance, Rubinstein (1976) has shown that the BS model does not hold under discrete trading conditions and risk aversion unless certain assumptions hold (e.g. no dividends, all investors agree upon a single value of the volatility, consumption and underlying asset are jointly lognormally distributed etc). In addition, the parametric models considered in this thesis have been developed under the assumption of no transaction costs. As shown in Leland (1985) and Bensaid et al. (1992), the existence of transaction

---

[3] Merton (1973) has relaxed this assumption.

costs would have produced bounds in which option prices should lie. In addition, some other assumptions are causing severe misspecification in the model. Specifically, the literature documents that BS model suffers from systematically biases because the Geometric Brownian Motion is a poor approximation of the unknown diffusion process that prevails in the market because, for example, it precludes the possibility for stochastic volatility and jumps. In other words, the BS assumption of a log-normal distribution is too limited since in practice implicit stock returns' distributions are negatively skewed with excess kurtosis (Bakshi et al., 1997). This creates the well known volatility *smile* anomaly (also known as volatility *smirk* or *sneer* anomaly) according to which the contract specific volatilities implied by the BS model exhibit certain patterns across moneyness (the ratio of the underlying asset to strike price) and maturity levels. In response to this, additional parametric option pricing models are also considered in this thesis. These models can be considered as generalizations of the BS formula because they generalize the Geometric Brownian Motion with more complex diffusion processes that imply more flexible and realistic distributions for the stock returns and can approximate better the unknown market diffusion process. Yet most of the other assumptions mentioned above hold also for these models. In this thesis the Stochastic Volatility as well as the Stochastic Volatility and Jump models of Bates (1996) are considered to be the most widely referenced generalizations of the BS model (see Bakshi et al., 1997). In addition, to the above two parametric models, in this thesis we also consider some heuristic extensions of the BS model like the Corrado and Su (1996) model as well as the Deterministic Volatility Functions approach proposed by Dumas et al. (1996).

Here we must note two things. First, that the BS, CS and the associated DVF models are based on the assumption of complete markets that rule out any arbitrage opportunities. Under such assumptions, the implied risk neutral parameters need no adjustment in order to reflect the ones obtained under the subjective diffusion process (the only adjustment is to add the market risk premium to the risk free rate). On the other hand, the SV and SVJ models are developed based on the incomplete markets where option pricing becomes tractable only under the assumption of a representative agent that has state independent utility of wealth. In other words, these models account for a premium induced by the risk of a jump and of randomly changing volatility.

According to these models, an adjustment based on a utility function is needed in order to go from the implied risk-neutral parameters to the subjective ones. For SV and SVJ, the implied risk neutral parameters need an adjustment in order to reflect the ones obtained under the subjective diffusion process (adding only the risk premium to the risk-free rate is not enough). Second, it is also important to note that our nonparametric models are developed under assumptions that also hold for the aforementioned parametric models (e.g. continuous trading, complete/incomplete and frictionless markets, etc).

The rest of the parametric models and methods used in this thesis are briefly explained below.

I.2.1.2. Corrado and Su model

The CS model constitutes an extension of the BS formula that accounts for additional skewness and kurtosis in stock returns in a heuristic manner. Corrado and Su base their extension on a methodology employed earlier by Jarrow and Rudd (1982). Using a Gram-Charlier series expansion of a normal density function they define their model as (see also the correction in Brown and Robinson, 2002; for further discussions see Jondeau and Rockinger, 2001, and Jurczenko et al., 2004):

$$c^{CS} = c^{BS} + \mu_3 Q_3 + (\mu_4 - 3) Q_4 \qquad \text{(I.2)}$$

where c^{BS} is the BS value for the European call option given in Eq. (I.1) and,

$$Q_3 = \frac{1}{3!} S e^{-d_y T} \sigma\sqrt{T}((2\sigma\sqrt{T} - d)n(d) + (\sigma\sqrt{T})^2 N(d)) \qquad \text{(I.2.1)}$$

$$Q_4 = \frac{1}{4!} S e^{-d_y T} \sigma\sqrt{T}((d^2 - 1 - 3\sigma\sqrt{T}(d - \sigma\sqrt{T}))n(d) + (\sigma\sqrt{T})^3 N(d)) \qquad \text{(I.2.2)}$$

In Eq. (I.2) Q_3 and Q_4 represent the marginal effect of non-normal skewness and kurtosis, respectively in the option price whereas μ_3 and μ_4 correspond to coefficients of skewness and kurtosis. In the above expressions,

$$n(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2 / 2) \qquad \text{(I.2.3)}$$

refers to the standard normal probability density function.

I.2.1.3. Stochastic volatility and stochastic volatility & jump models

Bakshi et al. (1997) found that the SVJ exhibited satisfactory out of sample performance for the S&P 500 index options when compared to other parametric option pricing models since it offers a quite flexible distributional structure. Specifically the correlation between the volatility and the returns of the underlying asset controls the level of skewness whilst the variability of volatility allows for non-normal kurtosis. Moreover, the addition of a jump component enhances the distributional flexibility and allows for more accurate pricing performance of the short term options. In this thesis the SVJ model of Bates (1996) is employed. In this model the instantaneous conditional variance V_t follows a mean-reverting square root process:

$$\frac{dS}{S} = (\mu - \lambda\bar{\kappa})dt + \sqrt{V}dZ + \kappa dq \qquad \text{(I.3)}$$

$$dV = (\alpha - \beta V)dt + \sigma_v \sqrt{V} dZ_v \qquad \text{(I.4)}$$

with

$$\text{cov}(dZ, dZ_v) = \rho dt$$

$$\ln(1+\kappa) \sim N(\ln(1+\bar{\kappa}) - 0.5\theta^2, \theta^2)$$

$$prob(dq = 1) = \lambda dt$$

where μ is the instantaneous drift of the underlying asset, λ is the annual frequency of jumps, κ is the random percentage jump conditional on a jump occurring, q is a Poison counter with intensity λ, θ^2 is the jump variance, and ρ is the correlation coefficient between the volatility shocks and the underlying asset movements. Moreover, β is the rate of mean reversion and α / β is the variance steady-state level (long run mean).

The value of a European call option is given as a function of state variables and parameters:

$$c^{SVJ} = e^{-rT}[F\Pi_1 - X\Pi_2] \tag{I.5}$$

with $F = E(S_T) = Se^{(r-d_y)T}$ the forward price of the underlying asset, $E(.)$ the expectation with respect to the risk-neutral probability measure and S_T the price of S at option's maturity. Evaluation of Π_1 and Π_2 is done by using the moment generating functions of $\ln(S_T / S)$. The following expressions are needed to compute Π_1 and Π_2:

$$\begin{aligned} F_j(\Phi \,|\, V,T) = \exp\{C_j(T;\Phi) + D_j(T;\Phi)V + \lambda T(1+\bar{\kappa})^{\mu_j + 0.5} \\ \times [(1+\bar{\kappa})^{\Phi} e^{\theta^2(\mu_j \Phi + \Phi^2/2)} - 1]\} \qquad j = 1, 2 \end{aligned} \tag{I.6}$$

$$\begin{aligned} C_j(T;\Phi) = (r - d_y - \lambda\bar{\kappa})\Phi T - \frac{\alpha T}{\sigma_v^2}(\rho\sigma_v \Phi - B_j - G_j) \\ - \frac{2\alpha}{\sigma_v^2}\ln\left[1 + 0.5(\rho\sigma_v\Phi - B_j - G_j)\frac{1-e^{G_j T}}{G_j}\right] \end{aligned} \tag{I.7}$$

$$D_j(T;\Phi) = -2\frac{\mu_j\Phi + 0.5\Phi^2}{\rho\sigma_v\Phi - B_j + G_j\dfrac{1+e^{G_jT}}{1-e^{G_jT}}} \tag{I.7.1}$$

$$G_j = \sqrt{(\rho\sigma_v\Phi - B_j)^2 - 2\sigma_v^2(\mu_j\Phi + 0.5\Phi^2)} \tag{I.7.2}$$

$$\mu_1 = 0.5, \quad \mu_2 = -0.5, \quad B_1 = \beta - \rho\sigma_v, \quad B_2 = \beta \tag{I.7.3}$$

and the resulting probabilities Π_1 and Π_2 are derived by numerically evaluating the imaginary part of the Fourier inversion:

$$prob(S_T e^{(r-d_y)T} > X \mid F_j) = 0.5 + \frac{1}{\pi}\int_0^{\infty} \frac{imag[F_j(i\Phi)e^{-i\Phi\chi}]}{\Phi} d\Phi \qquad (I.8)$$

with $\chi \equiv \ln(X/S)$ and the integrals to be evaluated with an adaptive Lobatto quadrature. By constraining the jump component values to equal zero someone can get European call prices for the SV model.

I.2.2. Deterministic volatility functions

I.2.2.1. Black and Scholes deterministic volatility functions

Dumas et al. (1998) estimate DVF of quadratic forms that provide unique per contract volatility estimates. According to Dumas et al. (1998), this approach of smoothing the BS implied volatilities across strike prices and maturities exhibits superior in and out of sample performance for pricing European options. According to Christoffersen and Jacobs (2004) the DVF approach does not constitute a proper and fully specified alternative to other structural option pricing models but is a convenient way to mitigate the BS deficiencies. In addition, Berkowitz (2004) has demonstrated theoretically that the DVF constitutes a reduced-form approximation to an unknown structural model which under frequent re-estimation can exhibit exceptional pricing performance. Christoffersen and Jacobs (2004) demonstrate that Ordinary Least Squares (OLS) estimates of the DVF parameters yield biased predictions of the observed option prices. They emphasize the importance of deriving the DVF by optimizing in respect to the option pricing function via Nonlinear Least Squares (NLS). For the analysis three different DVF model versions as in the thesis of Dumas et al. (1998) are considered:

DVF#1: $\sigma = \max(0.01, a_0 + a_1 X + a_2 X^2)$ (I.9)

DVF#2: $\sigma = \max(0.01, a_0 + a_1 X + a_2 X^2 + a_3 T + a_4 XT)$ (I.10)

$$\text{DVF\#3}: \sigma = \max(0.01, a_0 + a_1 X + a_2 X^2 + a_3 T + a_4 XT + a_5 T^2) \quad \text{(I.11)}$$

I.2.2.1. Extending the deterministic volatility functions

DVF is implemented not only for BS but also for a first time in this thesis for the CS (CS-DVF) and the SVJ (SVJ-DVF) models. We estimate the coefficients for the three different DVF models each day using OLS (*Lc*) and also using NLS (*NLc*). For the latter several initializations are used in order to minimize the risk of estimating coefficients based on a local minimum of the optimization function.

I.3. Nonparametric models

Researchers have drawn attention to the use of nonparametric techniques like feedforward artificial neural networks that can be used for nonlinear regression. The key power provided by this type of methods is that they rely on fairly simple algorithms and the underlying nonlinearity can be learned from transactions data (see Duda et al., 2001, for further details). In addition, they are universal function approximators with good out of sample generalization abilities (see Cybenko, 1989; for a general discussion of neural networks in financial econometrics see Tsay, 2002). Below there is a brief explanation about the different approaches developed and used in this thesis[4].

I.3.1. Standard and hybrid feedforward artificial neural networks

A feedforward artificial neural network is a collection of interconnected simple processing elements structured in successive layers and can be depicted as a network of arcs/connections and nodes/neurons (refer to Figure 1.1 of first

[4] All nonparametric models presented in this study can be also used for deriving the implied risk neutral density function from the cross section of option prices. This can be performed after estimating the nonparametric functional forms and taking twice the partial derivative of the option value with respect to the option's strike price. The methodology behind this idea has been suggested by Breeden and Litzenberger (1978) and has been recently employed with nonparametric techniques by Ait-Sahalia and Lo (1998).

essay). The traditional networks used in this thesis have three layers: an *input* layer with N input variables, a *hidden* layer with H neurons, and a single neuron *output* layer. Each connection is associated with a *weight*, w_{ki}, and a *bias*, b_k, in the hidden layer and a *weight*, v_k, and a *bias*, v_0, for the output layer (k = 1,2,…,H, i = 1,2,…,N). A particular neuron node is composed of: *i)* the vector of *input signals*, *ii)* the *vector weights* and the associated *bias*, *iii)* the *neuron* itself that *sums* the product of the input signal with the corresponding weights and bias, and finally, *iv)* the neuron *transfer function*. In addition, the outputs of the hidden layer ($y_1^{(1)}, y_2^{(1)} \ldots y_H^{(1)}$) are the inputs for the output layer. Inputs are set up in feature vectors, $\tilde{x}_q = [x_{1q}, x_{2q} \ldots, x_{Nq}]$ for which there is an associated and known target, $Y \equiv t_q$, with $q \equiv 1,2,\ldots,P$, where P is the number of the available sample features. The operation carried out for estimating output y, is the following:

$$y = f_0[v_0 + \sum_{k=1}^{H} v_k f_H(b_k + \sum_{i=1}^{N} w_{ki} x_i)] \qquad (I.12)$$

For the purpose of this thesis ANN architectures with only one hidden layer are considered since they can operate as a nonlinear regression tool and can be trained to approximate most functions arbitrarily well (Cybenko, 1989). High accuracy can be obtained by including enough processing nodes in the hidden layer.

The training of ANNs is a non-linear optimization process in which the network's weights are modified according to an error loss function. The error function between the estimated response y_q and the actual response t_q is defined as:

$$e_q(w) = y_q(w) - t_q \qquad (I.13)$$

where, w is an n-dimensional column vector containing the weights and biases given by: $w = [w_{10} \ldots, w_{H0}, w_{11}, \ldots, w_{HN}, v_0, \ldots, v_H]^T$. The modified Levenberg-

Marquardt (LM) algorithm is utilized for estimating the ANNs. According to LM, the weights and the biases of the network are updated in such a way so as to minimize the following sum of squares performance function:

$$F(w) = \sum_{q=1}^{P} e_q^2 \equiv \sum_{q=1}^{P} \left(y_q(w) - t_q(w)\right)^2 \qquad \text{(I.14)}$$

Then, at each iteration τ of the estimation algorithm, the weights vector w is updated as follows:

$$w_{\tau+1} = w_\tau - [J^T(w_\tau)J(w_\tau) + \mu_\tau I]^{-1} J^T(w_\tau)e(w_\tau) \qquad \text{(I.15)}$$

where I is an $n \times n$ identity matrix, $J(w)$ is the $P \times n$ Jacobian matrix of the *P-dimensional* output error column vector $e(w)$, and μ_τ is like a learning parameter that is adjusted in each iteration in order to secure convergence. Further technical details about the implementation of Levenberg-Marquardt algorithm can be found in Hagan and Menhaj (1994) and Hagan et al. (1996).

For the needs of this thesis, ANNs are implemented by using two different target (desired) function. The first one that is called the standard target function given by:

$$t = c^{mrk} / X \qquad \text{(I.16)}$$

where c^{mrk} is the market price for a call option. In addition, the so called hybrid target function is also used. This target function is comprised by the residual between the actual call market price and call option estimate given by a parametric model:

$$t \equiv c^{mrk} / X - c^{\Omega} / X \qquad \text{(I.17)}$$

with Ω defines inputs from the specific parametric models (usually from BS and CS models) .

I.3.2. Robust feedforward artificial neural networks

To get robust estimates for the ANNs' vector parameter w, the Huber function can be considered (i.e. Huber, 1981, Bandler et al., 1993):

$$E(w) = \sum_{q=1}^{P} \rho_k(e_q(w)) \tag{I.18}$$

where ρ_k is the Huber function specified as:

$$\rho_k(e) = \begin{cases} 0.5e^2 & \text{if } |e| \le k \\ k|e| - 0.5k^2 & \text{if } |e| > k \end{cases} \tag{I.19}$$

and k is a positive constant. It is obvious that when $|e| > k$ the Huber function treats the error in the l_1 (least absolute) sense and in the l_2 (least square) sense only if $|e| \le k$ depending on the value of threshold parameter k. The Huber function has a smooth transition between the two norms at $|e| = k$, so that the first derivative of ρ_k is continuous everywhere.

The choice of k defines the threshold between *large* and *small* errors. Different values of k determine the proportion of the errors to be treated in the l_1 or the l_2 norm. As seen, when k is sufficiently large the Huber function encompasses the widely used and conventional least squares (l_2) training of the ANNs. As the k parameter approaches zero, the Huber function approaches the l_1 function and the errors are penalized in the least absolute sense. The Huber function should be more robust to abnormal data since it penalizes them less compared to the l_2 norm.

I.3.3. Nonparametric generalized parameter functions

In this thesis in order to develop the Generalized Parameter Functions (GPFs) a more general network structure is used. In this case the proposed network model under scrutiny has four layers. The first three are typical layers of an ANN as explained before with the exception that more than one neurons can exist at the output layer. The addition of a fourth layer, which is called the *enhanced layer*, makes possible for a chosen POPM to be an *inseparable* part of the network's structure (obtaining in this sense the enhanced Parametric Options Pricing Models, ePOPMs). Under this setting it can be hypothesized that the network structure embeds knowledge from the parametric model during training. If we let X_S to denote the set of all input variables that are necessary for the parametric model to price options, then (refer to Figure 3.1, of the third essay) $X_{S2} \subseteq X_S$ corresponds to the *enhanced* variables coming from the network's output layer and $X_{S3} \subset X_S$ those variables that are passed to the parametric model directly, $X_{S3} = X_S - X_{S2}$. Moreover, X_{S1} represents inputs to the nonparametric model with $X_{S2} \subseteq X_{S1} \subseteq X_S$.

Under this approach, the operation carried out for computing the final estimated output, y, is the following:

$$y = f_{PM}(\upsilon, X_{S2}) \qquad \text{(I.20)}$$

and,

$$\upsilon = [\upsilon_1, \upsilon_2, \ldots, \upsilon_M] \qquad \text{(I.21)}$$

where υ represented the *enhanced* variable vector that is given by:

$$\begin{aligned} \upsilon_1 &= f_M[w_{10}^{(2)} y_0 + \sum_{i=1}^{H} w_{1i}^{(2)} f_H(w_{i0}^{(1)} x_{s0} + \sum_{n=1}^{N} w_{in}^{(1)} x_n)] \\ \vdots \quad & \vdots \qquad\qquad\qquad \vdots \\ \upsilon_M &= f_M[w_{M0}^{(2)} y_0 + \sum_{i=1}^{H} w_{Mi}^{(2)} f_H(w_{i0}^{(1)} x_{s0} + \sum_{n=1}^{N} w_{in}^{(1)} x_n)] \end{aligned} \qquad \text{(I.22)}$$

The above expression follows the functional form of a typical three-layer network with $f_M(.)$ and $f_H(.)$ to be smooth monotonically increasing transfer functions (like log-sigmoid and tangent sigmoid) associated with the output and hidden layer respectively, x_n, n=1,2,…,N, to be the inputs to the network, $w_{in}^{(1)}$, $w_{i0}^{(1)}$ (i=1,2,…,H, n=1,2,…N) to be the weights of the input layer and $w_{ji}^{(2)}$, $w_{j0}^{(2)}$ (j=1,2,…,M) to be the weights of the hidden layer. The M elements of Eq. (I.22) are estimated simultaneously. The vector defined by the right hand side of Eq. (I.22) is the called *generalized parameter function* which produces the enhanced variables. To let the network learn the underlying relationship, its weights are adjusted in order to minimize a sum of squares loss function of the error between the network output and the desire target values.

I.3.4. Support vector machines for function approximation

All methodologies employed in the first three chapters were estimated by minimizing an *empirical* risk functional of the form:

$$\arg\min_{f \in C_l} R_{emp}[f] = \frac{1}{n}\sum_{i=1}^{n} L(x_i, y_i, f(x_i)) \qquad (I.23)$$

where $L(x_i, y_i, f(x_i))$ represents a general loss function determining how estimation errors are penalized and C_l represents a general class of continuous functions. As mentioned before, depending on the application, $L(x_i, y_i, f(x_i))$ could be the sum-of-squared-errors or even the Huber (1981) function that can be used for robust estimation.

However if C_l has very high capacity/flexibility and someone is dealing with few data in high-dimensional spaces then to avoid over-fitting and secure good generalization properties then it might be better to minimize a *regularized* risk functional of the form (see Vapnik, 1995 and Smola et al., 1998):

$$\arg\min_{f \in C_l} R_{reg}[f] = R_{emp}[f] + C\|w\|^2 \qquad (I.24)$$

where $C > 0$ is the so called regularization constant that controls for the capacity and smoothens of the estimated approximation and $\|w\|^2$ defines the complexity of the model. Support Vector Machines (SVMs) is one promising candidate methodology that builds on this idea and is widely used in electrical engineering, bioinformatics, pattern recognition, text analysis, computer vision and widely neglected in financial econometrics. SVM can be implemented using the so called $\varepsilon-$insensitive loss function:

$$|y - f(x)|_{\varepsilon} = \max\{0, |y - f(x) - \varepsilon|\} \qquad \text{(I.25)}$$

which does not penalize errors below some $\varepsilon > 0$. SVM can be used for function approximation via linear and nonlinear regression and is has been evolved in the framework of *statistical learning theory* of Vapnik and Chervonenkis (1974) (so called VC theory) for learning machines (see Vapnik, 1995, for extensive details). The main advantage of SVMs over other nonparametric techniques is that they encompass statistical properties that enables them to generalize satisfactorily well to unseen data. One significant characteristic is that under SVMs someone solves a convex optimization problem with a unique global (and sparse) solution while other nonparametric methods can have non-convex error functions which entail the risk of having multiple local minima solutions. Another one significant feature is that SVMs employ VC theory to select function approximations based on the (out of sample) upper bound of the model's generalization error, which is defined in a strictly statistical framework without restricting the form of the data generating mechanism, and controlling in specific ways the model's parameterization to preclude overfitting.

I.4. Data and parameter estimates

The data for this research come from two dominant world markets, the New York Stock Exchange (NYSE) for the S&P 500 equity index and the Chicago Board of Options Exchange (CBOE) for call option contracts. The S&P 500 Index call options are considered because this option market is extremely liquid and one of the most popular index options traded on the CBOE. This market is the closest to the theoretical setting of the parametric models (see also Constantinides et al., 2008).

In the first two essays we use data for the period 1998 to 2001 while for the third and fourth essays we use data for the period 2002-2004. We implement filtering rules like in Bakshi et al. (1997). The following summarizes the filtering rules adopted in each essay (the superscript indicates the essay for which essay the filtering rule is effective) according to which certain observations are discarded:

Eliminate all zero volume transactions [#3, #4]

Eliminate an observation if call price at day t-1 is equal to call price at day t and if the open interest for these days stays unchanged and if the underlying asset has changed [#1, #2]

Eliminate call option prices less than 1 index point [#1, #2, #3, #4]

Eliminate all call option values that violate the upper and lower arbitrage bounds [#1, #2, #3, #4]

Eliminate observations based on the following moneyness criterions:

No eliminations based on a moneyness criterion [#1, #2]

Keep only observations with $S/X \in [0.85, 1.35]$ (and also with less than 180 trading days to maturity) [#1]

Keep only observations with $S/X \in [0.80, 1.20]$ [#3, #4]

Eliminate option transactions with less (more) than 6 (260) trading days until expiration [#1, #2, #3, #4]

Discard maturity with less than four option contracts [#1, #2, #3, #4]

Option transactions with implied volatility outside [5%, 70%] [#3, #4]

The following summarizes the most important characteristics of the data used in the four essays.

Essay	Period	S/X	T	σ_{imp}
# 1	1998 – 2001	all available & [0.85, 1.35]	[5, 260] & [5, 180]	All available
# 2	1998 – 2001	all available	[5, 260]	All available
# 3	2002 – 2004	[0.80, 1.20]	[5, 260]	[5%, 70%]
# 4	2002 - 2004	[0.80, 1.20]	[5, 260]	[5%, 70%]

Studies like Rubinstein (1994) and Jackwerth (2004) also filter options data for butterfly spread violations. We do not check our data for this type of arbitrage violation since the main literature we follow has ignored it (e.g. Bakshi et al., 1997, Bates 1996, 2000, etc). This filtering rule should be most significant for studies that estimate the implied risk neutral density function directly from options data. Nevertheless, we have run some checks on our data for this type of filtering rule for the dataset we used in the third and fourth essays. We found that butterfly spreads are violated for about 4% of the total sample but with an insignificant mean violation value of 0.083 (approximately equal to 0.1% of the call prices involved in the violations). We firmly believe that these violations would not affect the quality of the reported results.

Compared to the existing literature, this thesis examines more explanatory variables including historical, weighted average implied and pure implied parameters. Also, instead of constant maturity risk-less interest rate, nonlinear interpolation is used for extracting a continuous rate according to each option's time to maturity.

I.4.1. Observed and historically estimated parameters

Below follows an explanation for the relevant input and output variables used for fulfilling the current thesis. Some of these are used in all different chapters whilst other are used only in some of the different parts of the thesis.

I.4.1.1. Observed variables and parameters measures

Moneyness Ratio (S/X): The moneyness ratio may explicitly allow the nonparametric methods to better learn the moneyness bias associated with the BS (see also Garcia and Gencay, 2000). The dividend adjusted moneyness ratio $(Se^{-\delta T})/X$ is used in this thesis with ANNs and SVMs because it is more informative since dividends affect the options pricing mechanism.

Time to maturity (T): For each option contract, trading days are computed assuming 252 days in a year.

Risk-less interest rate (r): Most of the studies use 90-day T-bill rates (or similar when this is unavailable) as approximation of the interest rate. In this thesis nonlinear cubic spline interpolation is used for matching each option contract with a continuous interest rate that corresponds to the option's maturity. This is done by utilizing T-bill rates collected from the U.S. Federal Reserve Bank Statistical Releases.

Historical Volatilities (σ)*:* The 60-day historical volatility is calculated using all the past 60 index log-returns.

CBOE VIX Volatility Index: It was developed by CBOE in 1993 and is a measure of the volatility of the S&P 500 Index. VIX is calculated by taking the weighted average of the implied volatilities of eight S&P 500 Index call and put options with an average time to maturity of 30 days.

Skewness and Kurtosis: The 60-day skewness and kurtosis needed for the CS model are approximated from the sixty most recent log-returns of the S&P 500.

I.4.1.2. Implied parameters

For extracting the implied parameters for the POPMs, the Whaley's (1982) simultaneous equation procedure is considered in this thesis. This methodology minimizes a price deviation function with respect to the unobserved parameters. The market option prices (c^{mrk}) are assumed to be the corresponding model prices (c^k, k defining input from a parametric model – e.g. BS, CS, SVJ, SV) plus a random additive disturbance term. For any option set of size N_t (it refers to the number of different call option transaction datapoints available on a specific day), the difference:

$$\varepsilon_{N_t}^{k} = c_{N_t}^{mrk} - c_{N_t}^{k} \tag{I.26}$$

between the market and the model value of a certain option is a function of the values taken by the unknown parameters. To find implied parameter values the following unconstrained optimization problem is considered:

$$SSE(t) = \min_{\theta^k} \sum_{l=1}^{N_t} (\varepsilon_l^k)^2 \tag{I.27}$$

where t represents the time instance, and θ^k the unknown parameters associated with a specific POPM. To minimize the possibility of obtaining implied parameters that correspond to a local minimum of the error surface (see also Bates, 1991, and Bakshi et al., 1997), several starting values are used for the parameters of each model. This methodology is implemented under various schemes in order to derive daily implied parameter values for the POPMs considered.

I.4.1.3. Validation, testing and pricing performance measures

In order to estimate the nonparametric model, the available data points are divided/splitted into training, validation and testing sub-datasets in a chronological manner via a rolling-forward procedure. Depending on the case considered, the available dataset is divided into a number of different overlapping training (*Tr*) and validation (*Vd*) sets, each followed by separate and non-overlapping testing (*Ts*). Since a practitioner is faced with time-series data, it was decided to partition the available data based on this rationale since it allows frequent re-estimation of the nonparametric models so as to keep a reasonable track of the time-variation of the option valuation relationships between the input/output variable combinations. All testing sub-sets are pooled forming in this way an aggregate dataset (*AggTs*). For this aggregate dataset various error metrics are reported in order to determine the pricing accuracy of each model considered. The Root Mean Square Error (RMSE) is considered to be the most important error measure of this thesis since most of the parametric and nonparametric methodologies are effectively calibrated with respect to a sum-of-squares loss function. This treatment is in line with the intuition behind the study of Christoffersen and Jacobs (2004) that suggest that better estimation results can be obtained when the estimation and evaluation loss functions are aligned. Nevertheless, we also report the Mean Absolute Error (MAE) since for nonlinear models or data that exhibit nonlinearities MAE is sometimes considered as a better criterion given that it is more robust to extreme observations.

I.5. Four Essays on Empirical Options Pricing: Descriptions and Results

Below, a brief description of the methodologies developed and the results obtained in each of the four essays is provided.

I.5.1. Summarizing Essay #1: Pricing and Trading European Options by Combining Artificial Neural Networks and Parametric Models with Implied Parameters

In this essay the BS and CS models are compared with several ANN configurations with respect to pricing the S&P 500 European call options. The standard and hybrid function are implemented with ANNs. In previous studies the standard steepest descent backpropagation algorithm was (mostly) used for training the feedforward ANNs. As it is shown in Charalambous (1992) this learning algorithm is often unable to converge rapidly to the optimal solution. Thus in this essay the modified Levenberg-Marquardt algorithm is utilized which is much more sophisticated and efficient in terms of time capacity and accuracy (Hagan and Menhaj, 1994). In contrast to most previous studies, a different network configuration is used per period based on the early stopping technique and a thorough cross-validation strategy.

Although previous researchers have exploited BS or ANNs, little has been reported for the case of CS and nothing for the hybrid ANNs that use information derived by CS. To investigate the economic significance of the alternative option pricing approaches, trading strategies without and with the inclusion of transaction costs are utilized. These trading strategies are implemented with the standard single instrument delta-hedging values implied by each model, but also with the corrected values according to the (widely neglected) Chen and Johnson (1985) methodology. In order to check the robustness of the results, in addition to the full dataset that considers broad range of strike prices and time to maturity options, the whole analysis is repeated by using a reduced dataset that has been considered in previous studies (i.e. Hutchison et al., 1994).

Regarding the in sample pricing, CS performs better than the BS model (with the exception of the case of the contract specific implied parameters that practically eliminate the pricing error). Regarding the out of sample pricing, CS outperforms BS with the use of overall average (one per day) implied parameters, but BS is still a better model when the contract specific (one per contract) implied parameters are used; in general, implied parameters lead to better performance than the historical ones or the VIX volatility proxy; it is found that ANNs estimated based on the standard target function cannot outperform the parametric models in the full range of data, but this result does not necessarily holds for the reduced data set; hybrid neural networks that combine both neural network technology and the parametric models provide the best performance, especially when contract specific implied parameters are used. The BS based hybrid ANN (with contract specific parameters) is the overall best performer, and the equivalent CS hybrid often a good alternative.

In trading and before transaction costs, models using contract specific implied parameters provide the best performance. But they also lead to the highest number of trades. In trading when transaction costs are accounted for in a naive manner, profits practically in all cases disappear. On the contrary when dynamic cost-efficient strategies are implemented profits are present at reasonable levels of transaction costs hinting thus to potential market inefficiencies. The parametric BS with contract specific volatility is the best among the parametric models. The hybrid ANN based on BS with contract specific volatility is again the overall best.

In this essay it is also shown that by implementing the widely neglected Chen and Johnson (1985) modified hedging approach the profitability of trading strategies can be improved considerably for parametric models that use overall average (one per day) implied parameters (the models more consistent with the assumptions behind the modified hedging approach). This approach did not affect the choice of the overall best model in terms of trading with transaction costs. But it did demonstrate that reasonable alternatives for trading do exist without the need to resort to the extra sophistication of the ANNs technology.

I.5.2. Summarizing Essay #2: Robust Artificial Neural Networks for Pricing of European Options

The scope of this chapter is to compare alternative standard and robust ANN configurations with respect to pricing the S&P 500 European call options. Robust ANNs that use the Huber (1981) function are developed, and configurations that are optimized based solely on the least squares norm are compared with robust configurations that are closer to the least absolute norm. Like in the first essay, the standard and hybrid ANN target functions with historical and implied parameter measures are employed.

In previous empirical research on option pricing, ANNs have been optimized based on the l_2 (the least squares) norm. The l_2 norm is a convenient way to train ANNs. Of course, the least squares optimization is highly susceptible to the influence of large errors since some abnormal datapoints (or few outlier observations) can deliver non-reliable networks. On the contrary, robust optimization methods that exploit the l_1 (the least absolute) norm are unaffected by large (or catastrophic) errors but are doomed to fail when dealing with small variation errors.

In this essay the Huber function (Huber, 1981) is used as the loss function during the ANNs optimization process. The Huber function utilizes the robustness of l_1 and the unbiasedness of l_2 and has proved to be an efficient tool for robust optimization problems for various tasks (i.e. Bandler et al., 1993), albeit it does not constitute the mainstream. The Huber function has been considered because it is widely referenced on robust estimation (Bishop, 1995), it provides a simple generalization of the least squares approach; it avoids the need for any probabilistic assumptions, and does not lead to complex mathematical expressions when used with ANNs.

Regarding the out of sample pricing, the hybrid models outperform both the standard ANNs and the parametric ones. Huber optimization improves significantly the performance of both the standard and the hybrid ANNs. The non-hybrid ANNs are affected more by large errors. The overall best models are the Huber based hybrid ANNs. In general, within each class, the best performing Huber model has considerably smaller probability of large mispricing compared to the least squares counterpart. Regarding the economic significance of the models, the Huber models are the overall best models.

I.5.3. Summarizing Essay #3: Generalizing the Deterministic Volatility Functions for Enhanced Options Pricing

The broader scope of this essay is to propose a nonparametric enhancement of the implied parameter values used in the POPMs, generalizing thus the DVF method of Dumas et al. (1998) (see also Christoffersen and Jacobs, 2004). The proposed approach results in Generalized Parameter Functions (GPFs) that allow an enhancement of parameters without specifying a deterministic functional form. The nonparametric parameter enhancement provides the volatility to the BS and CS models. In addition, skewness or skewness and kurtosis can be enhanced for the CS model. A significant feature of the methodology is that it allows a set of the input variables to the parametric model to be jointly determined by the generalized parameter functions. The proposed approach has the following important features. First, it retains the theoretical properties of the parametric model being enhanced concerning the desire for: *i*) nonnegative option values (thus expecting satisfactory pricing performance at the boundary of option pricing areas, in both dense and sparse input areas), *ii*) theory consistent Greek letters, and *iii*) nonnegative implied state price densities. Second, as conjectured by Bandler et al. (1999), nonparametric techniques that incorporate knowledge regarding the nature of the problem should need a smaller amount of training samples and also reduce the number of free parameters needed for estimation to exhibit a satisfactory performance in out of sample testing as opposed to the case of standard nonparametric approaches. Third, the approach compared to the DVF and Whaley (1982) combines two important characteristics (see discussions in Christoffersen and Jacobs, 2004, p. 313). It has enhanced precision in parameter estimates due to long term estimation of the GPF and at the same time captures the time-variation of the option valuation relationship since input to the nonparametric structure is calibrated daily.

In this essay ePOPMs are developed for the case of BS and of CS models. These are then compared with their parametric alternatives using the overall average implied parameters and their DVF versions in pricing S&P 500 index call options. Part of the contribution is to apply the DVF approach to the CS model. Moreover, the SVJ model of Bates (1996) is used as benchmark since it is an effective remedy to the BS biases (see Bakshi et al., 1997, and Bates, 1996); results for the SV sub-model are also reported.

Regarding the results it is first shown that daily calibration of either SVJ or the DVF based BS and CS models requires careful search. In the sample, SVJ has the best fit while SV is inferior to the best DVF models. The out of sample results strongly support the proposed methodology. The first important finding is that the DVF approach when applied to CS provides results superior to CS (with overall average parameter estimates) and also to BS (with either overall average or DVF estimates). The second is that the SVJ model is the best model among the parametric models whilst the SV is inferior to DVF based BS and CS models. The third is that the increase in the pricing accuracy of the enhanced BS and CS models over the best performing BS and CS parametric ones is considerable and statistically significant. In general, the best enhanced models estimated monthly are comparable to the daily estimated SVJ model. In addition, it is shown that the enhanced methodology is robust both to the complexity of the generalized parameter functions, and to the pricing of contracts not used during estimation. Consistently with the recommendation in Christoffersen and Jacobs (2004) it is observed that hedging results using ePOPMs chosen using a hedging criterion outperform both the parametric models and the ePOPMs chosen using a pricing criterion.

I.5.4. Summarizing Essay #4: Functional Estimation for Options Pricing Via Support Vector Machines

The focus of this essay is to investigate the pricing performance of Support Vector Machines (SVMs) for pricing S&P 500 index call options. SVMs comprise is a novel nonparametric methodology that has been evolved in the framework of statistical learning theory (see Vapnik, 1995, for extensive details) and can be utilized for problems involving linear or nonlinear function approximations. Unlike ANNs, SVMs have not gained yet any significant popularity in financial econometric applications although they are widely used in electrical engineering, bioinformatics, pattern recognition, text analysis, computer vision etc. The main advantage of SVMs over other nonparametric techniques is that they encompass statistical properties that enables them to generalize satisfactorily well to unseen data. SVMs employ the so called VC theory (see Vapnik and Chervonenkis, 1974), which is defined in a strictly statistical framework, that controls in specific ways the model's estimation and parameterization to preclude overfitting so that to ensure good out of sample (generalization) results.

Based on the theory that underlies SVM, their superiority over ANNs should be more obvious in datasets of small and moderate size (see Vojislav, 2001). In addition, their estimation is much more efficient in terms of time spent on the training/optimization for small datasets. For this reason in this essay the SVMs are estimated with short-time span data sets.

The main contribution of this essay regards the application of SVM for options pricing and their comparison with other alternative pricing approaches. Two types of SVMs are considered. The first is the traditional SVMs that were originally developed by Vapnik and are based on the ε-insensitive loss function (see Vapnik, 1995) which are considered to be more robust when noise is non Gaussian. The second is the Least Squares Support Vector Machines (LS-SVM) which is a subsequent variant of the original SVMs methodology originally proposed by Suykens and co-workers (see Suykens et al., 2002). Compared to SVMs, LS-SVMs are more robust when noise is Gaussian and they rely on fewer tuning hyper-parameters. Most importantly LS-SVMs minimize a least squares loss function which is most common in empirical options pricing studies (see Christoffersen and Jacobs, 2004). Another contribution of this essay regards the application of ANNs in small datasets. Most previous studies reviewed in the first three essays employ ANNs with rather large datasets.

The dataset, the alternative POPMs and the methodology to derive the implied parameters are the same as the one used in the previous (third) essay. SVMs and LS-SVMs are developed by using the standard and the hybrid target functions and are compared with standard and hybrid ANNs (based on the estimation methods employed in the first essay). The empirical results show that SVMs can produce pricing results that are comparable to the more sophisticated parametric models like the SVJ model. In addition, it is found that the ANNs, especially the ones developed with the hybrid target function, perform exceptionally well with small datasets. This is a new observation since up to this moment ANNs have been tested using rather large datasets. Overall the performance of SVMs and ANNs seems to be comparable. As explained by Smola and Schölkoph (1998), it is possible for ANNs to achieve similar performance with SVMs. Nevertheless, as there are only two to three critical parameters in SVMs (compared to usually few dozens for ANNs), it may be more convenient and easier to use SVMs.

Four Essays on Empirical Options Pricing

1. Pricing and Trading European Options by Combining Artificial Neural Networks and Parametric Models with Implied Parameters

Abstract

We compare the ability of the parametric Black and Scholes, Corrado and Su models, and Artificial Neural Networks to price European call options on the S&P 500 using daily data for the period January 1998 to August 2001. We use several historical and implied parameter measures. Beyond the standard neural networks, in our analysis we include hybrid networks that incorporate information from the parametric models.

Our results are significant and differ from previous literature. We show that the Black and Scholes based hybrid artificial neural network models outperform the standard neural networks and the parametric ones. We also investigate the economic significance of the best models using trading strategies (extended with the Chen and Johnson modified hedging approach). We find that there exist profitable opportunities even in the presence of transaction costs.

The existing chapter had been submitted for publication in the **European Journal of Operational Research** and it is forthcoming in volume 185, issue 3, 16 March 2008, pg. 1415-1433.

1.1. Introduction

In this essay we compare parametric option pricing models (POPMs) -- Black and Scholes (1973) (BS) and the semi-parametric Corrado and Su (1996) (CS) -- with several artificial neural network (ANN) configurations. We compare them with respect to pricing the S&P 500 European call options, and trading strategies are implemented in the presence of transaction costs.

Black and Scholes introduced in 1973 their milestone POPM. Despite the fact that BS and its variants are considered as the most prominent achievements in financial theory in the last three decades, empirical research has shown that the formula suffers from *systematic biases* (see Black and Scholes, 1975, MacBeth and Merville, 1980, Gultekin et al., 1982, Rubinstein, 1994, Bates, 1991 and 2003, Bakshi et al., 1997, Andersen et al., 2002, and Cont and Fonseca, 2002). The BS bias stems from the fact that the model has been developed under a set of simplified assumptions such as geometric Brownian motion of stock price movements, constant variance of the underlying returns, continuous trading on the underlying asset, constant interest rates, etc.

Post-BS research (e.g. stochastic volatility, jump-diffusion, stochastic interest rates, etc.) has not managed to either generalize all the assumptions of BS or provide results truly consistent with the observed market data. These models are often too complex to implement, have poor out of sample pricing performance and have implausible and sometimes inconsistent implied parameters (see Bakshi et al., 1997). This justifies the severe time endurance of BS[5]. Together with the BS model, we also consider the semi-parametric CS model that allows for excess skewness and kurtosis, as a model that can proxy for other more complex parametric ones.

Nonparametric techniques such as *Artificial Neural Networks* are promising alternatives to the parametric OPMs. ANNs do not necessarily involve directly any financial theory because the option's price is estimated inductively using historical or implied input variables and option transactions data. Option-pricing functions are multivariate and highly nonlinear, so ANNs are desirable approximators of the *empirical option pricing function*. Parametric models describe

[5] According to Andersen et al., (2002), "*the option pricing formula associated with the Black and Scholes diffusion is routinely used to price European options, although it is known to produce systematic biases*".

a stationary nonlinear relationship between a theoretical option price and various variables. Since it is known that market participants change their option pricing attitudes from time to time (i.e. Rubinstein, 1994) a stationary model may fail to adjust to such rapidly changing market behavior (see also Cont and Fonseca, 2002, for evidence of noticeable variation in daily implied parameters). ANNs if frequently trained can adapt to changing market conditions, and can potentially correct the aforementioned BS bias (Hutchison et al., 1994, Lajbcygier et al., 1996, Garcia and Gencay, 2000, Yao and Tan, 2000).

Beyond the standard ANN target function we further examine the hybrid ANN target function suggested by Watson and Gupta (1996) and used for pricing options with ANNs in Lajbcygier et al. (1997). In the hybrid models the target function is the residual between the actual call market price and the parametric option price estimate. In previous studies the *standard steepest descent backpropagation* algorithm is (mostly) used for training the feedforward ANNs. It is shown in Charalambous (1992) that this learning algorithm is often unable to converge rapidly to the optimal solution. Here we utilize the modified Levenberg-Marquardt (LM) algorithm which is much more sophisticated and efficient in terms of time capacity and accuracy (Hagan and Menhaj, 1994). In contrast to most previous studies, thorough cross-validation allows us to use a different network configuration in different testing periods.

The data for this research come from two dominant world markets, the New York Stock Exchange (NYSE) for the S&P 500 equity index and the Chicago Board of Options Exchange (CBOE) for call option contracts, spanning a period from January 1998 to August 2001. To our knowledge, the resulting dataset is larger than the ones used in other published ANN studies. We also (similarly to Rubinstein, 1994, Bates, 1996, Bakshi et al., 1997; see discussion in Bates, 2003) *reserve* option datapoints that in several ANN studies were dropped out of the analysis. Note that in order to check the robustness of the results we repeated the analysis using a reduced dataset following Hutchison et al. (1994). We examine more explanatory variables including historical, weighted average implied and pure implied parameters. Also, instead of constant maturity riskless interest rate, we use nonlinear interpolation for extracting a continuous rate according to each option's time to maturity.

Lastly, although previous researchers have exploited BS or ANNs, little has been reported for the case of CS[6] and nothing for the hybrid ANNs that use information derived by CS. To investigate the economic significance of the alternative option pricing approaches, trading strategies without and with the inclusion of transaction costs are utilized. These trading strategies are implemented with the standard delta-hedging values implied by each model, but also with the corrected values according to the (widely neglected) Chen and Johnson (1985) methodology.

In the following we first review the BS and CS models, and the standard and hybrid ANN model configuration. Then we discuss the dataset, the historical and implied parameter estimates we derive, and we define the parametric and ANN models according to the parameters used. Subsequently we review the numerical results with respect to the in- and out of sample pricing errors; and we discuss the economic significance of dynamic trading strategies both in the absence and in the presence of transaction costs. The final section concludes. In general, our results are novel and significant. We identify the best hybrid ANN models, and we provide evidence that (even in the presence of transaction costs), profitable trading opportunities still exist.

1.2. The parametric models

The Black Scholes formula for European call options modified for dividend-paying underlying asset is:

$$c^{BS} = Se^{-\delta T} N(d_1) - Xe^{-rT} N(d_2) \quad (1.1)$$

$$d_1 = \frac{\ln(S/X) + (r - \delta)T + (\sigma\sqrt{T})^2/2}{\sigma\sqrt{T}} \quad (1.1.1)$$

$$d_2 = d_1 - \sigma\sqrt{T} \quad (1.1.2)$$

---

[6] An exception is the paper by Sami Vahamaa (2003) that examined the hedging performance of the CS model without including transaction costs.

where, $c^{BS} \equiv$ premium paid for the European call option; $S \equiv$ spot price of the underlying asset; $X \equiv$ exercise price of the option; $r \equiv$ continuously compounded riskless interest rate; $\delta \equiv$ continuous dividend yield paid by the underlying asset; $T \equiv$ time left until the option expiration; $\sigma^2 \equiv$ yearly variance rate of return for the underlying asset; $N(.) \equiv$ the standard normal cumulative distribution.

The standard deviation of continuous returns (σ^2) is the only variable in Eqs. (1.1.1) and (1.1.2) that cannot be directly observed in the market. For this study, we use both historical and implied volatility forecasts. For the Historical Volatility we use the past 60 days. The Implied Volatility (IVL) calculation involves solving Eq. (1.1) iteratively for σ given the values of the observable c^{mrk} (the most recently observed market price of a call option), and the relevant values of S, X, T, r and δ. Contrary to historical volatility, IVL has desirable properties that make it attractive to practitioners: it is forward looking, and avoids the assumption that past volatility will be repeated.

If BS is a well-specified model, then all IVLs on the same underlying asset should be the same, or at least deterministic functions of time. Unfortunately, many researchers have reported systematic biases. For example, Rubinstein (1994) has shown that IVL derived via BS as a function of the moneyness ratio (*S/X*) and time to expiration (*T*) often exhibits a **U** shape, the well known *volatility smile*. Bakshi et al. (1997) report that implicit stock returns' distributions are negatively skewed with more excess kurtosis than allowable in the BS lognormal distribution. This is why we usually refer to BS as being a misspecified model with an inherent source of bias (see also Latane and Rendleman, 1976, Bates, 1991, Canica and Figlewski, 1993, Bakshi et al., 2000, and Andersen et al., 2002). For the aforementioned reason we include in our analysis the Corrado and Su (1996) (see also the correction in Brown and Robinson, 2002) model that explicitly allows for excess skewness and kurtosis. The CS model is a semi-parametric model since it does not rely on specific assumptions about the underlying stochastic process. Corrado and Su define their model as:

$$c^{cs} = c^{BS} + \mu 4 Q_3 + (\mu 4 - 3) Q_4 \qquad (1.2)$$

$$Q_3 = \frac{1}{3!} S e^{-\delta T} \sigma \sqrt{T} ((2\sigma\sqrt{T} - d_1) n(d_1) - \sigma^2 T N(d_1)) \quad (1.2.1)$$

$$Q_4 = \frac{1}{4!} S e^{-\delta T} \sigma \sqrt{T} ((d_1^2 - 1 - 3\sigma\sqrt{T}(d_1 - \sigma\sqrt{T})) n(d_1) + \sigma^3 T^{3/2} N(d_1)) \quad (1.2.2)$$

$$n(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2/2) \quad (1.2.3)$$

where c^{BS} is the BS value for the European call option adjusted for dividends, and μ_3 and μ_4 are the coefficients of skewness and kurtosis of the returns.

1.3. Artificial neural networks

A Feedforward Artificial Neural Network is a collection of interconnected simple processing elements structured in successive layers and can be depicted as a network of arcs/connections and nodes/neurons. Fig. 1 depicts a fully-connected ANN architecture similar to the one applied in this study. This network has three layers: an *input* layer with N input variables, a *hidden* layer with H neurons, and a single neuron *output* layer. Each connection is associated with a *weight*, w_{ki}, and a *bias*, b_k, in the hidden layer and a *weight*, v_k, and a *bias*, v_0, for the output layer (k = 1,2,…,H, i = 1,2,…,N). A particular neuron node is composed of: *i)* the vector of *input signals*, *ii)* the *vector weights* and the associated *bias*, *iii)* the *neuron* itself that *sums* the product of the input signal with the corresponding weights and bias, and finally, *iv)* the neuron *transfer function*. In addition, the outputs of the hidden layer ($y_1^{(1)}, y_2^{(1)} \ldots y_H^{(1)}$) are the inputs for the output layer. Since we want to approximate the market options pricing function, ANNs operate as a non-linear regression tool:

$$Y = G(\tilde{x}) + \varepsilon_{ANN} \quad (1.3)$$

that maps the unknown relation, $G(.)$, between the input variable vector, $\tilde{x} = [x_1, x_2, \ldots, x_N]$, the target function, Y, and the error term, ε_{ANN}. Inputs are set up in feature vectors, $\tilde{x}_q = [x_{1q}, x_{2q} \ldots, x_{Nq}]$ for which there is an associated and known target, $Y \equiv t_q$ (in our case, $t_q \equiv c_q^{mrk} / X_q$), with $q \equiv 1,2,\ldots,P$, where P is the number of the available sample features. According to Fig. 1, the operation carried out for estimating output y (in our case, $y_q \equiv c_q^{ANN} / X_q$), is the following:

$$y = f_0[v_0 + \sum_{k=1}^{H} v_k f_H(b_k + \sum_{i=1}^{N} w_{ki} x_i)] \qquad (1.4)$$

For the purpose of this study, the hidden layer always uses the hyperbolic tangent sigmoid transfer function, while the output layer uses a linear transfer function. In addition, ANN architectures with only one hidden layer are considered since they operate as a nonlinear regression tool and can be trained to approximate most functions arbitrarily well. This is based on the universal approximation theorem provided by Cybenko (1989) (for further details see also Haykin, 1999):

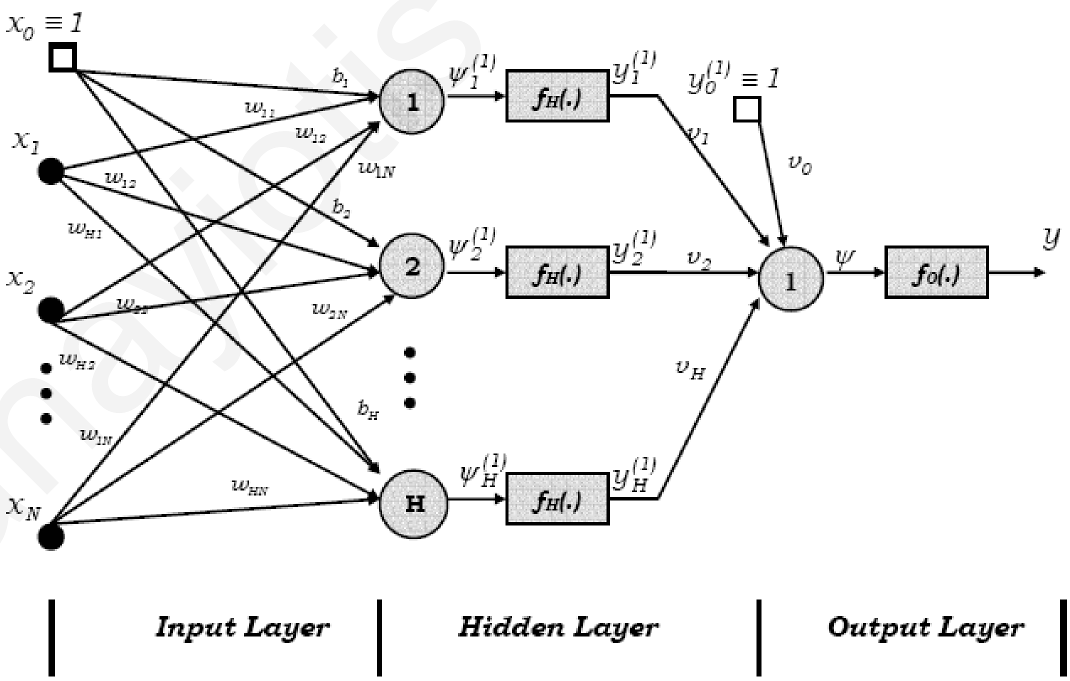

Figure 1.1. A single hidden layer feedforward neural network

Let $f_H(.)$ be a non-constant, bounded and monotone-increasing continuous function. Let l_N denote the N-dimensional unit hypercube $[0,1]^N$. The space of continuous functions on l_N is denoted by $C(l_N)$. Then, given any function $g \in C(l_N)$ and $\varepsilon > 0$, there exist an integer number H and sets of real constants, $w_{k0}, w_{ki}, v_k, k = 1,2,\ldots,H, i = 1,2,\ldots,N$ such that we may define,

$$y(x) = \sum_{k=1}^{H} v_k f_H(w_{k0} + \sum_{i=1}^{N} w_{ki} x_i)$$

as an approximate realization of the function $g(.)$; that is, $|y(x) - g(x)| < \varepsilon$ for all vectors x that lie in the input space. High accuracy can be obtained by including enough processing nodes in the hidden layer.

To train the ANNs, we utilized the modified LM algorithm. According to LM, the weights and the biases of the network are updated in such a way so as to minimize the following sum of squares performance function:

$$F(W) = \sum_{q=1}^{P} e_q^2 \equiv \sum_{q=1}^{P} (y_q - t_q)^2 \equiv \sum_{q=1}^{P} (f_0[v_0 + \sum_{k=1}^{H} v_k f_H(b_k + \sum_{i=1}^{N} w_{ki} x_{iq})] - t_q)^2 \quad (1.5)$$

where, W is an n-dimensional column vector containing the weights and biases: $W = [b_1,\ldots,b_H, w_{11},\ldots,w_{HN}, v_0,\ldots,v_H]^T$. Then, at each iteration τ of LM, the weights vector W is updated as follows:

$$W_{j+1} = W_j - [J^T(W_j)J(W_j) + \mu_j I]^{-1} J^T(W_j)e(W_j) \quad (1.6)$$

where I is an $n \times n$ identity matrix, $J(W)$ is the $P \times n$ Jacobian matrix of the *P-dimensional* output error column vector $e(W)$, and μ_τ is like a learning parameter that is adjusted in each iteration in order to secure convergence. Further technical details about the implementation of LM can be found in Hagan and Menhaj (1994) and Hagan et al. (1996). In addition, to the standard use of

ANNs where $t_q \equiv c_q^{mrk} / X_q$, we also try hybrid ANNs in which the target function is the residual between the actual call market price and the BS or CS call option estimation:

$$t_q \equiv c_q^{mrk} / X_q - c_q^k / X_q \tag{1.7}$$

with k defining inputs from a parametric model. To avoid neuron saturation, we scale input variables using the *mean-variance* transformation (z-score) defined as follows:

$$\tilde{z}_i = (\tilde{x}_i - \mu_i) / s_i \tag{1.8}$$

where $\tilde{x}_i$ is the vector containing all of the available observations related to a certain input/output variable for a specific training period, μ_i is the mean and s_i the standard deviation of this vector. Moreover, we also utilize the network initialization technique proposed by Nguyen and Windrow (see Hagan et al., 1996) that generates initial weights and bias values for a nonlinear transfer function so that the active regions of the layer's neurons are distributed roughly evenly over the input space.

In this study for each input variable set of each training sample, all the available networks having two to ten hidden neurons are cross-validated (in total nine). Moreover, since the initial network weights affect the final network performance, for a specific number of hidden neurons the network is initialized, trained and validated many times. Each network is estimated and optimized using the Mean Square Error (MSE) criterion shown in Eq. (1.5) for no more than two-hundred iterations. The dataset is divided into three sub-sets. The first is the *training (estimation) set*. The second is the *validation set* where the ANN model's error is monitored and the optimal number of hidden neurons and their weights are defined, via an early stopping procedure (MSE fails to decrease in 10 consecutive iterations). Given the optimal ANN structure, its pricing capability is tested in a third separate *testing dataset*.

1.4. Data, parameter estimates and model implementation

Our dataset covers the period January 1998 to August 2001. To our knowledge, the resulting dataset is larger than the one used in other published studies and reserves option data points that in most of the previous studies were dropped out of the analysis. After implementing the filtering rules, our dataset consists of 76,401 data points, with an average of 35,000 data points per (overlapping rolling training-validation-testing) sub-period (see Fig. 2). Hutchison et al. (1994) have an average of 6,246 data points per sub-period. Lajbcygier et al. (1996) include 3,308 data points, Yao et al. (2000) include 17,790 data points, and Schittenkopf and Dorffner (2001) include 33,633 data points. The S&P 500 Index call options are considered because this option market is extremely liquid and one of the most popular index options traded on the CBOE. This market is the closest to the theoretical setting of the parametric models. Along with the index, we have collected a daily dividend yield, δ, provided online by Datastream.

1.4.1. Observed and historically estimated parameters

Moneyness Ratio (S/X): The moneyness ratio may explicitly allow the ANNs to learn the moneyness bias associated with the BS (see also Garcia and Gencay, 2000). The dividend adjusted moneyness ratio $(Se^{-\delta T})/X$ is used in this study with ANNs because it is more informative. The simple moneyness ratio S/X is used in order to tabulate results as in Hutchison et al. (1994). We adopt the following terminology: very deep out of the money (VDOTM) when S/X<0.85, deep out the money (DOTM) when 0.85≤S/X<0.90, out the money (OTM) when 0.90≤S/X<0.95, just out the money (JOTM) when 0.95≤S/X<0.99, at the money (ATM) when 0.99≤S/X<1.01, just in the money (JITM) when 1.01≤ S/X <1.05, in the money (ITM) when 1.05≤S/X<1.10, deep in the money (DITM) when 1.10≤S/X<1.35, and very deep in the money (VDITM) when S/X≥1.35.

Time to maturity (T): For each option contract, trading days are computed assuming 252 days in a year. In terms of time length, an option contract is classified as *short term maturity* when its maturity is less than 60 days, as *medium term maturity* when its maturity is between 60 and 180 days and as *long term maturity* when it has maturity longer than (or equal to) 180 days.

Riskless interest rate (r): Most of the studies use 90-day T-bill rates (or similar when this is unavailable) as approximation of the interest rate. We use nonlinear cubic spline interpolation for matching each option contract with a continuous interest rate, r, that corresponds to the option's maturity, by utilizing the 3-month, 6-month and one-year T-bill rates collected from the U.S. Federal Reserve Bank Statistical Releases.

Historical Volatilities (σ)*:* The 60-day historical volatility is calculated using all the past 60 log-relative index returns and is symbolized as σ_{60}.

CBOE VIX Volatility Index: It was developed by CBOE in 1993 and is a measure of the volatility of the S&P 100 Index[7]. VIX is calculated by taking the weighted average of the implied volatilities of eight S&P 100 Index call and put options with an average time to maturity of 30 days. This volatility measure can only be used with BS and is symbolized as σ_{vix}^{BS}.

Skewness and Kurtosis: The 60-day skewness ($\mu_{3,60}^{CS}$) and kurtosis ($\mu_{4,60}^{CS}$) needed for the CS model are approximated from the sixty most recent log-returns of the S&P 500.

1.4.2. Implied parameters

We adopt the Whaley's (1982) simultaneous equation procedure to minimize a price deviation function with respect to the unobserved parameters. As with Bates (1991), market option prices (c^{mrk}) are assumed to be the corresponding model prices (c^k, k defining input from a parametric model) plus a random additive disturbance term. For any option set of size N_t (N_t refers to the number of different call option transaction datapoints available on a specific day), the difference:

$$\varepsilon_N^k = c_N^{mrk} - c_N^k \qquad (1.9)$$

between the market and the model value of a certain option is a function of the values taken by the unknown parameters. To find optimal implied parameter

---

[7] The S&P 100 Index and S&P 500 Index exhibit 30 day log-return average correlations for the period January 1998 to August 2002 of about 0.98.

values we solve an unconstrained optimization problem that has the following form:

$$SSE(t) = \min_{\theta^k} \sum_{n=1}^{N} (\varepsilon_n^k)^2 \qquad (1.10)$$

where t represents the time instance, and θ^k the unknown parameters associated with a specific POPM ($\theta^{BS} = \{\sigma\}, \theta^{CS} = \{\sigma, \mu_3, \mu_4\}$). The SSE is minimized via a non-linear least squares optimization based on the LM algorithm. To minimize the possibility of obtaining implied parameters that correspond to a local minimum of the error surface (see also Bates, 1991, and Bakshi et al., 1997), with each model we use three different starting values for the unknown parameters based on reported average values in Corrado and Su (1996).

A difference of our approach compared to previous studies is that the above minimization procedure is used daily to derive four different sets of implied parameters for each parametric model. The first optimization is performed by including all available options transaction data in a day to obtain *daily average* implied structural parameters. Alternatively, for a certain day we minimize the SSE of Eq. (1.10) by fitting the BS and CS for options that share the same maturity date as long as four different available call options exist. We thus get *daily average per maturity* parameters. In a third step, for every maturity each available option contract is grouped with its three nearest options in terms of the moneyness ratio in order to minimize the above SSE function, deriving thus parameters *average per the 4 closest* contracts; such estimates are ignored in previous research. We finally calibrate the implied structural parameters, by focusing on the Brownian volatility for each contract so as to drive the residual error to zero or to a negligible value. In the case of BS this is quite simple and we can easily obtain a *contract specific* volatility estimate. For CS we need three structural parameters, so for each call option we minimize Eq. (1.10) with respect to the Brownian volatility after fixing the skewness and kurtosis coefficients to the values obtained from the previous procedure that gave the *average per the 4 closest* implied parameters. Two kinds of constraints are included in the optimization process for practical reasons: nonnegative implied volatility parameters are obtained by using an exponential transformation; and the skewness of CS[8] is permitted to vary in the range [–10, 5] whereas kurtosis is constrained to be less than 30. Unlike previous studies, we include contract specific implied parameters since these are widely used by market practitioners (i.e. Bakshi et al., 1997, pg. 2019).

For notational reasons, implied parameters obtained from the first step are denoted by the subscript *av*, from the second step by the subscript *avT*, from the third step by the subscript *avT4*, and from the fourth step by the subscript *con*. The four different implied BS volatility estimates are symbolized as: σ_j^{BS}, $j = \{av, avT, avT4, con\}$, whilst the four different sets of CS parameters as: $\{\sigma_j^{CS}, \mu_{3,j}^{CS}\ \mu_{4,j}^{CS}\}$. For pricing and trading reasons at time instant *t*, the implied structural parameters derived at day *t*-1 are used together with all other needed information (*S*, *T*, *X*, *r*, and δ).

It is known that ANN input variables should be presented in a way that maximizes their information content. When we price options, the POPM formulas adjust those values to represent the appropriate value that corresponds to an option's expiration period. According to this rationale, volatility measures for use with the ANNs are transformed by multiplying each of the yearly volatility forecast with the square root of each option's time to maturity ($\tilde{\sigma}_j = \sigma_j \sqrt{T}$, where *j*={*60, vix, av, avT, avT4, con*}). We denote these volatility measures as $\tilde{\sigma}_j^{BS}$ and $\tilde{\sigma}_j^{CS}$; and we name them as *maturity (or expiration) adjusted volatilities*. Additionally, for the case of CS, skewness $\mu_{3,j}^{CS}$, $j = \{60, av, avT, avT4, con\}$, is transformed by multiplication with Q_3 that represents the marginal effect of nonnormal skewness. Similarly, $\mu_{4,j}^{CS}$ is multiplied with Q_4. We denote these adjusted parameters as $\tilde{\mu}_{3,j}^{CS}$ (adjusted skewness), and $\tilde{\mu}_{4,j}^{CS}$ (adjusted kurtosis).

---

[8] If not somehow constrained, skewness and kurtosis can take implausible values (i.e. Bates, 1991) due to model overfitting that will lead to enormous pricing errors on the next day (especially for deep in the money options). In our case these constraints were binding in less than 2% of the whole dataset.

1.4.3. Output variables, filtering and processing

The BS (c_q^{BS}) and CS (c_q^{CS}) outputs, are used as an estimate for the market call option, c_q^{mrk}. For training ANNs, the call standardized by the striking price, c_q^{mrk} / X_q, is used as the target function to be approximated. In addition, we implement the hybrid structure where the target function represents the pricing error between the option's market price and the parametric models estimate, $c_q^{mrk} / X_q - c_q^k / X_q$.

	VDOTM	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM	VDITM
S/X	<0.85	0.85-0.95	0.90-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.10	1.10-1.35	≥1.35
				Short Term Options <60 Days					
call	3.61	1.63	5.15	15.70	32.40	56.58	99.55	199.77	470.38
volatility	0.36	0.21	0.19	0.19	0.20	0.22	0.27	0.38	0.99
# obs	399	1,361	4,815	7,483	3,964	6,548	4,970	7,990	2,103
				Medium Term Options 60-180 Days					
Call	4.38	8.29	23.58	46.06	64.51	90.35	131.10	227.41	493.18
volatility	0.22	0.18	0.20	0.21	0.21	0.23	0.25	0.30	0.54
# obs	1,412	1,727	2,578	3,147	1,780	2,901	3,038	8,100	3,999
				Long Term Options ≥ 180 Days					
Call	9.65	42.09	74.03	106.24	126.03	150.99	185.87	267.12	495.82
Volatility	0.18	0.21	0.22	0.23	0.24	0.25	0.26	0.28	0.40
# obs	332	333	575	603	343	660	812	2,695	1,733

Table 1.1. Sample descriptive statistics
Sample characteristics for the period January 5, 1998 to August 24, 2001 concerning the average call option value, the average Black and Scholes contract specific implied volatility and the number of observations in each moneyness/maturity class.

Before filtering, more than 100,000 observations were included for the period January 1998 – August 2001. The filtering rules we adopt are: *i) eliminate an observation if the call contract price,* $c_{m,t}^{mrk}$*, m defining each traded contract, is greater than the underlying asset value,* S_t*; ii) exclude an observation if the call moneyness ratio is larger than unity,* $S_t/X_m>1$*, and the call price,* $c_{m,t}^{mrk}$*, is less than its lower bound,* $S_t e^{-\delta_{m,t}T_{m,t}} - X_m e^{-r_{m,t}T_{m,t}}$*; iii) eliminate all the options observations with time to maturity less than 6 trading days*. The latter filtering rule is adopted to avoid extreme option prices that are observed due to potential illiquidity problems; *iv) price quotes lower than 0.5 index points are not included; v) maturities with less than four call option observations are also eliminated, vi)* in addition, to remove impact from thin trading we eliminate observations according to the following rule: *eliminate an observation if the* $c^{mrk}_{m,t}$ *is equal to* $c^{mrk}_{m,t-1}$ *and if the open interest for these days stays unchanged and if the underlying asset S has changed.*

Our final dataset consists of 76,401 datapoints. Table 1.1 exhibits some of the properties of our sample tabulated according to moneyness ratio and time to maturity forming 27 different moneyness/maturity classes. We provide the average values for c^{mrk} and σ^{BS}_{con}, and the number of observations within each moneyness and maturity class. The implied volatility, σ^{BS}_{con}, presents a non-flat moneyness structure when fixing the time to maturity and vice versa revealing the bias associated with BS. Moreover, we should notice that DITM and VDITM options dominate in number of datapoints all other classes, so unlike studies that ignore these options we choose to include them in the dataset. For the training sub-periods, the observations vary between: 19,852-22,545; for the validation sub-periods between: 10,372-10,916; and for the testing sub-periods between: 3,797-4,264.

In order to check the robustness of the results, in addition to the *full* dataset just described, we repeat the analysis using a *reduced* dataset. In this reduced dataset we follow Hutchison et al. (1994), and we neither use long maturity (longer than 180 trading days) options, nor the VDOTM (S/X<0.85) or the VDITM (S/X≥1.35) options. The excluded observations (because of considerations of thin trading) comprise about 21% of the full dataset resulting in a total of 60,402 observations. The training-validation-testing splitting dates are the same as in the original dataset. For the training sub-periods, the observations vary between: 15,851-18,053; for the validation sub-periods: 7,728-9,638; and for the testing sub-periods: 2,689-3,983. To be consistent with Hutchison et al. (1994), in using the reduced dataset we retrain the ANNs. Our discussion will focus on the full dataset. In order to save space, we will only show selected results using the reduced dataset.

1.4.4. Validation, testing and pricing performance measures

Since a practitioner is faced with time-series data, it was decided to partition the available data into training, validation and testing datasets using a chronological manner, and via a rolling-forward procedure. Our dataset is divided into ten different overlapping training (*Tr*) and validation (*Vd*) sets, each followed by separate and non-overlapping testing (*Ts*) sets as exhibited by Fig. 2. The ten sequential testing sub-periods cover the last 25 months of the complete dataset.

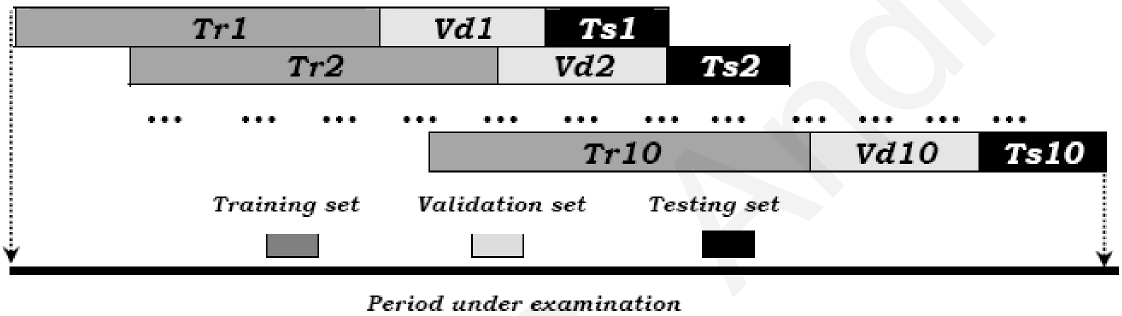

Figure 1.2. The rolling-over training-validation-testing procedure

There are *M* available call option contracts, for each of which there exist Ξ_m observations taken in consecutive time instances *t*, resulting in a total of *P* ($P = \sum_{m=1}^{M} \Xi_m$) available call option datapoints. To determine the pricing accuracy of each model's estimates c^k (*k* defining the model), we examine the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE):

$$RMSE = \sqrt{(1/p)\sum_{\upsilon=1}^{p}(c_{\upsilon}^{mrk} - \hat{c}_{\upsilon}^{k})^2} \qquad (1.11)$$

$$MAE = (1/p)\sum_{\upsilon=1}^{p}\left|c_{\upsilon}^{mrk} - \hat{c}_{\upsilon}^{k}\right|, \qquad (1.12)$$

where *p* indicates the number of observations. The error measures are computed for an aggregate testing period (*AggTs*) with 39,831 datapoints by pooling together the pricing estimates of all ten testing periods. For *AggTs* we also compute the Median of the Absolute Error (MeAE). Of course, since ANNs are effectively optimized with respect to the mean square error, the out of sample pricing performance should be similarly based on RMSE and in a lesser degree on MAE and MeAE.

1.4.5. The set of alternative BS, CS and ANN models

With the BS models we use as input *S, X, T, r,* δ, and any of the six different volatility measures: σ_{60}, σ_{vix}^{BS}, σ_{av}^{BS}, σ_{avT}^{BS}, σ_{avT4}^{BS} and σ_{con}^{BS}. Using *P* in the superscript to denote the parametric version of BS, the six different models are symbolized as: BS_{60}^{P}, BS_{vix}^{P}, BS_{av}^{P}, BS_{avT}^{P}, BS_{avT4}^{P}, and BS_{con}^{P}. In a similar way there are five different CS models according to the kind of parameters used: CS_{60}^{P}, CS_{av}^{P}, CS_{avT}^{P}, CS_{avT4}^{P}, and CS_{con}^{P}.

With ANNs, we also use three standard input variables/parameters: $(Se^{-\delta T})/X$, T and r. Additional input parameters depend on the parametric model considered. There are six ANN models that use as an additional input the above BS volatility measures to map the standard target function c^{mrk}/X. There are six more versions that utilize the maturity adjusted parameters. Each of the previous input parameter sets is also used with the hybrid target function. The ANNs that use the untransformed BS volatility forecast are denoted by *N* in the superscript, the transformed versions by *N**, while the corresponding hybrid versions by *Nh* and *Nh** respectively. For instance, BS_{con}^{N} (BS_{con}^{Nh}) is the ANN model that uses as additional input σ_{con}^{BS} and maps the standard (hybrid) target function, whilst BS_{con}^{N*} (BS_{con}^{Nh*}) the ANN model that uses as additional input $\tilde{\sigma}_{con}^{BS}$ and maps the standard (hybrid) target function. In total there are 24 different versions of ANNs related to the BS and 20 related to the CS model.

1.5. Pricing results and discussion

We briefly review the observed in sample fit of the parametric models as well as the in sample characteristics of the various implied parameters. Then we discuss the out of sample performance of the alternative OPMs. When we do not explicitly refer to the dataset, we imply the full one. The insights derived were not affected by the choice of dataset. When noteworthy differences exist, we state them explicitly.

1.5.1. BS and CS in sample fitting performance and implied parameters

Based on our (not reported in detail for brevity) statistics for the whole period (1998-2001) we have observed that CS is producing smaller fitting errors than the BS. The contract specific fitting procedure reduces the fitting errors so as to almost eliminate the residuals and obtain fully calibrated implied parameters. The in sample RMSE measures using the overall average set of implied parameters (*av*), the average per maturity (*avT*), and the closest four contracts (*avT4*), are: 11.63, 11.31, and 7.00 for the BS model; and 9.52, 8.52, and 5.35 for the CS model[9]. From unreported statistics we can also attest that the S&P 500 average σ_{con}^{BS} in 1998 was about 33%, in 1999 about 30%, in 2000 about 26% and in 2001 about 27%. It seems that the in sample fitting error of the models (diminishing in time) is positively correlated with the market volatility.

We can also provide some statistics about the implied parameter values for the whole period. The Brownian volatility varies between 22% and 30% in BS and between 27% and 31% in CS. For the BS model, the average implied volatility (σ_{av}^{BS}) estimates are smaller in magnitude (both in mean and in median values) from the contract specific implied volatility, σ_{con}^{BS}, although similar volatility estimates do not necessarily lead to similar pricing and hedging values (Bakshi et al., 1997). Regarding implied skewness and kurtosis, the implicit distributions are negatively skewed with excess kurtosis in almost all days, something that is probably attributed to the crash fears of the market participants after the Black Monday of 1987. Implied average skewness does not change significantly (from -1.19 to -1.20) if we move from {*av*} to {*avT*} but there is a shift in implied average kurtosis (from 6.91 to 6.19).

[9] The RMSE for CS in the fourth step (*con*) is 1.82 (caused by a tiny part of the dataset less than 0.1%) due to binding constraints on skewness and kurtosis. For this step, the MeAE is more appropriate, and is effectively zero. The RMSE and the MeAE for BS in the fourth step are effectively zero.

1.5.2. Out of sample pricing results

Table 1.2, exhibits the performance of all parametric and ANN models considered in this study in terms of RMSE, MAE and MeAE for the *AggTs* (aggregate) period. In Table 1.3 we tabulate statistics for a pairwise comparison of the (statistical significance of) pricing performance of a selection of models. Since the ten testing periods are disjoint and because we have pricing estimates coming from different OPMs we can assume (similarly to Hutchison et al, 1994 and Schittenkopf and Dorffner, 2001) that the pricing errors are independent and standard t-test can be applied. Similarly to the previous authors we need to report that these tests should be interpreted with caution. The upper diagonal of Table 1.3 reports the *t*-values taken by a two-tail matched-pair test about the MAE of the alternative models whilst the lower diagonal exhibits the two-tail matched-pair *t*-test values about the MSE of the compared OPMs. Table 1.4 provides (as a robustness check) the performance of the models when using the reduced dataset.

By looking at Tables 1.2 and 1.4 we can see that the use of implied instead of historical parameters improves performance, both for parametric and ANN models (in both datasets). Note that the 60-day historical volatility performed better than VIX with the parametric BS model, but the VIX volatility measure performed better with the ANN models. Using time adjusted parameters in the ANNs or using contract specific parameters {*avT4*, *con*} usually improves performance. The combination of time adjusted parameters and contract specific parameters always provided the best model within each class of ANNs (standard or hybrid, BS or CS based) in both datasets.

In comparing the parametric models and again looking at Tables 1.2 and 1.4, it is noteworthy that CS outperforms BS when average implied parameters are used. BS still works better with contract specific parameters. The overall best among the parametric models is the contract specific BS model. In other more complex parametric models that include jumps and stochastic volatility components (i.e. Bakshi et al., 1997), deriving implied parameters may lead to model overfitting. The contract specific approach we adopt in this study seems not to lead to model overfitting, retaining thus good out of sample properties. For the ANN models, the CS based may outperform the BS based in some cases, but when the best combinations are used (time adjusted parameters and contract specific parameters), the best model always is BS based in both the standard and hybrid networks.

	BS^P_{60}	BS^P_{vix}	BS^P_{av}	BS^P_{avT}	BS^P_{avT4}	BS^P_{con}	CS^P_{60}		CS^P_{av}	CS^P_{avT}	CS^P_{avT4}	CS^P_{con}
RMSE	11.18	12.57	9.72	9.47	8.03	7.04	11.25		8.89	8.87	8.11	7.71
MAE	6.83	8.60	5.32	5.00	3.10	2.70	6.89		3.86	3.72	3.27	3.10
MeAE	4.48	6.38	3.74	3.37	1.52	1.43	4.61		2.26	1.94	1.69	1.68
	BS^N_{60}	BS^N_{vix}	BS^N_{av}	BS^N_{avT}	BS^N_{avT4}	BS^N_{con}	BS^{N*}_{60}	BS^{N*}_{vix}	BS^{N*}_{av}	BS^{N*}_{avT}	BS^{N*}_{avT4}	BS^{N*}_{con}
RMSE	13.06	12.65	10.97	12.48	10.74	9.06	14.68	12.76	12.30	11.69	9.33	7.86
MAE	7.58	6.65	5.91	7.04	6.04	4.68	7.68	6.70	6.67	6.55	5.04	3.81
MeAE	5.13	3.83	3.65	4.11	3.69	2.88	4.71	3.65	3.99	3.94	2.94	2.44
	CS^N_{60}		CS^N_{av}	CS^N_{avT}	CS^N_{avT4}	CS^N_{con}	CS^{N*}_{60}		CS^{N*}_{av}	CS^{N*}_{avT}	CS^{N*}_{avT4}	CS^{N*}_{con}
RMSE	15.22		11.28	11.59	9.87	11.83	14.35		11.42	11.96	9.47	9.76
MAE	9.13		5.80	6.14	5.73	5.81	7.71		5.39	5.56	4.67	4.87
MeAE	6.43		3.48	3.96	3.65	3.65	4.27		3.26	3.15	2.93	3.03
	BS^{Nh}_{60}	BS^{Nh}_{vix}	BS^{Nh}_{av}	BS^{Nh}_{avT}	BS^{Nh}_{avT4}	BS^{Nh}_{con}	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{av}	BS^{Nh*}_{avT}	BS^{Nh*}_{avT4}	BS^{Nh*}_{con}
RMSE	9.05	8.35	8.57	8.29	7.79	6.38	9.03	8.27	8.87	7.84	7.68	6.01
MAE	5.40	4.55	4.35	4.09	3.30	2.68	5.46	4.53	4.35	3.91	3.17	2.61
MeAE	3.73	2.98	2.83	2.51	1.80	1.60	3.98	3.00	2.69	2.53	1.67	1.58
	CS^{Nh}_{60}		CS^{Nh}_{av}	CS^{Nh}_{avT}	CS^{Nh}_{avT4}	CS^{Nh}_{con}	CS^{Nh*}_{60}		CS^{Nh*}_{av}	CS^{Nh*}_{avT}	CS^{Nh*}_{avT4}	CS^{Nh*}_{con}
RMSE	10.33		8.68	8.63	7.97	7.60	9.68		8.83	8.66	7.60	7.39
MAE	6.38		4.12	3.84	3.42	3.14	6.20		3.95	3.94	3.39	3.11
MeAE	4.46		2.42	2.17	1.93	1.77	4.56		2.33	2.35	1.96	1.82

Table 1.2. Pricing error measures in the aggregate testing period (*AggTs*)
RMSE is the Root Mean Square Error, MAE the Mean Absolute Deviation and MeAE the Median of the Absolute Error. The superscripts refer to the kind of the model: *P* refers to parametric models, *N* to the simple neural networks and *Nh* to the hybrid neural networks. The asterisk (*) refers to neural network models that use transformed variables. The subscripts refer to the kind of historical/implied parameters used.

In comparing the parametric models with the standard ANNs, in the full dataset the ANNs never outperform the equivalent parametric ones. Apparently, the standard ANNs cannot perform well in the extreme data regions. In the reduced dataset (see Table 1.4), we observe the opposite since the standard ANNs always outperform the equivalent parametric ones.

In comparing the hybrid with the standard ANNs, in the full dataset the hybrid are always better. In the reduced dataset this may not always be the case, but the best combinations (time adjusted parameters and contract specific parameters) give as the best model always a hybrid one.

In both the full and the reduced dataset, the hybrid always outperform the equivalent parametric ones. Finally, in both the full and the reduced dataset, the overall best model is the BS based hybrid with time adjusted and contract specific volatility.

From Table 1.3, we can confirm the statistical significance of the best models. The comparative results we discuss with tests using the full dataset, and they also hold for the reduced dataset (statistics not reported for brevity). We can see that BS_{con}^{Nh*} outperforms all other models. Specifically, BS_{con}^{Nh*} is producing a RMSE equal to 6.01 and a MAE equal to 2.61, pricing measures that are smaller that any other model at the 5% significance level.

	BS_{60}^{P}	BS_{vix}^{P}	BS_{con}^{P}	CS_{60}^{P}	CS_{con}^{P}	BS_{60}^{N*}	CS_{60}^{N*}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{con}^{Nh*}	CS_{60}^{Nh*}	CS_{con}^{Nh*}
BS_{60}^{P}		-27.74	75.12	-0.94	65.72	-11.07	-11.72	23.90	40.83	81.22	10.80	66.92
BS_{vix}^{P}	7.17		104.84	26.74	94.84	11.84	11.71	53.72	70.71	112.32	40.52	96.53
BS_{con}^{P}	-16.13	-25.08		-75.91	-8.43	-70.51	-72.82	-56.94	-38.56	2.12	-70.87	-8.76
CS_{60}^{P}	0.34	-6.72	16.31		66.53	-10.28	-10.91	24.85	41.74	82.02	11.78	67.74
CS_{con}^{P}	-13.38	-21.60	2.14	-13.58		-63.58	-65.63	-46.76	-28.85	11.11	-60.38	-0.10
BS_{60}^{N*}	7.24	4.64	13.37	7.09	12.48		-0.34	30.64	43.94	74.23	20.23	64.28
CS_{60}^{N*}	7.77	4.67	15.19	7.59	14.09	-0.62		31.84	45.50	76.80	21.15	66.39
BS_{60}^{Nh*}	-9.55	-18.30	7.57	-9.81	4.95	-10.83	-12.15		18.64	63.30	-14.28	47.82
BS_{vix}^{Nh*}	-12.54	-21.61	4.48	-12.75	2.02	-11.91	-13.46	-3.25		43.70	-32.87	29.50
BS_{con}^{Nh*}	-21.16	-32.03	-3.45	-21.26	-5.62	-14.65	-16.83	-12.27	-8.78		-78.03	-11.58
CS_{60}^{Nh*}	-6.86	-15.36	10.42	-7.15	7.65	-9.84	-10.96	2.97	6.24	15.52		61.73
CS_{con}^{Nh*}	-14.98	-23.78	1.15	-15.16	-1.04	-12.95	-14.69	-6.34	-3.26	4.73	-9.18	

Table 1.3. Matched-pair student t-tests for square and absolute differences

Reported matched-pair *t*-tests concerning the absolute differences are in the upper diagonal, whilst the matched-pair *t*-tests concerning the square differences in the lower diagonal. Both tests compare the MAE and MSE between models in the vertical heading versus models in the horizontal heading. In general, a positive t-value larger than 1.645 (2.325) means that the model in the vertical heading has a larger MAE or MSE than the model in the horizontal heading at 5% (1%) significance level.

The BS based hybrid ANNs even with historical or the VIX volatility measure are considerably better than the equivalent parametric alternatives at a statistically significant level. Specifically, BS_{60}^{Nh*} is producing 1.23 (1.25) times smaller MSE (MAE) compared to BS_{60}^{P}. Also BS_{vix}^{Nh*} produces 1.52 (1.90) times smaller MSE (MAE) compared to BS_{vix}^{P}.

Comparing the out of sample pricing performance of BS_{con}^{Nh*} to CS_{con}^{Nh*} we observe that the extra ANN flexibility of the latter due to the two additional input parameters does not lead to increased accuracy. The BS_{con}^{Nh*} is better than the CS_{con}^{Nh*} model at 1% significance level.

We can similarly see the statistical significance of the superiority of the BS based models with contract specific volatility versus the equivalent CS based models (both parametric and hybrid); and the superiority of the models using the implied volatility versus the equivalent ones using the historical volatility measures.

	BS_{60}^{P}	BS_{vix}^{P}	BS_{av}^{P}	BS_{avT}^{P}	BS_{avT4}^{P}	BS_{con}^{P}	CS_{60}^{P}		CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
RMSE	9.83	11.82	8.41	8.25	7.08	7.06	9.74		7.56	7.55	7.55	7.52
MAE	6.35	8.43	4.82	4.54	2.65	2.65	6.32		3.38	3.12	2.99	3.04
MeAE	4.50	6.57	3.63	3.27	1.48	1.46	4.59		2.17	1.83	1.69	1.71
	BS_{60}^{N}	BS_{vix}^{N}	BS_{av}^{N}	BS_{avT}^{N}	BS_{avT4}^{N}	BS_{con}^{N}	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
RMSE	8.05	6.56	7.34	6.94	6.64	6.69	7.14	6.60	6.82	6.91	6.25	6.12
MAE	5.07	3.34	4.02	3.72	3.42	3.37	4.11	3.43	3.46	3.59	3.01	3.00
MeAE	3.80	2.32	2.99	2.56	2.33	2.24	3.09	2.41	2.44	2.56	1.99	2.02
	CS_{60}^{N}		CS_{av}^{N}	CS_{avT}^{N}	CS_{avT4}^{N}	CS_{con}^{N}	CS_{60}^{N*}		CS_{av}^{N*}	CS_{avT}^{N*}	CS_{avT4}^{N*}	CS_{con}^{N*}
RMSE	9.05		7.18	6.93	6.94	6.88	8.35		6.97	6.59	6.50	6.77
MAE	5.74		3.95	3.61	3.73	3.62	4.94		3.68	3.26	3.23	3.45
MeAE	4.25		2.74	2.41	2.60	2.55	3.43		2.62	2.22	2.25	2.36
	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
RMSE	8.45	6.70	7.29	7.01	6.58	6.78	7.35	6.40	7.05	6.83	5.94	5.64
MAE	5.11	3.58	3.62	3.38	2.62	2.69	4.27	3.21	3.32	3.30	2.45	2.44
MeAE	3.44	2.59	2.55	2.35	1.55	1.65	3.13	2.26	2.30	2.33	1.51	1.54
	CS_{60}^{Nh}		CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}	CS_{60}^{Nh*}		CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
RMSE	7.80		7.29	6.83	7.31	7.35	7.69		6.90	6.80	6.51	6.46
MAE	4.65		3.20	3.08	3.03	3.03	4.58		3.13	2.92	2.83	2.87
MeAE	3.41		2.13	2.02	1.82	1.80	3.23		2.03	1.80	1.79	1.81

Table 1.4. Pricing error measures in the aggregate period (AggTs) for the reduced dataset

RMSE is the Root Mean Square Error, MAE the Mean Absolute Error and MeAE the Median of the Absolute Error. The superscripts refer to the kind of the model: *P* refers to parametric models, *N* to the simple neural networks and *Nh* to the hybrid neural networks. The asterisk (*) refers to neural network models that use the transformed variables. The subscripts refer to the kind of historical/implied parameters used.

1.5.3. Other statistics

We tabulate in Table 1.5 the MSE of a selective (but representative) choice of models, according to the various moneyness and maturity classes for the aggregate (*AggTs*) period. We demonstrate results for the two best performing parametric models which serve as benchmark (BS_{con}^{P}, CS_{con}^{P},) and the two best performing (in their respective class) hybrid ANN models (BS_{con}^{Nh*}, CS_{con}^{Nh*}). We also demonstrate results for the reduced dataset (BS_{con}^{Nh*}, CS_{con}^{Nh*}). The relevant information for the parametric models in the reduced dataset can be taken from the information concerning the full if we ignore the long maturities, and the VDOTM and the VDITM classes. Very briefly, what can be seen is that BS_{con}^{P} has a smaller RMSE in all data classes compared to CS_{con}^{P}. The same holds for BS_{con}^{Nh*} over CS_{con}^{Nh*}. If we compare the BS and CS based hybrid models with the equivalent parametric ones, the hybrid ANN models rarely underperform the parametric ones, and they do so only in some classes far away from ATM. This we attribute to the scarcity of such call option datapoints in the training samples compared to other moneyness and maturity classes.

	Short	Medium	Long	Short	Medium	Long
Results for the full dataset						
		BS^{P}_{con}			CS^{P}_{con}	
VDOTM	3.60	4.91	0.56	8.34	10.61	0.66
DOTM	2.27	4.50	2.82	3.02	5.24	4.47
OTM	5.78	8.37	3.97	6.29	9.68	5.08
JOTM	7.81	6.68	6.15	8.13	7.64	7.65
ATM	6.67	9.46	5.86	7.30	10.14	7.29
JITM	6.71	9.41	4.34	7.29	9.21	5.97
ITM	7.70	7.13	4.43	8.24	7.59	5.18
DITM	7.07	7.93	7.27	7.20	8.50	7.50
VDITM	8.26	9.46	8.74	8.29	10.05	9.05
		$BS^{Nh^*}_{con}$			$CS^{Nh^*}_{con}$	
VDOTM	3.60	4.97	1.15	6.13	10.22	6.04
DOTM	2.46	4.83	2.32	2.96	5.28	5.03
OTM	5.50	7.75	3.98	6.19	9.41	5.36
JOTM	5.89	5.36	5.78	7.83	7.30	7.66
ATM	4.73	8.18	5.38	6.94	9.86	7.13
JITM	5.59	7.39	4.10	6.89	8.68	6.64
ITM	6.24	6.05	3.95	7.58	7.16	5.69
DITM	5.80	7.15	6.74	6.64	8.04	7.17
VDITM	8.03	9.29	8.46	8.96	10.33	9.26
Results for the reduced dataset						
		$BS^{Nh^*}_{con}$			$CS^{Nh^*}_{con}$	
VDOTM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
DOTM	2.36	4.07	n.a.	2.54	5.22	n.a.
OTM	5.08	7.25	n.a.	5.69	8.74	n.a.
JOTM	5.82	5.59	n.a.	6.76	7.09	n.a.
ATM	4.65	8.37	n.a.	5.68	9.53	n.a.
JITM	5.50	7.68	n.a.	6.20	8.16	n.a.
ITM	5.98	5.84	n.a.	6.73	6.75	n.a.
DITM	5.45	6.59	n.a.	5.95	7.67	n.a.
VDITM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.

Table 1.5. Root Mean Square Errors for selected models (clustered by moneyness and maturity)

We should finally comment on the complexity of each neural network configuration. Since we have a constant number of inputs within each model class, the larger the number of hidden neurons the more complex the ANN model architecture, and the more complex the target function to be approximated. Firstly, we observe that the number of hidden neurons changes significantly between sub-periods. This contradicts many previous studies that employ the assumption that the market's options pricing mechanism is the same for all periods examined and that a constant ANN structure is sufficient. Secondly, the standard target function is more complex compared to the hybrid one, hence this hybrid category of networks can perform better in out of sample pricing. Thus, it is not surprising that the best performing ANN model, BS_{con}^{Nh*}, demonstrates the simplest structure with an average of 3.2 hidden layer neurons, compared to the 8 hidden layer neurons in the case of the equivalent standard ANN (BS_{con}^{N*}). Similarly for the CS-based ANNs, we have 4.9 (for CS_{con}^{Nh*}) and 7.7 (for CS_{con}^{N*}) hidden layer neurons respectively. Similar network complexities (not reported) were observed in the reduced dataset.

1.6. Delta neutral trading strategies

We now investigate the economic significance of the best performing models in options trading. In order to save space we discuss the parametric versions of BS and CS which are usually the benchmark, and the hybrid ANN models which provided the overall best performance. Other studies usually restrict their analysis only to a hedging investigation of various alternative POPM models (i.e. Hutchison et al., 1994, Garcia and Gencay, 2000, Schittenkopf and Dorffner, 2001) and avoid exploiting trading strategies. It is known from previous studies that the best POPM in terms of out of sample pricing performance does not always prove to be the best solution when we consider delta hedging, since ANNs are optimized based on a pricing error criterion. Instead, and following the spirit of Black and Scholes (1972), Galai (1977), and Whaley (1982), we investigate the economic significance of the OPMs by implementing trading strategies. “A model that consistently achieves to identify mispriced options and within a time period produces an amount of trading profits will always be preferred by a practitioner” (Black and Scholes, 1972). The trading profitability that we will document, indirectly also hints to potential option market inefficiencies, although testing market efficiency is beyond the scope of our study. We implement trading strategies based on single instrument hedging, as for example in Bakshi et al. (1997). In addition, we consider various levels of transaction costs, and we focus on dynamic strategies that are cost-effective. We later extend the analysis by implementing a modified approach for trading using hedging ratios obtained via the (widely neglected) Chen and Johnson (1985) method. To our knowledge, this is the first effort to validate this modified trading strategy using both parametric and ANN OPMs.

In the trading strategy we implement, we create portfolios by buying (selling) options undervalued (overvalued) relative to a model's prediction and taking a delta hedging position in the underlying asset. This (single-instrument) delta hedging follows the no-arbitrage strategy of Black and Scholes (1973), where a portfolio including a short (long) position in a call is hedged via a long (short) position in the underlying asset, and the hedged portfolio rebalancing takes place in discrete time intervals (in an optimal manner, not necessarily daily). At time *t*, if according to the model the m^{th} call option contract is overvalued (undervalued) relative to its market value, $c_{m,t}^{mrk}$, we go short (long) in this contract and we go long (short) in $\Delta_{m,t}^{k}$ "index shares[10]", where *k* denotes the relevant model. Then we invest the residual, $B_{m,t}$, in a riskless bond. Note that $\Delta_{m,t}^{k}$ is the partial derivative of the option price with respect to the underlying asset, $\partial c_{m,t}^{k} / \partial S_t$, depending on the POPM under consideration. $\Delta_{m,t}^{ANN}$ can be calculated by differentiating Eq. (1.4) via the chain rule. The expression for $\Delta_{m,t}^{BS}$ is $e^{-\delta T} N(d_1)$ and is derived from Eq. (1.2). The expression for $\Delta_{m,t}^{CS}$ includes $\Delta_{m,t}^{BS}$ and is:

$$\Delta_{m,t}^{CS} = \Delta_{m,t}^{BS} + \mu 3 \Phi_3 + (\mu 4 - 3)\Phi_4 , \qquad (1.13)$$

where $\Phi_3 = \dfrac{\partial Q_3}{\partial S}$ and $\Phi_4 = \dfrac{\partial Q_4}{\partial S}$ are given below:

$$\Phi_3 = \frac{1}{3!} e^{-\delta T} ((\sigma\sqrt{T})^3 N(d_1) + n(d_1)[3(\sigma\sqrt{T})^2 - 3 d_1 \sigma \sqrt{T} + (d_1)^2 - 1]) \qquad (1.13.1)$$

---

10 Similarly to Bakshi et al. (1997) we assume that the spot S&P 500 index is a traded security.

$$\Phi_4 = \frac{1}{4!} e^{-\delta T} ((\sigma\sqrt{T})^4 N(d_1) + 4n(d_1)((\sigma\sqrt{T})^3 - 6(d_1)n(d_1)(\sigma\sqrt{T})^2 - 4n(d_1)(\sigma\sqrt{T}) + 4(d_1)^2 n(d_1)(\sigma\sqrt{T}) + 3(d_1)n(d_1) - (d_1)^3 n(d_1)) \quad (1.13.2)$$

In general we avoid a *naive* (expensive) trading strategy with daily rebalancing, since in the presence of transaction costs this would become prohibitively expensive. Instead, the position is held *as long as* the call is undervalued (overvalued) without necessarily daily rebalancing. Then the position is liquidated and the profit or loss is computed, tabulated separately and a new position is generated according to the prevailing conditions in the options market. This procedure is carried out for all contracts included in the dataset. We rebalance our position in the underlying asset to keep the appropriate hedge ratio. Rebalanced positions in the index, $V_{m,t+\Delta t}$, and the bond, $B_{m,t+\Delta t}$, are according to:

$$V_{m,t+\Delta t} = \pm S_{t+\Delta t}(\Delta_{m,t+\Delta t} - \Delta_{m,t}) \quad (1.14)$$

$$B_{m,t+\Delta t} = B_{m,t} e^{r\Delta t} + V_{m,t+\Delta t}, \quad (1.15)$$

where the positive sign is considered when we treat undervalued and the negative sign when we treat overvalued options. Note that in all trading strategies, when we need to invest money we borrow and pay the riskless rate; similarly we do for as long as a strategy provides losses. Thus, when we present profits they are always above the dollar return on the riskless rate.

Computed statistics include the total profit or loss (P&L), the number of trades (# Trades), the total profit or loss at 0.2% transaction costs, P&L (0.2%), and 0.4% transaction costs, P&L (0.4%). The (proportional) transaction costs are paid for both positions (in the call option and in the "index shares")[11]. We also implement strategies with enhanced cost-effectiveness by ignoring trades that

[11] For example, assume that the index is at 1300 and a call option has a market price equal to 25 index points and a delta value of 0.60. Under 0.4% transaction costs the total commissions paid (for a single trade) will be 3.22 index points. In the *AggTs* period the S&P 500 was in a range from about 1100 to 1500. This level of transaction costs is low but attainable by professional traders and market makers.

involve call options whose absolute percentage mispricing error, $|c^k - c^{mrk}|/c^k$, is less than a mispricing margin d = 15%, found as P&L (d = 15%). In addition, for these strategies, we also calculate P&L under aggregate transaction costs for the "index shares". With such aggregation, transactions in the underlying assets are paid on the net (aggregate) exposure of $V_{m,t+\Delta t}$ and not on each position individually. Under this strategy, we expect additional cost savings that may provide profits even at rather high transaction cost levels. We use the prefix *Agg.* in front of P&L to indicate this strategy. The following observations refer to the full dataset, but they also hold for the reduced one (unreported due to brevity considerations).

	BS^P_{60}	BS^P_{vix}	BS^P_{av}	BS^P_{avT}	BS^P_{avT4}	BS^P_{con}
Panel A: Black and Scholes trading strategy with standard delta values						
P&L	7,447	13,518	14,088	13,069	32,040	35,026
# Trades	3,361	3,878	4,858	5,477	13,539	15,644
P&L 0.2% (d=0%)	-6,829	-6,847	-5,348	-7,512	-17,911	-23,307
Agg P&L 0.2% (d=0%)	-1,861	-266	737	-1,394	-5,638	-8,437
P&L 0.2% (d=15%)	3,320	4,134	7,527	6,841	7,907	7,369
Agg P&L 0.2% (d=15%)	5,003	5,019	8,344	7,657	8,384	7,873
P&L 0.4% (d=0%)	-21,105	-27,211	-24,785	-28,093	-67,863	-81,640
Agg P&L 0.4% (d=0%)	-11,170	-14,049	-12,614	-15,858	-43,316	-51,899
P&L 0.4% (d=15%)	-1,468	-508	3,241	2,269	4,691	4,212
Agg P&L 0.4% (d=15%)	1,897	1,262	4,875	3,901	5,645	5,221
Panel B: Black and Scholes trading strategy with modified delta values						
P&L	7,916	14,367	14,232	13,441	32,281	35,229
# Trades	3,361	3,878	4,858	5,477	13,539	15,644
P&L 0.2% (d=0%)	-6,169	-5,599	-4,958	-6,946	-17,788	-23,080
Agg P&L 0.2% (d=0%)	-1,392	1,342	1,225	-778	-5,534	-8,259
P&L 0.2% (d=15%)	4,044	5,534	8,182	7,546	8,306	7,713
Agg P&L 0.2% (d=15%)	5,515	6,558	9,115	8,524	8,815	8,198
P&L 0.4% (d=0%)	-20,254	-25,564	-24,148	-27,334	-67,858	-81,390
Agg P&L 0.4% (d=0%)	-10,700	-11,682	-11,782	-14,998	-43,348	-51,748
P&L 0.4% (d=15%)	-685	1,284	4,143	3,180	4,883	4,339
Agg P&L 0.4% (d=15%)	2,257	3,333	6,007	5,137	5,900	5,308

Table 1.6. Trading strategies for the Black and Scholes models
P&L is the total profit and loss without transaction costs; # Trades is the number of trades. P&L (d=0 and 15%) represents the P&L at 0.2% or 0.4% transaction costs when we ignore trades whose absolute percentage of mispricing error between model estimates and market values is at least 0% and 15% respectively. *Agg.* refers to aggregating the position on the underlying asset to reduce transaction costs. Panel A tabulates results with standard delta values whilst Panel B tabulates results with Chen and Johnson modified delta values.

The results for the parametric BS and CS models are tabulated in Panel A of Tables 1.6 and 1.7 respectively. We observe that all models before transaction costs produce significant profits, implying that both BS and CS can successfully identify mispriced options. Within BS models the magnitude of P&L is larger for

BS_{con}^{P} that employs a more *sophisticated* implied volatility forecast. Note though that the more sophisticated volatility forecast that is used with BS, the larger the number of trades. So, when 0.2% transaction costs are taken into consideration, all models produce significant losses and the previous profit dominance of BS_{con}^{P} over BS_{60}^{P} reverts because the latter model incurs less transaction costs (since it engages in a smaller number of trades).

	CS_{60}^{P}	CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
Panel A: Corrado and Su trading strategy with standard delta values					
P&L	7,603	28,816	32,803	37,072	36,777
# Trades	3,430	11,178	13,306	14,911	15,219
P&L 0.2% (d=0%)	-7,658	-15,867	-19,045	-22,750	-24,414
Agg P&L 0.2% (d=0%)	-2,532	-4,495	-5,641	-6,685	-6,909
P&L 0.2% (d=15%)	2,868	7,960	6,791	6,606	6,422
Agg P&L 0.2% (d=15%)	4,533	8,739	7,483	7,418	7,311
P&L 0.4% (d=0%)	-22,919	-60,550	-70,894	-82,572	-85,604
Agg P&L 0.4% (d=0%)	-12,667	-37,805	-44,085	-50,441	-50,595
P&L 0.4% (d=15%)	-1,949	2,797	1,935	1,371	1,124
Agg P&L 0.4% (d=15%)	1,383	4,355	3,319	2,993	2,901
Panel B: Corrado and Su trading strategy with modified delta values					
P&L	7,837	29,208	33,219	37,044	37,097
# Trades	3,430	11,178	13,306	14,911	15,219
P&L 0.2% (d=0%)	-7,209	-15,317	-18,610	-22,828	-24,203
Agg P&L 0.2% (d=0%)	-2,332	-3,843	-5,186	-6,708	-6,615
P&L 0.2% (d=15%)	3,512	8,685	7,322	6,740	6,778
Agg P&L 0.2% (d=15%)	4,943	9,539	8,024	7,594	7,720
P&L 0.4% (d=0%)	-22,255	-59,841	-70,439	-82,700	-85,503
Agg P&L 0.4% (d=0%)	-12,501	-36,893	-43,590	-50,460	-50,328
P&L 0.4% (d=15%)	-1,218	3,521	2,303	1,172	1,074
Agg P&L 0.4% (d=15%)	1,646	5,229	3,707	2,881	2,958

Table 1.7. Trading strategies for the Corrado and Su models
P&L is the total profit and loss without transaction costs; # Trades is the number of trades. P&L (*d*=0 and 15%) represents the P&L at 0.2% or 0.4% transaction costs when we ignore trades whose absolute percentage of mispricing error between model estimates and market values is at least 0% and 15% respectively. *Agg.* refers to aggregating the position on the underlying asset to reduce transaction costs. Panel A tabulates results with standard delta values whilst Panel B tabulates results with Chen and Johnson modified delta values

Similar results hold for the CS models although CS_{avT4}^{P} generates slightly higher profits compared to CS_{con}^{P}. Realizing that our simpler trading strategy does not discriminate between high or low expected trading profits, we compute P&L when trades occur only when an expected profit of at least d = 15% is expected. Now we observe that all models can be profitable even under 0.4% transaction costs. Overall we may conclude the following. First, without transaction costs, the CS models produce higher P&L than their counterpart BS models. This is expected since the delta values generated by CS models are consistently higher than those of BS models (for example the median delta values of BS_{con}^{P} for *AggTs* is 0.632 whilst for CS^{P}_{con} is 0.697), making CS based trading more aggressive. Moreover, CS with {*av*} and {*avT*} volatility measures, outperforms significantly the equivalent BS models since it generates more than twice the number of trades; this may happen because unlike the BS models whose implied volatility changes more smoothly, CS models implied skewness and kurtosis can change more erratically. Secondly, and for the same reason, CS models under 0.2% or 0.4% transaction costs become inferior to their BS counterparts. Thirdly, from unreported calculations we have seen that as *d* increases we generally observe P&L to increase in a diminishing fashion indicating that there is an optimal *d* for maximizing trading profits. Finally, trading "in aggregate" positions leads to significant further savings on transaction costs.

	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{av}	BS^{Nh*}_{avT}	BS^{Nh*}_{avT4}	BS^{Nh*}_{con}
			Panel A: Black and Scholes based hybrid ANNs			
P&L	27,024	29,529	32,908	33,514	35,774	37,281
# Trades	5,675	8,246	8,907	9,457	11,995	12,650
P&L 0.2% (d=0%)	1,694	-4,193	-2,435	-4,134	-11,484	-12,939
Agg P&L 0.2% (d=0%)	10,552	6,053	7,871	7,086	837	1,066
P&L 0.2% (d=15%)	6,593	5,147	8,162	8,579	7,910	8,427
Agg P&L 0.2% (d=15%)	8,247	6,977	9,890	9,957	8,689	9,237
P&L 0.4% (d=0%)	-23,637	-37,914	-37,778	-41,782	-58,741	-63,158
Agg P&L 0.4% (d=0%)	-5,920	-17,424	-17,166	-19,343	-34,100	-35,148
P&L 0.4% (d=15%)	1,804	-277	2,232	3,156	4,364	4,812
Agg P&L 0.4% (d=15%)	5,112	3,382	5,687	5,911	5,922	6,432
	CS^{Nh*}_{60}		CS^{Nh*}_{av}	CS^{Nh*}_{avT}	CS^{Nh*}_{avT4}	CS^{Nh*}_{con}
			Panel B: Corrado and Su based hybrid ANNs			
P&L	26,691		32,915	31,943	34,907	37,975
# Trades	5,140		10,043	10,377	12,537	12,947
P&L 0.2% (d=0%)	3,590		-8,721	-12,019	-17,527	-16,084
Agg P&L 0.2% (d=0%)	11,032		3,734	898	-1,586	735
P&L 0.2% (d=15%)	7,337		6,653	5,601	6,052	7,826
Agg P&L 0.2% (d=15%)	8,861		8,231	7,114	7,439	8,960
P&L 0.4% (d=0%)	-19,511		-50,356	-55,980	-69,962	-70,143
Agg P&L 0.4% (d=0%)	-4,626		-25,446	-30,146	-38,078	-36,505
P&L 0.4% (d=15%)	2,433		724	457	612	2,605
Agg P&L 0.4% (d=15%)	5,481		3,879	3,484	3,387	4,873

Table 1.8. Trading strategies for the hybrid ANN models
P&L is the total profit and loss without transaction costs; # Trades is the number of trades. P&L (*d*=0 and 15%) represents the P&L at 0.2% or 0.4% transaction costs when we ignore trades whose absolute percentage of mispricing error between model estimates and market values is at least 0% and 15% respectively. *Agg.* refers to aggregating the position on the underlying asset to reduce transaction costs. Panel A tabulates results for the hybrid BS based ANN model whilst Panel B tabulates results for the hybrid CS based ANN models.

In Table 1.8 we present results for the trading strategies based on ANNs (only for the hybrid models with time adjusted parameters). In general we observe similar results to those of the parametric models. Contrary though to the parametric OPMs, the ANNs offer significant improvement in the cases of less sophisticated parameter estimates. For example, $BS_{av}^{Nh^*}$ produces a P&L equal to 32,908 compared to a P&L equal to 14,088 in the case of BS_{av}^{P}. The best models provide profits in 77%-82% of transactions (detailed figures not reported for brevity) using both the full and the reduced dataset. Finally, in the presence of transaction costs the BS based hybrid model with contract specific volatility is not only the best performing ANN model, but also the overall best. A final observation is that the ability to generate profits even under a considerable level of transaction costs (we do not report here, but the best strategies retained profitability even up to a level of 0.5% of transaction costs) provides some evidence of inefficiency in these options markets. Our study however is not intended to be a test of market efficiency.

1.6.1. Improving trading with the Chen and Johnson approach

We now extend the trading strategies by utilizing with all models the improved hedging scheme suggested by Chen and Johnson (1985). This is a widely neglected (see Roon et al., 1998 for a rare exception in the use of parametric models) approach that deals with deriving hedge parameters under the assumption of mispriced options. According to this hedging scheme and when an option is mispriced, the delta hedge parameter, $\Delta_{m,t}^{k}$, should be derived in a different way. If a mispriced option has been identified, then the riskless hedge will not earn *r*, the riskless rate, but some other rate, *r**. Chen and Johnson obtain the expression for a European call option that is the same as BS presented in Eqs (1, 1.a) and (1.1.b), by replacing *r* with *r**. In order to derive the correct hedge ratio, Equation 1 must be solved numerically for *r** using the observed market price of c^{mrk} (like retrieving the implied interest rate). We implement this approach with the parametric BS and CS models, and the ANNs.

Finding the implied interest rate, *r**, for the case of BS or CS is a simple numerical task and we employ the repeated cubic interpolation technique according to Charalambous (1992). Finding the implied interest rate, *r**, for ANNs is a more involved task, since in the case of hybrid models we need to jointly optimize with respect to the interest rate input to the neural networks and to the interest rate in the parametric model that is used to create the hybrid target function; this introduces many jagged ridge regions in the optimization surface.

Thus, in the case of hybrid ANNs we adopt a more computationally intensive methodology according to which we again use the cubic interpolation technique with ten different initial starting points.

After finding *r** for all models considered we rerun the trading strategies. Results for the parametric BS and CS models appear in Panel B of Tables 1.6 and 1.7. The most important observation is that before transaction costs are accounted for, in all BS models under consideration there is a slight (only) improvement in their profitability (P&L). Under aggregate 0.4% transaction costs and for d = 15%, the improvement in BS_{60}^{P} is about 19%, in BS_{vix}^{P} is surprisingly about 164% and for the more sophisticated BS_{con}^{P} model only 1.67%. We remind that BS_{vix}^{P} exhibited both, the poorest out of sample pricing performance and only a modest profitability (under 0.4% transaction costs) among the BS models. Under the adjusted deltas, this seems to be partly alleviated. Somewhat similar results we observe for the semi-parametric CS model. For both parametric models, the modified hedging approach under transaction costs gave the best results when using the average (not contract specific) parameters. In the case of ANNs (results unreported for brevity) and under no transaction costs, we also observe a slight tendency for increased performance, but the results are mixed. With transaction costs the technique was unable to improve the profitability of ANNs. The above observations refer to the full dataset, but they also hold for the reduced one (again not reported due to brevity).

A general observation for the use of the modified hedging approach in trading strategies is that it significantly improves trading performance when it is applied with POPM models under assumptions consistent with the assumptions under which this approach was developed. Thus, it performs well with the parametric models when either historical, or average implied parameters are used. The use of this approach did not reverse our previous findings about the best performing models when trading in the presence of transaction costs. Still, it demonstrated that simple models can be efficient alternatives to the more sophisticated and computationally intensive hybrid ANN methods.

1.6.2. Delta hedging

We have also considered hedging as a testing tool. Our results here coincide with previous literature – model ranking may differ if testing is based on hedging instead of pricing. Bakshi et al. (1997) compare alternative parametric models and state that the hedging-based ranking of the models is in sharp contrast with that obtained based on out of sample pricing. They also state that (delta-hedging) performance is virtually indistinguishable among models. Quite similar results are reported in papers where non-parametric methods were used, like Garcia and Gencay (2000), and Gencay and Qi (2001). Schittenkopf and Dorffner (2001) find the results (marginally) better for the parametric models, but practically indistinguishable. Hutchison et al. (1994) also report that the learning networks they use have a better hedging performance compared to BS but they find it difficult to infer which network type performs best. We attribute this difference of model ranking to the fact that models are usually optimized with respect to pricing. An exception is Carverhill and Cheuk (2003) who focus more on hedging performance by optimizing with respect to the hedge parameters. Optimizing the "hedging performance" is beyond the scope of this essay. Furthermore, hedging performance is not a substitute for trading performance, since hedging tests fail to account for the difference between overpriced and underpriced options.

We have calculated the mean hedging error (MHE) and the mean absolute hedging error (MAHE) of a standard hedging strategy with daily rebalancing. For brevity we do not report the full results here, but we have found according to MHE that the best parametric model is the CS^{P}_{con}. Among the ANN models the best performing one is CS^{Nh*}_{con}, with an identical error for the parametric CS model (equal for both models to 0.26). In addition, the error equals 0.30 for both the BS^{P}_{con} and the BS^{Nh*}_{con} models. In general, from the MHE we cannot tell which POPM is the best since their difference in this measure is practically indistinguishable. Continuing with the MAHE we have the same picture, and we find it hard to observe a certain POPM that dominates in this measure since many models have "almost identical" MAHE values. It is true that BS^{P}_{con} and BS^{P}_{avT4} are the overall best models (with MAHE equal to 2.57 for both) and perform relatively better than the ANN models (their hybrid ANN counterparts both having an error equal to 2.63).

In general, we can conclude that the hedging error performance is not in line with the models' pricing performance. That is, our best model in pricing accuracy, BS_{con}^{Nh*}, does not produce the smallest hedging errors. But again, it is truly hard to differentiate among models. The above discussion pertains to the full dataset, but we have observed that ranking models using hedging performance is not affected by the choice of dataset.

1.7. Conclusions

Our effort has focused in developing European option pricing and trading tools by combining the use of ANN methodology and information provided by parametric OPMs (the BS and the CS model). For our empirical tests we have used European call options on the S&P 500 Index from January 1998 to August 2001. In our analysis we have included historical parameters, a VIX volatility proxy derived by weighting implied volatilities (for the case of BS only), and implied parameters (an overall average, an average per maturity, the 4-point closest in moneyness, and a contract-specific parameter set). Neural networks are optimized using a modified Levenberg-Marquardt training algorithm. We include in the analysis simple ANNs (with input supplemented by historical or implied parameters specific either to BS or the CS model), and hybrid ANNs that in addition use pricing information derived by any of the two parametric models. In order to check the robustness of the results, in addition to our *full* dataset we repeat the analysis using a *reduced* dataset (following Hutchison et al., 1994). The economic significance of the models is investigated through trading strategies with transaction costs. Instead of *naive* trading strategies we implement improved (dynamic and cost-effective) ones. Furthermore, we also refine these strategies with the Chen and Johnson (1985) modified hedging approach. Our results can be synopsized as follows:

Regarding the in sample pricing, CS performs better than the BS model (with the exception of the case of the contract specific implied parameters that practically eliminate the pricing error).

Regarding out of sample pricing, CS outperforms BS with the use of average implied parameters, but BS is still a better model when the contract specific implied parameters are used; in general, implied parameters lead to better performance than the historical ones or the VIX volatility proxy; the simple neural networks cannot outperform the parametric models in the full range of data, but we verified allegations to the contrary found in the literature with the use of a reduced data set; hybrid neural networks that combine both neural network technology and the parametric models provide the best performance, especially when contract specific and adjusted parameters are used. The BS based hybrid ANN (with contract specific parameters) is the overall best performer, and the equivalent CS hybrid often a good alternative.

In trading and before transaction costs, models using contract specific implied parameters provide the best performance. But they also lead to the highest number of trades. In trading when transaction costs are accounted for in a *naive* manner, profits practically in all cases disappear. In trading and even with 0.4% transaction costs, when dynamic cost-efficient strategies are implemented, profits are still feasible hinting thus to potential market inefficiencies. The parametric BS with contract specific volatility is the best among the parametric models. The hybrid ANN based on BS with contract specific volatility is the overall best.

Implementing the widely neglected Chen and Johnson (1985) modified hedging approach, improves significantly the profitability of trading strategies that are based on the parametric models with average implied parameters (the models more consistent with the assumptions behind the modified hedging approach). This approach did not affect the choice of the overall best model in terms of trading with transaction costs. But it did demonstrate that reasonable alternatives for trading do exist without the need to resort to the extra sophistication of ANN technology.

1.A. Appendix with full dataset results

	VDOTM	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM	VDITM
S/X	**<0.85**	**0.85-0.95**	**0.90-0.95**	**0.95-0.99**	**0.99-1.01**	**1.01-1.05**	**1.05-1.10**	**1.10-1.35**	**≥1.35**
				Short Term Options <60 Days					
Call	3.61	1.63	5.15	15.70	32.40	56.58	99.55	199.77	470.38
σ_{con}^{BS}	0.36	0.21	0.19	0.19	0.20	0.22	0.27	0.38	0.99
# obs	399	1,361	4,815	7,483	3,964	6,548	4,970	7,990	2,103
				Medium Term Options 60-180 Days					
Call	4.38	8.29	23.58	46.06	64.51	90.35	131.10	227.41	493.18
σ_{con}^{BS}	0.22	0.18	0.20	0.21	0.21	0.23	0.25	0.30	0.54
# obs	1,412	1,727	2,578	3,147	1,780	2,901	3,038	8,100	3,999
				Long Term Options ≥ 180 Days					
Call	9.65	42.09	74.03	106.24	126.03	150.99	185.87	267.12	495.82
σ_{con}^{BS}	0.18	0.21	0.22	0.23	0.24	0.25	0.26	0.28	0.40
# obs	332	333	575	603	343	660	812	2,695	1,733

Table F1: Sample descriptive statistics

Sample characteristics for the period January 5, 1998 to August 24, 2001 concerning the average call option value, the average Black and Scholes implied volatility and the number of observations in each moneyness/maturity class.

Set	*Starting*	*Ending*	*# obs*	*Set*	*Starting*	*Ending*	*# obs*
Tr1	5-Jan-98	8-Mar-99	22,545	**Tr6**	9-Mar-99	20-Jan-00	20,637
Vd1	9-Mar-99	12-Jul-99	10,916	**Vd6**	21-Jan-00	17-May-00	10,511
Ts1	13-Jul-99	24-Sep-99	4,092	**Ts6**	18-May-00	17-Jul-00	3,959
Tr2	24-Apr-98	16-Apr-99	22,038	**Tr7**	20-Apr-99	28-Feb-00	20,050
Vd2	19-Apr-99	23-Sep-99	10,579	**Vd7**	29-Feb-00	17-Jul-00	10,589
Ts2	24-Sep-99	5-Jan-00	4,122	**Ts7**	18-Jul-00	10-Oct-00	4,264
Tr3	23-Jun-98	3-Jun-99	21,304	**Tr8**	7-Jun-99	11-Apr-00	20,037
Vd3	4-Jun-99	5-Jan-00	10,660	**Vd8**	12-Apr-00	6-Oct-00	10,711
Ts3	6-Jan-00	10-Feb-00	3,963	**Ts8**	9-Oct-00	24-Jan-01	3,797
Tr4	3-Sep-98	24-Aug-99	20,950	**Tr9**	26-Aug-99	5-Jun-00	19,852
Vd4	25-Aug-99	11-Feb-00	10,616	**Vd9**	6-Jun-00	24-Jan-01	10,504
Ts4	14-Feb-00	27-Mar-00	3,813	**Ts9**	25-Jan-01	29-Mar-01	3,945
Tr5	29-Jan-99	18-Oct-99	20,631	**Tr10**	21-Oct-99	11-Aug-00	20,042
Vd5	19-Oct-99	24-Mar-00	10,405	**Vd10**	14-Aug-00	30-Mar-01	10,372
Ts5	28-Mar-00	16-May-00	4,037	**Ts10**	2-Apr-01	24-Aug-01	3,839

Table F2: Training (Tr), validation (Vd) and testing (Ts) dates

	BS_{av}^{P}	BS_{avT}^{P}	BS_{avT4}^{P}	BS_{con}^{P}	CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
	In sample descriptive statistics for 1998							
RMSE	18.85	18.71	11.79	0.00	16.33	15.06	9.34	3.20
MAE	7.48	6.96	3.78	0.00	7.39	5.51	2.89	0.29
RMeSE	3.61	2.92	0.71	0.00	3.26	1.12	0.24	0.00
	In sample descriptive statistics for 1999							
RMSE	9.16	8.61	4.53	0.00	6.44	5.20	2.99	0.63
MAE	6.51	5.87	1.73	0.00	3.59	2.01	0.85	0.03
RMeSE	5.30	4.26	0.72	0.00	2.53	0.82	0.11	0.00
	In sample descriptive statistics for 2000							
RMSE	8.55	8.10	5.59	0.04	6.90	5.93	4.19	1.64
MAE	5.05	4.53	1.57	0.00	2.61	1.84	0.84	0.07
RMeSE	3.74	3.18	0.61	0.00	1.42	0.58	0.12	0.00
	In sample descriptive statistics for 2001							
RMSE	4.83	4.46	1.95	0.00	2.49	1.97	1.06	0.25
MAE	3.46	3.12	1.03	0.00	1.41	0.99	0.43	0.01
RMeSE	2.68	2.37	0.54	0.00	0.85	0.53	0.16	0.00
	Total in sample descriptive statistics (1998-2001)							
RMSE	11.63	11.31	7.00	0.02	9.52	8.52	5.35	1.82
MAE	5.86	5.33	2.06	0.00	3.87	2.63	1.26	0.10
RMeSE	3.95	3.24	0.65	0.00	1.89	0.74	0.14	0.00

Table F3: Parametric models in sample fitting errors
Fitting errors for all versions of the Black-Scholes and Corrado and Su models. RMSE is the Root Mean Square Error, MAE the Mean Absolute Deviation and RMeSE the Root Median Square Error.

	σ_{av}^{BS}	σ_{avT}^{BS}	σ_{avT4}^{BS}	σ_{con}^{BS}	σ_{av}^{CS}	$\mu3_{av}^{CS}$	$\mu4_{av}^{CS}$	σ_{avT}^{CS}
Min	0.10	0.05	0.05	0.02	0.12	-9.70	0.00	0.06
5th Perc	0.18	0.17	0.16	0.16	0.19	-1.73	3.31	0.18
Mean	0.23	0.22	0.28	0.29	0.30	-1.19	6.91	0.27
Median	0.22	0.22	0.24	0.24	0.26	-1.21	5.20	0.24
95th Perc	0.29	0.29	0.52	0.56	0.56	-0.39	18.02	0.55
Max	0.71	1.11	4.58	5.34	1.85	1.33	30.00	1.85

	$\mu3_{avT}^{CS}$	$\mu4_{avT}^{CS}$	σ_{avT4}^{CS}	$\mu3_{avT4}^{CS}$	$\mu4_{avT4}^{CS}$	σ_{con}^{CS}	$\mu3_{con}^{CS}$	$\mu4_{con}^{CS}$
Min	-9.70	0.00	0.00	-10.00	0.00	0.00	-10.00	0.00
5th Perc	-2.30	2.84	0.14	-3.71	0.00	0.14	-3.71	0.00
Mean	-1.20	6.19	0.30	-0.79	7.14	0.31	-0.79	7.14
Median	-1.21	5.20	0.26	-0.98	5.74	0.26	-0.98	5.74
95th Perc	-0.17	13.81	0.57	2.90	25.25	0.71	2.90	25.25
Max	5.00	30.00	2.00	5.00	30.00	2.50	5.00	30.00

Table F4: Implied parameters descriptive statistics

Descriptive statistics for the parametric models implied parameters for the period January 5, 1998 to August 24, 2001. For each model it is tabulated the minimum, the 5th percentile, the mean and median, the 95th percentile and the maximum values. Brownian volatility is symbolized with σ, skewness with $\mu 3$, whilst kurtosis with $\mu 4$. The superscripts refer to the kind of the parametric model whilst the subscripts refer to the kind of implied parameter.

	BS^P_{60}	BS^P_{vix}	BS^P_{av}	BS^P_{avT}	BS^P_{avT4}	BS^P_{con}	CS^P_{60}		CS^P_{av}	CS^P_{avT}	CS^P_{avT4}	CS^P_{con}
RMSE	11.18	12.57	9.72	9.47	8.03	7.04	11.25		8.89	8.87	8.11	7.71
MAE	6.83	8.60	5.32	5.00	3.10	2.70	6.89		3.86	3.72	3.27	3.10
RMeSE	4.48	6.38	3.74	3.37	1.52	1.43	4.61		2.26	1.94	1.69	1.68

	BS^N_{60}	BS^N_{vix}	BS^N_{av}	BS^N_{avT}	BS^N_{avT4}	BS^N_{con}	BS^{N*}_{60}	BS^{N*}_{vix}	BS^{N*}_{av}	BS^{N*}_{avT}	BS^{N*}_{avT4}	BS^{N*}_{con}
RMSE	13.06	12.65	10.97	12.48	10.74	9.06	14.68	12.76	12.30	11.69	9.33	7.86
MAE	7.58	6.65	5.91	7.04	6.04	4.68	7.68	6.70	6.67	6.55	5.04	3.81
RMeSE	5.13	3.83	3.65	4.11	3.69	2.88	4.71	3.65	3.99	3.94	2.94	2.44

	CS^N_{60}		CS^N_{av}	CS^N_{avT}	CS^N_{avT4}	CS^N_{con}	CS^{N*}_{60}		CS^{N*}_{av}	CS^{N*}_{avT}	CS^{N*}_{avT4}	CS^{N*}_{con}
RMSE	15.22		11.28	11.59	9.87	11.83	14.35		11.42	11.96	9.47	9.76
MAE	9.13		5.80	6.14	5.73	5.81	7.71		5.39	5.56	4.67	4.87
RMeSE	6.43		3.48	3.96	3.65	3.65	4.27		3.26	3.15	2.93	3.03

	BS^{Nh}_{60}	BS^{Nh}_{vix}	BS^{Nh}_{av}	BS^{Nh}_{avT}	BS^{Nh}_{avT4}	BS^{Nh}_{con}	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{av}	BS^{Nh*}_{avT}	BS^{Nh*}_{avT4}	BS^{Nh*}_{con}
RMSE	9.05	8.35	8.57	8.29	7.79	6.38	9.03	8.27	8.87	7.84	7.68	6.01
MAE	5.40	4.55	4.35	4.09	3.30	2.68	5.46	4.53	4.35	3.91	3.17	2.61
RMeSE	3.73	2.98	2.83	2.51	1.80	1.60	3.98	3.00	2.69	2.53	1.67	1.58

	CS^{Nh}_{60}		CS^{Nh}_{av}	CS^{Nh}_{avT}	CS^{Nh}_{avT4}	CS^{Nh}_{con}	CS^{Nh*}_{60}		CS^{Nh*}_{av}	CS^{Nh*}_{avT}	CS^{Nh*}_{avT4}	CS^{Nh*}_{con}
RMSE	10.33		8.68	8.63	7.97	7.60	9.68		8.83	8.66	7.60	7.39
MAE	6.38		4.12	3.84	3.42	3.14	6.20		3.95	3.94	3.39	3.11
RMeSE	4.46		2.42	2.17	1.93	1.77	4.56		2.33	2.35	1.96	1.82

Table F5: Error pricing measures for all models in the aggregate testing period (AggTs)
RMSE is the Root Mean Square Error, MAE the Mean Absolute Deviation and RMeSE the Root Median Square Error. The superscripts refer to the kind of the model: *P* refers to parametric models, *N* to the simple neural networks and *Nh* to the hybrid neural networks. The asterisk (*) refers to neural network models that use transformed variables. The subscripts refer to kind of historical/implied parameters used to each model per se.

	BS^P_{60}	BS^P_{vix}	BS^P_{av}	BS^P_{avT}	BS^P_{avT4}	BS^P_{con}	CS^P_{60}		CS^P_{av}	CS^P_{avT}	CS^P_{avT4}	CS^P_{con}
MHE	0.24	0.27	0.25	0.24	0.25	0.25	0.24		0.24	0.24	0.23	0.22
MAHE	2.74	2.89	2.74	2.72	2.62	2.61	2.74		2.90	2.88	2.91	2.91
MPE	5.93	5.99	5.93	5.92	5.86	5.86	5.95		6.08	6.08	6.13	6.14
MD	0.592	0.597	0.594	0.592	0.571	0.569	0.592		0.616	0.613	0.610	0.607

	BS^N_{60}	BS^N_{vix}	BS^N_{av}	BS^N_{avT}	BS^N_{avT4}	BS^N_{con}	BS^{N*}_{60}	BS^{N*}_{vix}	BS^{N*}_{av}	BS^{N*}_{avT}	BS^{N*}_{avT4}	BS^{N*}_{con}
MHE	0.35	0.31	0.29	0.29	0.28	0.27	0.34	0.29	0.26	0.30	0.31	0.28
MAHE	3.45	3.28	3.29	3.09	2.92	2.88	3.27	3.16	3.01	3.18	2.98	2.86
MPE	6.67	6.52	6.52	6.27	6.17	6.12	6.52	6.39	6.20	6.40	6.21	6.13
MD	0.647	0.637	0.640	0.622	0.598	0.594	0.635	0.626	0.617	0.627	0.601	0.593

	CS^N_{60}		CS^N_{av}	CS^N_{avT}	CS^N_{avT4}	CS^N_{con}	CS^{N*}_{60}		CS^{N*}_{av}	CS^{N*}_{avT}	CS^{N*}_{avT4}	CS^{N*}_{con}
MHE	0.30		0.30	0.31	0.24	0.25	0.31		0.31	0.28	0.27	0.25
MAHE	3.42		3.17	3.19	2.98	3.01	3.23		3.20	3.04	2.98	2.94
MPE	6.62		6.40	6.43	6.19	6.23	6.46		6.40	6.24	6.20	6.14
MD	0.64		0.632	0.630	0.612	0.616	0.630		0.624	0.611	0.607	0.602

	BS^{Nh}_{60}	BS^{Nh}_{vix}	BS^{Nh}_{av}	BS^{Nh}_{avT}	BS^{Nh}_{avT4}	BS^{Nh}_{con}	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{av}	BS^{Nh*}_{avT}	BS^{Nh*}_{avT4}	BS^{Nh*}_{con}
MHE	0.25	0.26	0.25	0.25	0.26	0.25	0.26	0.28	0.25	0.25	0.26	0.26
MAHE	2.96	2.97	2.95	2.94	2.66	2.64	2.96	3.00	2.93	2.93	2.66	2.69
MPE	6.12	6.12	6.12	6.09	5.89	5.89	6.11	6.17	6.09	6.08	5.90	5.94
MD	0.621	0.621	0.623	0.621	0.582	0.580	0.617	0.626	0.621	0.619	0.583	0.586

	CS^{Nh}_{60}		CS^{Nh}_{av}	CS^{Nh}_{avT}	CS^{Nh}_{avT4}	CS^{Nh}_{con}	CS^{Nh*}_{60}		CS^{Nh*}_{av}	CS^{Nh*}_{avT}	CS^{Nh*}_{avT4}	CS^{Nh*}_{con}
MHE	0.25		0.24	0.24	0.23	0.23	0.24		0.25	0.24	0.23	0.22
MAHE	2.87		2.94	2.91	2.92	2.92	2.87		2.99	2.94	2.95	2.96
MPE	6.07		6.12	6.11	6.14	6.15	6.07		6.16	6.12	6.17	6.20
MD	0.612		0.623	0.619	0.613	0.609	0.613		0.629	0.621	0.618	0.616

Table F6: Hedging error measures for all models in the aggregate testing period (AggTs)
MHE is the Mean Hedging Error, MAHE the Mean Absolute Hedging Error, MPE the Mean Prediction Error and MD the Mean Delta of each model. The superscripts refer to the kind of the model: *P* refers to parametric models, *N* to the simple neural networks and *Nh* to the hybrid neural networks. The asterisk (*) refers to neural network models that use the transformed variables. The subscripts refer to kind of historical/implied parameters used to each model per se.

	BS^{P}_{60}	BS^{P}_{vix}	BS^{P}_{con}	CS^{P}_{60}	CS^{P}_{con}	BS^{N*}_{60}	CS^{N*}_{60}	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{con}	CS^{Nh*}_{60}	CS^{Nh*}_{con}
BS^{P}_{60}		-27.74	75.12	-0.94	65.72	-11.07	-11.72	23.90	40.83	81.22	10.80	66.92
BS^{P}_{vix}	7.17		104.84	26.74	94.84	11.84	11.71	53.72	70.71	112.32	40.52	96.53
BS^{P}_{con}	-16.13	-25.08		-75.91	-8.43	-70.51	-72.82	-56.94	-38.56	2.12	-70.87	-8.76
CS^{P}_{60}	0.34	-6.72	16.31		66.53	-10.28	-10.91	24.85	41.74	82.02	11.78	67.74
CS^{P}_{con}	-13.38	-21.60	2.14	-13.58		-63.58	-65.63	-46.76	-28.85	11.11	-60.38	-0.10
BS^{N*}_{60}	7.24	4.64	13.37	7.09	12.48		-0.34	30.64	43.94	74.23	20.23	64.28
CS^{N*}_{60}	7.77	4.67	15.19	7.59	14.09	-0.62		31.84	45.50	76.80	21.15	66.39
BS^{Nh*}_{60}	-9.55	-18.30	7.57	-9.81	4.95	-10.83	-12.15		18.64	63.30	-14.28	47.82
BS^{Nh*}_{vix}	-12.54	-21.61	4.48	-12.75	2.02	-11.91	-13.46	-3.25		43.70	-32.87	29.50
BS^{Nh*}_{con}	-21.16	-32.03	-3.45	-21.26	-5.62	-14.65	-16.83	-12.27	-8.78		-78.03	-11.58
CS^{Nh*}_{60}	-6.86	-15.36	10.42	-7.15	7.65	-9.84	-10.96	2.97	6.24	15.52		61.73
CS^{Nh*}_{con}	-14.98	-23.78	1.15	-15.16	-1.04	-12.95	-14.69	-6.34	-3.26	4.73	-9.18	

Table F7a: Matched pair student-t tests for square and absolute differences

Matched pair t-tests concerning the absolute differences are reported in the upper diagonal whilst on the lower diagonal the matched pair t-tests concerning the square differences are tabulated. Both tests compare the MAE and MSE between models in the vertical heading versus models in the horizontal heading. In general, a positive t-value larger than 1.645 (2.325) means that the model in the vertical heading has a larger MAE or MSE than the model in the horizontal heading at 5% (1%) significance level.

	BS_{60}^{P}	BS_{vix}^{P}	BS_{con}^{P}	CS_{60}^{P}	CS_{con}^{P}	BS_{60}^{N*}	CS_{60}^{N*}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{con}^{Nh*}	CS_{60}^{Nh*}	CS_{con}^{Nh*}
BS_{60}^{P}		-21.23	23.69	-4.37	19.79	-7.72	-8.57	36.11	50.44	37.08	29.37	23.50
BS_{vix}^{P}	7.17		35.66	18.88	30.60	-4.88	-5.04	53.19	69.94	55.02	47.51	35.68
BS_{con}^{P}	-16.13	-25.08		-23.68	-6.03	-13.84	-16.00	-11.56	-6.77	7.47	-15.91	-3.41
CS_{60}^{P}	0.34	-6.72	16.31		19.89	-7.57	-8.40	35.16	48.95	36.36	27.77	23.48
CS_{con}^{P}	-13.38	-21.60	2.14	-13.58		-12.97	-14.90	-7.52	-3.02	10.22	-11.61	8.22
BS_{60}^{N*}	7.24	4.64	13.37	7.09	12.48		2.15	11.50	12.60	15.19	10.45	13.46
CS_{60}^{N*}	7.77	4.67	15.19	7.59	14.09	-0.62		13.31	14.69	17.79	11.98	15.56
BS_{60}^{Nh*}	-9.55	-18.30	7.57	-9.81	4.95	-10.83	-12.15		18.82	25.05	-19.39	10.42
BS_{vix}^{Nh*}	-12.54	-21.61	4.48	-12.75	2.02	-11.91	-13.46	-3.25		17.48	-42.22	5.27
BS_{con}^{Nh*}	-21.16	-32.03	-3.45	-21.26	-5.62	-14.65	-16.83	-12.27	-8.78		-31.61	-9.62
CS_{60}^{Nh*}	-6.86	-15.36	10.42	-7.15	7.65	-9.84	-10.96	2.97	6.24	15.52		15.05
CS_{con}^{Nh*}	-14.98	-23.78	1.15	-15.16	-1.04	-12.95	-14.69	-6.34	-3.26	4.73	-9.18	

Table F6b: Matched pair student-t and Johnson t-tests for the square differences
Matched pair student-t (lower diagonal) and Johnson modified t (upper diagonal) tests concerning square differences are tabulated. Both tests compare the MSE between models in the vertical heading versus models in the horizontal heading. In general, a positive t-value larger than 1.645 (2.325) means that the model in the vertical heading has a larger MSE than the model in the horizontal heading at 5% (1%) significance level.

	BS_{60}^{P}	BS_{vix}^{P}	BS_{av}^{P}	BS_{avT}^{P}	BS_{avT4}^{P}	BS_{con}^{P}	CS_{60}^{P}	CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
Ts1	16.69	14.71	18.06	17.99	17.16	17.15	16.68	17.78	17.99	18.33	18.25
Ts2	10.95	11.50	9.80	9.76	8.00	4.52	11.17	9.38	9.60	6.90	5.99
Ts3	13.87	13.28	9.94	9.25	5.92	4.07	15.24	7.91	7.72	6.07	5.10
Ts4	14.18	14.86	12.93	12.37	9.67	5.77	14.46	11.89	11.85	9.45	7.53
Ts5	10.09	16.97	7.65	7.39	5.25	4.73	8.53	6.01	5.94	5.49	5.31
Ts6	11.95	10.71	7.97	7.64	6.18	4.14	9.82	6.58	5.66	4.81	4.25
Ts7	7.59	8.09	5.69	5.46	4.62	2.77	7.61	4.66	4.30	3.38	3.47
Ts8	8.21	12.66	6.15	6.09	5.23	5.98	9.34	5.52	5.52	5.74	5.39
Ts9	6.97	11.25	6.00	5.73	4.45	4.73	8.33	4.97	5.07	4.85	5.05
Ts10	6.12	9.13	4.96	4.72	4.18	4.48	6.17	4.74	4.91	4.60	4.80
	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}	CS_{60}^{Nh*}	CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
Ts1	14.06	13.51	17.21	14.50	16.00	13.70	13.98	18.82	17.35	15.74	17.49
Ts2	9.14	8.42	8.55	8.07	7.61	4.12	9.08	8.71	9.51	7.38	5.69
Ts3	10.45	9.62	10.89	7.35	5.74	3.77	12.04	7.51	7.71	7.06	5.09
Ts4	11.51	11.81	10.81	11.01	9.46	5.75	12.74	10.80	11.39	9.30	7.50
Ts5	8.85	7.71	6.34	5.83	5.17	4.53	9.93	6.01	5.73	5.51	5.07
Ts6	8.02	6.68	6.32	6.53	6.43	4.01	8.37	6.08	5.96	4.80	4.21
Ts7	6.45	4.91	4.97	4.70	4.48	2.62	6.97	5.22	5.12	3.47	3.35
Ts8	6.84	5.87	5.45	5.07	4.98	5.34	6.78	5.22	5.05	5.48	5.12
Ts9	5.48	4.37	4.80	4.44	4.38	4.38	6.26	4.61	4.85	4.58	4.37
Ts10	5.47	4.01	4.43	4.25	4.07	3.92	7.03	4.45	4.54	4.29	4.43

Table F7: Testing periods RMSE for the best performing models

	BS^{N}_{60}	BS^{N}_{vix}	BS^{N}_{av}	BS^{N}_{avT}	BS^{N}_{avT4}	BS^{N}_{con}	BS^{N*}_{60}	BS^{N*}_{vix}	BS^{N*}_{av}	BS^{N*}_{avT}	BS^{N*}_{avT4}	BS^{N*}_{con}
Min	2	4	2	5	6	3	2	4	3	4	4	2
Mean	6.7	6.4	6.7	8.3	8.5	7.7	6.4	6.3	6.7	7.3	7.4	8
Max	10	10	10	10	10	10	10	9	10	10	10	10

	CS^{N}_{60}		CS^{N}_{av}	CS^{N}_{avT}	CS^{N}_{avT4}	CS^{N}_{con}	CS^{N*}_{60}		CS^{N*}_{av}	CS^{N*}_{avT}	CS^{N*}_{avT4}	CS^{N*}_{con}
Min	2		2	2	5	6	3		3	4	3	4
Mean	6.1		6.5	7.7	8.2	8.6	6.9		5	7.1	7.2	7.9
Max	9		10	10	10	10	9		9	10	10	10

	BS^{Nh}_{60}	BS^{Nh}_{vix}	BS^{Nh}_{av}	BS^{Nh}_{avT}	BS^{Nh}_{avT4}	BS^{Nh}_{con}	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{av}	BS^{Nh*}_{avT}	BS^{Nh*}_{avT4}	BS^{Nh*}_{con}
Min	2	3	2	2	2	2	2	2	2	2	2	2
Mean	4.7	5.3	4.1	3.5	4	4.5	4.5	5	4.2	4	4.1	3.2
Max	9	9	9	6	7	7	10	8	8	7	7	6

	CS^{Nh}_{60}		CS^{Nh}_{av}	CS^{Nh}_{avT}	CS^{Nh}_{avT4}	CS^{Nh}_{con}	CS^{Nh*}_{60}		CS^{Nh*}_{av}	CS^{Nh*}_{avT}	CS^{Nh*}_{avT4}	CS^{Nh*}_{con}
Min	3		2	2	2	2	4		2	2	2	2
Mean	6.5		5	4.7	6.2	4.8	7.4		4.5	4.8	5.3	4.9
Max	10		10	9	10	7	10		10	9	8	10

Table F8: ANNs complexity
Minimum, mean and maximum number of the hidden layer neurons for the ten different training periods.

	BS_{con}^{P}			CS_{con}^{P}		
	Short	**Medium**	**Long**	**Short**	**Medium**	**Long**
VDOTM	3.60	4.91	0.56	8.34	10.61	0.66
DOTM	2.27	4.50	2.82	3.02	5.24	4.47
OTM	5.78	8.37	3.97	6.29	9.68	5.08
JOTM	7.81	6.68	6.15	8.13	7.64	7.65
ATM	6.67	9.46	5.86	7.30	10.14	7.29
JITM	6.71	9.41	4.34	7.29	9.21	5.97
ITM	7.70	7.13	4.43	8.24	7.59	5.18
DITM	7.07	7.93	7.27	7.20	8.50	7.50
VDITM	8.26	9.46	8.74	8.29	10.05	9.05
	BS_{con}^{Nh*}			CS_{con}^{Nh*}		
VDOTM	3.60	4.97	1.15	6.13	10.22	6.04
DOTM	2.46	4.83	2.32	2.96	5.28	5.03
OTM	5.50	7.75	3.98	6.19	9.41	5.36
JOTM	5.89	5.36	5.78	7.83	7.30	7.66
ATM	4.73	8.18	5.38	6.94	9.86	7.13
JITM	5.59	7.39	4.10	6.89	8.68	6.64
ITM	6.24	6.05	3.95	7.58	7.16	5.69
DITM	5.80	7.15	6.74	6.64	8.04	7.17
VDITM	8.03	9.29	8.46	8.96	10.33	9.26

Table F9: Root Mean Square Errors for the four best models

	BS^{P}_{60}	BS^{P}_{vix}	BS^{P}_{av}	BS^{P}_{avT}	BS^{P}_{avT4}	BS^{P}_{con}
P&L	7,447	13,518	14,088	13,069	32,040	35,026
# Trades	3,361	3,878	4,858	5,477	13,539	15,644
P&L 0.2% (d=0%)	-6,829	-6,847	-5,348	-7,512	-17,911	-23,307
Agg P&L 0.2% (d=0%)	-1,861	-266	737	-1,394	-5,638	-8,437
P&L 0.2% (d=5%)	-153	3,587	4,206	1,731	5,971	5,716
Agg P&L 0.2% (d=5%)	2,717	5,756	7,217	4,693	8,319	8,304
P&L 0.2% (d=10%)	1,349	3,723	7,478	5,879	7,334	7,206
Agg P&L 0.2% (d=10%)	3,433	4,924	8,928	7,373	8,251	8,201
P&L 0.2% (d=15%)	3,320	4,134	7,527	6,841	7,907	7,369
Agg P&L 0.2% (d=15%)	5,003	5,019	8,344	7,657	8,384	7,873
P&L 0.4% (d=0%)	-21,105	-27,211	-24,785	-28,093	-67,863	-81,640
Agg P&L 0.4% (d=0%)	-11,170	-14,049	-12,614	-15,858	-43,316	-51,899
P&L 0.4% (d=5%)	-8,636	-5,025	-9,117	-12,571	-9,059	-9,177
Agg P&L 0.4% (d=5%)	-2,897	-688	-3,093	-6,647	-4,364	-4,001
P&L 0.4% (d=10%)	-4,748	-1,988	455	-2,141	1,070	995
Agg P&L 0.4% (d=10%)	-580	414	3,354	848	2,903	2,986
P&L 0.4% (d=15%)	-1,468	-508	3,241	2,269	4,691	4,212
Agg P&L 0.4% (d=15%)	1,897	1,262	4,875	3,901	5,645	5,221

Table F10: Trading strategies for Black and Scholes models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS^P_{60}	BS^P_{vix}	BS^P_{av}	BS^P_{avT}	BS^P_{avT4}	BS^P_{con}
P&L	7,916	14,367	14,232	13,441	32,281	35,229
# Trades	3,361	3,878	4,858	5,477	13,539	15,644
P&L 0.2% (d=0%)	-6,169	-5,599	-4,958	-6,946	-17,788	-23,080
Agg P&L 0.2% (d=0%)	-1,392	1,342	1,225	-778	-5,534	-8,259
P&L 0.2% (d=5%)	492	4,867	4,642	2,293	6,271	6,088
Agg P&L 0.2% (d=5%)	3,044	7,352	7,758	5,278	8,653	8,625
P&L 0.2% (d=10%)	2,069	5,060	8,044	6,453	7,673	7,522
Agg P&L 0.2% (d=10%)	3,863	6,457	9,622	8,086	8,615	8,477
P&L 0.2% (d=15%)	4,044	5,534	8,182	7,546	8,306	7,713
Agg P&L 0.2% (d=15%)	5,515	6,558	9,115	8,524	8,815	8,198
P&L 0.4% (d=0%)	-20,254	-25,564	-24,148	-27,334	-67,858	-81,390
Agg P&L 0.4% (d=0%)	-10,700	-11,682	-11,782	-14,998	-43,348	-51,748
P&L 0.4% (d=5%)	-7,807	-3,352	-8,451	-11,817	-8,924	-8,883
Agg P&L 0.4% (d=5%)	-2,702	1,617	-2,219	-5,847	-4,160	-3,808
P&L 0.4% (d=10%)	-3,891	-246	1,287	-1,365	1,262	1,177
Agg P&L 0.4% (d=10%)	-304	2,549	4,444	1,901	3,147	3,087
P&L 0.4% (d=15%)	-685	1,284	4,143	3,180	4,883	4,339
Agg P&L 0.4% (d=15%)	2,257	3,333	6,007	5,137	5,900	5,308

Table F11: Chen and Johnson trading strategies for Black and Scholes models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{P}	CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
P&L	7,603	28,816	32,803	37,072	36,777
# Trades	3,430	11,178	13,306	14,911	15,219
P&L 0.2% (d=0%)	-7,658	-15,867	-19,045	-22,750	-24,414
Agg P&L 0.2% (d=0%)	-2,532	-4,495	-5,641	-6,685	-6,909
P&L 0.2% (d=5%)	-468	4,150	3,604	3,871	3,612
Agg P&L 0.2% (d=5%)	2,351	6,795	6,096	6,815	7,151
P&L 0.2% (d=10%)	776	7,497	6,660	5,959	5,827
Agg P&L 0.2% (d=10%)	2,934	8,752	7,810	7,268	7,345
P&L 0.2% (d=15%)	2,868	7,960	6,791	6,606	6,422
Agg P&L 0.2% (d=15%)	4,533	8,739	7,483	7,418	7,311
P&L 0.4% (d=0%)	-22,919	-60,550	-70,894	-82,572	-85,604
Agg P&L 0.4% (d=0%)	-12,667	-37,805	-44,085	-50,441	-50,595
P&L 0.4% (d=5%)	-9,026	-14,215	-15,549	-15,088	-15,701
Agg P&L 0.4% (d=5%)	-3,388	-8,924	-10,566	-9,200	-8,622
P&L 0.4% (d=10%)	-5,271	-1,324	-2,135	-3,175	-3,570
Agg P&L 0.4% (d=10%)	-955	1,186	165	-557	-534
P&L 0.4% (d=15%)	-1,949	2,797	1,935	1,371	1,124
Agg P&L 0.4% (d=15%)	1,383	4,355	3,319	2,993	2,901

Table F12: Trading strategies for Corrado and Su models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS^P_{60}	CS^P_{av}	CS^P_{avT}	CS^P_{avT4}	CS^P_{con}
P&L	7,837	29,208	33,219	37,044	37,097
# Trades	3,430	11,178	13,306	14,911	15,219
P&L 0.2% (d=0%)	-7,209	-15,317	-18,610	-22,828	-24,203
Agg P&L 0.2% (d=0%)	-2,332	-3,843	-5,186	-6,708	-6,615
P&L 0.2% (d=5%)	-42	4,787	4,174	3,968	3,999
Agg P&L 0.2% (d=5%)	2,417	7,515	6,680	6,979	7,595
P&L 0.2% (d=10%)	1,276	8,138	7,202	6,098	6,170
Agg P&L 0.2% (d=10%)	3,112	9,475	8,353	7,432	7,728
P&L 0.2% (d=15%)	3,512	8,685	7,322	6,740	6,778
Agg P&L 0.2% (d=15%)	4,943	9,539	8,024	7,594	7,720
P&L 0.4% (d=0%)	-22,255	-59,841	-70,439	-82,700	-85,503
Agg P&L 0.4% (d=0%)	-12,501	-36,893	-43,590	-50,460	-50,328
P&L 0.4% (d=5%)	-8,404	-13,462	-15,087	-15,231	-15,601
Agg P&L 0.4% (d=5%)	-3,486	-8,006	-10,074	-9,211	-8,409
P&L 0.4% (d=10%)	-4,603	-597	-1,689	-3,301	-3,549
Agg P&L 0.4% (d=10%)	-931	2,076	613	-632	-433
P&L 0.4% (d=15%)	-1,218	3,521	2,303	1,172	1,074
Agg P&L 0.4% (d=15%)	1,646	5,229	3,707	2,881	2,958

Table F13: Chen and Johnson trading strategies for Corrado and Su models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^N	BS_{vix}^N	BS_{av}^N	BS_{avT}^N	BS_{avT4}^N	BS_{con}^N
P&L	25,759	25,627	28,829	24,290	25,980	30,421
# Trades	4,790	6,159	7,097	6,151	6,777	8,016
P&L 0.2%	3,825	-721	-2,079	-2,769	-3,017	-4,754
Agg P&L 0.2%	11,911	7,957	8,150	6,769	7,471	7,381
P&L 0.2% (d=5%)	10,084	6,187	6,513	4,528	6,353	8,668
Agg P&L 0.2% (d=5%)	14,093	9,853	11,019	8,629	11,028	12,745
P&L 0.2% (d=10%)	9,911	6,822	7,280	5,690	6,782	8,465
Agg P&L 0.2% (d=10%)	12,707	9,060	10,345	8,286	9,470	10,755
P&L 0.2% (d=15%)	9,139	7,394	7,324	5,630	7,023	8,006
Agg P&L 0.2% (d=15%)	11,257	9,070	9,740	7,586	8,983	9,571
P&L 0.4%	-18,108	-27,070	-32,987	-29,828	-32,015	-39,929
Agg P&L 0.4%	-1,936	-9,714	-12,528	-10,753	-11,039	-15,659
P&L 0.4% (d=5%)	-2,108	-7,610	-9,055	-8,627	-7,465	-5,130
Agg P&L 0.4% (d=5%)	5,909	-278	-41	-425	1,884	3,023
P&L 0.4% (d=10%)	1,721	-1,469	-2,202	-2,246	-569	1,021
Agg P&L 0.4% (d=10%)	7,313	3,008	3,927	2,946	4,807	5,600
P&L 0.4% (d=15%)	2,704	1,737	452	-172	1,963	3,227
Agg P&L 0.4% (d=15%)	6,939	5,090	5,284	3,740	5,884	6,356

Table F14.1: Trading strategies for standard BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
P&L	21,829	26,992	25,975	28,890	26,762	31,164
# Trades	4,683	6,151	6,024	6,088	7,598	8,793
P&L 0.2%	334	808	-158	1,980	-4,336	-6,229
Agg P&L 0.2%	9,399	9,796	8,726	11,299	5,971	5,699
P&L 0.2% (d=5%)	6,710	8,263	8,675	9,035	4,663	7,322
Agg P&L 0.2% (d=5%)	11,084	12,017	12,785	13,237	8,653	11,305
P&L 0.2% (d=10%)	8,231	7,301	8,413	8,886	6,364	8,017
Agg P&L 0.2% (d=10%)	11,078	9,474	10,941	11,506	8,782	10,249
P&L 0.2% (d=15%)	7,685	6,796	8,181	8,144	6,704	8,125
Agg P&L 0.2% (d=15%)	9,994	8,464	10,052	10,075	8,361	9,663
P&L 0.4%	-21,160	-25,376	-26,290	-24,930	-35,433	-43,622
Agg P&L 0.4%	-3,030	-7,400	-8,523	-6,292	-14,820	-19,766
P&L 0.4% (d=5%)	-4,956	-5,489	-4,705	-4,929	-9,232	-6,375
Agg P&L 0.4% (d=5%)	3,793	2,020	3,513	3,475	-1,252	1,590
P&L 0.4% (d=10%)	474	-489	222	695	-1,239	581
Agg P&L 0.4% (d=10%)	6,168	3,856	5,277	5,935	3,596	5,043
P&L 0.4% (d=15%)	1,660	1,568	2,250	2,137	1,615	3,436
Agg P&L 0.4% (d=15%)	6,277	4,904	5,992	6,000	4,928	6,514

Table F14.2: Trading strategies for standard BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS^N_{60}	CS^N_{av}	CS^N_{avT}	CS^N_{avT4}	CS^N_{con}
P&L	21,244	31,526	27,045	29,141	28,387
# Trades	4,158	6,833	6,354	7,338	7,093
P&L 0.2%	2,293	2,298	-1,036	-3,602	-3,063
Agg P&L 0.2%	9,413	12,590	9,032	10,290	10,525
P&L 0.2% (d=5%)	6,570	10,917	7,307	7,589	6,728
Agg P&L 0.2% (d=5%)	10,676	15,307	11,750	13,226	12,558
P&L 0.2% (d=10%)	7,536	11,081	8,576	8,466	7,619
Agg P&L 0.2% (d=10%)	10,482	13,716	11,382	11,878	11,137
P&L 0.2% (d=15%)	7,146	10,018	7,621	8,557	7,716
Agg P&L 0.2% (d=15%)	9,393	12,053	9,682	10,966	10,200
P&L 0.4%	-16,659	-26,929	-29,116	-36,345	-34,513
Agg P&L 0.4%	-2,418	-6,345	-8,980	-8,561	-7,337
P&L 0.4% (d=5%)	-5,672	-3,648	-6,394	-7,107	-8,106
Agg P&L 0.4% (d=5%)	2,540	5,132	2,493	4,168	3,554
P&L 0.4% (d=10%)	-1,108	2,197	156	-224	-929
Agg P&L 0.4% (d=10%)	4,783	7,468	5,768	6,601	6,106
P&L 0.4% (d=15%)	701	3,760	1,498	2,521	1,769
Agg P&L 0.4% (d=15%)	5,196	7,830	5,621	7,340	6,737

Table F15.1: Trading strategies for standard CS-based ANN models
P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{N*}	CS_{av}^{N*}	CS_{avT}^{N*}	CS_{avT4}^{N*}	CS_{con}^{N*}
P&L	27,632	29,837	30,074	30,433	28,721
# Trades	5,224	7,451	8,415	8,502	8,150
P&L 0.2%	3,771	-2,324	-4,684	-5,330	-5,180
Agg P&L 0.2%	13,371	9,248	7,181	8,518	7,892
P&L 0.2% (d=5%)	10,194	7,904	6,118	7,788	6,052
Agg P&L 0.2% (d=5%)	14,997	12,374	10,643	12,666	11,031
P&L 0.2% (d=10%)	9,963	8,225	8,127	8,687	7,092
Agg P&L 0.2% (d=10%)	13,079	11,170	10,595	11,490	10,057
P&L 0.2% (d=15%)	9,563	8,007	8,104	8,715	6,607
Agg P&L 0.2% (d=15%)	12,101	10,199	9,756	10,924	8,855
P&L 0.4%	-20,091	-34,486	-39,441	-41,093	-39,080
Agg P&L 0.4%	-890	-11,341	-15,712	-13,396	-12,937
P&L 0.4% (d=5%)	-2,409	-6,442	-10,254	-6,965	-8,346
Agg P&L 0.4% (d=5%)	7,196	2,498	-1,205	2,792	1,613
P&L 0.4% (d=10%)	1,569	-630	-680	471	-1,056
Agg P&L 0.4% (d=10%)	7,801	5,260	4,255	6,076	4,874
P&L 0.4% (d=15%)	3,265	1,571	2,471	2,811	846
Agg P&L 0.4% (d=15%)	8,341	5,956	5,776	7,229	5,343

Table F15.2: Trading strategies for standard CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}
P&L	29,466	29,731	33,402	33,314	34,564	36,881
# Trades	6,193	8,058	8,722	9,169	11,714	12,953
P&L 0.2%	2,663	-4,411	-2,528	-2,397	-11,896	-14,050
Agg P&L 0.2%	11,447	6,292	8,487	8,511	-187	-439
P&L 0.2% (d=5%)	9,499	4,506	7,511	7,004	6,067	8,222
Agg P&L 0.2% (d=5%)	12,753	8,060	11,089	10,314	8,509	10,810
P&L 0.2% (d=10%)	8,735	4,974	7,945	7,654	6,890	9,230
Agg P&L 0.2% (d=10%)	10,894	7,261	10,268	9,807	8,060	10,342
P&L 0.2% (d=15%)	7,976	4,597	7,752	7,726	6,946	8,929
Agg P&L 0.2% (d=15%)	9,661	6,363	9,478	9,285	7,701	9,595
P&L 0.4%	-24,139	-38,553	-38,459	-38,108	-58,356	-64,981
Agg P&L 0.4%	-6,572	-17,148	-16,427	-16,291	-34,939	-37,759
P&L 0.4% (d=5%)	-1,868	-8,959	-7,101	-8,253	-8,090	-5,910
Agg P&L 0.4% (d=5%)	4,641	-1,851	54	-1,633	-3,205	-734
P&L 0.4% (d=10%)	1,912	-2,924	-535	-842	625	3,171
Agg P&L 0.4% (d=10%)	6,231	1,649	4,109	3,463	2,965	5,394
P&L 0.4% (d=15%)	2,916	-978	2,079	2,323	3,437	5,624
Agg P&L 0.4% (d=15%)	6,288	2,554	5,531	5,442	4,946	6,956

Table F16.1: Trading strategies for hybrid BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
P&L	27,024	29,529	32,908	33,514	35,774	37,281
# Trades	5,675	8,246	8,907	9,457	11,995	12,650
P&L 0.2%	1,694	-4,193	-2,435	-4,134	-11,484	-12,939
Agg P&L 0.2%	10,552	6,053	7,871	7,086	837	1,066
P&L 0.2% (d=5%)	8,620	5,446	7,439	7,101	6,488	7,054
Agg P&L 0.2% (d=5%)	11,948	8,816	10,872	10,344	9,020	9,897
P&L 0.2% (d=10%)	7,587	5,034	7,532	8,837	7,572	8,764
Agg P&L 0.2% (d=10%)	9,718	7,254	9,841	10,740	8,764	10,107
P&L 0.2% (d=15%)	6,593	5,147	8,162	8,579	7,910	8,427
Agg P&L 0.2% (d=15%)	8,247	6,977	9,890	9,957	8,689	9,237
P&L 0.4%	-23,637	-37,914	-37,778	-41,782	-58,741	-63,158
Agg P&L 0.4%	-5,920	-17,424	-17,166	-19,343	-34,100	-35,148
P&L 0.4% (d=5%)	-2,339	-8,601	-7,626	-8,667	-8,087	-7,131
Agg P&L 0.4% (d=5%)	4,319	-1,861	-760	-2,180	-3,023	-1,443
P&L 0.4% (d=10%)	1,079	-3,096	-1,479	476	1,225	2,297
Agg P&L 0.4% (d=10%)	5,340	1,343	3,139	4,280	3,609	4,983
P&L 0.4% (d=15%)	1,804	-277	2,232	3,156	4,364	4,812
Agg P&L 0.4% (d=15%)	5,112	3,382	5,687	5,911	5,922	6,432

Table F16.2: Trading strategies for hybrid BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{Nh}	CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}
P&L	24,205	28,938	36,057	35,097	36,718
# Trades	5,117	9,529	11,513	13,042	14,082
P&L 0.2%	2,068	-9,244	-10,716	-18,188	-21,211
Agg P&L 0.2%	9,791	2,572	3,118	-1,952	-3,139
P&L 0.2% (d=5%)	8,919	4,457	5,975	2,658	3,883
Agg P&L 0.2% (d=5%)	12,359	8,168	9,550	6,440	7,731
P&L 0.2% (d=10%)	8,518	6,136	6,937	5,173	6,111
Agg P&L 0.2% (d=10%)	10,835	8,300	8,775	7,154	8,011
P&L 0.2% (d=15%)	7,515	6,267	7,613	5,888	6,359
Agg P&L 0.2% (d=15%)	9,272	7,825	8,839	7,185	7,519
P&L 0.4%	-20,068	-47,426	-57,489	-71,472	-79,140
Agg P&L 0.4%	-4,622	-23,793	-29,822	-39,001	-42,996
P&L 0.4% (d=5%)	-2,649	-12,290	-12,275	-16,135	-14,934
Agg P&L 0.4% (d=5%)	4,230	-4,868	-5,125	-8,570	-7,239
P&L 0.4% (d=10%)	1,139	-2,722	-1,734	-4,201	-3,184
Agg P&L 0.4% (d=10%)	5,774	1,606	1,943	-239	615
P&L 0.4% (d=15%)	2,148	801	2,492	242	936
Agg P&L 0.4% (d=15%)	5,662	3,918	4,943	2,836	3,256

Table F17.1: Trading strategies for hybrid CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{Nh*}	CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
P&L	26,691	32,915	31,943	34,907	37,975
# Trades	5,140	10,043	10,377	12,537	12,947
P&L 0.2%	3,590	-8,721	-12,019	-17,527	-16,084
Agg P&L 0.2%	11,032	3,734	898	-1,586	735
P&L 0.2% (d=5%)	9,328	5,059	3,188	3,854	5,017
Agg P&L 0.2% (d=5%)	12,243	8,188	6,783	7,447	8,732
P&L 0.2% (d=10%)	8,398	6,611	4,664	5,718	7,480
Agg P&L 0.2% (d=10%)	10,353	8,644	6,734	7,752	9,236
P&L 0.2% (d=15%)	7,337	6,653	5,601	6,052	7,826
Agg P&L 0.2% (d=15%)	8,861	8,231	7,114	7,439	8,960
P&L 0.4%	-19,511	-50,356	-55,980	-69,962	-70,143
Agg P&L 0.4%	-4,626	-25,446	-30,146	-38,078	-36,505
P&L 0.4% (d=5%)	-1,215	-11,691	-14,095	-14,040	-12,799
Agg P&L 0.4% (d=5%)	4,615	-5,433	-6,906	-6,852	-5,369
P&L 0.4% (d=10%)	1,570	-2,378	-3,918	-3,252	-1,363
Agg P&L 0.4% (d=10%)	5,479	1,688	220	816	2,149
P&L 0.4% (d=15%)	2,433	724	457	612	2,605
Agg P&L 0.4% (d=15%)	5,481	3,879	3,484	3,387	4,873

Table F17.2: Trading strategies for hybrid CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

1.B. Appendix with reduced dataset results

Set	*Starting*	*Ending*	*# obs*	*Set*	*Starting*	*Ending*	*# obs*
Tr1	5-Jan-98	8-Mar-99	18,053	**Tr6**	9-Mar-99	20-Jan-00	17,352
Vd1	9-Mar-99	12-Jul-99	9,146	**Vd6**	21-Jan-00	17-May-00	8,003
Ts1	13-Jul-99	24-Sep-99	3,174	**Ts6**	18-May-00	17-Jul-00	3,543
Tr2	24-Apr-98	16-Apr-99	17,313	**Tr7**	20-Apr-99	28-Feb-00	16,947
Vd2	19-Apr-99	23-Sep-99	9,638	**Vd7**	29-Feb-00	17-Jul-00	8,781
Ts2	24-Sep-99	5-Jan-00	3,223	**Ts7**	18-Jul-00	10-Oct-00	3,982
Tr3	23-Jun-98	3-Jun-99	16,945	**Tr8**	7-Jun-99	11-Apr-00	16,247
Vd3	4-Jun-99	5-Jan-00	9,474	**Vd8**	12-Apr-00	6-Oct-00	9,644
Ts3	6-Jan-00	10-Feb-00	2,835	**Ts8**	9-Oct-00	24-Jan-01	3,418
Tr4	3-Sep-98	24-Aug-99	16,782	**Tr9**	26-Aug-99	5-Jun-00	15,851
Vd4	25-Aug-99	11-Feb-00	8,600	**Vd9**	6-Jun-00	24-Jan-01	9,615
Ts4	14-Feb-00	27-Mar-00	2,689	**Ts9**	25-Jan-01	29-Mar-01	3,486
Tr5	29-Jan-99	18-Oct-99	17,015	**Tr10**	21-Oct-99	11-Aug-00	16,165
Vd5	19-Oct-99	24-Mar-00	7,728	**Vd10**	14-Aug-00	30-Mar-01	9,349
Ts5	28-Mar-00	16-May-00	3,370	**Ts10**	2-Apr-01	24-Aug-01	3,594

Table R1: Training (Tr), validation (Vd) and testing (Ts) dates

	BS_{60}^{P}	BS_{vix}^{P}	BS_{av}^{P}	BS_{avT}^{P}	BS_{avT4}^{P}	BS_{con}^{P}	CS_{60}^{P}		CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
RMSE	9.83	11.82	8.41	8.25	7.08	7.06	9.74		7.56	7.55	7.55	7.52
MAE	6.35	8.43	4.82	4.54	2.65	2.65	6.32		3.38	3.12	2.99	3.04
RMeSE	4.50	6.57	3.63	3.27	1.48	1.46	4.59		2.17	1.83	1.69	1.71

	BS_{60}^{N}	BS_{vix}^{N}	BS_{av}^{N}	BS_{avT}^{N}	BS_{avT4}^{N}	BS_{con}^{N}	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
RMSE	8.05	6.56	7.34	6.94	6.64	6.69	7.14	6.60	6.82	6.91	6.25	6.12
MAE	5.07	3.34	4.02	3.72	3.42	3.37	4.11	3.43	3.46	3.59	3.01	3.00
RMeSE	3.80	2.32	2.99	2.56	2.33	2.24	3.09	2.41	2.44	2.56	1.99	2.02

	CS_{60}^{N}		CS_{av}^{N}	CS_{avT}^{N}	CS_{avT4}^{N}	CS_{con}^{N}	CS_{60}^{N*}		CS_{av}^{N*}	CS_{avT}^{N*}	CS_{avT4}^{N*}	CS_{con}^{N*}
RMSE	9.05		7.18	6.93	6.94	6.88	8.35		6.97	6.59	6.50	6.77
MAE	5.74		3.95	3.61	3.73	3.62	4.94		3.68	3.26	3.23	3.45
RMeSE	4.25		2.74	2.41	2.60	2.55	3.43		2.62	2.22	2.25	2.36

	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
RMSE	8.45	6.70	7.29	7.01	6.58	6.78	7.35	6.40	7.05	6.83	5.94	5.64
MAE	5.11	3.58	3.62	3.38	2.62	2.69	4.27	3.21	3.32	3.30	2.45	2.44
RMeSE	3.44	2.59	2.55	2.35	1.55	1.65	3.13	2.26	2.30	2.33	1.51	1.54

	CS_{60}^{Nh}		CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}	CS_{60}^{Nh*}		CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
RMSE	7.80		7.29	6.83	7.31	7.35	7.69		6.90	6.80	6.51	6.46
MAE	4.65		3.20	3.08	3.03	3.03	4.58		3.13	2.92	2.83	2.87
RMeSE	3.41		2.13	2.02	1.82	1.80	3.23		2.03	1.80	1.79	1.81

Table R2: Pricing error measures for all models in the aggregate testing period (AggTs)
RMSE is the Root Mean Square Error, MAE the Mean Absolute Error and RMeSE the Root Median Square Error. The superscripts refer to the kind of the model: *P* refers to parametric models, *N* to the simple neural networks and *Nh* to the hybrid neural networks. The asterisk (*) refers to neural network models that use the transformed variables. The subscripts refer to kind of historical/implied parameters used to each model per se.

	BS_{60}^{P}	BS_{vix}^{P}	BS_{av}^{P}	BS_{avT}^{P}	BS_{avT4}^{P}	BS_{con}^{P}	CS_{60}^{P}		CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
MHE	0.28	0.32	0.29	0.29	0.29	0.30	0.29		0.28	0.28	0.27	0.26
MAHE	2.68	2.82	2.68	2.66	2.57	2.57	2.68		2.86	2.83	2.87	2.88
MPE	5.80	5.86	5.80	5.80	5.74	5.75	5.83		5.97	5.97	6.03	6.04
MD	0.581	0.585	0.584	0.582	0.561	0.558	0.580		0.609	0.606	0.605	0.602

	BS_{60}^{N}	BS_{vix}^{N}	BS_{av}^{N}	BS_{avT}^{N}	BS_{avT4}^{N}	BS_{con}^{N}	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
MHE	0.33	0.30	0.28	0.29	0.28	0.30	0.29	0.29	0.28	0.28	0.29	0.29
MAHE	3.00	2.92	2.93	2.91	2.71	2.77	2.91	2.89	2.88	2.86	2.68	2.71
MPE	6.08	6.04	6.08	6.04	5.88	5.91	6.03	6.02	6.02	6.00	5.84	5.86
MD	0.618	0.614	0.615	0.610	0.578	0.581	0.606	0.611	0.610	0.608	0.579	0.582

	CS_{60}^{N}		CS_{av}^{N}	CS_{avT}^{N}	CS_{avT4}^{N}	CS_{con}^{N}	CS_{60}^{N*}		CS_{av}^{N*}	CS_{avT}^{N*}	CS_{avT4}^{N*}	CS_{con}^{N*}
MHE	0.30		0.29	0.29	0.27	0.27	0.31		0.32	0.30	0.30	0.29
MAHE	3.07		2.90	2.92	2.84	2.86	2.97		2.97	2.81	2.78	2.81
MPE	6.14		6.04	6.05	6.00	5.99	6.06		6.09	5.93	5.91	5.94
MD	0.615		0.611	0.613	0.602	0.605	0.609		0.607	0.593	0.592	0.596

	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
MHE	0.30	0.29	0.28	0.30	0.29	0.29	0.31	0.29	0.28	0.29	0.30	0.30
MAHE	2.94	2.88	2.88	2.91	2.63	2.61	2.93	2.90	2.89	2.90	2.63	2.63
MPE	6.03	6.00	5.99	5.99	5.79	5.78	6.02	6.01	6.00	6.00	5.79	5.80
MD	0.616	0.610	0.614	0.614	0.574	0.570	0.615	0.613	0.615	0.615	0.574	0.574

	CS_{60}^{Nh}		CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}	CS_{60}^{Nh*}		CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
MHE	0.30		0.28	0.28	0.27	0.26	0.29		0.29	0.29	0.26	0.26
MAHE	2.85		2.90	2.85	2.87	2.88	2.83		2.95	2.91	2.95	2.96
MPE	5.98		6.00	5.98	6.03	6.04	5.95		6.05	6.03	6.11	6.12
MD	0.605		0.613	0.609	0.605	0.602	0.606		0.621	0.616	0.617	0.615

Table R3: Hedging error measures for all models in the aggregate testing period (AggTs)
MHE is the Mean Hedging Error, MAHE the Mean Absolute Hedging Error, MPE the Mean Prediction Error and MD the Mean Delta of each model. The superscripts refer to the kind of the model: *P* refers to parametric models, *N* to the simple neural networks and *Nh* to the hybrid neural networks. The asterisk (*) refers to neural network models that use the transformed variables. The subscripts refer to kind of historical/implied parameters used to each model per se.

	BS_{60}^{P}	BS_{vix}^{P}	BS_{con}^{P}	CS_{60}^{P}	CS_{con}^{P}	BS_{60}^{N*}	CS_{60}^{N*}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{con}^{Nh*}	CS_{60}^{Nh*}	CS_{con}^{Nh*}
BS_{60}^{P}		-33.91	67.72	0.56	59.35	42.88	25.46	39.53	61.45	78.64	33.14	66.94
BS_{vix}^{P}	9.23		99.72	34.63	91.26	77.59	59.53	74.19	95.49	112.26	67.82	100.24
BS_{con}^{P}	-9.26	-18.95		-67.60	-7.35	-30.38	-44.47	-33.27	-11.78	4.67	-39.12	-4.59
CS_{60}^{P}	-0.36	-9.58	8.87		59.17	42.59	25.05	39.21	61.31	78.66	32.78	66.84
CS_{con}^{P}	-7.79	-17.20	1.30	-7.42		-21.78	-36.11	-24.68	-3.53	12.68	-30.51	3.31
BS_{60}^{N*}	-9.60	-20.05	0.24	-9.19	-1.14		-16.95	-3.40	20.55	39.41	-10.06	27.55
CS_{60}^{N*}	-5.54	-15.46	4.05	-5.15	2.63	4.07		13.61	36.30	54.07	7.16	42.53
BS_{60}^{Nh*}	-9.06	-19.60	0.87	-8.65	-0.54	0.67	-3.46		23.78	42.50	-6.64	30.64
BS_{vix}^{Nh*}	-12.00	-22.94	-1.89	-11.56	-3.26	-2.29	-6.41	-3.00		18.58	-30.25	7.61
BS_{con}^{Nh*}	-14.73	-26.67	-4.03	-14.26	-5.43	-4.65	-8.94	-5.44	-2.29		-48.82	-10.24
CS_{60}^{Nh*}	-7.95	-18.38	1.95	-7.53	0.53	1.83	-2.31	1.18	4.19	6.69		36.87
CS_{con}^{Nh*}	-11.98	-23.07	-1.76	-11.54	-3.15	-2.16	-6.33	-2.88	0.16	2.50	-4.08	

Table R4a: Matched pair student-t tests for square and absolute differences

Matched pair t-tests concerning the absolute differences are reported in the upper diagonal whilst on the lower diagonal the matched pair t-tests concerning the square differences are tabulated. Both tests compare the MAE and MSE between models in the vertical heading versus models in the horizontal heading. In general, a positive t-value larger than 1.645 (2.325) means that the model in the vertical heading has a larger MAE or MSE than the model in the horizontal heading at 5% (1%) significance level.

	BS^{P}_{60}	BS^{P}_{vix}	BS^{P}_{con}	CS^{P}_{60}	CS^{P}_{con}	BS^{N*}_{60}	CS^{N*}_{60}	BS^{Nh*}_{60}	BS^{Nh*}_{vix}	BS^{Nh*}_{con}	CS^{Nh*}_{60}	CS^{Nh*}_{con}
BS^{P}_{60}		-27.29	15.19	5.00	13.66	56.05	25.49	52.42	59.33	38.80	49.71	35.26
BS^{P}_{vix}	9.23		28.88	27.42	27.37	60.81	44.58	62.12	78.15	60.32	59.53	53.50
BS^{P}_{con}	-9.26	-18.95		-14.53	-5.43	-0.40	-6.42	-1.42	3.08	7.61	-3.19	4.02
CS^{P}_{60}	-0.36	-9.58	8.87		12.99	54.57	23.86	50.11	57.84	37.25	45.03	33.67
CS^{P}_{con}	-7.79	-17.20	1.30	-7.42		1.96	-4.40	0.94	5.62	10.18	-0.92	8.32
BS^{N*}_{60}	-9.60	-20.05	0.24	-9.19	-1.14		-27.71	-7.51	23.55	13.25	-18.57	6.63
CS^{N*}_{60}	-5.54	-15.46	4.05	-5.15	2.63	4.07		19.90	36.18	22.35	12.84	16.92
BS^{Nh*}_{60}	-9.06	-19.60	0.87	-8.65	-0.54	0.67	-3.46		30.72	16.80	-13.27	9.53
BS^{Nh*}_{vix}	-12.00	-22.94	-1.89	-11.56	-3.26	-2.29	-6.41	-3.00		7.04	-37.74	-0.52
BS^{Nh*}_{con}	-14.73	-26.67	-4.03	-14.26	-5.43	-4.65	-8.94	-5.44	-2.29		-19.94	-10.44
CS^{Nh*}_{60}	-7.95	-18.38	1.95	-7.53	0.53	1.83	-2.31	1.18	4.19	6.69		13.08
CS^{Nh*}_{con}	-11.98	-23.07	-1.76	-11.54	-3.15	-2.16	-6.33	-2.88	0.16	2.50	-4.08	

Table R4b: Matched pair student-t and Johnson t-tests for the square differences

Matched pair student-t (lower diagonal) and Johnson modified t (upper diagonal) tests concerning square differences are tabulated. Both tests compare the MSE between models in the vertical heading versus models in the horizontal heading. In general, a positive t-value larger than 1.645 (2.325) means that the model in the vertical heading has a larger MSE than the model in the horizontal heading at 5% (1%) significance level.

	BS_{60}^{N}	BS_{vix}^{N}	BS_{av}^{N}	BS_{avT}^{N}	BS_{avT4}^{N}	BS_{con}^{N}	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
Min	5	6	3	3	5	5	3	4	3	3	4	3
Mean	5.7	7.8	6.7	7.3	6.8	7.5	6.2	6.7	5.6	6.9	6.4	6.1
Max	8	10	10	10	10	10	10	9	7	9	10	10

	CS_{60}^{N}		CS_{av}^{N}	CS_{avT}^{N}	CS_{avT4}^{N}	CS_{con}^{N}	CS_{60}^{N*}		CS_{av}^{N*}	CS_{avT}^{N*}	CS_{avT4}^{N*}	CS_{con}^{N*}
Min	3		4	3	4	4	3		5	4	3	3
Mean	5.9		7.3	7	7.4	6.9	4.8		7.1	6.8	5.4	6.7
Max	9		10	10	10	10	10		9	10	9	10

	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
Min	3	4	4	2	3	2	3	4	3	2	2	2
Mean	5.6	6.9	5.7	6	3.7	5.7	4.9	6.7	5	5.2	3.8	3.9
Max	10	10	8	10	6	10	9	10	6	9	6	6

	CS_{60}^{Nh}		CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}	CS_{60}^{Nh*}		CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
Min	2		2	2	2	2	2		2	2	2	2
Mean	5		4.1	4.4	4.2	4.4	5.2		4.6	4.7	3.9	4
Max	10		7	8	7	7	9		10	9	7	10

Table R5: ANNs Complexity

Minimum, mean and maximum number of the hidden layer neurons for the ten different training periods.

	BS^{P}_{con}			CS^{P}_{con}		
	Short	**Medium**	**Long**	**Short**	**Medium**	**Long**
VDOTM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
DOTM	2.27	4.50	n.a.	3.02	5.24	n.a.
OTM	5.78	8.37	n.a.	6.29	9.68	n.a.
JOTM	7.81	6.68	n.a.	8.13	7.64	n.a.
ATM	6.67	9.46	n.a.	7.30	10.14	n.a.
JITM	6.71	9.41	n.a.	7.29	9.21	n.a.
ITM	7.70	7.13	n.a.	8.24	7.59	n.a.
DITM	7.07	7.93	n.a.	7.20	8.50	n.a.
VDITM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
	BS^{Nh*}_{con}			CS^{Nh*}_{con}		
VDOTM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.
DOTM	2.36	4.07	n.a.	2.54	5.22	n.a.
OTM	5.08	7.25	n.a.	5.69	8.74	n.a.
JOTM	5.82	5.59	n.a.	6.76	7.09	n.a.
ATM	4.65	8.37	n.a.	5.68	9.53	n.a.
JITM	5.50	7.68	n.a.	6.20	8.16	n.a.
ITM	5.98	5.84	n.a.	6.73	6.75	n.a.
DITM	5.45	6.59	n.a.	5.95	7.67	n.a.
VDITM	n.a.	n.a.	n.a.	n.a.	n.a.	n.a.

Table R6: Root Mean Square Errors for the four best models for the full dataset

	BS_{60}^{P}	BS_{vix}^{P}	BS_{av}^{P}	BS_{avT}^{P}	BS_{avT4}^{P}	BS_{con}^{P}
P&L	6,617	11,502	12,852	11,398	27,418	29,784
# Trades	3,128	3,481	4,584	5,130	11,944	13,255
P&L 0.2% (d=0%)	-6,826	-6,744	-5,483	-7,698	-15,038	-17,010
Agg P&L 0.2% (d=0%)	-2,195	-733	320	-1,957	-5,267	-5,467
P&L 0.2% (d=5%)	-1,481	1,774	2,672	-66	4,045	4,276
Agg P&L 0.2% (d=5%)	1,143	3,743	5,460	2,721	6,009	6,506
P&L 0.2% (d=10%)	245	2,081	5,896	4,376	6,102	6,229
Agg P&L 0.2% (d=10%)	2,153	3,116	7,182	5,727	6,834	7,049
P&L 0.2% (d=15%)	2,552	2,863	6,400	5,741	7,023	6,675
Agg P&L 0.2% (d=15%)	4,127	3,642	7,144	6,484	7,396	7,075
P&L 0.4% (d=0%)	-20,270	-24,989	-23,819	-26,793	-57,494	-63,805
Agg P&L 0.4% (d=0%)	-11,006	-12,967	-12,211	-15,311	-37,953	-40,717
P&L 0.4% (d=5%)	-9,559	-6,229	-10,067	-13,797	-10,120	-9,779
Agg P&L 0.4% (d=5%)	-4,310	-2,291	-4,490	-8,222	-6,191	-5,320
P&L 0.4% (d=10%)	-5,534	-3,254	-807	-3,352	184	372
Agg P&L 0.4% (d=10%)	-1,717	-1,185	1,764	-651	1,649	2,012
P&L 0.4% (d=15%)	-2,012	-1,535	2,275	1,332	3,998	3,696
Agg P&L 0.4% (d=15%)	1,137	24	3,763	2,819	4,743	4,496

Table R7: Trading strategies for Black and Scholes models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS^P_{60}	BS^P_{vix}	BS^P_{av}	BS^P_{avT}	BS^P_{avT4}	BS^P_{con}
P&L	7,110	12,486	13,068	11,800	27,591	29,886
# Trades	3,128	3,481	4,584	5,130	11,944	13,255
P&L 0.2% (d=0%)	-6,132	-5,394	-5,017	-7,097	-14,951	-16,884
Agg P&L 0.2% (d=0%)	-1,671	989	885	-1,331	-5,237	-5,387
P&L 0.2% (d=5%)	-798	3,156	3,179	538	4,308	4,596
Agg P&L 0.2% (d=5%)	1,528	5,413	6,066	3,315	6,253	6,778
P&L 0.2% (d=10%)	1,010	3,529	6,556	4,992	6,409	6,509
Agg P&L 0.2% (d=10%)	2,644	4,738	7,951	6,473	7,121	7,289
P&L 0.2% (d=15%)	3,330	4,400	7,165	6,525	7,379	6,982
Agg P&L 0.2% (d=15%)	4,689	5,312	7,997	7,403	7,743	7,367
P&L 0.4% (d=0%)	-19,373	-23,274	-23,102	-25,995	-57,494	-63,655
Agg P&L 0.4% (d=0%)	-10,453	-10,509	-11,297	-14,462	-38,066	-40,661
P&L 0.4% (d=5%)	-8,680	-4,491	-9,329	-12,998	-9,989	-9,528
Agg P&L 0.4% (d=5%)	-4,028	23	-3,555	-7,444	-6,099	-5,164
P&L 0.4% (d=10%)	-4,621	-1,440	110	-2,536	375	527
Agg P&L 0.4% (d=10%)	-1,353	978	2,900	426	1,800	2,086
P&L 0.4% (d=15%)	-1,163	357	3,275	2,315	4,188	3,803
Agg P&L 0.4% (d=15%)	1,555	2,181	4,939	4,072	4,916	4,572

Table R8: Chen and Johnson modified trading strategies for Black and Scholes
P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{P}	CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
P&L	6,575	26,969	29,811	31,291	30,886
# Trades	3,160	10,136	11,967	12,796	12,893
P&L 0.2% (d=0%)	-7,494	-13,607	-16,185	-18,062	-18,632
Agg P&L 0.2% (d=0%)	-2,708	-3,537	-4,955	-5,645	-5,178
P&L 0.2% (d=5%)	-1,879	2,811	2,038	2,061	2,033
Agg P&L 0.2% (d=5%)	697	5,077	4,066	4,567	5,160
P&L 0.2% (d=10%)	-369	6,426	5,503	4,754	4,622
Agg P&L 0.2% (d=10%)	1,625	7,454	6,378	5,893	5,968
P&L 0.2% (d=15%)	2,020	7,223	6,180	5,708	5,537
Agg P&L 0.2% (d=15%)	3,609	7,845	6,693	6,400	6,329
P&L 0.4% (d=0%)	-21,563	-54,183	-62,182	-67,415	-68,150
Agg P&L 0.4% (d=0%)	-11,990	-34,042	-39,721	-42,580	-41,241
P&L 0.4% (d=5%)	-10,032	-14,585	-16,016	-15,830	-16,196
Agg P&L 0.4% (d=5%)	-4,882	-10,054	-11,960	-10,817	-9,941
P&L 0.4% (d=10%)	-6,108	-1,969	-2,776	-3,921	-4,292
Agg P&L 0.4% (d=10%)	-2,118	88	-1,027	-1,644	-1,600
P&L 0.4% (d=15%)	-2,578	2,328	1,626	741	483
Agg P&L 0.4% (d=15%)	600	3,573	2,653	2,126	2,068

Table R9: Trading strategies for Corrado and Su models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{P}	CS_{av}^{P}	CS_{avT}^{P}	CS_{avT4}^{P}	CS_{con}^{P}
P&L	6,920	27,251	30,031	31,279	31,025
# Trades	3,160	10,136	11,967	12,796	12,893
P&L 0.2% (d=0%)	-6,914	-13,122	-15,867	-18,090	-18,570
Agg P&L 0.2% (d=0%)	-2,369	-2,965	-4,662	-5,633	-5,060
P&L 0.2% (d=5%)	-1,315	3,385	2,497	2,230	2,285
Agg P&L 0.2% (d=5%)	909	5,701	4,492	4,755	5,435
P&L 0.2% (d=10%)	222	6,987	5,928	4,950	4,844
Agg P&L 0.2% (d=10%)	1,902	8,056	6,766	6,075	6,204
P&L 0.2% (d=15%)	2,741	7,877	6,602	5,867	5,759
Agg P&L 0.2% (d=15%)	4,076	8,528	7,087	6,576	6,582
P&L 0.4% (d=0%)	-20,747	-53,496	-61,765	-67,459	-68,165
Agg P&L 0.4% (d=0%)	-11,658	-33,181	-39,355	-42,545	-41,145
P&L 0.4% (d=5%)	-9,252	-13,850	-15,597	-15,855	-16,192
Agg P&L 0.4% (d=5%)	-4,803	-9,219	-11,606	-10,803	-9,892
P&L 0.4% (d=10%)	-5,332	-1,277	-2,386	-3,945	-4,355
Agg P&L 0.4% (d=10%)	-1,973	859	-710	-1,694	-1,637
P&L 0.4% (d=15%)	-1,754	3,025	1,946	610	344
Agg P&L 0.4% (d=15%)	916	4,328	2,916	2,029	1,989

Table R10: Chen and Johnson modified trading for Corrado and Su models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{N}	BS_{vix}^{N}	BS_{av}^{N}	BS_{avT}^{N}	BS_{avT4}^{N}	BS_{con}^{N}
P&L	22,992	29,737	25,424	27,340	26,012	26,687
# Trades	4,836	7,615	6,168	7,069	7,770	8,128
P&L 0.2%	1,604	-2,031	-440	-1,779	-5,467	-5,414
Agg P&L 0.2%	8,560	6,343	7,812	6,824	4,266	4,686
P&L 0.2% (d=5%)	8,375	7,155	6,829	7,142	5,866	5,912
Agg P&L 0.2% (d=5%)	11,289	9,721	10,303	10,385	8,963	9,379
P&L 0.2% (d=10%)	8,781	6,388	6,030	7,532	6,447	7,221
Agg P&L 0.2% (d=10%)	10,744	7,774	8,468	9,550	8,278	9,121
P&L 0.2% (d=15%)	7,985	6,870	7,454	7,482	7,166	7,253
Agg P&L 0.2% (d=15%)	9,502	7,855	9,347	8,980	8,372	8,553
P&L 0.4%	-19,784	-33,798	-26,305	-30,899	-36,946	-37,516
Agg P&L 0.4%	-5,872	-17,052	-9,800	-13,692	-17,480	-17,315
P&L 0.4% (d=5%)	-2,369	-6,249	-5,494	-6,151	-6,619	-7,032
Agg P&L 0.4% (d=5%)	3,459	-1,116	1,454	335	-425	-96
P&L 0.4% (d=10%)	1,817	-319	-1,987	-154	175	457
Agg P&L 0.4% (d=10%)	5,744	2,452	2,889	3,883	3,836	4,257
P&L 0.4% (d=15%)	2,764	2,439	1,551	2,318	3,150	3,098
Agg P&L 0.4% (d=15%)	5,797	4,408	5,337	5,314	5,563	5,700

Table R11.1: Trading strategies for standard BS-based ANN models
P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
P&L	25,747	28,401	27,359	26,910	27,547	28,591
# Trades	5,398	7,058	7,096	7,077	8,502	8,663
P&L 0.2%	576	-1,847	-2,388	-2,923	-5,325	-5,531
Agg P&L 0.2%	8,570	6,338	6,078	6,061	3,879	4,767
P&L 0.2% (d=5%)	8,771	7,142	6,300	6,013	6,252	6,660
Agg P&L 0.2% (d=5%)	11,503	9,740	9,238	8,768	8,854	9,834
P&L 0.2% (d=10%)	7,945	7,205	5,982	5,751	7,222	6,581
Agg P&L 0.2% (d=10%)	9,609	8,747	7,909	7,459	8,615	8,082
P&L 0.2% (d=15%)	7,088	7,634	6,343	6,450	7,292	7,172
Agg P&L 0.2% (d=15%)	8,358	8,721	7,778	7,702	8,218	8,099
P&L 0.4%	-24,594	-32,096	-32,135	-32,757	-38,196	-39,653
Agg P&L 0.4%	-8,607	-15,725	-15,204	-14,788	-19,788	-19,057
P&L 0.4% (d=5%)	-1,515	-5,588	-6,407	-7,159	-6,587	-6,740
Agg P&L 0.4% (d=5%)	3,950	-394	-530	-1,649	-1,381	-393
P&L 0.4% (d=10%)	2,094	549	-1,178	-1,592	1,227	550
Agg P&L 0.4% (d=10%)	5,423	3,634	2,677	1,822	4,014	3,552
P&L 0.4% (d=15%)	2,765	3,383	1,432	1,578	3,732	3,571
Agg P&L 0.4% (d=15%)	5,306	5,557	4,302	4,081	5,582	5,424

Table R11.2: Trading strategies for standard BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS^N_{60}	CS^N_{av}	CS^N_{avT}	CS^N_{avT4}	CS^N_{con}
P&L	20,675	26,166	27,584	27,124	26,776
# Trades	4,513	6,514	7,362	6,979	6,964
P&L 0.2%	673	-560	-2,697	-2,459	-2,680
Agg P&L 0.2%	8,516	8,386	6,766	8,191	8,130
P&L 0.2% (d=5%)	6,106	7,415	6,517	7,030	6,978
Agg P&L 0.2% (d=5%)	9,972	10,848	10,186	10,953	10,904
P&L 0.2% (d=10%)	6,916	7,353	6,747	7,299	7,783
Agg P&L 0.2% (d=10%)	9,461	9,575	8,808	9,812	10,259
P&L 0.2% (d=15%)	6,875	7,229	6,722	7,164	7,585
Agg P&L 0.2% (d=15%)	8,865	8,847	8,197	9,082	9,445
P&L 0.4%	-19,329	-27,287	-32,978	-32,043	-32,137
Agg P&L 0.4%	-3,643	-9,395	-14,053	-10,743	-10,517
P&L 0.4% (d=5%)	-4,802	-4,985	-7,526	-5,222	-5,349
Agg P&L 0.4% (d=5%)	2,930	1,882	-188	2,625	2,503
P&L 0.4% (d=10%)	-557	-479	-838	50	546
Agg P&L 0.4% (d=10%)	4,533	3,965	3,284	5,075	5,498
P&L 0.4% (d=15%)	1,286	1,582	1,709	2,154	2,598
Agg P&L 0.4% (d=15%)	5,266	4,819	4,660	5,988	6,318

Table R12.1: Trading strategies for standard CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS^{N*}_{60}	CS^{N*}_{av}	CS^{N*}_{avT}	CS^{N*}_{avT4}	CS^{N*}_{con}
P&L	22,739	27,299	30,079	26,912	26,703
# Trades	5,173	7,133	8,531	8,106	8,030
P&L 0.2%	-492	-1,118	-3,802	-5,609	-4,992
Agg P&L 0.2%	7,689	7,923	6,227	4,973	5,413
P&L 0.2% (d=5%)	7,138	6,880	6,460	6,036	5,479
Agg P&L 0.2% (d=5%)	10,397	10,163	9,662	9,348	9,427
P&L 0.2% (d=10%)	7,517	7,117	6,671	7,211	6,405
Agg P&L 0.2% (d=10%)	9,556	9,178	8,341	9,176	8,807
P&L 0.2% (d=15%)	6,978	7,048	7,416	7,374	6,379
Agg P&L 0.2% (d=15%)	8,433	8,659	8,566	8,774	8,212
P&L 0.4%	-23,723	-29,535	-37,683	-38,129	-36,687
Agg P&L 0.4%	-7,360	-11,453	-17,624	-16,965	-15,878
P&L 0.4% (d=5%)	-3,137	-6,142	-7,934	-7,264	-8,493
Agg P&L 0.4% (d=5%)	3,383	425	-1,530	-640	-599
P&L 0.4% (d=10%)	1,139	-428	-264	-133	-1,312
Agg P&L 0.4% (d=10%)	5,218	3,695	3,076	3,796	3,492
P&L 0.4% (d=15%)	2,490	1,700	3,024	2,729	1,373
Agg P&L 0.4% (d=15%)	5,402	4,924	5,325	5,528	5,040

Table R12.2: Trading strategies for standard CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}
P&L	21,946	28,851	25,382	28,451	29,642	29,127
# Trades	5,090	7,507	7,821	8,162	10,747	10,429
P&L 0.2%	347	-1,714	-6,099	-4,333	-10,412	-11,602
Agg P&L 0.2%	7,186	6,659	3,041	4,287	-188	-545
P&L 0.2% (d=5%)	6,747	7,576	3,233	4,837	5,216	5,376
Agg P&L 0.2% (d=5%)	9,406	10,228	6,042	7,569	7,318	7,797
P&L 0.2% (d=10%)	6,213	6,778	4,464	5,560	6,736	6,577
Agg P&L 0.2% (d=10%)	8,060	8,442	6,547	7,245	7,664	7,763
P&L 0.2% (d=15%)	6,084	6,404	5,524	6,555	7,608	6,818
Agg P&L 0.2% (d=15%)	7,557	7,674	7,129	7,774	8,191	7,543
P&L 0.4%	-21,252	-32,279	-37,579	-37,117	-50,467	-52,331
Agg P&L 0.4%	-7,573	-15,532	-19,301	-19,876	-30,018	-30,217
P&L 0.4% (d=5%)	-3,615	-5,370	-10,392	-9,957	-8,608	-7,940
Agg P&L 0.4% (d=5%)	1,702	-65	-4,774	-4,492	-4,404	-3,097
P&L 0.4% (d=10%)	-401	-255	-3,769	-2,524	705	513
Agg P&L 0.4% (d=10%)	3,293	3,072	399	847	2,560	2,884
P&L 0.4% (d=15%)	1,245	1,711	-150	1,259	4,328	3,464
Agg P&L 0.4% (d=15%)	4,191	4,252	3,060	3,697	5,494	4,915

Table R13.1: Trading strategies for hybrid BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
P&L	24,143	30,438	29,540	29,053	31,567	31,379
# Trades	5,621	8,469	8,338	8,337	10,965	10,758
P&L 0.2%	460	-3,910	-3,697	-3,775	-10,122	-9,717
Agg P&L 0.2%	7,793	5,211	5,808	4,700	513	1,216
P&L 0.2% (d=5%)	7,060	5,085	5,735	6,350	5,621	5,209
Agg P&L 0.2% (d=5%)	9,819	7,635	8,324	9,081	7,702	7,729
P&L 0.2% (d=10%)	6,537	5,801	6,202	6,123	6,844	6,461
Agg P&L 0.2% (d=10%)	8,401	7,352	8,012	7,915	7,796	7,607
P&L 0.2% (d=15%)	6,383	5,488	6,786	6,738	7,564	7,027
Agg P&L 0.2% (d=15%)	7,817	6,641	8,165	8,096	8,154	7,728
P&L 0.4%	-23,223	-38,258	-36,933	-36,603	-51,812	-50,812
Agg P&L 0.4%	-8,557	-20,016	-17,923	-19,652	-30,541	-28,947
P&L 0.4% (d=5%)	-3,115	-8,607	-8,606	-8,095	-7,613	-8,019
Agg P&L 0.4% (d=5%)	2,403	-3,507	-3,429	-2,633	-3,452	-2,980
P&L 0.4% (d=10%)	316	-1,318	-1,989	-1,772	1,126	563
Agg P&L 0.4% (d=10%)	4,045	1,783	1,633	1,811	3,031	2,853
P&L 0.4% (d=15%)	1,779	945	1,372	1,557	4,411	3,715
Agg P&L 0.4% (d=15%)	4,648	3,250	4,131	4,272	5,591	5,117

Table R13.2: Trading strategies for hybrid BS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	CS_{60}^{Nh}	CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}
P&L	20,002	29,247	28,793	30,691	31,231
# Trades	5,202	9,157	9,604	11,678	11,871
P&L 0.2%	-1,869	-7,909	-10,106	-15,394	-15,064
Agg P&L 0.2%	5,336	2,981	1,305	-2,108	-1,449
P&L 0.2% (d=5%)	5,784	5,846	3,699	2,441	2,884
Agg P&L 0.2% (d=5%)	8,594	8,882	6,637	5,426	6,367
P&L 0.2% (d=10%)	5,694	7,401	5,762	5,118	5,023
Agg P&L 0.2% (d=10%)	7,601	9,184	7,293	6,704	6,625
P&L 0.2% (d=15%)	5,407	7,448	6,385	5,606	6,073
Agg P&L 0.2% (d=15%)	6,913	8,732	7,399	6,691	7,033
P&L 0.4%	-23,741	-45,066	-49,004	-61,479	-61,358
Agg P&L 0.4%	-9,329	-23,285	-26,182	-34,907	-34,129
P&L 0.4% (d=5%)	-4,775	-9,338	-12,195	-14,870	-14,853
Agg P&L 0.4% (d=5%)	844	-3,265	-6,319	-8,901	-7,886
P&L 0.4% (d=10%)	-555	-397	-1,785	-3,370	-3,597
Agg P&L 0.4% (d=10%)	3,258	3,170	1,276	-198	-393
P&L 0.4% (d=15%)	818	2,340	1,826	587	1,056
Agg P&L 0.4% (d=15%)	3,829	4,907	3,855	2,756	2,976

Table R14.2: Trading strategies for hybrid CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	$CS_{60}^{Nh^*}$	$CS_{av}^{Nh^*}$	$CS_{avT}^{Nh^*}$	$CS_{avT4}^{Nh^*}$	$CS_{con}^{Nh^*}$
P&L	22,708	28,886	31,663	32,810	32,718
# Trades	5,358	8,898	10,463	10,927	11,098
P&L 0.2%	-1,162	-7,138	-10,387	-10,983	-11,897
Agg P&L 0.2%	6,521	3,451	1,658	1,490	1,836
P&L 0.2% (d=5%)	6,317	6,143	4,316	4,713	4,041
Agg P&L 0.2% (d=5%)	9,012	8,849	7,219	7,633	7,401
P&L 0.2% (d=10%)	6,681	6,737	6,378	6,503	6,091
Agg P&L 0.2% (d=10%)	8,364	8,435	7,984	8,017	7,757
P&L 0.2% (d=15%)	6,688	7,560	7,230	6,814	6,972
Agg P&L 0.2% (d=15%)	8,045	8,847	8,379	7,862	8,040
P&L 0.4%	-25,031	-43,163	-52,437	-54,776	-56,511
Agg P&L 0.4%	-9,666	-21,984	-28,347	-29,829	-29,047
P&L 0.4% (d=5%)	-4,141	-8,962	-11,909	-11,765	-12,765
Agg P&L 0.4% (d=5%)	1,248	-3,550	-6,102	-5,923	-6,046
P&L 0.4% (d=10%)	617	-1,176	-1,401	-1,762	-2,165
Agg P&L 0.4% (d=10%)	3,984	2,220	1,811	1,267	1,168
P&L 0.4% (d=15%)	2,141	2,633	2,717	1,863	2,128
Agg P&L 0.4% (d=15%)	4,855	5,208	5,017	3,959	4,266

Table R14.2: Trading strategies for hybrid CS-based ANN models

P&L is the total profit and loss without transaction costs; # Trades is the number of trades forgone. P&L (d=0%, 5%, 10% and 15%) represents the total profit and loss at 0.2% or 0.4% transaction costs when trading strategies are implemented by ignoring trades that involve call options whose absolute percentage of mispricing error between their models estimates and their market value is less than 0%, 5%, 10% and 15% respectively. The abbreviation *Agg.* refers to trading strategy results with aggregate transaction costs of the underlying asset.

	BS_{60}^{N}	BS_{vix}^{N}	BS_{av}^{N}	BS_{avT}^{N}	BS_{avT4}^{N}	BS_{con}^{N}	BS_{60}^{N*}	BS_{vix}^{N*}	BS_{av}^{N*}	BS_{avT}^{N*}	BS_{avT4}^{N*}	BS_{con}^{N*}
t (Sq)	7.03	9.13	2.40	8.87	6.68	1.75	9.54	7.22	6.24	5.97	3.88	1.32
John. t (Sq)	38.80	46.00	16.72	25.24	16.36	7.77	31.41	46.76	22.61	34.83	27.90	6.36
t (abs)	27.57	43.07	13.62	35.36	31.42	11.66	37.80	35.54	34.84	30.71	24.80	9.00

	CS_{60}^{N}		CS_{av}^{N}	CS_{avT}^{N}	CS_{avT4}^{N}	CS_{con}^{N}	CS_{60}^{N*}		CS_{av}^{N*}	CS_{avT}^{N*}	CS_{avT4}^{N*}	CS_{con}^{N*}
t (Sq)	9.19		2.40	4.30	3.44	2.72	5.95		2.98	4.89	2.70	2.31
John. t (Sq)	34.09		16.16	25.07	23.91	18.10	25.53		16.79	18.45	17.81	14.94
t (abs)	36.20		11.44	25.99	20.96	21.44	19.87		14.97	21.53	16.24	15.22

	BS_{60}^{Nh}	BS_{vix}^{Nh}	BS_{av}^{Nh}	BS_{avT}^{Nh}	BS_{avT4}^{Nh}	BS_{con}^{Nh}	BS_{60}^{Nh*}	BS_{vix}^{Nh*}	BS_{av}^{Nh*}	BS_{avT}^{Nh*}	BS_{avT4}^{Nh*}	BS_{con}^{Nh*}
t (Sq)	-2.89	0.92	-0.08	0.44	0.96	-1.35	1.19	1.35	1.93	-0.88	2.41	0.68
John. t (Sq)	-21.80	11.09	-1.01	4.07	8.47	-6.21	17.43	20.50	16.81	-5.73	8.87	3.98
t (abs)	-9.13	4.96	0.66	-0.17	4.78	-1.81	10.27	11.71	7.57	-1.41	5.81	1.61

	CS_{60}^{Nh}		CS_{av}^{Nh}	CS_{avT}^{Nh}	CS_{avT4}^{Nh}	CS_{con}^{Nh}	CS_{60}^{Nh*}		CS_{av}^{Nh*}	CS_{avT}^{Nh*}	CS_{avT4}^{Nh*}	CS_{con}^{Nh*}
t (Sq)	2.85		0.53	1.84	0.07	0.15	3.18		2.65	1.78	0.98	2.06
John. t (Sq)	26.62		7.46	9.91	1.51	5.79	31.41		8.43	9.77	6.20	6.54
t (abs)	16.92		7.00	3.29	1.88	0.54	20.86		4.74	9.27	4.93	1.33

Table R15: Test statistics that compare the pricing accuracy between the reduced datasets without and with retraining of the ANNs

The t(Sq) compares the MSE of the ANNs trained with the full dataset but examined on the reduced region with the MSE of the ANNs trained on the reduced region. John. (Sq) does the same but with the Johnson modified t-test; t(abs) compares the MAE of the ANNs trained with the full dataset but examined on the reduced region with the MAE of the ANNs trained on the reduced region. In general, a positive value larger than 1.645 (2.325) means that the ANN model that was trained in the full dataset and examined in the reduced region has a larger MSE or MAE than the ANN model that was trained and examined in the reduced region.

2. Robust Artificial Neural Networks for Pricing of European Options

Abstract

The option pricing ability of Robust Artificial Neural Networks optimized with the Huber function is compared against those optimized with Least Squares. Comparison is in respect to pricing European call options on the S&P 500 using daily data for the period April 1998 to August 2001. The analysis is augmented with the use of several historical and implied volatility measures. Implied volatilities are the overall average, and the average per maturity. Beyond the standard neural networks, hybrid networks that directly incorporate information from the parametric model are included in the analysis. It is shown that the artificial neural network models with the use of the Huber function outperform the ones optimized with least squares.

The existing essay had been published in the **Computational Economics,** volume 27, issues 2-3, April-May 2006, pg. 329-351.

2.1. Introduction

The scope of this work is to compare alternative feed-forward Artificial Neural Network (ANN) configurations in respect to pricing the S&P 500 European call options. Robust ANNs that use the Huber function are developed, and configurations that are optimized based solely on the least squares norm are compared with robust[12] configurations that are closer to the least absolute norm. The data for this research come from the New York Stock Exchange (NYSE) for the S&P 500 equity index and the Chicago Board of Options Exchange (CBOE) for call option contracts, spanning a period from April 1998 to August 2001.

Black and Scholes introduced in 1973 their milestone (BS) formula which is still a most prominent conventional Parametric Options Pricing Model (POPM). The options we price are on the S&P 500 index, which is extremely liquid and is the closest to the theoretical setting of the Black and Scholes model (Garcia and Gencay, 2000). Empirical research in the last three decades has shown that the formula suffers from *systematic biases* for various reasons (for details see Black and Scholes, 1975, Rubinstein, 1994, Bates, 1991 and 2003, Bakshi et al., 1997, Andersen et al., 2002, and Cont and Fonseca, 2002). Despite this, BS is frequently used to price European options[13] mainly because alternative parametric models (e.g. stochastic volatility, jump-diffusion, stochastic interest rates, etc.) have drastically failed to provide results truly consistent with the observed market data. Additionally, these models are often too complex to implement and be used for real trading (see Bakshi et al., 1997). On the other hand, artificial neural networks are promising alternatives to the parametric OPMs; they do not necessarily rely on any financial theory and are trained inductively using

---

[12] Huber (1981) and Hampel et al. (1986) offer an overview for the tools and concepts of the theory of robust statistics. As pointed out for example by Franses et al. (1999), parametric estimators that are derived under the assumption of normally distributed errors are very sensitive to outliers and other departures from the normality assumption (see also Krishnakumar and Ronchetti, 1997, and Ortelli and Trojani, 2005). They show that the results obtained under a robust analysis can differ significantly from the ones obtained under similar techniques that are based on the Gaussian analysis. Chang (2005) has found that the use of the Huber estimation can significantly reduce the influence of outliers for the estimation of block-angular linear regression model.

[13] According to Andersen et al., (2002), "*the option pricing formula associated with the Black and Scholes diffusion is routinely used to price European options*".

historical or implied input variables and option transactions data. Their popularity is constantly increasing, and contemporary financial econometric textbooks (e.g. Tsay, 2002) dedicate special sections or even whole chapters to this topic.

It is well known that market participants change their option pricing attitudes from time to time (i.e. Rubinstein, 1994), so a parametric model might fail to adjust to such rapidly changing market behavior. ANNs can potentially correct the aforementioned BS bias (Hutchison et al., 1994, Lajbcygier et al., 1996, Garcia and Gencay, 2000, Yao and Tan, 2000). Neural networks trained on the least squares error criterion are highly influenced by outliers, especially in the presence of non-Gaussian noise (Bishop, 1995). Options data are known to be heavily influenced *at least* by noise due either to thin trading or to abnormal volume trading (Long and Officer, 1997, and Ederington and Guan, 2005) and exhibit a strong time-varying element (Dumas et al., 1995, and Cont and Fonseca, 2002). Consecutively, robust estimation is expected to improve out of sample pricing of options.

In previous empirical research on option pricing, ANNs have been optimized based on the l_2 (the least squares) norm. The l_2 norm is a convenient way to train ANNs, since ready to use statistical packages are widely available for this purpose. Of course, the least squares optimization is highly susceptible to the influence of large errors since some abnormal datapoints (or few outlier observations) can deliver non-reliable networks. On the contrary, robust optimization methods that exploit the l_1 (the least absolute) norm are unaffected by large (or catastrophic) errors but are doomed to fail when dealing with small variation errors (e.g. Bandler et al., 1993, and Devabhaktuni et al., 2001, for applications in the electrical engineering field).
Here the Huber function (Huber, 1981) is used as the error penalty criterion during the ANNs optimization process to immunize the adaptable weights in the presence of data-point peculiarities. The Huber function utilizes the robustness of l_1 and the unbiasedness of l_2 and has proved to be an efficient tool for robust optimization problems for various tasks (Bandler et al., 1993, Jabr, 2004, Chang, 2005), albeit it does not constitute the mainstream. The training of ANNs with the Huber technique has recently gained attention in electrical engineering (i.e. Devabhaktuni et al., 2001, Xi et al., 1999), but to our knowledge has not gained attention in options pricing, where it is possible

to observe both small and large errors for a variety of reasons (e.g. Bakshi et al., 1997). Our choice of the Huber function is because it is widely referenced on robust estimation (Bishop, 1995), it provides a simple generalization of the least squares approach, it avoids the need for any probabilistic assumptions, and it is not difficult to implement with neural networks. Comparison with other estimators, like the *MM* estimators (Yohai, 1987), the *S* estimators (Rousseeuw and Yohai, 1984), and the redescenting estimators (Morgenthaler, 1990), is beyond the scope of this work, but can be part of future research.

The standard ANN target functions are employed that are comprised by the market value of the call option standardized with the strike price. Furthermore, the hybrid ANN target function suggested by Watson and Gupta (1996) and used for pricing options with ANNs in Lajbcygier et al. (1997) are examined. In the hybrid models the target function is the residual between the actual call market price and the parametric option price estimate standardized with the strike price. It can capture the potential misspecification of the BS assumptions of geometric Brownian motion (see for example, Lim et al., 1997). Unlike Hutchison et al. (1994), in the parametric as well as in the nonparametric models both historical and implied volatility measures are used. To train the ANNs, the modified *Levenberg-Marquardt* (LM) algorithm which is efficient in terms of time capacity and accuracy (Hagan and Menhaj, 1994) is utilized. In contrast to many previous studies, thorough cross-validation allows the use of a different network configuration in different testing periods.

In the following, first the parametric BS model, and the standard and hybrid ANN model configuration with the Huber function and with least squares (mean square error to be precise) are reviewed. Then, the dataset, and the historical and implied parameter estimates are discussed, and the parametric and ANN models are defined according to the parameters used. Subsequently, the numerical results are reviewed with respect to the in- and out of sample pricing errors; the economic significance of dynamic trading strategies both in the absence and in the presence of transaction costs is also discussed. The final section concludes. It is demonstrated that with the use of the Huber function, ANNs outperform their counterparts optimized with least squares. The best (hybrid and standard) ANN models with the Huber function are identified, and evidence is provided that, even in the presence of transaction costs, profitable trading opportunities still exist.

2.2. Option pricing models: the parametric BS and ANNs

2.2.1. The Black and Scholes option pricing model

The Black Scholes formula for European call options modified for dividend-paying underlying asset is:

$$c^{BS} = Se^{-\delta T}N(d_1) - Xe^{-rT}N(d_2) \qquad (2.1)$$

$$d_1 = \frac{\ln(S/X) + (r - \delta)T + (\sigma\sqrt{T})^2/2}{\sigma\sqrt{\mathrm{T}}} \qquad (2.1.1)$$

$$d_2 = d_1 - \sigma\sqrt{T} \qquad (2.1.2)$$

In the above, $c^{BS} \equiv$ estimated premium for the European call option; $S \equiv$ spot price of the underlying asset; $X \equiv$ exercise price of the option; $r \equiv$ continuously compounded riskless interest rate; $\delta \equiv$ continuous dividend yield paid by the underlying asset; $T \equiv$ time left until the option expiration; $\sigma^2 \equiv$ yearly variance rate of return for the underlying asset; $N(.) \equiv$ the standard normal cumulative distribution.

The standard deviation of continuous returns (σ) is not observed and an appropriate forecast should be used. The literature has used both *historical* and *implied volatility* forecasts. Contrary to the historical estimates, the implied volatility forecasts have desirable properties that make them attractive to practitioners: they are forward looking and avoid the assumption that past volatility will be repeated. In this study, similarly with Hutchison et al. (1994) and in addition to the other volatility measures, the 60 days historical volatility which is a widely used historical volatility benchmark is also employed.

If BS is a well-specified model, then all implied volatilities of the same underlying asset should be the same or at least some deterministic functions of time. Unfortunately, this is far from being empirically true. For example, Rubinstein (1994) has shown that the implied volatilities derived via BS as a function of the moneyness ratio (*S/X*) and time to expiration (*T*) often exhibit a U shape, known as the *volatility smile*. This is why BS is usually referred to as being a misspecified model with an inherent source of bias (see also Latane and Jr., 1976, Bates, 1991, Canica and Figlewski, 1993, Bakshi et al., 2000, and Andersen et al., 2002). Under the existence of this anomaly, any historical volatility measure is doomed to fail, while measures (like the implied ones) that mitigate this bias could perform better.

2.2.2. Neural networks

A neural network is a collection of interconnected simple processing elements structured in successive layers and can be depicted as a network of arcs/connections and nodes/neurons. The network has the input layer, one or more hidden layers and an output layer. Each interconnection corresponds to a numerical value named weight, which is modified according to the faced problem via an optimization algorithm. The particularity of ANNs relies on the fact that the neurons on each layer operate collectively and in a parallel manner on all input data and that each neuron behaves as a summing vessel that works, under certain conditions, as a non-linear mapping junction for the forward layer.

Figure 2.1 depicts an ANN architecture similar to the one applied for the purposes of this study. This network has three layers: an *input* layer with *N* input variables, a *hidden* layer with *H* neurons, and a *single* neuron *output* layer. Each neuron is connected with all neurons in the previous and the forward layer. Each connection is associated with a *weight*, w_{ki}, and a *bias*, w_{k0}, in the hidden layer and a *weight*, v_k, and a *bias*, v_0, in the output layer ($k=1,2,\dots,H$, $i=1,2,\dots N$). In addition, the outputs of the hidden layer ($y_1^{(1)}, y_2^{(1)} \dots y_H^{(1)}$) are the inputs for the output layer.

Inputs are set up in feature vectors, $\tilde{x}_q = [x_{1q}, x_{2q} \dots, x_{Nq}]$ for which there is an associated and known target, t_q, $q \equiv 1,2,\dots,P$, where P is the number of the available sample feature vectors for a particular training sample. According to Figure 2.1, the operation carried out for computing the final estimated output, y, is the following:

$$y = f_O[v_O + \sum_{k=1}^{H} v_k f_H(w_{kO} + \sum_{i=1}^{N} w_{ki} x_i)] \qquad (2.2)$$

where f_0 and f_H are the transfer functions associated with the output and hidden layers respectively.

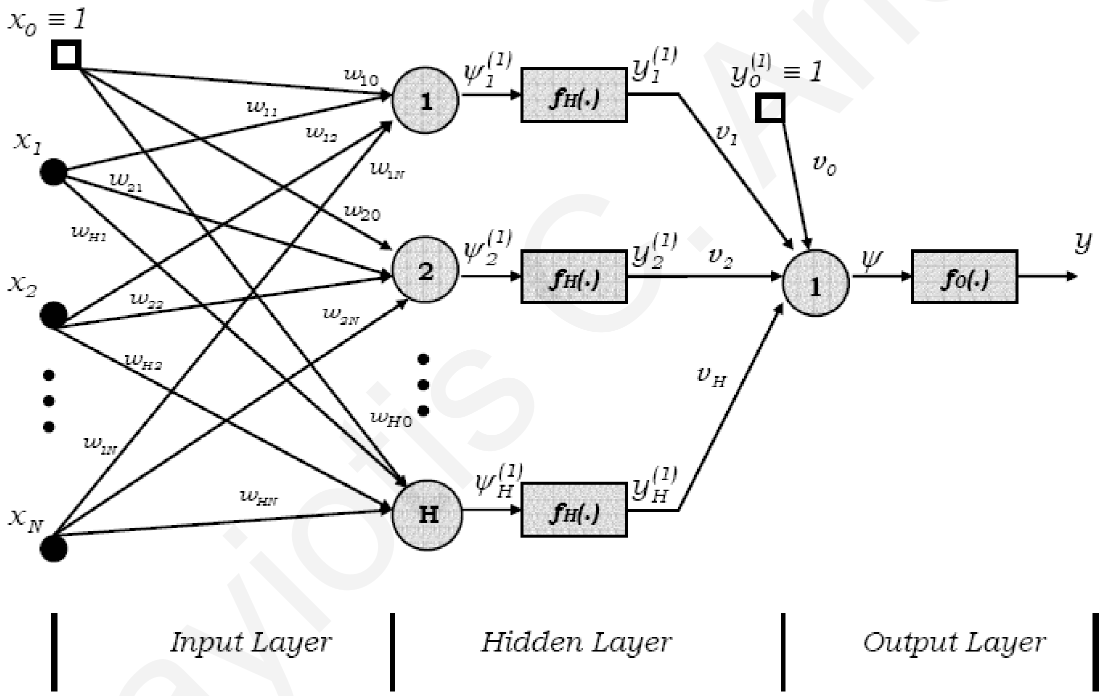

Figure 2.1. A single hidden layer feedforward neural network

For the purpose of this study, the hidden layer always uses the hyperbolic tangent sigmoid transfer function, while the output layer uses a linear transfer function. In addition, ANN architectures with only one hidden layer are considered since research has shown that this is adequate in order to approximate most functions arbitrarily well. This is based on the universal approximation theorem provided by Cybenko (1989) (theorem can be fund in section 1.3., for further details see also Haykin, 1999):

Training ANNs is a non-linear optimization process in which the network's weights are modified according to an error function. For the case that the ANN model has only one output neuron, the error function between the estimated response y_q and the actual response t_q is defined as:

$$e_q(w) = y_q(w) - t_q \qquad (2.3)$$

where, w is an n-dimensional column vector containing the weights and biases given by: $w = [w_{10} \ldots, w_{H0}, w_{11}, \ldots, w_{HN}, v_0, \ldots, v_H]^T$. The Huber function that is used to optimize the trainable parameters w is defined as (i.e. Huber, 1981, Bandler et al., 1993):

$$E(w) = \sum_{q=1}^{P} \rho_k(e_q(w)) \qquad (2.4)$$

where ρ_k is the Huber function defined as:

$$\rho_k(e) = \begin{cases} 0.5e^2 & if\ |e| \le k \\ k|e| - 0.5k^2 & if\ |e| > k \end{cases} \qquad (2.5)$$

where k is a positive constant. It is obvious that when $|e| > k$ the Huber function treats the error in the l_1 sense and in the l_2 sense only if $|e| \le k$ depending on the value of threshold parameter k. Figure 2.2 depicts the Huber function along with the least absolute (l_1) and least squares (l_2) error functions. The Huber function has a smooth transition between the two norms at $|e| = k$, so that the first derivative of ρ_k is continuous everywhere.

The choice of k defines the threshold between *large* and *small* errors. Different values of k determine the proportion of the errors to be treated in the l_1 or the l_2 norm. As seen, when k is sufficiently large the Huber function encompasses the widely used and conventional least squares (l_2) training of the ANNs. As the k parameter approaches zero, the Huber function approaches the l_1 function and the errors are penalized in the least absolute sense. Figure 2.2 makes obvious that the Huber function should be more robust to abnormal data since it penalizes them less compared to the l_2 norm.

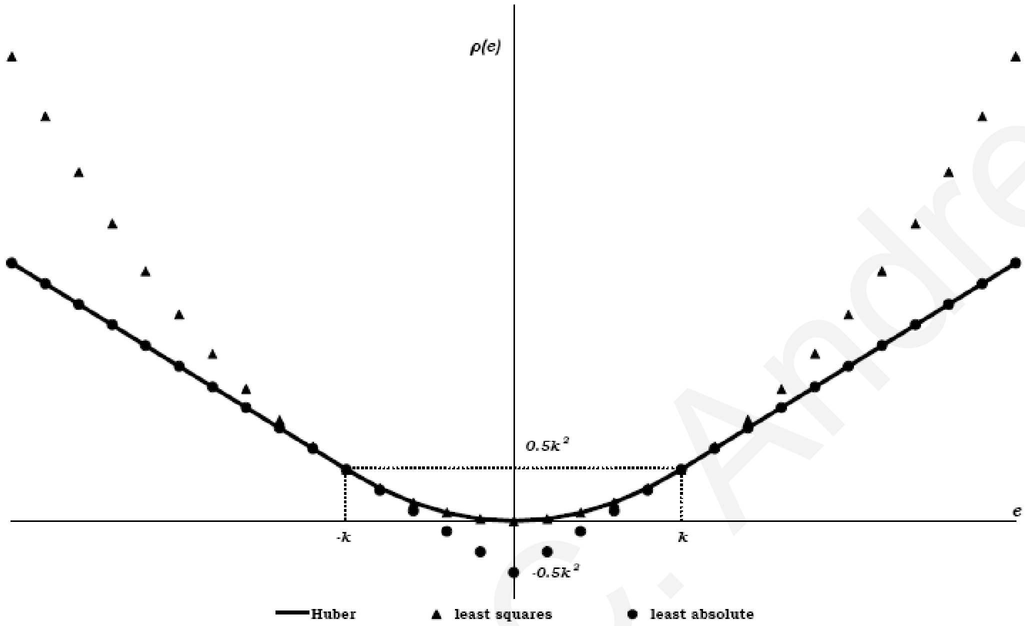

Figure 2.2. The Huber, the least absolute (l_1) and the least squares (l_2) error functions

The nice properties of the Huber function compared to the l_2 norm are more distinct when they are compared according to their gradient vector. The gradient vector of the least squares error function is:

$$\nabla E_{l_2}(w) = \sum_{q=1}^{P} e_q \nabla e_q(w) \tag{2.6}$$

whilst the gradient for the Huber function is:

$$\nabla E(w) = \sum_{q=1}^{P} \zeta_q \nabla e_q(w) \tag{2.7}$$

where,

$$\zeta_q = \frac{\partial \rho_k(e_q)}{\partial e_q} = \begin{cases} e_q & if \ |e_q| \le k \\ -k & if \ e_q < k \\ +k & if \ e_q > k \end{cases} \tag{2.8}$$

The $P \times n$ Jacobian matrix, $J(w)$, of the *P-dimensional* output error column vector is given by:

$$J(w) = \begin{bmatrix} \nabla e_1^T(w) \\ \vdots \\ \nabla e_P^T(w) \end{bmatrix} \tag{2.9}$$

Using this notation, (2.7) can be written in the form:

$$\nabla E(w) = J(w)^T \zeta(w) \tag{2.10}$$

where ζ is a *P-dimensional* column vector with elements the ζ_q values.

Quantity $\nabla e_q(w)$ is the gradient vector of $e_q(w)$ with respect to the trainable parameter vector w. The difference between (2.6) and (2.7) depends on the weighting factor of the $\nabla e_q(w)$. The weighting factor of $\nabla e_q(w)$ for the Huber gradient is the same with the least squares gradient only when $|e_q| \le k$. In all other cases the weighing factor for the Huber gradient is held constant at the value of the threshold k unlike in the l_2 case that gives more weight to large errors. This is how the Huber function immunizes against the influence of large errors.

Moreover, the Hessian matrix in the case of the Huber function is given by:

$$\nabla^2 E(w) = \sum_{q=1}^{P} d_q \nabla e_q(w) \nabla e_q(w)^T + \sum_{q=1}^{P} \zeta_q \nabla^2 e_q(w) \tag{2.11}$$

where

$$d_q = \frac{\partial^2 \rho_k(e_q)}{\partial e_q{}^2} = \begin{cases} 1 & if \ \left|e_q\right| \le k \\ 0 & if \left|e_q\right| > k \end{cases} \qquad (2.12)$$

The quantity $\nabla e_q(w)$ is computed based on the back-propagation algorithm that is commonly used in the context of feed-forward ANNs. Based on the neural network depicted in Figure 2.1, the partial derivative of the error function (2.3) with respect to the weight υ_k at the hidden layer is:

$$\frac{\partial e_q}{\partial \upsilon_k} = y_k^{(1)} f_O'(\psi) \qquad (2.13)$$

where $f_O'(\psi)$ is the differential of the output neuron transfer function at point ψ. Since a linear transfer function is used at the output neuron, the $f_O'(\psi)$ is equal to unity. Furthermore, the partial derivative of the error function (3) with respect to the weight w_{ki} at the input layer is:

$$\frac{\partial e_q}{\partial w_{ki}} = x_i f_H'(\psi_\kappa^{(1)}) \upsilon_k f_0'(\psi) \qquad (2.14)$$

where $f_H'(\psi_\kappa^{(1)})$ is the differential of the transfer function associated with the k^{th} hidden neuron at point $\psi_\kappa^{(1)}$. For our case, we always use the hyperbolic tangent as a transfer function:

$$f_H(a) = \frac{2}{1+e^{-2a}} - 1 \equiv tanh(a) \qquad (2.15)$$

The differential of this function with respect to a can be expressed in a particularly simple form:

$$f_H'(a) = 1 - (f_H(a))^2 \tag{2.16}$$

To optimize the weights, the modified *Levenberg-Marquardt* (LM) algorithm is employed. According to LM, the weights and the biases of the network are updated in order to minimize $E(w)$. At each iteration τ of the LM, the weights vector w is updated as follows:

$$w_{\tau+1} = w_\tau - [G_\tau + \mu_\tau I]^{-1} J(w_\tau)^T \zeta(w_\tau) \tag{2.17}$$

where G is an *approximation* of the $n \times n$ Hessian matrix defined as:

$$G = \sum_{q=1}^{P} d_q \nabla e_q(w) \nabla e_q(w)^T \tag{2.18}$$

and d_q is defined in (2.12). The matrix G is obtained from the Hessian matrix by deleting its second term which is usually considered small. Moreover, I is an $n \times n$ identity matrix, $J(w_\tau)$ is the Jacobian matrix at the τ^{th} iteration, and μ_τ is like a learning parameter that is automatically adjusted in each iteration in order to secure convergence. Large values of μ_τ lead to directions that approach the steepest descent, while small values lead to directions that approach the Gauss-Newton algorithm. Further technical details about the implementation of LM can be found in Hagan and Menhaj (1994) and Hagan et al. (1996). Based on (2.17), weights and biases update takes place in a batch mode manner where update occurs only when all input vectors have been presented to the network.

In addition to the standard ANNs with $t \equiv c^{mrk} / X$ (call market values standardized with their strike price), hybrid ANNs according to which the target function is the residual between the actual call market price and the BS call option estimation $t \equiv c^{mrk} / X - \hat{c}^{\Theta} / \mathrm{X}$ (again standardized with the strike price) are also investigated, where $\hat{c}^{\Theta}$ should define a pricing estimate taken by the BS under a certain volatility forecast (this is explained further in the following section). For effective training, the input/output variables are scaled using the *z*-score transformation $\tilde{z} = (\tilde{x} - \mu) / s$, where $\tilde{x}$ is the vector of an input/output variable, μ is the mean and s the standard deviation of this vector. Moreover, the network initialization technique proposed by Nguyen and Windrow (see Hagan et al., 1996) is utilized that generates initial weights and bias values for a nonlinear transfer function so that the active regions of the layer's neurons are distributed roughly evenly over the input space.

For a given set of training data and for a given value of the Huber *k* value, the optimal number of hidden neurons is chosen via a cross-validation procedure. ANN structures with 2 to 10 hidden neurons are trained, and the one that performs the best in the validation period is selected. Since the initial network weights affect the final network performance, for a specific number of hidden neurons and Huber *k* value, the network is initialized, trained and validated five separate times. Huber (1981) gives a formula for deriving the optimal *k* value, but this formula was not derived with applications of neural networks in mind. Most importantly, restrictive probabilistic assumptions (of symmetrically contaminated Gaussian distributions) are made. In addition, (as pointed out also in Koenker, 1982, p. 232), we need to know the degree of contamination (i.e., the percent of abnormal observations). With neural networks we neither make any probabilistic assumptions, nor we know a priori the degree of contamination. Thus, we follow an empirical approach. The optimal *k* value is shown from the data after investigating a wide range of potential values. Different networks are developed for the following Huber *k*-values: 0.1, 0.2, 0.30, 0.40, 0.5, 0.60, 0.70, 0.80, 0.90, 1, 1.5, 2 and *Inf* (that corresponds to the optimization of the ANNs based on the l_2 norm). After defining the optimal ANN structure, its weights are frozen and its pricing capability is tested (out of sample) in a third separate *testing dataset* in order to verify the ANNs ability to generalize to unseen data.

2.3. Data, parameter estimates and model implementation

The dataset covers the period from April 1998 to August 2001. The S&P 500 Index call options are considered because the CBOE option market is extremely liquid and these index options among the most popular. This market is the closest to the theoretical setting of the parametric models (Garcia and Gencay, 2000). Our prices are closing quotes. The majority (around 75%) of our call options lies in the +/-15% moneyness area. As suggested by Day and Lewis (1988), because trading volume tends to concentrate in the options that are around at-the-money and just in-the-money, any lack of synchronization between closing index level and the closing option price will be minimized for these options (pg. 107). Of course, it is not the first time that non intra-day option and index prices are used in analysis (see for example, Day and Lewis, 1988, Hutchison et al., 1994, Ackert and Tian, 2001, and Ederington and Guan, 2005). Specifically, Ackert and Tian (2001) argue that closing prices, which are non-synchronous, constitute the best alternative solution to examine the options arbitrage violations for the S&P index. Kamara and Miller (1995) compare intraday and closing option pricing results for market efficiency tests and argue that closing option prices are appropriate for analysis because they are representative of the transaction prices that prevailed during the day. This suggests that it is not unreasonable to use closing data in empirical options research. In our case, the Huber function is helpful in treating the options data according to the noise level.

Along with the index, a daily dividend yield, δ, is collected (provided online by Datastream). After applying various filtering rules, the dataset consists of 64,627 data points, with an approximate average of 35,000 data points per sub-period. Hutchison et al. (1994) have an average of 6,246 data points per sub-period. Lajbcygier et al. (1996) include in total 3,308 data points, Yao et al. (2000) include in total 17,790 data points, and Schittenkopf and Dorffner (2001) include 33,633 data points.

2.3.1. Observed and historically estimated parameters

The moneyness ratio, *S/X*, is the basic input to be used with ANNs since it is highly related with the pricing bias associated with the BS. The moneyness ratio S/X is calculated and used with ANNs like in Hutchison et al. (1994) (see also Garcia and Gencay, 2000). The dividend adjusted moneyness ratio $(Se^{-\delta T})/X$ is preferred here since dividends are relevant. In addition, the time to maturity (T) is computed assuming 252 days in a year. Previous studies have used 90-day T-bill rates as approximation of the interest rate. In this study we use nonlinear cubic spline interpolation for matching each option contract with a continuous interest rate, r, that corresponds to the option's maturity. For this purpose, the 3-month, 6-month and one-year T-bill rates collected from the U.S. Federal Reserve Bank Statistical Releases are used.

Moreover, the 60-days volatility is a widely used historical estimate (see Hutchison et al., 1994, and Lajbcygier et al., 1997). This estimate is calculated using all the past 60 log-relative index returns and is symbolized as σ_{60}. In addition, the VIX Volatility Index is an estimate that can be directly observed from the CBOE. It was developed by CBOE in 1993 and is a measure of the volatility of the S&P 100 Index and is frequently used to approximate the volatility of the S&P 500 as well. In our dataset the 30-day returns of the two indexes were strongly correlated (with Pearson correlation coefficient between 0.94 and 0.99). VIX is calculated as a weighted average of S&P 100 option with an average time to maturity of 30 days and emphasis on at-the-money options. This volatility measure is symbolized as σ_{vix}.

2.3.2. Implied parameters

The Whaley's (1982) simultaneous equation procedure is adopted to minimize a price deviation function with respect to the unobserved parameters. For a given day the optimal implied parameter values correspond to the solution of an unconstrained optimization problem that minimizes the sum of squares residuals between the actual call option market values and the BS estimates. The optimization is done via a non-linear least squares optimization based on the Levenberg-Marquardt algorithm. His approach is an alternative to using the methodology proposed by Chiras and Manaster (1978) (CM), or Latane and Rendleman (1976) (LR). His reasoning is that: "rather than explicitly weighting the implied standard deviations of a particular stock where the weights are assigned in an ad-hoc fashion, the call prices are allowed to provide an implicit weighting scheme that yields an estimate of standard deviation which has little prediction error as is possible" (pg. 39). Bates remarks that the Whaley's (1982) least squares weighting scheme effectively assigns heavier weights on the near the money options than CM and LR. His approach is widely applied even in more recent research; for instance Bakshi et al. (1997). Nevertheless, we tried these two weighting schemes (the CM and the modified LR as recommended by CM), and at least in our dataset the results are inferior to those of the overall average approach (or its per-maturity variant). The per-maturity versions worked even better since they can capture time-varying volatility effects (Bakshi, 1997, and Bates, 2003).

Similarly to Bakshi et al. (1997), two different implied volatility measures are taken from the above procedure. The first optimization is performed by including all available options transaction data in a day to obtain *daily average* implied parameters (σ_{av}). Second, *daily average per maturity* parameters (σ_{avT}) can be obtained by fitting the BS to all options that share the same maturity date as long as four different available call options exist.

For pricing and trading reasons at time instant *t*, the implied structural parameters derived at day *t*-1 are used together with all other needed information (*S*, *T*, *δ*, *X* and *r*). The same reasoning holds for the historical (σ_{60}) and the weighted implied average (σ_{vix}) estimates.

It is known that ANN input variables should be presented in a way that maximizes their information content (Garcia and Gencay, 2000). When options are priced, the POPM formulas adjust those values to represent the appropriate value that corresponds to an option's expiration period. According to this rationale, for use with the ANNs, the volatility measures are transformed by multiplying each of the yearly volatility forecast with the square root of each option's maturity ($\tilde{\sigma}_j = \sigma_j \sqrt{T_{252}}$, where *j*={*60, vix, av, avT*}). They are named *maturity (or expiration) adjusted volatilities*. Also, and following the advice by a referee, we have constructed tables (not included for brevity) for all nine sub-periods (in several moneyness and maturity ranges) in order to compare between the volatilities of the training and the volatilities of the testing sub-periods. If these estimates differ considerably, this may imply saturation of the neural network with poor performance as a consequence. On average, we have observed no volatility jumps. Furthermore, the superior out of sample performance of the neural networks (see section 4) is additional evidence that the saturation problem mentioned by the referee does not seem to be present.

2.3.3. Output variables, filtering and processing

For training ANNs, the call standardized by the striking price, c_q^{mrk} / X_q, is used as one target function to be approximated. In addition, the hybrid structure is implemented, where the target function represents the pricing error between the option's market price and the parametric models estimate, $c^{mrk} / X - \hat{c}^{\Theta} / \mathrm{X}$, where Θ is one of BS_{60}, BS_{vix}, BS_{av}, and BS_{avT}. In both cases, target residuals are standardized using the mean-variance scaling; hence the output neuron transfer function is linear.

Before filtering, more than 90,000 observations were included for the period April 1998 – August 2001. The final dataset consists of 64,627 datapoints. The filtering rules adopted are: *i) eliminate an observation if the call contract price is greater than the underlying asset value*; *ii) exclude an observation if the call moneyness ratio is larger than unity and the call price is less than its dividend adjusted lower bound; iii) eliminate all the options observations with time to maturity less than 6 trading days* (adopted to avoid extreme option prices that are observed due to potential illiquidity problems); *iv) price quotes lower than 0.5 index points are not included*; *v) maturities with less than four call option observations are also eliminated; vi)* in addition, to remove impact from thin trading observations are eliminated according to the rule: *eliminate an observation if the call option price at day t is the same as with day t*-1 *and if the open interest for these days stays unchanged and if the underlying asset has changed.* We filter data when we believe that they are "bad data" (filtering rules *i*, *ii*, *iv*, *vi*), or that they come from a different "data generating process" (filtering rule *iii*, following Bakshi et al., 1997). Filtering rule *v* was perceived as necessary in order to get an average volatility per maturity (Bakshi et al., 1997, recommend no less than two observations).

2.3.4. Validation, testing and pricing performance measures

The available data are partitioned into training, validation and testing datasets using a chronological splitting, and via a rolling-forward procedure. Our dataset is divided into nine overlapping sub-periods in chronological order. Each sub-period is divided into a training (*Tr*), a validation (*Vd*) and a testing (*Ts*) set, again in chronological sequence. In each sub-period the training, validation and testing sets are non-overlapping. The nine testing sets are non-overlapping and they cover completely the last 20 months of the dataset.

There are M available call option contracts, for each of which there exist Ξ_m observations taken in consecutive time instances t, resulting in a total of P ($P = \sum_{m=1}^{M} \Xi_m$) available call option datapoints (P is the total number of call option transactions that cover the whole period and is equal to 64,627). To determine the pricing accuracy of each model's estimates, $\hat{c}$, the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE) are examined:

$$RMSE = \sqrt{1/p\sum_{q=1}^{p}(c_q^{mrk} - \hat{c}_q)^2} \qquad (2.19)$$

$$\text{MAE} = 1/p\sum_{q=1}^{p}| c_q^{mrk} - \hat{c}_q | \qquad (2.20)$$

where $p \leq P$ indicates the number of observations used per case. These error measures are computed for an aggregate testing period (*AggTs*) with 35,734 (so p is equal to 35,734) datapoints by simply pooling together the pricing estimates of all nine testing periods. For *AggTs*, the Median Absolute Error (MdAE) as well as the 5^{th} (5^{th} APE) and 95^{th} (95^{th} APE) percentile Absolute Pricing Error values derived from the whole pricing error distribution are also computed and tabulated. Since ANNs are not optimized solely based on the mean square error and there are cases that the ANNs are optimized with the Huber function, it is wise to take into consideration various error measures.

2.3.5. The parametric and nonparametric models used

With the BS models input includes *S, X, T, δ, r*, and any of the four different volatility measures: σ_{60}, σ_{vix} σ_{av}, and σ_{avT}; the four different models are symbolized as: BS_{60}, BS_{vix}, BS_{av}, and BS_{avT}.

To train ANNs inputs of the parametric BS model are also used. These include the three standard input variables/parameters: $(Se^{-\delta T})/X$, T and r. The various versions of the ANNs also depend on the BS volatility estimate considered, the kind of the target function, and the *k* value of the Huber function.

As mentioned before, ANNs are trained based on twelve different values of the Huber function ($k \in$ [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1, 1.5, 2]). Additionally, ANN structures trained with the use of the mean square error (l_2 norm) which is equivalent to the case where the Huber *k* value is set to a very large value that approaches infinity (*k* = Inf) are included.

Specifically, for each of the four different BS volatility measures, there are thirteen ANN models that are trained to map the standard target function c^{mrk}/X (fifty-two models). Furthermore, each of the previous ANN structures is rebuild based on the hybrid target function, $c^{mrk}/X - \hat{c}^{\Theta}/X$ where Θ is one of BS_{60}, BS_{vix}, BS_{av}, and BS_{avT}. In total, there are 104 different ANN versions.

The standard ANNs are denoted by *Ns*, and the hybrid versions by *Nh*. To distinguish between various Huber function versions, the corresponding value of the *k* parameter is used in the superscript and the BS volatility reference is used in the subscript. For instance, Ns_{avT}^{Inf} (Nh_{avT}^{Inf}) is the ANN model that uses as fourth input the (maturity adjusted) volatility, $\tilde{\sigma}_{avT}$, maps the standard (hybrid) target function and is trained based on the mean square error.

2.4. Pricing results and discussion

Table 2.1 exhibits the out-of sample pricing performance of BS and ANN models with alternative volatility measures. As mentioned before, the various models are compared in terms of RMSE, MAE, MdAE and the 5th and 95th Absolute Pricing Errors. All statistics are reported for the *AggTs* (aggregate) period; for the neural networks the aggregate results are created by selecting the optimal Huber *k*-value in the RMSE measure for each sub-period, aggregating, and then comparing with least squares (*inf*) estimation.

	Parametric Models			
	BS_{60}	BS_{vix}	BS_{av}	BS_{avT}
RMSE	10.360	12.302	8.266	7.952
MAE	6.620	8.631	4.989	4.646
MdAE	4.458	6.386	3.630	3.274
5th APE	0.302	0.482	0.323	0.256
95th APE	19.448	23.732	12.399	11.672
	Standard Neural Networks (Optimal *k*, *Inf*)			
	Ns_{60}	Ns_{vix}	Ns_{av}	Ns_{avT}
RMSE	10.52, 15.38	10.08, 12.70	11.25, 11.92	10.76, 12.07
MAE	5.73, 9.51	4.67, 6.44	5.18, 6.62	5.42, 5.90
MdAE	4.06, 6.58	2.99, 3.98	3.35, 4.28	3.40, 3.53
5th APE	0.44, 0.50	0.30, 0.41	0.34, 0.44	0.33, 0.36
95th APE	14.90, 26.54	12.71, 18.92	13.29, 20.20	15.10, 17.39
	Hybrid Neural Networks (Optimal *k*, *Inf*)			
	Nh_{60}	Nh_{vix}	Nh_{av}	Nh_{avT}
RMSE	8.16, 8.58	7.88, 7.79	7.21, 7.73	6.83, 7.15
MAE	5.05, 5.59	3.95, 4.60	4.13, 4.52	3.56, 4.02
MdAE	3.54, 4.02	2.50, 3.07	2.87, 3.03	2.38, 2.58
5th APE	0.29, 0.30	0.22, 0.29	0.24, 0.26	0.20, 0.20
95th APE	13.84, 15.21	10.32, 13.40	10.91, 12.65	9.13, 11.37

Table 2.1. Pricing results with standard and robust ANNS
RMSE is the Root Mean Square Error, MAE the Mean Absolute Error, MdAE the Median Absolute Error, 5th APE is the fifth percentile Absolute Pricing Error and 95th APE the 95th percentile Absolute Pricing Error. The right hand side subscripts refer to the kind of historical/implied parameters used. For the neural networks, the information provided is first under optimal *k*-value in each sub-period, and then under least squares estimation.

It is obvious that the implied volatility measures lead to lower pricing errors in the case of BS. Looking at the parametric models and similarly to Bakshi et al. (1997), the overall best BS model is the one that utilizes the implied average per maturity volatility, BS_{avT}, followed by BS_{av} that utilizes the overall average. The BS_{avT} model outperforms significantly all others in all error measures. Specifically, BS_{avT} has RMSE equal to 7.952, MAE equal to

4.646 and MdAE equal to 3.274. In addition, this model has a higher chance for small pricing errors and considerably smaller chance for large pricing errors compared to the other models (see the 5th and 95th APE statistics).

	σ_{60}	σ_{vix}	σ_{av}	σ_{avT}
ANN RMSE MAE	0.1-0.3, 0.1-0.3	0.1-0.1, 0.1-0.1	0.1-0.2, 0.1-0.2	0.1-0.2, 0.1-0.2
Hybrid RMSE MAE	0.2-0.4, 0.2-0.5	0.1-0.5, 0.1-0.3	0.1-0.8, 0.1-0.7	0.5-0.9, 0.2-0.6

Table 2.2. Range of observed optimal *k* values
The above figures include at least the 66.66% of observed optimal *k* values for the 9 testing sub-periods, after the 3 out of the 9 were removed. The first range is for the RMSE and the second for the MAE error measures.

In comparing the parametric models with the standard (non-hybrid) ANNs that were trained based on the mean square error criterion, it is true that in general, the standard ANN models underperform the equivalent parametric ones (see also Lajbcygier et al, 1996). But Huber standard ANN models perform better than the equivalent least squares ones. The significance of the improvement provided by the Huber approach is obvious from the APE error measures. In some cases (Ns_{vix}) the improvement provides a model better than the equivalent parametric one.

Before considering the impact of the Huber approach, it is evident that the hybrid least squares ANNs outperform significantly both the respective parametric ones, and the standard ANNs, in all measures considered in practically all cases. Similarly to the parametric OPMs, the out of sample pricing accuracy of ANNs seems to be highly dependent on the implied parameters used; that is, as we move from Nh_{60}^{Inf} to Nh_{avT}^{Inf} the pricing accuracy improves significantly. The hybrid least squares ANNs even with historical or weighted average input parameters are considerably better than the equivalent parametric alternatives. Furthermore, it can be observed that Nh_{avT}^{Inf} outperforms all other parametric and least squares ANN models.

The Huber optimized hybrid ANN models outperform significantly all equivalent standard ANNs (Huber and least squares) in all error measures considered. The Huber optimized hybrid ANN models outperform significantly all equivalent least squares hybrid ANNs, in all measures considered in practically all cases. The only exception is when *vix* volatility is used and in a small difference among the RMSE measures; in all other measures, this model with the Huber approach proved to be superior to the least squares one. Again, the Huber optimized hybrid ANN model with *avT* volatility is the overall best, with RMSE equal to 6.83, MAE equal to 3.56, MdAE equal to 2.38, and 5th APE equal to 0.20. We should feel confident in selecting this model, since its 95th APE is equal to 9.13, compared to 11.37 of the equivalent least squares ANN.

Since in each testing sub-period we used the optimal Huber *k*-value determined from the validation set, Table 2.2 demonstrates a clustering summary for standard and hybrid ANNs, in the RMSE and the MAE error measure. It shows the range that includes the majority of observed optimal *k* values (six out of the nine). For the standard ANNs we have a strong clustering around 0.1 and 0.2, and for the hybrid ANNs values around 0.3 and 0.6 are the most likely ones.

S/X	<0.85	0.85-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.15	1.15-1.35	≥1.35
Ns60								
<60 Days	1.13	0.76	0.27	0.09	0.13	0.18	0.32	9.30
60-180 Days	0.48	0.20	0.15	0.17	0.34	0.18	0.51	11.48
≥ 180 Days	0.00	0.06	2.14	4.17	1.56	0.08	0.89	1.62
Nsvix								
<60 Days	0.56	0.22	0.27	0.09	0.15	0.17	0.24	7.57
60-180 Days	0.14	0.10	0.17	0.21	0.40	0.29	0.55	13.59
≥ 180 Days	1.25	0.06	2.31	4.33	1.47	0.20	1.14	1.48
Nsav								
<60 Days	0.71	0.00	0.00	0.00	0.01	0.07	0.06	8.03
60-180 Days	0.07	0.05	0.06	0.13	0.24	0.06	0.37	14.73
≥ 180 Days	0.16	0.13	2.82	4.98	1.91	0.20	1.46	1.96
NsavT								
<60 Days	1.69	0.73	0.28	0.11	0.17	0.18	0.26	9.10
60-180 Days	0.95	0.22	0.15	0.17	0.32	0.22	0.41	10.79
≥ 180 Days	0.00	0.06	2.14	3.69	1.04	0.16	0.76	1.35

Table 2.3. Percentage of outliers for standard robust ANNs
In each cell tabulated (per maturity and degree of moneyness) the percentage of observations that behave as outliers when the RMSE is used as error measure in the *standard* robust neural network. The information is grouped vertically for the four volatility measures, starting with the 60 days maturity, then the *VIX*, the overall average (*av*), and finally the average per maturity (*avT*).

S/X	<0.85	0.85-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.15	1.15-1.35	≥1.35
Ns60								
<60 Days	53.7	13.2	16.8	21.0	19.8	21.0	19.5	28.7
60-180 Days	40.8	22.0	27.3	30.1	31.6	32.9	22.1	40.4
≥ 180 Days	47.3	43.7	55.4	57.6	56.0	55.1	48.7	61.5
Nsvix								
<60 Days	21.0	7.8	13.8	16.3	16.7	16.6	14.3	27.9
60-180 Days	16.2	10.9	22.1	22.3	22.6	24.2	21.9	37.1
≥ 180 Days	8.6	22.0	31.0	33.5	31.9	30.5	28.9	34.5
Nsav								
<60 Days	44.8	14.1	17.2	18.3	17.8	16.2	17.4	30.7
60-180 Days	29.2	13.7	21.6	20.7	21.3	24.0	25.4	39.0
≥ 180 Days	33.2	25.9	30.3	29.7	28.2	26.0	33.4	41.3
NsavT								
<60 Days	25.3	2.0	1.7	3.4	3.3	3.1	2.5	18.8
60-180 Days	13.0	1.5	4.3	5.4	6.2	6.4	6.5	23.3
≥ 180 Days	16.1	3.4	9.8	11.4	8.8	7.0	9.8	10.4

Table 2.4. Percentage of outliers for hybrid robust ANNs
In each cell tabulated (per maturity and degree of moneyness) the percentage of observations that behave as outliers when the RMSE is used as error measure in the *hybrid* (robust) neural network. The information is grouped vertically for the four volatility measures, starting with the 60 days maturity, then the *VIX*, the overall average (*av*), and finally the average per maturity (*avT*).

Tables 2.3 (for the standard ANN) and 2.4 (for the hybrid ANN) present information about the percent of observations treated as outliers by the use of the Huber function (using the RMSE as the error measure). Each cell is for a maturity and degree of moneyness classification the following line the percent of those observations treated as outliers. For the standard neural networks we observe outliers heavily concentrated in the in-the-money observations of short and medium maturity options. There is also evidence of outliers present in at-the-money long maturity options. Drawing on Long and Officer (1997) the long-maturity at-the-money outliers instead, may be attributed to microstructure effects. As Long and Officer show, excessive demand for certain options may also induce the presence of outliers. For the hybrid neural networks we observe that the Huber technique is even more important since outliers are heavily concentrated not only in in-the-money but also in out-of-the-money observations; furthermore, other cells also often show significant evidence of outliers. The wide range of outliers in the hybrid model is a hint that the misspecification of the BS model is in general rather significant in all ranges of moneyness and maturity. Heavily out-of-money outliers may also be due to thin (non-synchronous) trading effects (Day and

Lewis, 1988). For the hybrid model, the choice of volatility used with BS seems to be more important than for the standard neural network.

In the spirit of Black and Scholes (1972), Galai (1977), and Whaley (1982), the economic significance of the OPMs has also been investigated by implementing trading strategies. Trading strategies are implemented based on single instrument hedging as for example in Bakshi et al. (1997), and in addition, transaction costs and cost-effective heuristics are incorporated (see results in Essay #1). Portfolios are created by buying (selling) options undervalued (overvalued) relative to a model's prediction and taking a delta hedging position in the underlying asset. This (single-instrument) delta hedging follows the no-arbitrage strategy of Black and Scholes (1973), where a portfolio including a short (long) position in a call is hedged via a long (short) position in the underlying asset, and the hedged portfolio rebalancing takes place in discrete time intervals. Rebalancing is done in an optimal manner, not necessarily daily; the position is held *as* long *as* the call is undervalued (overvalued) without necessarily daily rebalancing. Proportional transaction costs of 0.2% are also paid for both positions (in the call option and in the "index shares"). Strategies with enhanced cost-effectiveness are also implemented by ignoring trades that involve call options whose absolute percentage mispricing error is less than a mispricing margin of 15%. Even with transaction costs, there still exist opportunities for profitable trading. Again, the hybrid neural networks outperform all other models, and when estimated via the Huber approach they outperform the ones estimated via least squares.

2.5. Conclusions

The option pricing ability of Robust ANNs optimized with the Huber function is compared with that of ANNs optimized with Least Squares. Comparison is in respect to pricing European call options on the S&P 500 Index from April 1998 to August 2001. In the analysis, a historical parameter, a VIX volatility proxy derived by a weighted implied, and implied parameters (an overall average, and an average per maturity) are used. Simple ANNs (with input supplemented by historical or implied parameters), and hybrid ANNs that in addition use pricing information directly derived by the parametric model are considered. The economic significance of the models is investigated through trading strategies with transaction costs. Instead of *naïve* trading strategies, improved (dynamic and cost-effective) ones are implemented. The use of the robust Huber technique has delivered better ANN structures. The results can be synopsized as follows:

Regarding out of sample pricing, the hybrid models outperform both the standard ANNs and the parametric ones. Huber optimization improves significantly the performance of both the standard and the hybrid ANNs. The non-hybrid ANNs are affected more by large errors, and thus require smaller Huber *k*-value. The overall best models were the Huber based hybrid ANNs. In general, within each class, the best performing Huber model has considerably smaller probability of large mispricing compared to the least squares counterpart. Lye and Martin (1993) identify the importance of the generalized exponential distributions for the error function, in the presence of skewed fat-tailed error distribution. Future work could consider option pricing with robust ANNs that explicitly account for such error distributions. Regarding the economic significance of the models, the Huber models are the overall best models. We have also found that profitable opportunities still exist with single-instrument cost-effective trading strategies and 0.2% proportional transaction costs.

3. Generalizing the Deterministic Volatility Functions for Enhanced Options Pricing

Abstract

We extend the Deterministic Volatility Functions of Dumas et al. (1998) to provide a semi-parametric approach where an enhancement of the implied parameter values is used in the parametric option pricing models. We enhance not only volatility but also skewness and kurtosis. Empirical results using three years of S&P 500 index call option prices strongly support our approach and compares very favorably to stochastic volatility and jump models. The economic significance of the approach is tested in terms of hedging where the evaluation and estimation loss functions are aligned: hedging results when enhancing skewness and kurtosis parameters are significantly improved.

3.1. Introduction

In this essay we price S&P 500 index call options by extending the Deterministic Volatility Functions[14] (DVF) of Dumas et al. (1998) to provide a nonparametric enhancement of the implied parameter values to be used in Parametric Option Pricing Models (POPMs). The proposed method allows us to estimate generalized parameter functions in the sense that not only volatility but other parameters (like skewness and kurtosis) can also be estimated. The resulting semi-parametric models, which we call the enhanced Parametric Option Pricing Models (ePOPMs), outperform (in respect to out of sample pricing) by a large margin the counterpart DVF based parametric ones. With respect to hedging, our results confirm the intuition in Christoffersen and Jacobs (2004) that better out of sample performance can be obtained when optimization is based on a hedging criterion.

The Black and Scholes (1973) (BS) model is an options pricing formula that is built on a set of unrealistic assumptions and exhibits systematic biases like the volatility smile (i.e. Black and Scholes, 1975, Rubinstein, 1994, Bakshi et al., 1997, Bates, 2000). Recent POPMs that incorporate Stochastic Volatility (SV) or Stochastic Volatility and Jump (SVJ) risk factors (e.g. Andersen et al., 2002, Bakshi et al., 1997, Bates, 1991, 1996 and 2000, Heston, 1993, Eraker, 2004), mitigate much of the bias associated with the original BS. A similar effect is achieved indirectly with the Corrado and Su (1996, 1997) (CS) model, an important alternative due to its ease of use[15]. According to Bakshi et al. (1997), SV and SVJ parametric models offer flexible distributional structures with adequate ability to capture negative skewness and excess kurtosis in option market prices. This results to better out of sample pricing performance compared to the simple BS model, with SVJ being superior to SV; yet both models are clearly misspecified (p.g. 2026) with SV

---

[14] The DVF approach relaxes the BS assumption of having a single volatility per day.

[15] The CS model is an extension of BS using a Gram-Charlier (Type A) series expansion that allows for non-normal skewness and kurtosis. Backus et al. (1997) conjecture that the CS formula exhibits good performance for pricing options when the underlying asset follows a jump-diffusion process (see also Jurczenko et al., 1997).

producing implied parameters that can be statistically inconsistent with those implied by historical time series. According to Dumas et al. (1998) and Hull and Suo (2002) both models are difficult to be estimated on a daily basis. BS has shown severe time endurance[16] and is still widely used by practitioners since it generates reasonable prices for a wide spectrum of European financial options.

Dumas et al. (1998) estimate DVF of quadratic forms that provide unique per contract volatility estimates (in contrast to the overall average volatility estimates of Whaley, 1982) and examine how well they predict option prices. This methodological framework is conceptually similar to the one developed with the Space Mapping techniques in Bandler et al. (1994) and Bandler et al. (1999) where several parameter values to an imperfect model are adjusted so as to make the imperfect model (for example a simple POPM) approximate the performance of a finer but more expensive or inaccessible one to use (for example the market prices). Berkowitz (2004) demonstrates theoretically that the DVF constitutes a reduced-form approximation to an unknown structural model which under frequent re-estimation can exhibit exceptional pricing performance. Dumas et al. (1998) conclude by suggesting that the DVF approach should be extended and generalized. Our approach extends DVF by also retaining the intuition in Christoffersen and Jacobs (2004) that while calculating implied parameters optimization should be in respect to the option pricing function.

Researchers have also addressed attention to the use of nonparametric techniques like artificial neural networks that can be used for nonlinear regression. The key power provided by this type of methods is that they rely on fairly simple algorithms and the underlying nonlinearity can be learned from transactions data (see Duda et al., 2001, for further details). Standard applications of artificial neural networks do not involve any financial theory and can be used to estimate directly the empirical options pricing function (thereinafter termed as the standard/traditional neural network approach). Evidence concerning their out of sample pricing performance is mixed.

---

[16] According to Andersen et al., (2002), "*the option pricing formula associated with the Black and Scholes diffusion is routinely used to price European options, although it is known to produce systematic biases*".

Hutchinson et al. (1994) apply them on market transactions of the S&P 500 futures call options from 1987 to 1991 to conclude that although the networks do not constitute a substitute for the more traditional BS formula, they can be more accurate and computationally more efficient alternatives when the underlying asset's price dynamics are unknown. Garcia and Gencay (2000) find that the BS with historical volatility underperforms significantly the standard artificial neural networks. Of course, the application of standard artificial neural networks for pricing of options has also its limitations. First of all, standard neural networks are usually applied in cases where there is lack of knowledge about an adequate functional form; so they are commonly interpreted as "black boxes" since they learn the empirical functions inductively from transactions data without embedding any information related to the problem under scrutiny. Second, standard artificial neural networks are very sensitive to time-varying input variables and this problem is exaggerated in option pricing since key variables, such as (implied) volatility, can be very volatile. Finally, the use of standard neural networks can deliver options prices that violate fundamental financial principles; for instance they might return negative option values or irrational Greek letters (these are the partial derivatives of the option with respect to a parametric model's structural parameters). Herrmann and Narr (1997) show that standard neural networks return negative implied state price densities in regions that available options data is scarce and non informative.

The scope of this essay is to propose a nonparametric enhancement of the parameter values used in the POPMs, generalizing thus Dumas et al. (1998) DVF (see also Christoffersen and Jacobs, 2004). With our approach we estimate Generalized Parameter Functions (GPF) that allow enhancement of parameters beyond volatility without specifying a-priori a deterministic parametric functional form. In our case, the parameter enhancement provides the volatility to the BS and CS models. In addition, skewness or skewness and kurtosis can be enhanced for the CS model. A significant feature of the methodology is that it allows a set of the input variables to the parametric model to be jointly determined by the GPF. Thus, the neural networks are not used in the standard (black box) approach but they incorporate existing theoretical knowledge arising from parametric models. The proposed semi-parametric approach has the following important features. First, it retains the theoretical properties[17] of the parametric model being enhanced concerning the desire for: *i*) nonnegative option values (we thus expect satisfactory pricing performance at the boundary of option pricing areas, in both dense and sparse input areas), *ii*) theory consistent Greek letters, and *iii*) nonnegative implied state price densities. Second, as conjectured by Bandler et al. (1999), semi-parametric techniques that incorporate knowledge regarding the nature of the problem should need a smaller amount of estimation samples and also reduce the number of free parameters needed for estimation to exhibit satisfactory out of sample performance as opposed to the case of standard nonparametric approaches (a similar conjecture is also made by Aït-Sahalia and Lo, 1998, pg. 510). Third, compared to the DVF and Whaley (1982) (see also discussions in Christoffersen and Jacobs, 2004, p. 313) we use long term estimation (twelve months) of the GPF. At the same time though, we capture the time-variation of the option valuation relationship since both in the estimation of the GPF and for the out of sample application we use daily implied parameters. This is in the same spirit with Christoffersen et al. (2007) where they use long term (twelve months) for the estimation of most parameters but with frequent reestimation of implied spot volatilities.

We build ePOPMs for both the BS and the CS model. We compare them with their parametric alternatives using the overall average implied parameters and their DVF versions in pricing S&P 500 index call options. Part of our contribution is to apply the DVF approach to the CS model. Moreover, we include in the comparison the SVJ model of Bates (1996) since it is an effective remedy to the BS biases (see Bakshi et al., 1997, and Bates, 1996) but we also report results for the SV sub-model. We first show that daily calibration of either SVJ or the DVF based BS and CS models requires careful daily parameter search. In the sample, SVJ has the best fit while SV is inferior to the best DVF models. Concerning the out of sample pricing performance we

[17] In the case of Corrado and Su in some extreme regions of skewness and kurtosis the model may give negative option values and/or negative sensitivity of the call to the underlying asset (see also Jondeau and Rockinger, 2001). In our sample this was extremely rare (the worst case was 0.1% of the sample where the negative values were slightly only below zero). Furthermore, our methodology could easily constrain skewness and kurtosis to prohibit such inconsistencies.

find that extending the DVF approach to the case of CS can significantly improve the model's performance, yet SVJ dominates in the family of parametric models. We find that all semi-parametric models (ePOPMs) have excellent performance and outperform their DVF parametric counterparts. The enhanced CS model is the overall best ePOPM and is competitive in performance to SVJ since it is found to have a rich distributional flexibility in generating skewness and kurtosis patterns across time to maturity and strike prices. Our results show that ePOPMs exhibit superior out of sample robustness, and the enhanced models can significantly outperform SVJ in moneyness regions not used in estimation. The hedging performance of all models is in line with previous literature when models are optimized with a pricing criterion. Better out of sample results are obtained when optimization is based on a hedging criterion, where the resulting enhanced parameters differ significantly from those used in pricing. Parameters enhanced for hedging exhibit positive skewness and high kurtosis hedging against the prospect of extreme (negative) returns.

In the following we review the parametric models and we explain the implementation of the ePOPM structure via the GPF. We then discuss the data, filtering and the alternative versions of the models under comparison. Finally we discuss the pricing results, we provide various pricing robustness checks, we implement a single instrument hedging strategy for the best models considered and then we conclude. The Appendix shows the necessary Greeks for the POPMs used during calibration and hedging.

3.2. Parametric models used

Below we briefly discuss the different POPMs we employ in this study. The first model examined is the Black and Scholes (1973) since it is a benchmark and widely referenced model. The BS formula for European call options modified for dividend-paying (see also Merton, 1973) underlying asset is:

$$c^{BS} = Se^{d_y T} N(d) - Xe^{-rT} N(d - \sigma\sqrt{T}) \qquad (3.1)$$

$$d = \frac{\ln(S/X) + (r - d_y)T + (\sigma\sqrt{T})^2/2}{\sigma\sqrt{\mathrm{T}}} \qquad (3.1.1)$$

where c^{BS} is the price of the European call option, S is the spot price of the underlying asset, X is the exercise price of the call option, r is the continuously compounded risk free interest rate, d_y is the continuous dividend yield paid by the underlying asset, T is the time left until the option expiration date, σ^2 is the yearly variance of the rate of return for the underlying asset and $N(.)$ stands for the standard normal cumulative distribution. Vega, which is the partial derivative of the BS call options with respect to the volatility will be necessary for our application of the ePOPMs:

$$\frac{\partial c^{BS}}{\partial \sigma} = Se^{-d_y T}\sqrt{T}n(d) \qquad (3.1.2)$$

In addition for hedging purposes the BS delta value is also needed:

$$\frac{\partial c^{BS}}{\partial S} = e^{-d_y T}N(d) \qquad (3.1.3)$$

The abundant empirical evidence regarding the *smile/smirk* behavior of the BS implied volatility is indicative of implied return distributions that are negatively skewed with higher kurtosis than what the BS log-normal distribution allows (see Bakshi et al., 1997 and Bates, 2000). For this reason we include in the parametric analysis more general option pricing models. We use the Corrado and Su (1996) model which is an extension of the BS model able to capture non-normal skewness and kurtosis for the underlying returns' distribution. Corrado and Su derived an extension of the BS model based on a methodology employed earlier by Jarrow and Rudd (1982). Using a Gram-Charlier series expansion of a normal density function they define their model as (see also the correction in Brown and Robinson, 2002; for further discussions see Jondeau and Rockinger, 2001, and Jurczenko et al., 2004):

$$c^{CS} = c^{BS} + \mu_3 Q_3 + (\mu_4 - 3)Q_4 \qquad (3.2)$$

where c^{BS} is the BS value for the European call option given in Eq. (3.1) and,

$$Q_3 = \frac{1}{3!} Se^{-d_y T} \sigma\sqrt{T}((2\sigma\sqrt{T} - d)n(d) + (\sigma\sqrt{T})^2 N(d)) \qquad (3.2.1)$$

$$Q_4 = \frac{1}{4!} Se^{-d_y T} \sigma\sqrt{T}((d^2 - 1 - 3\sigma\sqrt{T}(d - \sigma\sqrt{T}))n(d) + (\sigma\sqrt{T})^3 N(d)) \qquad (3.2.2)$$

In Eq. (3.2) Q_3 and Q_4 represent the marginal effect of non-normal skewness and kurtosis respectively in the option price whereas μ_3 and μ_4 correspond to coefficients of skewness and kurtosis. In the above expressions,

$$n(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2 / 2) \qquad (3.2.3)$$

refers to the standard normal probability density function. The following partial derivatves (Greek letters) are necessary for our application of ePOPMs. The CS Vega is given by:

$$\frac{\partial c^{CS}}{\partial \sigma} = \frac{\partial c^{BS}}{\partial \sigma} + e^{-d_y T}\left(\frac{1}{3!}\mu_3 \frac{\partial Q_3}{\partial \sigma} + \frac{1}{4!}(\mu_4 - 3)\frac{\partial Q_4}{\partial \sigma}\right) \qquad (3.2.4)$$

where,

$$\frac{\partial Q_3}{\partial \sigma} = S\,n(d)\sigma\sqrt{T}\left[3d\sigma\sqrt{T} + 3d^2 + \sigma\sqrt{T} + 3\right] - S\sqrt{T}d^3\,n(d) + 3S\sigma^2 T^{3/2} N(d) \qquad (3.2.5)$$

$$\frac{\partial Q_4}{\partial \sigma} = S\sqrt{T}\,d\,n(d)\left[-2d + d^3 - 4\sigma\sqrt{T}\,d^2 + 6d\left(\sigma\sqrt{T}\right)^2 - 4\left(\sigma\sqrt{T}\right)^3\right] - S\sqrt{T}\,n(d) + 6S\sigma^2 T^{3/2}\,n(d) + 4S\sigma^3 T^2 N(d) + S\,n(d)\sigma^4 T^{3/2} \tag{3.2.6}$$

The CS partial derivative of call with respect to skewness is given by:

$$\frac{\partial c^{CS}}{\partial \mu_3} = Q_3 \tag{3.2.7}$$

The CS partial derivative of call with respect to kurtosis is given by:

$$\frac{\partial c^{CS}}{\partial \mu 4} = Q_4 \tag{3.2.8}$$

In addition, for hedging purposes the CS delta value is also needed:

$$\frac{\partial c^{CS}}{\partial S} = e^{-d_y T}\left(\begin{array}{l} N(d) + \frac{1}{3!}\mu_3\left(\left(\sigma\sqrt{T}\right)^3 N(d) + \left(3\left(\sigma\sqrt{T}\right)^2 - 3d\sigma\sqrt{T} + d^2 - 1\right)n(d)\right) + \\ \frac{1}{4!}(\mu_4 - 3)\left(\begin{array}{l} \left(\sigma\sqrt{T}\right)^4 N(d) + 4\left(\sigma\sqrt{T}\right)^3 n(d) - 6d\left(\sigma\sqrt{T}\right)^2 n(d) + 4d^2\sigma\sqrt{T}n(d) + \\ 3dn(d) - 4\sigma\sqrt{T}n(d) - d^3 n(d) \end{array}\right) \end{array}\right)$$

(3.2.9)

Motivated by empirical evidence (Bakshi et al., 1997, Das and Sundaram, 1999, Bates 2000), and unlike Christoffersen and Jacobs (2004) that concentrate on the SV model, we consider SVJ as the benchmark model (but we also report results for SV). Bakshi et al. (1997) found that the SVJ exhibited better out of sample pricing performance for the S&P 500 index options when compared to the SV and BS models. Here we must note that SV and SVJ models are not widely used by traders for pricing options (see Hull and Suo, 2002, p. 300). Traders usually rely on simpler models and more intuitive methodologies that are closer to DVF (see also Brandt and Wu, 2002). We employ the SVJ model of Bates (1996) where the instantaneous conditional variance V_t follows a mean-reverting square root process:

$$\frac{dS}{S} = (\mu - \lambda\bar{\kappa})dt + \sqrt{V}dZ + \kappa dq \quad (3.3)$$

$$dV = (\alpha - \beta V)dt + \sigma_v \sqrt{V} dZ_v \quad (3.4)$$

with

$$\text{cov}(dZ, dZ_v) = \rho dt$$

$$\ln(1+\kappa) \sim N(\ln(1+\bar{\kappa}) - 0.5\theta^2, \theta^2)$$

$$prob(dq = 1) = \lambda dt$$

Here μ is the instantaneous drift of the underlying asset, λ is the annual frequency of jumps, κ is the random percentage jump conditional on a jump occurring, q is a Poison counter with intensity λ, θ^2 is the jump variance, and ρ is the correlation coefficient between the volatility shocks and the underlying asset movements. Moreover, β is the rate of mean reversion and α / β is the variance steady-state level (long run mean). In the above diffusion specification the correlation between the volatility and the returns of the underlying asset controls the level of skewness whilst the variability of volatility allows for non-normal kurtosis. Moreover, the addition of a jump component enhances the distributional flexibility and allows for more accurate pricing performance especially for the short term options.

The value of the European call option is given as a function of state variables and parameters:

$$c^{SVJ} = e^{-rT}[F\Pi_1 - X\Pi_2] \quad (3.5)$$

with $F = E(S_T) = Se^{(r-d_y)T}$ the forward price of the underlying asset, $E(.)$ the expectation with respect to the risk-neutral probability measure and S_T the price of S at option's maturity. Evaluation of the probabilities Π_1 and Π_2 is done by using the moment generating functions of $\ln(S_T / S)$. The following expressions are needed to compute Π_1 and Π_2:

$$F_j(\Phi \,|V,T) = \exp\{C_j(T;\Phi) + D_j(T;\Phi)V + \lambda T(1+\bar{\kappa})^{\mu_j+0.5} \times [(1+\bar{\kappa})^{\Phi} e^{\theta^2(\mu_j\Phi+\Phi^2/2)} - 1]\} \qquad j = 1, 2 \tag{3.6}$$

$$C_j(T;\Phi) = (r - d_y - \lambda\bar{\kappa})\Phi T - \frac{\alpha T}{\sigma_v^2}(\rho\sigma_v\Phi - B_j - G_j) - \frac{2\alpha}{\sigma_v^2}\ln\left[1 + 0.5(\rho\sigma_v\Phi - B_j - G_j)\frac{1-e^{G_jT}}{G_j}\right] \tag{3.7}$$

$$D_j(T;\Phi) = -2\frac{\mu_j\Phi + 0.5\Phi^2}{\rho\sigma_v\Phi - B_j + G_j\dfrac{1+e^{G_jT}}{1-e^{G_jT}}} \tag{3.7.1}$$

$$G_j = \sqrt{(\rho\sigma_v\Phi - B_j)^2 - 2\sigma_v^2(\mu_j\Phi + 0.5\Phi^2)} \tag{3.7.2}$$

$$\mu_1 = 0.5\,, \quad \mu_2 = -0.5\,, \quad B_1 = \beta - \rho\sigma_v\,, \quad B_2 = \beta \tag{3.7.3}$$

and Π_1 and Π_2 are derived by numerically evaluating the imaginary part of the Fourier inversion:

$$prob(S_T e^{(r-d_y)T} > X \mid F_j) = 0.5 + \frac{1}{\pi}\int_0^\infty \frac{imag[F_j(i\Phi)e^{-i\Phi\chi}]}{\Phi}d\Phi \tag{3.8}$$

with $\chi \equiv \ln(X/S)$ and the integrals to be evaluated with an adaptive Lobatto quadrature. By constraining the jump component values equal to zero we get European call prices for the SV model.

In this work, we fit the POPMs to obtain daily the implied parameters that minimize the sum of squared pricing deviations from daily market prices, so these (risk-neutral) parameters indirectly account for any pricing of jump and volatility risk. The proposed methodology should be compared to the DVF based BS and CS alternatives[18], but for completeness we also provide results for SVJ.

3.2.1. Deterministic volatility functions for BS and CS

According to Dumas et al. (1998), this approach of smoothing the BS implied volatilities across strike prices and maturities exhibits superior in and out of sample performance for pricing European options. For our analysis we estimate the three different DVF models:

DVF#1: $\sigma = \max(0.01, a_0 + a_1 X + a_2 X^2)$

DVF#2: $\sigma = \max(0.01, a_0 + a_1 X + a_2 X^2 + a_3 T + a_4 XT)$

DVF#3: $\sigma = \max(0.01, a_0 + a_1 X + a_2 X^2 + a_3 T + a_4 XT + a_5 T^2)$

As in Dumas et al. (1998), parameters can be estimated using either Ordinary Least Squares (OLS) where the loss function is the difference between the estimated volatility and the contract specific implied volatility or Nonlinear Least Squares where the loss function is the difference between the estimated and the actual option price. Aït-Sahalia and Lo (1998, Eq. 12 in pg. 511)

---

[18] Christoffersen and Jacobs (2004) conclude that the DVF based BS model, which does not require additional assumptions about investors' preferences for risk, represents a new and tougher benchmark against which the performance of future structural models can be measured.

examine a semi-parametric[19] approach where they use the BS volatility loss function but estimation is through a nonparametric kernel regression instead of OLS. Christoffersen and Jacobs (2004) demonstrate that the OLS estimates of the DVF parameters yield biased option pricing and that a price loss function should be used.

In this essay we implement the DVF not only for BS but also for the first time for the CS model, using both loss functions. For CS this is done in two steps. We first fit daily the CS model to market option prices to obtain overall average implied parameters values (similarly to the Whaley, 1982 method). Then we fix the skewness and kurtosis values to those obtained earlier (in contrast to the BS where these parameters are always fixed to the values of zero and three respectively) and further calibrate the model's volatility parameter in order to obtain a daily contract specific implied volatility value. Subsequently, for both BS and CS, we estimate the coefficients for the three different DVF models each day using OLS (*Lc*) and also using Nonlinear Least Squares (*NLc*). For the latter we use several initializations to minimize the risk of estimating coefficients based on a local minimum of the optimization function.

3.3. ePOPM structure

In order to estimate the enhanced parameters nonparametrically we employ artificial neural networks. They are universal function approximators with good out of sample generalization abilities (see Cybenko, 1989; for a general discussion of neural networks in financial econometrics see Tsay, 2002). An artificial neural network is a collection of interconnected simple processing elements structured in successive layers and can be depicted as a network of links (termed as *synapses*) and nodes (termed as *neurons*) between layers. A typical feedforward neural network has an input layer, one or more

[19] Our approach (and similarly DVF and Aït-Sahalia and Lo, 1998) should be considered as semi-parametric since a parametric option pricing model is involved in the process but estimation implies deviations from the theoretical model, when for example volatility is assumed to be a function of moneyness, etc.

hidden layers and an output layer. Each interconnection corresponds to a modifiable weight, which is adjusted according to the faced problem via optimization (the training algorithm).

Figure 3.1 depicts the general idea of the ePOPM structure we propose while Figure 3.2 depicts the exact network structure developed for the purposes of this study. For our analysis, inputs are set up in feature vectors, $\tilde{x}_p = [x_{1p}, x_{2p} \ldots, x_{Np}]$ for which there is an associated and known target characterizing our problem, t_p, $p = 1,2,\ldots,P$, where P is the number of the available sample feature vectors for a particular estimation sample and N the number of input variables. The network's outputs are obtained when the data are presented to the input layer and after evaluating the signals at each node. To let the network learn the underlying relationship, its weights are adjusted in order to minimize a loss function of the error between the network output and the desired target values.

The proposed network model has four layers. The first three are typical layers of a feedforward artificial neural network: an *input* layer with N input variables, a *hidden* layer with H neurons, and a layer with M *output* neurons. For these three layers, each node is connected with all neurons in the previous and the forward layer. Each connection is associated with a *weight*, $w_{in}^{(1)}$, and a *bias*, $w_{i0}^{(1)}$, in the input layer (*i=1,2,…,H, n=1,2,…N*) and a *weight*, $w_{ji}^{(2)}$, and a *bias*, $w_{j0}^{(2)}$, in the hidden layer (*j=1,2,…,M*). Each neuron behaves as a summing vessel that computes the weighted sum of its inputs to form a scalar term and with the use of the transfer/activation function it eventually works as a non-linear mapping junction for the forward layer. The part of the network that is outside the bold-dotted line in Figure 3.2 is a typical three-layer feedforward artificial neural network with a single output that under proper treatment can be used for nonlinear regression (Hutchinson et al., 1994 discuss the approach for option pricing).

The fourth layer, which hereafter will be termed as the *enhanced layer*, allows a certain POPM to be part of the network's structure. In this setting the network structure embeds knowledge from the parametric model during estimation (resulting thus to a semi-parametric options pricing methodology). If we let $X_I = X_E \cup X_D$ denote the set of all input parameters that are necessary for the parametric model, then (see Figure 3.1) $X_E \subseteq X_I$ corresponds to the *enhanced* parameters provided non-parametrically and $X_D \subset X_I$ those that are passed to the parametric model directly. The enhanced parameter set X_E is basically a choice of the researcher and effectively manifests the number of neurons at the output layer and the type of activation function to be used at the enhanced layer. In addition, X_S represents the set of inputs to the nonparametric model.

According to Figure 3.2, the operation carried out for computing the final estimated output, y, is the following:

$$y = f_{PM}(\upsilon, X_D) \tag{3.9}$$

and,

$$\upsilon = [f_{d_1}(d_1), f_{d_2}(d_2), \ldots, f_{d_M}(d_M)] \tag{3.10}$$

where $f_{PM}(.,.)$ refers to the functional form of the parametric options pricing model, $f_{dj}(.)$ are smooth monotonically increasing activation functions and d_j are simply the descaled values of $y_j^{(2)}$, where $j = 1, 2, \ldots, M$. We use the term *enhanced* for the υ_j variables (see enhanced layer in Figure 3.2) that are determined by the *Generalized Parameter Function* (GPF) mechanism. In our application, $\upsilon \equiv X_E$.

The elements of vector $[y_1^{(2)} \ldots y_M^{(2)}]^T$ follow the functional form of a typical three-layer network:

$$\begin{aligned}
y_1^{(2)} &= f_M[w_{10}^{(2)} y_0 + \sum_{i=1}^{H} w_{1i}^{(2)} f_H(w_{i0}^{(1)} x_{s0} + \sum_{n=1}^{N} w_{in}^{(1)} x_{sn})] \\
\vdots \quad &\vdots \qquad\qquad\qquad \vdots \\
y_M^{(2)} &= f_M[w_{M0}^{(2)} y_0 + \sum_{i=1}^{H} w_{Mi}^{(2)} f_H(w_{i0}^{(1)} x_{s0} + \sum_{n=1}^{N} w_{in}^{(1)} x_{sn})]
\end{aligned} \tag{3.11}$$

where $f_M(.)$ and $f_H(.)$ are smooth monotonically increasing activation functions associated with the output and hidden layer respectively and x_{sn} (represented by X_S in Figure 3.1), n=1,2,...,N, is just the scaled value of the input x_n. The M elements of Eq. (3.11) are estimated simultaneously using information propagated by the POPMs. The vector defined by the right hand side of Eq. (3.11) is the GPF which (with the appropriate descaling and the transformation depicted in Table 3.1 Panel A) produces the enhanced variables.

As shown in Figure 3.2, the proposed network structure can accommodate a scaling scheme for both the inputs and the enhanced variables. This can be essential since it increases the effectiveness of the optimization algorithm and minimizes the significance of differing dimensions of the input/output signals (see Haykin, 1999). In the current study we choose to apply a standard *z-score* scaling for the input signals: $\tilde{z} = (\tilde{x} - m)/s$, where $\tilde{x}$ is the vector of an input, m is the mean and s the standard deviation of this vector.

In our case, the smooth monotonically increasing activation functions are among the hyperbolic tangent sigmoid,

$$f(\eta) = \alpha \left[\frac{e^{b\eta} - e^{-b\eta}}{e^{b\eta} + e^{-b\eta}} \right] \tag{3.12}$$

the logistic,

$$f(\eta) = \frac{\alpha}{1 + e^{-b\eta}} \tag{3.13}$$

or the linear one,

$$f(\eta) = \eta \tag{3.14}$$

In the above expressions, with $a, b \in \Re$, a controls the output range and b the slope of the activation function[20]. In the hidden layer we always use the standard hyperbolic tangent sigmoid activation function (with a and b equal unity) for $f_H(.)$, while in the output layer we use a linear activation function for $f_M(.)$. The choice of the activation function at the enhanced layer is dictated by the type of the parametric model we use and the kind of enhanced variable(s) we choose; thus it is possible for $f_{d_1}(.)$, $f_{d_2}(.), \ldots, f_{d_M}(.)$ to be different depending on the case considered. This set of activation functions are necessary during the implementation of the method in order to ensure that the values of each of the enhanced variables are within an acceptable range for use with the parametric model[21]. Table 3.1 (Panel A) describes the different activation functions we use at the enhanced level for all cases considered. We use activation functions that truncate the enhanced variable value range. For instance in the case of BS and CS we do not allow volatility to be larger than 70%, and for the case of CS, skewness is confined in the $[-15,15]$ range and kurtosis is set to smaller than 30. The choice of the truncation point is not crucial for the implementation of the models as long as we allow each enhanced variable to vary within a plausible value range. This choice can be guided by empirical investigation. For example we rarely observe volatility to be above 70% or skewness to be below -15 or above 15 and kurtosis to be above 30 (e.g. Christoffersen and Jacobs, 2004, Ait-Sahalia and Lo, 1998, Corrado and Su, 1997, Bates, 1991).

---

[20] As in Duda et al. (2001, pg. 308), the overall range and slope are not important because it is their relationship to parameters such as the learning rate and magnitudes of the inputs and targets that affect learning. According to Haykin (1999) these transfer functions work well with feedforward artificial neural networks.

[21] For instance, if BS is the chosen parametric model and volatility is the enhanced variable, then our activation function should be a logistic that allows only positive values whilst if the enhanced variable is the skewness of CS then the activation function should be a hyperbolic tangent one that allows both positive and negative values.

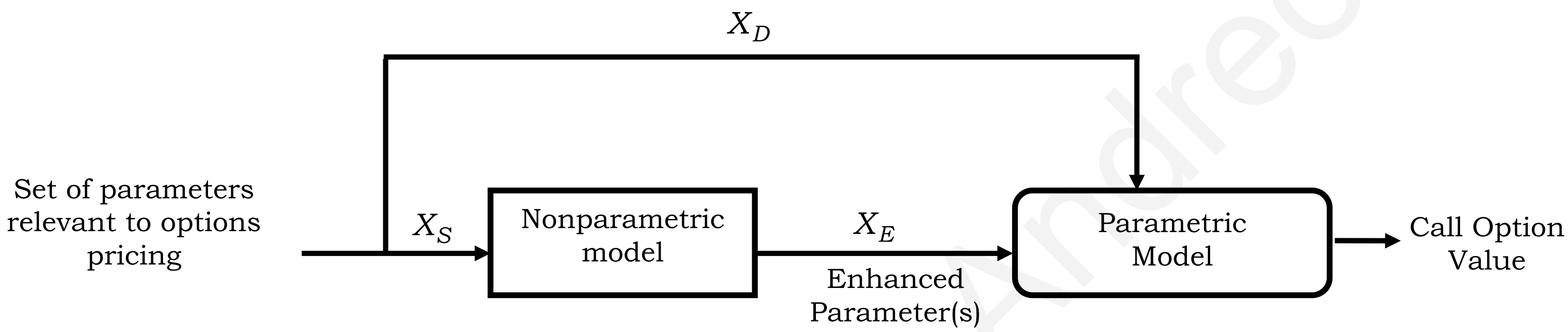

Figure 3.1. Schematic description of the enhanced models (ePOPMs)

In the proposed semi-parametric methodology the call option value is provided by a parametric model. Let $X_I = X_E \cup X_D$ denote the set of all input parameters that are necessary for the parametric model. $X_E \subseteq X_I$ corresponds to the enhanced parameters estimated nonparametrically and $X_D \subset X_I$ those that are passed directly to the parametric model. In addition, X_S denotes the set of inputs to the nonparametric model.

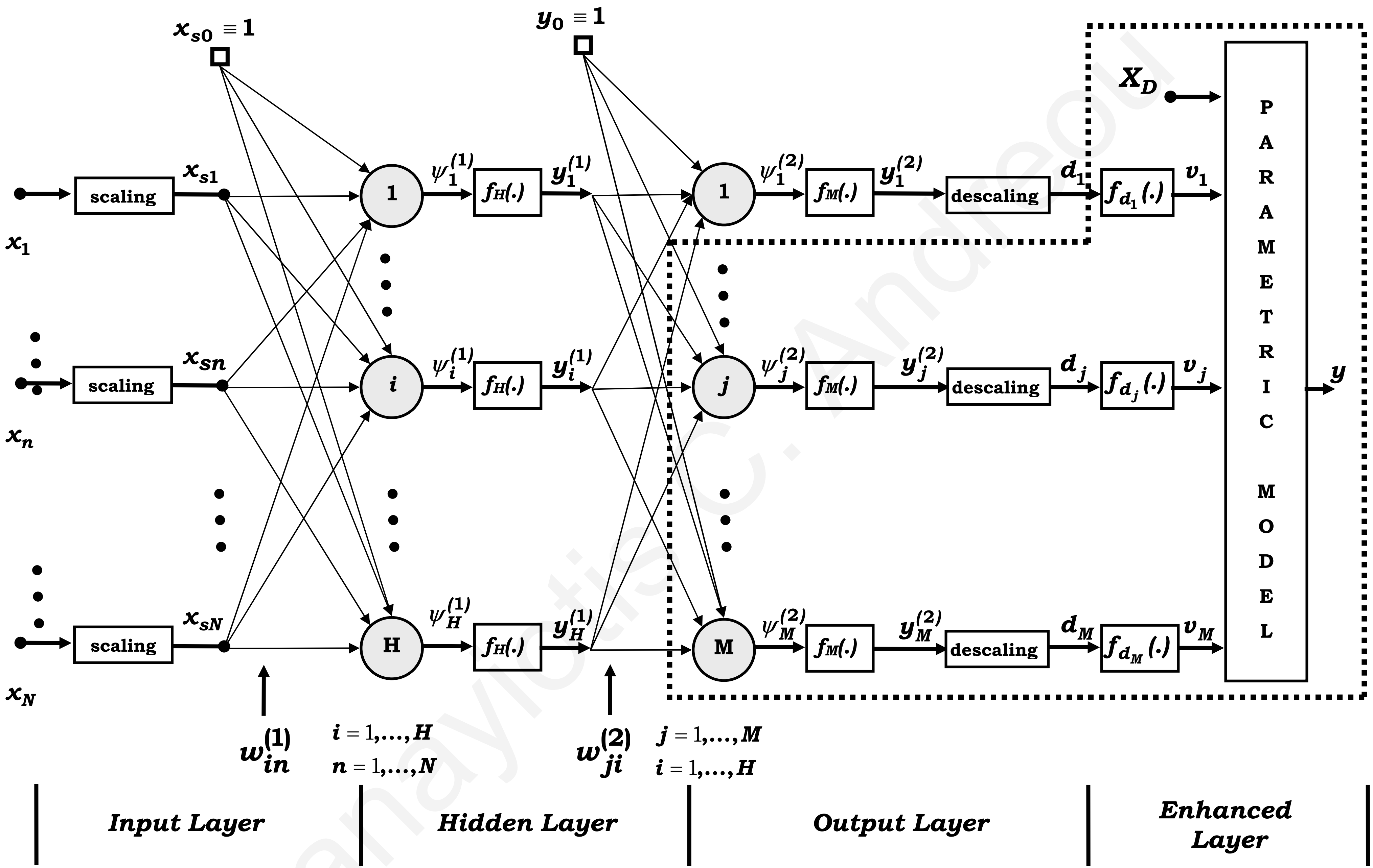

Figure 3.2. A detailed diagram of the enhanced models (ePOPMs)

The estimation of any type of network model is formulated as a highly non-linear optimization process in which the network's weights are modified according to a loss function. The loss function (discrepancy between the estimated response y_p and the actual response t_p) is defined as:

$$e_p(w) = y_p(w) - t_p \tag{3.15}$$

where w is an ν-dimensional column vector with the weights and biases given by:

$$w = [w_{10}^{(1)},\ldots,w_{1N}^{(1)},\ldots,w_{H0}^{(1)},\ldots,w_{HN}^{(1)},w_{10}^{(2)},\ldots,w_{1H}^{(2)},\ldots,w_{M0}^{(2)},\ldots,w_{MH}^{(2)}]^T .$$

Model	**Enhanced Variable**	**Activation Function**	**Parameter Values** $(a,\ b)$
BS	Volatility	Logistic	(0.70,1)
CS	Volatility	Logistic	(0.70,1)
CS	Skewness	Tangent	(15,0.15)
CS	Kurtosis	Logistic	(30,0.15)

Panel A: *Activation functions used with enhanced variables*

Model	**Input Variables to GPF** (X_S)	**Enhanced Variables** (X_E)
eBS_{av}	$(Se^{-d_yT})/X,\ T,\ \sigma_{av}^{BS}$	Volatility
eBS_{NL2}	$(Se^{-d_yT})/X,\ T,\ \sigma_{NL2}^{BS}$	
eCS_{av}^1	$(Se^{-d_yT})/X,\ T,\ \sigma_{av}^{CS}$	Volatility
eCS_{NL2}^1	$(Se^{-d_yT})/X,\ T,\ \sigma_{NL2}^{CS}$	
eCS_{av}^2	$(Se^{-d_yT})/X,\ T,\ \sigma_{av}^{CS},\ \mu_3$	Volatility and skewness
eCS_{NL2}^2	$(Se^{-d_yT})/X,\ T,\ \sigma_{NL2}^{CS},\ \mu_3$	
eCS_{av}^3	$(Se^{-d_yT})/X,\ T,\ \sigma_{av}^{CS},\ \mu_3,\ \mu_4$	Volatility, skewness and kurtosis
eCS_{NL2}^3	$(Se^{-d_yT})/X,\ T,\ \sigma_{NL2}^{CS},\ \mu_3,\ \mu_4$	

Panel B: *Description of all enhanced parametric models (ePOPMs)*

Table 3.1. Structure characteristics for the Enhanced Parametric Option Pricing Models (ePOPMs)

The traditional backpropagation algorithm which is based on the gradient descent vector is the most popular method for estimating feedforward artificial neural networks. It is shown in Charalambous (1992) that this algorithm is often unable to converge rapidly to the optimal solution. So, in this essay we rely on the Levenberg-Marquardt algorithm which is much more efficient estimation method in terms of time and convergence rate. The weights and the biases of the network are updated in such a way so as to minimize the following sum of squares error performance function[22]:

$$F(w) = \sum_{p=1}^{P} e_p^2(w) \equiv \sum_{p=1}^{P} (y_p - t_p)^2 \qquad (3.16)$$

Then, at each iteration τ of the algorithm, the weights vector w is updated as follows:

$$w_{\tau+1} = w_\tau + \left[J^T(w_\tau)J(w_\tau) + l_\tau I\right]^{-1} J^T(w_\tau)e(w_\tau) \qquad (3.17)$$

where, $J(w_\tau)$ is the $P \times \nu$ Jacobian matrix of the *P-dimensional* output error column vector at τ^{th} iteration, and is given by:

$$J(w) = \begin{bmatrix} \nabla e_1^T(w) \\ \vdots \\ \nabla e_P^T(w) \end{bmatrix} \qquad (3.18)$$

In the above, I is $\nu \times \nu$ identity matrix, and l_τ is a learning parameter that is automatically adjusted at each iteration in order to secure convergence. Large values of l_τ lead to directions that approximate the steepest descent, while small values lead to directions that approximate the Gauss-Newton algorithm.

---

[22] The use of Sum of Squared Errors (SSE) is common in empirical option pricing studies since Whaley (1982) and is supported by Christoffersen and Jacobs (2004).

Further technical details about the implementation of the Levenberg-Marquardt algorithm can be found in Hagan and Menhaj (1994) and Hagan et al. (1996). Based on Eq. (3.17), the weights and biases update takes place in a batch mode and only when all input vectors have been presented to the network.

The quantity $\nabla e_p(w)$ is the gradient vector of $e_p(w)$ with respect to the optimized parameter vector w. The partial derivative of the error function in Eq. (3.15) with respect to the weight $w_{ji}^{(2)}$ at the hidden layer is:

$$\frac{\partial e_p}{\partial w_{ji}^{(2)}} = \delta_j^{(2)} y_i^{(2)} \tag{3.19}$$

and,

$$\delta_j^{(2)} = \frac{\partial f_{PM}}{\partial v_j} f'_{d_j}(d_j) s_j f'_M(\psi_j^{(2)}) \tag{3.20}$$

where $f'_M(\psi_j^{(2)})$ and $f'_{d_j}(d_j)$ are the differentials at points $\psi_j^{(2)}$ and d_j respectively, and s_j the standard deviation of the enhanced variable given that a *z*-score scaling has also been applied at the enhanced layer.

The quantity $\frac{\partial f_{PM}}{\partial v_j}$ is the partial derivative of the parametric model with respect to input v_j creating a semi-parametric method dedicated to pricing European call options. This quantity is very important during the estimation because it incorporates theoretical knowledge from a parametric model. The partial derivative of the error function in Eq. (3.15) with respect to the weight $w_{in}^{(1)}$ at the input layer is:

$$\frac{\partial e_p}{\partial w_{in}^{(1)}} = \delta_i^{(1)} x_{sn} \tag{3.21}$$

where,

$$\delta_i^{(1)} = \varepsilon_i^{(1)} f'_H(\psi_i^{(1)}) \tag{3.22}$$

$$\varepsilon_i^{(1)} = \sum_{j=1}^{M} w_{ji}^{(2)} \delta_j^{(2)} \tag{3.23}$$

and x_{sn} is simply the *z*-score scaled value of x_n.

The optimal number of hidden neurons is chosen via a cross-validation procedure. All ePOPM structures with 1 to 10 hidden neurons are estimated, and the one that performs the best in the validation period is selected. The model is initialized, estimated and cross-validated with twenty different initializations (trying thirty initializations did not improve results). We employ the network initialization technique proposed by Nguyen and Windrow (see Hagan et al., 1996) that generates initial weights and bias values for a nonlinear activation function so that the active regions of the layer's neurons are distributed roughly evenly over the input space. After defining the optimal network structure, its weights are frozen and its pricing capability is tested (out of sample) in a third separate testing dataset.

3.4. Data and methodology

Our dataset covers the period January 2002 to August 2004 for a total of 671 trading days. The S&P 500 index call options are used because this option market is extremely liquid. They are the most popular index options traded in the CBOE and the closest to the theoretical setting of the parametric models (see Garcia and Gencay, 2000). For each trading day we have the last bid and ask call price for all available contracts, along with the strike price[23], X, date of expiration[24], volume and open interest. We have collected a daily dividend yield[25], d_y, provided online by Datastream. In our analysis we use the midpoint of the call option bid-ask spread since as noted by Dumas et al. (1998), using bid-ask midpoints rather that trade prices reduces noise in the cross sectional estimation of implied parameters. Each day the midpoint of the call option bid ask spread at the close of the market, c^{mrk}, is matched with the closing value of S&P 500 index[26].

We used a chronological data partitioning via a rolling-forward procedure in order to have a better simulation of the actual options trading conditions. The data is divided into eighteen different *overlapping* training/estimation (*trn*) and validation (*vld*) sets, each followed by separate and *non-overlapping* testing (*tst*) set. Each *trn*, *vld* and *tst* period has 12, 2 and 1 month spanning period respectively[27,28]. For instance, the first *trn* set covers

---

[23] For the purposes of this study we use the following moneyness categories: *deep out the money* (DOTM) when $S/X \leq 0.90$, *out the money* (OTM) when $0.90 < S/X \leq 0.95$, *just out the money* (JOTM) when $0.95 < S/X \leq 0.99$, *at the money* (ATM) when $0.99 < S/X \leq 1.01$, *just in the money* (JITM) when $1.01 < S/X \leq 1.05$, *in the money* (ITM) when $1.05 < S/X \leq 1.10$, *deep in the money* (DITM) when $S/X > 1.10$.

[24] In terms of time length, an option contract is classified as *short term maturity* (when maturity ≤ 60 calendar days), as *medium term maturity* (when maturity is between 61 and 180 calendar days) and as *long term maturity* (when maturity > 180 calendar days).

[25] Jackwerth (2000) also assumes that the present value of expected future dividends for the S&P 500 index can be approximated by a dividend yield. In addition, Chernov and Ghysels (2000) use a constant dividend yield for the whole period they examine.

[26] Data synchronicity is a minimal issue for this highly active market (see also Garcia and Gencay, 2000, and Ait-Sahalia and Lo, 1998). Among others, Christoffersen and Jacobs (2004) and Chernov and Ghysels (2000) use daily closing prices of European call options written on the S&P 500 index.

[27] In contrast to the implied trees methodology and the daily calibrated DVF models where data from a single day are used, using a training set of twelve months should alleviate overfitting concerns.

[28] Keeping the model's weights the same for one month period is consistent with the reasoning of Bates (2000, pg. 184). Daily recalibration of the weights would imply that

the period January to December 2002, the first *vld* set covers the period January to February 2003, the first *tst* set covers the period March 2003, etc. The eighteen testing (out of sample) monthly periods are non-overlapping. For the needs of the analysis, we created (after the use of filtering rules explained below) an aggregate testing period (*AggTs*) with 21644 data points by simply pooling together the pricing estimates of all eighteen *tst* periods. For *AggTs* we compute and tabulate: the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the Median Absolute Error (MdAE) and the 5th Percentile of Absolute Error (P_5AE) and 95th Percentile of Absolute Error (P_{95}AE). The main analysis is based on the RMSE measure. As pointed by Christoffersen and Jacobs (2004) estimation and evaluation of a model should be based on the same error measure. In addition, they conclude that RMSE estimates perform the best among different loss functions. Finally, Bates (2000, p. 202) points out that the RMSE is a relatively intuitive error measure and is useful for comparison purposes.

3.4.1. Observed structural parameters

The moneyness ratio, *S/X*, is a common input to non-parametric models since it is highly related to the pricing bias associated with the POPMs (see Hutchinson et al., 1994, and Garcia and Gencay, 2000). The dividend adjusted moneyness ratio $(Se^{-d_y T})/X$ is preferred here since dividends are relevant. Finally, time to maturity (T) is computed assuming 252 days per year. Previous studies have used 90-day T-bill rates as an approximation of the interest rate. In this study we use nonlinear cubic spline interpolation for matching each option contract with a continuous interest rate, r, that corresponds to the option's maturity. For this purpose, 1, 3, 6, and 12 months constant maturity T-bills rates (collected daily from the U.S. Federal Reserve Bank Statistical Releases) were considered.

---

the enhanced models are never to be taken seriously as a genuine data generating mechanism.

3.4.2. Data and filtering rules

To create our dataset we rely on the following filtering rules (see also Bakshi et al., 1997): We first eliminate all observations that have zero trading volume since they do not represent actual trades. Second, we eliminate observations that violate either the lower or the upper arbitrage options bounds. Third, we eliminate all options with less than six or more than 260 days to expiration to avoid extreme option prices where an illiquidity problem[29] may be present. Similarly, option price quotes of less than 1.0 index points are not included. Finally, we demand at least four datapoints per maturity to secure that during the implied parameters extraction process, every maturity period is satisfactorily represented. The final dataset has a total of 37202 observations (from which 21644 are used out of sample) and compares favourably with previous studies that test nonparametric methods. For instance Hutchinson et al. (1994) have an average of 6246 data points per sub-period; Aït-Sahalia and Lo (1988) have a total of 14431 data points; Schittenkopf and Dorffner (2001) include a total of 33633 data points. Sample characteristics for the dataset can be found in Table 3.2 where the average implied parameters are also reported (see explanations in section 5). The volatility anomaly is obvious both for the BS and the CS model.

3.4.3. Implied parameters

The methodology employed here for the extraction of daily overall average implied parameters is similar to that in previous studies (Bates, 1991, Bakshi et al., 1997, Christoffersen et al., 2006) that adopt the Whaley's (1982) simultaneous equation procedure to minimize a price deviation function with respect to the unobserved parameters[30]. Market option prices (c^{mrk}) are

---

[29] Dumas et al. (1998) drop observations with more than 100 days, and Bates (2000) and Christoffersen et al. (2006) choose to drop observations with more than 180 days. We choose to keep them since these options are not necessarily illiquid and comprise a significant part of the total number of observations.

[30] As noted by Das and Sundaram (1999), Chernov and Ghysels (2000) and Christoffersen et al. (2006) for the purpose of option valuation, parameters estimated

assumed to be the corresponding POPM prices (c^k) plus a random additive disturbance term (ε^k), k = BS, CS, or SVJ:

$$c^{mrk} = c^k + \varepsilon^k \tag{3.24}$$

	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM
S/X	**<0.90**	**0.90-0.95**	**0.95-0.99**	**0.99-1.01**	**1.01-1.05**	**1.05-1.10**	**≥1.10**
			Short Term Options <60 Days				
Call	2.749	4.742	9.589	20.483	38.920	72.966	119.305
BS Implied Volatility	0.258	0.212	0.179	0.184	0.206	0.254	0.339
CS Implied Volatility	0.429	0.290	0.214	0.200	0.202	0.219	0.266
# total sample obs	633	2525	5338	3080	4070	2172	1088
# out of sample obs	87	926	3308	2111	2745	1400	573
			Medium Term Options 60-180 Days				
Call	5.913	12.585	26.234	40.451	57.694	87.608	132.080
BS Implied Volatility	0.206	0.183	0.190	0.198	0.214	0.229	0.259
CS Implied Volatility	0.304	0.233	0.223	0.219	0.229	0.226	0.233
# total sample obs	2100	2802	2618	1474	1628	1014	701
# out of sample obs	684	1580	1569	927	1021	710	524
			Long Term Options ≥ 180 Days				
Call	14.034	31.002	50.150	64.452	80.207	105.852	147.541
BS Implied Volatility	0.183	0.185	0.193	0.196	0.209	0.217	0.234
CS Implied Volatility	0.253	0.231	0.230	0.225	0.236	0.233	0.246
# total sample obs	1606	1273	1114	633	630	375	328
# out of sample obs	761	743	667	429	391	259	229

Table 3.2. Sample characteristics
Figures refer to average market values of call options (first line), Black and Scholes implied volatility (second line), Corrado and Su implied volatility (third line), the total number of observations for the (whole) period 2 January 2002 to 31 August 2004 (fourth line) and the total observations used in the out of sample period (aggregate – *AggTs*) for the period 03 March 2003 to 31 August 2004 (fifth line).

To find optimal implied parameter values per model k we solve an optimization problem that has the following form:

$$SSE(t) = \min_{\xi^k} \sum_{j=1}^{P_t} (\varepsilon_j^k)^2 \tag{3.25}$$

---

from option prices are preferable to parameters estimated from the underlying returns. See also Pan (2002) for such kind of applications.

where P_t refers to the number of different call option transaction datapoints available in day *t*, and ξ^k to the unknown parameters associated with a specific POPM (*k* = BS, CS and SVJ). The SSE is minimized via Nonlinear Least Squares with a subspace trust region method based on the Newton approach offered by the MATLAB® Optimization Toolbox. To minimize the possibility to obtain implied parameters that correspond to a local minimum of the error surface with each model we use several starting values for the unknown parameters based on daily average values reported by previous literature (see section 5 for details).

From the above we obtain the following sets of daily overall average (*av*) implied (risk-neutral) parameters:

Daily overall average implied BS volatility estimates $\xi^{BS} = \left\{\sigma_{av}^{BS}\right\}$

Daily overall average implied CS estimates $\xi^{CS} = \left\{\sigma_{av}^{CS}, \mu_3, \mu_4\right\}$.

Daily overall average implied SVJ estimates[31] $\xi^{SVJ} = \left\{\sigma_{av}^{SVJ}, \lambda, \bar{\kappa}, \theta, \alpha, \beta, \sigma_v, \rho\right\}$.

In order to have a pure unconstrained optimization problem and avoid implausible implied values we enforce certain transformations to each model parameters via smooth, strictly increasing and differentiable functions. Specifically: *i*) via log transformations we constrain σ_{av}^{BS}, σ_{av}^{CS}, and σ_{av}^{SVJ} to be positive, μ_4 to be smaller than 30, λ to be smaller than 10, θ, α and σ_v to be smaller than 2, and β to be smaller than 20, and *ii*) via the hyperbolic tangent sigmoid functions we constrain μ_3 to lie between -15 and +15, $\bar{\kappa}$ to lie between -0.99 and 0.99 and ρ to be between -1 and +1. For similar treatments of the optimization phase see Bates (1991 and 2000) and Jondeau and Rockinger (2001).

---

[31] Similarly with the SVJ we also calibrate the SV model. In addition, like Bakshi et al. (1997) and Christoffersen and Jacobs (2004) we also calibrate the instantaneous conditional variance V_t daily (where its square root for consistency is also denoted as σ_{av}^{SVJ}).

We also estimate the three DVF models (DVF#1, DVF#2, and DVF#3) defined earlier. For BS this is straightforward; for CS we first estimate the overall average implied parameters and then we fix skewness and kurtosis to compute the contract specific implied volatility. We differentiate among the DVF models by using appropriate subscripts σ_{NL1}^{k}, σ_{NL2}^{k} and σ_{NL3}^{k} for the Nonlinear Least Squares estimation and σ_{L1}^{k}, σ_{L2}^{k} and σ_{L3}^{k} for OLS estimation ($k \in$ {BS, CS}). In addition, the DVF parameter estimates obtained via the Nonlinear Least Squares based on initial values obtained from the OLS are[32]: σ_{NLL1}^{k}, σ_{NLL2}^{k} and σ_{NLL3}^{k}.

For pricing and hedging reasons at time instant *t*, the implied structural parameters derived[33] at day t -1 are used together with all other needed information. Daily recalibration of the implied parameters (DVF and overall average) for POPMs is also adopted by Bakshi et al. (1997) and Christoffersen and Jacobs (2004) (see also discussions in Hull and Suo, 2002, and Berkowitz, 2004).

3.4.4. The set of alternative models

With the BS_j models we use as input *S, X, T,* d_y, *r*, and any of the following ten volatility estimates: σ_j^{BS} where $j \in$ {*av, L1, L2, L3, NL1, NL2, NL3, NLL1, NLL2, NLL3*}. Similarly we denote the ten parametric CS alternatives. Finally note that for the SV and SVJ models we use the overall average parameter estimates.

The notation for the enhanced models depends on the parametric model considered. We use eBS_j, with $j \in$ {*av, NL2*}, to denote the two

---

[32] It is quite tedious to find starting values for the nonlinear estimation of the DVF. Possible candidates for this are, among others, the estimates of the DVF coefficients obtained from OLS.

[33] Following the results in Christoffersen and Jacobs (2004) we use estimation with t - 1 day information; these authors have also used larger estimation period for their models in order to increase precision in the parameters estimation but they observed inferior out of sample results.

enhanced models where the BS volatilities σ_{av}^{BS} and σ_{NL2}^{BS} are being enhanced. Likewise we use eCS_j^1, with $j \in \{av, NL2\}$, to denote the two ePOPMs where the CS volatilities σ_{av}^{CS} and σ_{NL2}^{CS} are being enhanced. We also use eCS_j^2, with $j \in \{av, NL2\}$, to denote the two ePOPMs where the CS parameters (σ_{av}^{CS}, μ_3) and (σ_{NL2}^{CS}, μ_3) are being enhanced. Finally, we use eCS_j^3, with $j \in \{av, NL2\}$, to denote the two enhanced models where the CS parameters (σ_{av}^{CS}, μ_3, μ_4) and (σ_{NL2}^{CS}, μ_3, μ_4) are being enhanced. In addition to these parameters, the dividend adjusted moneyness ratio $(Se^{-d_y T})/X$ and the time to maturity (T) are also used as inputs to estimate the GPF. When we make reference to an enhanced model and we drop the subscript we refer to any model using daily either the *av* or *NL2* volatility inputs. All enhanced models examined are exhibited in Panel B of Table 3.1.

3.5. Model calibration and analysis of pricing results

3.5.1. Model calibration

To obtain the best daily overall average implied parameters for each model we use five different starting values in each case. For BS we choose five starting volatility values in the 6%-70% range. For CS we choose five starting sets of parameter values from the Corrado and Su (1996, 1997) studies. For SVJ we use five starting sets of parameter values (using more initializations for daily estimation of the SVJ model would be impractical). Three are based on the results reported in Bakshi et al. (1997, pg. 2018, Table III): initialization #1 is their average SVJ implied parameters taken by using all available options, initialization #2 is their average SV implied parameters taken by using all available options and initialization #3 is their average SVJ implied parameters taken by using all available at the money options. Moreover, two additional initializations are used: initializations #4 and #5 are created by adding noise from a uniform distribution to each of the average implied parameters values of the initializations #1 and #3. A similar approach is adopted for the case of SV.

Table 3.3 presents results from the daily optimization process for SVJ. We present the in and the out of sample pricing performance for each initialization and the number of cases where it produces the smallest daily RMSE. There is a variation in their out of sample pricing performance despite that four out of five initializations have similar in sample fitting. Initialization #2 is less successful indicating that calibrating properly the jump component is crucial. It is notable that initialization #3 which produces the least in sample RMSE in 178 out of 671 days is not the best model. The results indicate the existence of many local minima where different implied parameters reach the same in sample RMSE but result to significantly different out of sample RMSE. As noted by Bates in many of his works the complex parametric option pricing models appear to suffer from a nonlinear identification problem in that quite different parameter values can yield virtually identical in sample option prices. Finally, although initialization #1 seems to perform well in and out of sample, still it is not safe to conjecture that this will always be the case. Our calibrating results for SVJ indicate that it is better to try various initializations and choose the one with the smallest daily (in sample) RMSE. Calibrating either BS or CS is much easier; in each model, the different starting values result in almost the same final parameter values (results not shown due to brevity).

	Init. #1	Init. #2	Init. #3	Init. #4	Init. #5
In the sample RMSE	0.454	0.673	0.455	0.464	0.461
Out the sample RMSE	1.523	2.792	1.915	1.887	1.545
# times optimal	126	38	178	161	168

Table 3.3. Summary statistics for SVJ optimization
Daily in sample Root Mean Square Error (RMSE) pricing performance with the corresponding out of sample performance regarding the stochastic volatility and jump (SVJ) options pricing model. The first three initializations (Init. #1, #2 and #3) are accordingly three sets of overall average implied parameters reported in Bakshi et al. (1997) while Init. #4 (Init. #5) is taken by Init. #1 (Init. #3) after adding noise from a uniform distribution to each of the parameters. The last line reports the number of times/days that each of the five initializations has been the optimal one (has returned the lowest RMSE).

	Black & Scholes		Corrado & Su	
Type of Initialization	Times Optimal	In sample RMSE	Times Optimal	In sample RMSE
Lc	18	1.331	5	1.021
0.2Lc	20	1.614	58	0.863
0.5Lc	207	1.149	409	0.876
0.8Lc	191	1.079	130	0.901
0.9Lc	75	1.189	6	0.955
Rest	160	-	63	-

Table 3.4. Summary statistics for DVF optimization
Daily in sample Root Mean Square Error (RMSE) pricing performance is reported for the optimization process regarding the Deterministic Volatility Functions (DVF) used with Black and Scholes and the Corrado and Su models for the period 2 January 2002 to 31 August 2004. Only 5 out of 21 initializations used are reported here. The first and third columns of numerical results report the number of days that each of the 21 initializations used has been the optimal one (has resulted in the lowest RMSE). The results reported concern the DVF#2 specification according to which the volatility is linear in strike price, the squared of strike price, time to maturity and the cross product of strike price with time to maturity. *Lc* is the coefficient vector taken by OLS after regressing implied volatility on the variables included in DVF#2.

To obtain the daily optimal values for the DVF based BS and CS coefficients (OLS ones, *Lc*, and Nonlinear Least Squares, *NLc*) we similarly rely on a thorough optimization search regarding starting values. Although not reported in previous literature, finding “proper” starting values and optimizing the DVF models is not a trivial task. For each DVF model we try twenty-one different initial starting values. The first initialization is by using the *Lc* values. Another eight initializations are created by multiplying each of the elements of *Lc* by a value in the range 0.1 to 2 (specifically 0.1, 0.2, 0.5, 0.8, 0.9, 1.2, 1.5, 2). Additional three initializations are created by random numbers coming out of the normal distribution $N(0,0.1)$ and three more from $N(0,0.01)$. Finally, six initializations are created by using *Lc* + *Lc* x (random_sign) where random_sign is a vector of randomly chosen numbers (±0.2, ±0.5 or ±1 with equal probability between plus or minus sign).

In Table 3.4 we present partial results regarding the in sample performance of the most successful initializations for the DVF#2 based BS and CS (the optimization results for DVF#1 and DVF#3 are similar). The second and fourth columns present the number of days where a certain initialization produces the smallest RMSE for BS and CS respectively, while the third and fifth columns exhibit the in sample RMSE obtained by using every day the same starting values. As can be seen the most successful initializations are the ones using the *Lc* multiplied by 0.2, 0.5 and 0.8. This kind of initialization is supported by the results of Christoffersen and Jacobs (2004, figures in p. 310) who find that in their sample, non-linear DVF coefficients (*NLc*) are smaller than the linear ones (*Lc*) by a constant.

	σ	μ_3	μ_4	λ	$\bar{\kappa}$	θ	α	β	σ_v	ρ
BS	0.198 (0.002) **[0.182]**									
CS	0.232 (0.003)	-1.099 (0.013)	4.179 (0.048)							
SV	0.212 (0.003) **[0.187]**						0.186 (0.003) **[0.04]**	3.274 (0.067) **[1.15]**	0.809 (0.011) **[0.39]**	-0.639 (0.003) **[-0.64]**
SVJ	0.183 (0.003) **[0.194]**			0.679 (0.018) **[0.59]**	-0.149 (0.008) **[-0.05]**	0.192 (0.013) **[0.07]**	0.057 (0.002) **[0.04]**	2.762 (0.085) **[2.03]**	0.400 (0.013) **[0.38]**	-0.614 (0.013) **[-0.57]**

Table 3.5. Daily average implied parameters for the parametric models
Daily average implied parameter values obtained by jointly minimizing the sum of squared pricing deviations between a parametric model's estimates and the actual market value of the call options for the period 2 January 2002 to 31 August 2004. Standard error of each parameter is reported in parenthesis. Bold figures in square brackets are the corresponding values reported by Bakshi et al. (1997). The structural parameter σ is the Brownian volatility, μ_3 and μ_4 the skewness and kurtosis coefficients for CS. β is the rate of mean reversion, α / β is long run mean, σ_v is the volatility of volatility and ρ is the correlation coefficient between the volatility shocks and the underlying asset movements for the stochastic volatility process. λ is the frequency of jumps per year, $\bar{\kappa}$ the mean jump size and θ the volatility of the logarithm of $1+\kappa$ for the jump process.

Despite the fact that some initializations often produce the smallest daily RMSE, yet their in sample fitting performance is by far worst compared to the case where each day we select the initialization with the lowest RMSE. Using 0.5*Lc* to obtain the *NLc* for BS and CS proves to produce the smallest RMSE in 207 and 409 days respectively. DVF initializations created from random values were almost never the optimal choice. Finally, *Lc* very rarely provided the optimal choice. The conclusion again is that a thorough initialization for the DVF models is important[34].

---

[34] In sample fitting of Dumas et al. (1998) models for call options is 0.651, 0.300, 0.222 and 0.218 for the BS using the overall average volatility and nonlinear DVF#1, DVF#2 and DVF#3 respectively. The ratios of their overall average RMSE divided by their RMSE obtained for the three DVF models are 2.17, 2.93 and 2.99. The

3.5.2. Implied parameter estimates

In Table 3.5 we report the mean daily estimates for the overall average parameter values of the POPMs along with their standard errors in parentheses. In addition the bold figures in the square brackets report the associated values found in the study of Bakshi et al. (1997). The results for CS demonstrate that the implied index return distributions are negatively skewed with higher kurtosis than permitted by the BS assumptions. Thus, the BS model with *av* volatility is expected to perform poorly compared to other models that allow for more flexible distributions (see also Bakshi et al., 1997). Regarding the volatility process parameters, similarly to Bakshi et al. (1997) we observe implied volatilities of BS, SV and SVJ extremely close to each other, and we find that the implied long run mean volatility $\sqrt{\alpha / \beta}$ for SV equals 0.238 and is higher to 0.144 of SVJ. In addition, we similarly find the volatility of volatility σ_v and the magnitude of the correlation coefficient ρ in SV to be higher relative to SVJ indicating that SVJ captures part of the excess kurtosis and negative skewness with the jump component. In contrast though, the variation coefficient σ_v for SV is almost double compared to the one obtained for SVJ (a similar finding is also obtained in Bates, 2000, p. 203). According to Bates (2000, p. 226, see also Bates, 1996 and 2003), the negative correlation coefficient in SV is not enough to generate sufficiently negative implicit skewness, so a very high volatility of volatility (implausibly high compared to the time series properties of asset prices) might be necessary to match the observed option values[35] (similar conclusions are obtained by Bakshi et al., 1997, pg. 2043). Regarding the jump components of SVJ, we find that the average yearly frequency of jumps is 0.678, the average jump size is -14.9% and the jump size volatility is 19.2%. The jump size

---

respective ratios for our sample are 1.89, 3.91 and 4.88 indicating a successful optimization search for the DVF models coefficients (at least for DVF#2 and DVF#3; the ratio for our DVF#1 appears inferior since in our dataset we have also included long maturities).

[35] Bates (2000) favours SVJ by arguing that in the presence of a jump component, the model provides option prices more compatible with market prices and generates more plausible implied stochastic volatility parameter values.

parameter values are higher compared to those of Bakshi[36] et al. (1997) and closer to those of Bates (2000).

	Intercept	*X*	X^2	*T*	*XT*
			Black and Scholes		
Lc	2.105 (60.921)	-0.003 (-53.965)	1.307E-06 (49.310)	-0.372 (-18.200)	3.668E-04 (19.411)
NLc	1.404 (79.634)	-0.002 (-61.423)	7.451E-07 (49.912)	-0.228 (-19.056)	2.304E-04 (20.649)
			Corrado and Su		
Lc	1.081 25.984	-0.002 -21.527	8.312E-07 24.414	0.403 25.037	-3.917E-04 -22.795
NLc	0.571 24.906	-0.001 -14.728	3.366E-07 16.332	0.249 26.957	-2.331E-04 -23.436

Table 3.6. Summary statistics of coefficient estimates for DVF#2 model
Daily average coefficients obtained from fitting DVF#2 for the Black and Scholes and for the Corrado and Su model for the period January 2, 2002 to August 31, 2004. The *t*-statistics of each average coefficient value (computed from the daily values of the estimated coefficients) are reported below in parenthesis. *Lc* (*NLc*) is the ordinary least squares (nonlinear least squares) coefficient vector taken by regressing implied volatility on the variables included in DVF#2 specifications.

Table 3.6 tabulates the in sample coefficient estimates for the DVF#2 model of BS and CS (means and below in parenthesis the *t*-statistic). The first observation is that the sign of the average coefficient values for BS coincides with the ones obtained by Dumas et al. (1998) and Christoffersen and Jacobs (2004): negative for strike price (*X*) and time to maturity (*T*) and positive for X^2 and *XT*. Interestingly, the sign of the coefficients of X^2 and *XT* for CS are opposite to those for BS. This is evident also from Table 3.1 where we see that CS implied volatility is larger for out of the money options and smaller for in the money (the opposite pattern compared to BS). Second, similarly with Christoffersen and Jacobs (2004) we observe for all DVF#2 parameters (both for BS and CS) that the average coefficient values of *NLc* are significantly

[36] S&P 500 in the case of Bakshi et al. (1997) exhibited a major uptrend move in the whole period they examine while in our case, the market experienced a major down-trend move in the first 15 months and a steady upward movement afterwards.

smaller compared to *Lc* and more importantly less volatile[37] (as implied by larger *t*-statistic values for *NLc*). According to Christoffersen and Jacobs (2004) this manifests their better out of sample performance (as we also show below). Finally, by comparing BS with CS we can see that for both *Lc* and *NLc* models, BS coefficient values and *t*-values are larger. This indicates that the volatility smile is generally “flatter” for CS after controlling for skewness and kurtosis.

3.5.3. Pricing results

Table 3.7A provides the in sample RMSE and Table 3.7B demonstrates out of sample the pricing performance of all models considered in terms of RMSE, MAE, RMeSE, P_5AE and P_{95}AE for the aggregate period (*AggTs*). Before we compare the in and out of sample pricing performance of the alternative models we should note that the comparison can be biased against our semi-parametric approach since unlike the daily parameter (re-)calibration for the POPMs, GPFs are estimated only once a month.

	BS_{av}	BS_{L1}	BS_{NL1}	BS_{NLL1}	BS_{L2}	BS_{NL2}	BS_{NLL2}	BS_{L3}	BS_{NL3}	BS_{NLL3}
RMSE	3.365	2.964	1.783	2.139	1.492	0.861	1.331	1.305	0.690	1.219

	CS_{av}	CS_{L1}	CS_{NL1}	CS_{NLL1}	CS_{L2}	CS_{NL2}	CS_{NLL2}	CS_{L3}	CS_{NL3}	CS_{NLL3}
RMSE	1.572	2.098	1.467	1.585	1.119	0.782	1.025	0.954	0.636	0.897

	SVJ	SV
RMSE	0.437	0.696

Table 3.7A. In sample pricing performance of the parametric models
Root Mean Square Error (RMSE) values regarding the in the sample pricing performance for all parametric (overall average and DVF) models obtained by minimizing the sum of squared pricing deviations between a model’s estimates and the actual market value of the call options for the period January 2, 2002 to August 31, 2004.

[37] In general we have verified that the DVF#2 BS coefficient values plotted across the 671 days in our sample closely resemble (for the common coefficients) the plots presented in Christoffersen and Jacobs (2004, p. 310).

We first concentrate our attention to Table 3.7A (in sample) for the parametric BS, CS, SV and SVJ models. Before the alternative DVF versions are considered, the more complex parametric models with daily overall average values (*av*) exhibit superior performance, and thus SVJ is the best, followed by SV. The DVF approach improves the pricing performance of the BS and CS models considerably, with the nonlinear DVF#3 being superior; yet the SVJ is the overall best model in sample.

	BS_{av}	BS_{L1}	BS_{NL1}	BS_{NLL1}	BS_{L2}	BS_{NL2}	BS_{NLL2}	BS_{L3}	BS_{NL3}	BS_{NLL3}
RMSE	3.285	3.128	1.984	2.508	2.921	2.008	2.800	3.260	2.382	3.174
MAE	2.579	1.908	1.509	1.664	1.530	1.186	1.437	1.468	1.139	1.412
MeAE	2.172	1.164	1.213	1.242	0.962	0.833	0.923	0.834	0.739	0.826
P_5AE	0.242	0.091	0.115	0.124	0.082	0.078	0.085	0.072	0.067	0.073
P_{95}AE	6.396	6.440	3.796	4.375	4.364	3.100	3.944	4.161	2.983	3.931
	CS_{av}	CS_{L1}	CS_{NL1}	CS_{NLL1}	CS_{L2}	CS_{NL2}	CS_{NLL2}	CS_{L3}	CS_{NL3}	CS_{NLL3}
RMSE	2.245	2.794	2.110	2.262	2.248	1.766	2.136	2.667	2.189	2.627
MAE	1.709	1.890	1.609	1.679	1.451	1.257	1.390	1.438	1.252	1.411
MeAE	1.358	1.233	1.276	1.299	1.002	0.929	0.972	0.945	0.881	0.929
P_5AE	0.118	0.106	0.115	0.112	0.085	0.085	0.088	0.085	0.085	0.084
P_{95}AE	4.370	6.107	4.144	4.470	3.997	3.422	3.835	3.879	3.328	3.794
	SVJ	*SV**	*SV*							
RMSE	1.498	4.541	2.488							
MAE	1.071	1.551	1.318							
MeAE	0.796	0.900	0.904							
P_5AE	0.065	0.077	0.078							
P_{95}AE	2.996	3.413	3.362							

Table 3.7B. Out of sample pricing performance of the parametric models
Error performance results (out of sample pricing) for all parametric models for the aggregate period March 3, 2003 to August 31, 2004. *SV** results are the original ones before replacing extreme mispricing observations. *SV* results are obtained after replacing, with BS_{NL2}, *SV** values that differ by more than 50% compared to BS_{NL2}. In total, 747 observations are replaced. RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error, MeAE is the Median Absolute Error and P_5AE (P_{95}AE) is the 5^{th} (95^{th}) Percentile of Absolute Errors.

We then concentrate on Table 3.7B (out of sample performance of the parametric models)[38]. We see that the DVF based CS models provide better

[38] Our RMSE results are larger compared for instance to the ones of Bakshi et al. (1997) and Dumas et al., (1998) since for the period we examine the average index

performance than the corresponding DVF based BS ones. The best model is CS_{NL2} which improves RMSE performance over BS_{NL2} by 14%. The nonlinear DVF#2 model provides the best out of sample performance for both BS (this is consistent with the results in Dumas et al., 1998) and CS (although for the BS case the nonlinear DVF#1 was equally good in terms of the RMSE but inferior by far in terms of the other measures). The inferior performance of the nonlinear DVF#3 model is not surprising since as Bates (2000) notes, over-parameterized models entail the risk of overfitting the options data and start explaining white noise (see also Dumas et al., 1998 who argue in favor of the greater parsimony in the volatility function provided by DVF#2). Note that an overfitting problem does not appear to be present in the enhanced models where a large dataset is used for estimation. Similarly with Christoffersen and Jacobs (2004) we find that SV[39] underperforms BS_{NL2}. Still, among parametric models, the SVJ model is the top performer in all metrics.

We then look at Table 3.8 with the out of sample performance for the ePOPMs. We see that all enhanced models have excellent performance. The best BS version is eBS_{NL2} which is the enhancement of BS with nonlinear DVF#2 input and 16% RMSE improvement. The best CS version is eCS^2_{NL2} that enhances two parameters of CS (volatility and skewness) and improves RMSE by 19%. eCS^2_{NL2} is also the overall best ePOPM in terms of the RMSE metric. The eCS^2_{av} model is the second best ePOPM. Models eCS^3_{av} and eCS^3_{NL2} are also good performers. We must make the comment that the enhanced models

---

option prices are most of the time double or triple than theirs (compare our Table 2 with Table I of Bakshi et al., 1997).

[39] We noticed that the stochastic volatility model produced large out of sample mispricings for many cases. The problematic observations include options with maturities longer than 180 days. Table 7B presents two sets of pricing results for the stochastic volatility model. *SV** results are the original ones before replacing extreme mispricing observations. *SV* results are obtained after replacing, with BS_{NL2}, *SV** values that differ by more than 50% compared to BS_{NL2}. In total, 747 observations are replaced for the out of sample period. Christoffersen and Jacobs (2004, p. 307) mention that they have removed from their out of sample results certain problematic SV pricing observations.

using as input the overall average parameters show performance slightly only inferior but practically comparable to the enhanced models with DVF#2 volatility input; and they outperform by far the equivalent parametric models with overall average parameters. A final comment is that the best enhanced models[40] are also competitive to the SVJ model which is too expensive to properly calibrate daily.

	eBS_{av}	eBS_{NL2}	eCS^1_{av}	eCS^1_{NL2}	eCS^2_{av}	eCS^2_{NL2}	eCS^3_{av}	eCS^3_{NL2}
RMSE	1.754	1.732	1.646	1.601	1.532	1.489	1.568	1.535
MAE	1.327	1.157	1.243	1.176	1.167	1.087	1.176	1.131
MeAE	1.046	0.834	0.957	0.886	0.925	0.813	0.908	0.856
P_5AE	0.097	0.069	0.084	0.078	0.079	0.075	0.079	0.072
P_{95}AE	3.473	3.190	3.345	3.220	3.081	2.955	3.169	3.075

Table 3.8. Out of sample pricing performance of the non-parametrically enhanced models (ePOPMs)
Error performance results (out of sample pricing) for selected enhanced parametric models for the aggregate period March 3, 2003 to August 31, 2004. RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error, MeAE is the Median Absolute Error and P_5AE (P_{95}AE) is the 5^{th} (95^{th}) Percentile of Absolute Errors.

We see using both statistics that the ePOPMs outperform the equivalent POPMs (both with overall average and DVF parameter estimates), and the difference is statistically significant at the 1% level. The best ePOPM model is eCS^2_{NL2} and is competitive to SVJ (any difference in performance is not statistically significant). Our second best ePOPM is eCS^2_{av} which although appears marginally inferior to the SVJ is much easier to estimate.

[40] We have also checked the performance of standard neural networks like the ones used in previous studies (i.e. Hutchinson et al., 1994, Garcia and Gencay, 2000). The optimization/training methodology setup is similar to the one employed for the enhanced models. The results for standard feedforward artificial neural networks are always inferior to that of the enhanced models. Specifically their RMSE is between 2.013 and 2.743 which is quite large compared to the enhanced models whose RMSE is consistently below 1.754.

	BS_{NL2}	CS_{NL2}	SVJ	eBS_{av}	eBS_{NL2}	eCS_{av}^{2}	eCS_{NL2}^{2}	eCS_{av}^{3}	eCS_{NL2}^{3}
BS_{NL2}		3.18	6.02	3.40	3.45	6.03	6.47	5.62	5.98
CS_{NL2}	-3.68		6.85	0.52	0.91	9.90	10.96	8.29	9.32
SVJ	-6.26	-7.25		-7.38	-4.92	-0.96	0.24	-1.97	-1.01
eBS_{av}	-3.43	-0.57	7.61		0.66	14.68	15.28	11.83	12.97
eBS_{NL2}	-4.62	-1.17	5.22	-0.71		5.65	6.61	4.62	5.45
eCS_{av}^{2}	-6.07	-10.77	0.98	-23.49	-5.87		2.71	-2.60	-0.18
eCS_{NL2}^{2}	-6.85	-18.10	-0.25	-18.12	-7.92	-3.41		-4.80	-2.57
eCS_{av}^{3}	-5.65	-9.02	2.03	-19.99	-4.90	5.26	5.97		2.09
eCS_{NL2}^{3}	-6.32	-13.51	1.04	-15.51	-6.68	0.23	6.04	-2.69	

Table 3.9. *t*-tests for out of sample model performance comparison
Values in the upper (lower) diagonal report the Student *t*-value (Johnson, 1978, modified *t*-value) regarding the comparison of means of the squared residuals between models in the vertical heading versus models in the horizontal heading. In general, a positive (negative) *t*-value larger (smaller) than 1.96 (-1.96) indicates that the model in the vertical (horizontal) heading has a larger MSE than the model in the horizontal (vertical) heading at 5% significance level (for 1% significance level use 2.325 and -2.325 respectively).

In Tables 3.10 and 3.11 we analyze the RMSE of the best performing models in terms of moneyness and time to maturity (7x3=21 classes). In addition, the bottom panel of each table reports RMSE per moneyness class (aggregating time to maturity) while the last column reports RMSE per time to maturity (aggregating moneyness). First, we concentrate on the best performing parametric models (Table 3.10). Comparing moneyness class performance (bottom panel) we see that SVJ is superior in five out of seven moneyness classes while BS_{NL2} exhibits the best performance in JITM and ITM options. BS_{NL2} is superior to SV in all moneyness classes (a similar conclusion is reached by Christoffersen and Jacobs, 2004) except for DOTM. In addition, we see that CS_{NL2} which produces the overall best RMSE among DVF models is not superior to BS_{NL2} in all moneyness classes. Specifically, it is performing well in out-of-the-money options while BS_{NL2} is superior in at- and in-the-money options. In terms of time to maturity only (last column) we see that SVJ (BS_{NL2}) is the first (second) best performing in short term options, BS_{NL2} (SVJ) is the first (second) best performing in medium term options and SVJ (CS_{NL2}) is the first (second) best performing in long term options. Concluding on parametric models' results, although SVJ has the overall best performance on aggregate in terms of RMSE still it does not produce the least RMSE in all moneyness and time to maturity classes.

CS_{NL2} dominance over BS_{NL2} is coming from long term, out-of-the-money options. Finally note that SV fails over DVF models because it exhibits poor performance for long term options.

In Table 3.11 we see the results for the ePOPMs. In general the CS ones produce the best results. By comparing eCS_{av}^{2} with eCS_{av}^{3} we see that the enhancement of kurtosis in the latter model helps to improve the DOTM and DITM options but does not offer any improvement in the other cases. This result is very intuitive since kurtosis affects the tails. In comparing eCS_{av}^{2} with eCS_{NL2}^{2} we see that the latter model has better performance for JOTM, ATM and JITM options. Another significant observation is that enhanced models with DVF input perform better in short and medium term options while enhanced models with overall average implied parameters as input have significantly better performance in long term options. Lastly if we compare SVJ with eCS_{NL2}^{2} (eCS_{av}^{2}) we can see that in many cases the proposed semi-parametric methodology is better. Specifically eCS_{NL2}^{2} (eCS_{av}^{2}) has lower RMSE in eleven (six) out of twenty-one classes.

To investigate whether our semi-parametric approach imposes any discipline on the models and to preclude the possibility that the enhanced parameters are just moving around excessively through time we used graphical diagnostics (plots are not displayed due to brevity). Specifically, for each of the 379 out of sample days of the period 3 March 2003 to 31 August 2004 we use the already estimated GPF and DVF and get predictions for the daily volatility values for moneyness equal to S/X = 0.90, 1.00 and 1.10 and time to maturity equal to T = 21, 63, 126, 189 and 252 trading days (in total 15 combinations per day per model). The conclusion is that the enhanced volatility estimates derived by the GPF are in general less volatile compared to the DVF volatility estimates for the out of sample period and track better the evolution of the actual daily implied volatility[41]. A representative summary of the graphical diagnostics is exhibited in Table 3.12, where we report the

[41] The actual implied volatility is extracted daily from market option prices that are closest to each combination of moneyness and time to maturity.

RMSE between the daily actual contract specific volatility implied by market option prices and the volatility estimates obtained by the DVF based BS_{NL2}

	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM	Maturity classes
S/X	<0.90	0.90-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.10	≥1.10	
Model	**Short term options - ≤ 60 days**							
BS_{av}	0.682	1.793	2.587	2.180	1.779	2.629	2.911	2.289
BS_{NL2}	0.607	0.509	1.057	1.465	1.438	1.561	1.773	1.319
CS_{av}	1.331	0.976	1.955	2.632	2.395	1.842	1.904	2.136
CS_{NL2}	0.913	0.753	1.050	1.581	1.768	1.750	1.858	1.478
SV	0.523	0.606	1.080	1.639	1.697	1.566	1.729	1.435
SVJ	0.410	0.556	0.985	1.355	1.524	1.604	1.417	1.289
	Medium term options - 60-180 days							
BS_{av}	2.533	2.964	2.518	2.087	3.349	5.554	5.991	3.469
BS_{NL2}	1.199	1.018	1.213	1.514	1.726	2.062	2.149	1.488
CS_{av}	0.844	1.026	1.754	1.934	2.133	2.499	2.735	1.828
CS_{NL2}	0.833	1.040	1.452	1.764	1.979	2.211	2.166	1.615
SV	1.184	1.133	1.484	1.884	2.054	2.187	2.043	1.673
SVJ	0.565	0.935	1.430	1.686	1.880	2.012	1.830	1.495
	Long term options – > 180 days							
BS_{av}	4.633	3.762	3.057	3.879	5.293	8.317	9.862	5.105
BS_{NL2}	5.491	3.816	3.130	2.770	1.912	2.921	4.741	3.880
CS_{av}	1.476	2.092	3.177	3.738	3.876	4.688	4.944	3.163
CS_{NL2}	2.014	2.288	2.620	2.933	2.382	2.957	4.737	2.674
SV	5.352	4.572	4.485	4.994	4.779	6.221	6.948	5.127
SVJ	1.040	1.665	2.208	1.982	2.183	3.837	1.924	2.030
	Moneyness classes (aggregating maturity)							
BS_{av}	3.681	2.903	2.629	2.433	2.735	4.571	5.886	
BS_{NL2}	3.955	1.977	1.504	1.694	1.562	1.909	2.658	
CS_{av}	1.225	1.336	2.089	2.635	2.514	2.506	2.957	
CS_{NL2}	1.540	1.373	1.442	1.849	1.887	2.057	2.687	
SV	3.856	2.347	1.934	2.381	2.255	2.667	3.358	
SVJ	0.830	1.071	1.321	1.538	1.688	2.084	1.683	

Table 3.10. Tabulation (moneyness vs. maturity) of out of sample pricing RMSE for selected parametric models

Root Mean Square Error (RMSE) values regarding the out of sample pricing performance for selected parametric models for the aggregate period March 3, 2003 to August 31, 2004. RMSE is tabulated into moneyness and time to maturity. The bottom panel reports RMSE per moneyness classes (aggregating time to maturity) while the last column reports RMSE per time to maturity (aggregating moneyness).

	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM	Maturity classes
S/X	<0.90	0.90-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.10	≥1.10	
Model				**Short term options (≤ 60)**				
eBS_{av}	0.656	0.808	1.452	1.732	1.591	1.553	1.593	1.519
eBS_{NL2}	0.467	0.579	1.109	1.474	1.452	1.503	1.558	1.316
eCS^2_{av}	0.534	0.743	1.261	1.611	1.570	1.456	1.529	1.416
eCS^2_{NL2}	0.440	0.652	1.000	1.337	1.421	1.435	1.442	1.239
eCS^3_{av}	0.557	0.723	1.172	1.509	1.599	1.478	1.411	1.375
eCS^3_{NL2}	0.664	0.676	1.021	1.367	1.480	1.434	1.313	1.261
				Medium term options (60-180]				
eBS_{av}	0.684	1.186	1.790	1.928	1.958	1.941	1.710	1.652
eBS_{NL2}	1.062	1.010	1.237	1.464	1.646	1.921	1.979	1.423
eCS^2_{av}	0.700	0.992	1.498	1.719	1.730	1.662	1.639	1.441
eCS^2_{NL2}	0.746	1.018	1.372	1.602	1.675	1.709	1.811	1.413
eCS^3_{av}	0.627	0.986	1.486	1.721	1.871	1.768	1.559	1.466
eCS^3_{NL2}	0.757	1.028	1.460	1.726	1.797	1.778	1.737	1.476
				Long term options – (> 180)				
eBS_{av}	1.488	2.371	2.786	2.874	3.005	2.722	2.721	2.499
eBS_{NL2}	3.523	3.002	2.672	2.541	2.410	2.869	3.791	3.003
eCS^2_{av}	1.437	1.838	2.158	2.224	2.331	2.303	2.167	1.998
eCS^2_{NL2}	1.728	2.045	2.178	2.336	2.146	2.422	3.396	2.200
eCS^3_{av}	1.196	1.936	2.467	2.664	2.731	2.722	2.355	2.216
eCS^3_{NL2}	1.985	2.070	2.192	2.373	2.312	2.506	3.270	2.273
				Moneyness classes (aggregating maturity)				
eBS_{av}	1.155	1.468	1.760	1.960	1.861	1.835	1.879	
eBS_{NL2}	2.585	1.629	1.423	1.641	1.614	1.828	2.254	
eCS^2_{av}	1.123	1.186	1.464	1.727	1.695	1.631	1.698	
eCS^2_{NL2}	1.320	1.257	1.304	1.565	1.567	1.654	2.046	
eCS^3_{av}	0.951	1.216	1.475	1.748	1.803	1.743	1.667	
eCS^3_{NL2}	1.496	1.274	1.343	1.622	1.656	1.688	1.945	

Table 3.11. Tabulation (moneyness vs. maturity) of out of sample pricing RMSE for selected non-parametrically enhanced models (ePOPMs)

Root Mean Square Error (RMSE) values regarding the out of sample pricing performance for selected enhanced parametric models (ePOPMs) for the aggregate period March 3, 2003 to August 31, 2004. RMSE is tabulated into moneyness and time to maturity. The bottom panel reports RMSE per moneyness classes (aggregating time to maturity) while the last column reports RMSE per time to maturity (aggregating moneyness).

and CS_{NL2} models and the enhanced models: eBS_{av}, eBS_{NL2}, eCS^1_{av}, eCS^1_{NL2}. The first three rows of numerical results report RMSE for three specific moneyness cases (always aggregating time to maturity) while the last row of numerical results exhibits the aggregate RMSE values. The general conclusion is that the enhanced models provide more accurate predictions of the implied volatility surface compared to the DVF counterparts.

	BS_{NL2}	eBS_{av}	eBS_{NL2}	CS_{NL2}	eCS^1_{av}	eCS^1_{NL2}
RMSE accuracy across specific moneyness cases						
S/X=0.9	0.021369	0.009754	0.015801	0.020567	0.018049	0.019725
S/X=1.0	0.010938	0.010813	0.00985	0.020244	0.017309	0.018067
S/X=1.1	0.02933	0.026033	0.02701	0.026925	0.020244	0.021972
RMSE accuracy (aggregate)						
	0.0219	0.0172	0.0189	0.0228	0.0186	0.0200

Table 3.12. Out of sample RMSE of volatility predicted by selected models
Root Mean Square Error (RMSE) values regarding the out of sample volatility prediction for selected models for the period March 3, 2003 to August 31, 2004. Volatility estimates for each day for specific moneyness (for five representative maturities) cases are compared with contract specific implied volatility taken using the market call option prices closest to each moneyness/maturity combination. Contract specific implied volatility for the Corrado and Su model is computed after fixing the skewness and kurtosis coefficients to their daily overall average values taken with the Whaley (1982) method.

Table 3.13 shows for the aggregate testing period the mean values of the enhanced parameters (volatility, skewness and kurtosis) for the eBS_{av}, eCS^1_{av}, eCS^2_{av} and eCS^3_{av} models for different maturity and moneyness classes. These parameters are provided by the GPF (see υ_j variables in the enhanced layer in Figure 2). We concentrate on enhancement created using overall average parameters as input since these are the models that have shown superior robustness (see next section for robustness analysis). For eCS^3_{av} the enhanced volatilities preserve a smile effect in the short and medium term options, the enhanced skewness is increasing in moneyness and decreasing in maturity, and the enhanced kurtosis exhibits a hump shape in moneyness. For eCS^2_{av} enhanced volatilities preserve a similar smile effect in the short and medium term options, and skewness exhibits a hump shape in moneyness for short and medium maturity options and similarly decreasing in maturity. Das and Sundaram (1999) compare the stochastic volatility model with jump-diffusion and conclude that "*it is less obvious whether the theoretical predictions of either class of models are – or can be made – consistent with the observed term structures of these deviations*" (see also Andersen et al., 2002). The authors wonder whether SV and Jump models can fit the data well by capturing the level of skewness and kurtosis implied by the data for *all maturities*. They state that (compared with empirical observations) Jump models allow too rapid decay in skewness and kurtosis, and stochastic volatility models exhibit a hump shape overly pronounced for intermediate maturities. In contrast, the results for the enhanced CS model show that is allows more flexibility in all estimated parameters, not only in terms of maturity but also in terms of moneyness.

3.6. Robustness analysis for pricing results

We check the robustness of the performance of the enhanced models in several ways. First, we check in terms of the complexity of the nonparametric GPF. Then we check in terms of pricing data not used in the estimation. Finally we compare the performance of the enhanced models (which are estimated monthly) with parametric benchmarks estimated weekly instead of daily. These robustness tests can also be seen as tests of overfitting. We check the robustness of the enhanced models in out of sample performance by using fewer hidden neurons in the validation phase (results not reported for brevity). We use one-to-eight and one-to-six hidden neurons and the results show significant robustness to the case of one-to-ten hidden neurons used in the analysis. Specifically, with one-to-eight hidden neurons the out of sample RMSE deteriorates at most 3.8% for eBS_{NL2} and less than 1% for the other models while for one-to-six hidden neurons RMSE deteriorates around 6% for eBS_{NL2} and less than 4% for the other models. Enhanced models that employ an overall average implied parameter input exhibit the greatest robustness. We also calculated (not reported for brevity) out of sample RMSE by fixing the number of hidden neurons during estimation across periods. BS based enhanced models exhibit similar performance for seven to ten hidden neurons with RMSE deterioration around 3.7%-6.7% while between three and seven hidden neurons RMSE is below 1.90. CS based enhanced models with five or more hidden neurons exhibit quite similar out of sample RMSEs which are close to the optimal ones as in Table 3.8

		DOTM	OTM	JOTM	ATM	JITM	ITM	DITM
S/X		<0.90	0.90-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.10	≥1.10
Short Term Options - <60 Days								
eBS_{av}	volatility	0.220	0.171	0.158	0.167	0.180	0.207	0.254
eCS^1_{av}	volatility	0.314	0.211	0.177	0.175	0.177	0.188	0.211
eCS^2_{av}	volatility	0.236	0.182	0.166	0.172	0.181	0.198	0.228
eCS^2_{av}	skewness	-0.550	-0.374	-0.347	-0.396	-0.432	-0.489	-0.496
eCS^3_{av}	volatility	0.252	0.193	0.179	0.186	0.194	0.212	0.237
eCS^3_{av}	skewness	-0.784	-0.651	-0.511	-0.391	-0.287	-0.194	-0.105
eCS^3_{av}	kurtosis	5.008	5.517	6.089	6.030	5.854	5.742	5.635
Medium Term Options - 60-180 Days								
eBS_{av}	volatility	0.173	0.158	0.166	0.176	0.187	0.204	0.234
eCS^1_{av}	volatility	0.226	0.190	0.187	0.190	0.193	0.196	0.206
eCS^2_{av}	volatility	0.187	0.170	0.177	0.185	0.192	0.200	0.215
eCS^2_{av}	skewness	-0.499	-0.449	-0.487	-0.532	-0.547	-0.604	-0.626
eCS^3_{av}	volatility	0.202	0.185	0.193	0.202	0.207	0.216	0.231
eCS^3_{av}	skewness	-0.849	-0.747	-0.623	-0.522	-0.475	-0.360	-0.208
eCS^3_{av}	kurtosis	5.488	5.894	5.933	5.873	5.694	5.711	5.722
Long Term Options - ≥ 180 Days								
eBS_{av}	volatility	0.164	0.168	0.174	0.181	0.192	0.201	0.220
eCS^1_{av}	volatility	0.210	0.199	0.197	0.200	0.206	0.207	0.217
eCS^2_{av}	volatility	0.185	0.187	0.190	0.195	0.201	0.203	0.214
eCS^2_{av}	skewness	-0.627	-0.655	-0.710	-0.758	-0.686	-0.717	-0.757
eCS^3_{av}	volatility	0.195	0.199	0.203	0.207	0.214	0.218	0.229
eCS^3_{av}	skewness	-0.881	-0.801	-0.741	-0.678	-0.645	-0.543	-0.453
eCS^3_{av}	kurtosis	5.451	5.596	5.622	5.557	5.475	5.433	5.429

Table 3.13. Summary statistics regarding the enhanced parameters for models (ePOPMs) optimized and selected using a pricing criterion
Moneyness and time to maturity tabulation of enhanced parameters implied by some of the enhanced models for the aggregate period March 3, 2003 to August 31, 2004.

.

The next test was to disaggregate each model RMSE into two components: the RMSE of observations (in total 16609) that are common both in day t-1 (day used to extract the implied parameters) and in day t (out of sample day) and the RMSE of observations (in total 5035) that can be thought of as unseen data (exist in *t* but not in *t*-1). Results are presented in Table 3.14 (columns 2-3 with numerical results). The first column reports again the RMSE for the *AggTs* dataset for comparison purposes. It is important to see whether the models' performance in the unseen data is close to the performance in the common data or if it deviates significantly. As expected, the RMSE for the common observations (unseen) is always lower (higher) than the RMSE in *AggTs*. We can see that DVF models loose more accuracy when used to price unseen observations compared to the BS and CS with overall average implied parameters. We also confirm that SV is highly inaccurate when used to price unseen data (especially of long maturity) compared to SVJ.

	AggTs	Common in *t*-1 and *t* day	Unseen in *t*-1	Extra (new) Observations
BS_{av}	3.285	2.825	4.478	5.967
BS_{NL2}	2.008	1.298	3.432	4.802
CS_{av}	2.245	2.056	2.779	2.660
CS_{NL2}	1.766	1.482	2.483	3.089
SV	2.488	1.451	4.436	7.112
SVJ	1.498	1.348	1.910	1.709
eBS_{av}	1.754	1.680	1.978	1.838
eBS_{NL2}	1.732	1.393	2.548	3.203
eCS^2_{av}	1.532	1.480	1.693	1.836
eCS^2_{NL2}	1.489	1.316	1.953	2.327
eCS^3_{av}	1.568	1.497	1.785	1.341
eCS^3_{NL2}	1.535	1.377	1.969	2.803
#obs		16609	5035	1237

Table 3.14. Robustness analysis - RMSE for common/unseen and totally new observations

The results in the second and third numerical columns disaggregate each model Root Mean Square Error (RMSE) into two components: *i)* the RMSE of observations (in total 16609) that are common both in day *t*-1 (day used to extract the implied parameters) and in day *t* (out of sample day), and *ii)* the RMSE of observations (in total 5035) that can be thought of as unseen data (exist in *t* but not in *t*-1). The RMSE in the last column refers to the out of sample pricing performance of each model for 1237 totally unseen observations (outside the moneyness range used in estimation). The first column of results repeats RMSE for the aggregate dataset (*AggTs*) for comparison purposes.

As can been seen, the enhanced models with overall average implied parameters as input (eBS_{av}, eCS^2_{av} and eCS^3_{av}) are very robust with unseen data and some even outperform the SVJ model (for example, the RMSE of

eCS_{av}^{2} in the unseen dataset is 1.693, quite smaller than 1.910 for SVJ). The robustness of our proposed enhanced models is very important since unseen data comprise a considerable part of the contracts traded every day (about 23% of out of sample observations in our dataset were contracts not traded the day before).

We also examine the models' performance to price totally unseen contracts (outside the moneyness range used in estimation). Specifically for the period January 2002 – August 2004 there are 784 observations with moneyness in the range of 0.70-0.80 and 453 observations with moneyness in the range of 1.20-1.30 (in total 1237 new observations). Results are reported in the last column of Table 3.14**.** First, we note that CS based parametric models significantly outperform the equivalent BS ones. The SV model (in contrast to SVJ) performs very poorly (again we have verified that large errors are mostly produced from long maturity options). The enhanced models with overall average implied parameters are the best performers. Actually, the RMSE of eCS_{av}^{3} is 1.341 outperforming by far that of SVJ (1.709) and as we have verified this performance is consistent in all moneyness and maturity ranges of the new observations.

As a final check we calculate the RMSE of the parametric models five days ahead. Remember that the enhanced models are estimated and then used for a whole month, whereas the parametric ones (overall average and DVF) are estimated every day. In that comparison there is obviously a bias against the proposed semi-parametric methodology, since some models and especially the SVJ are very computationally expensive to calibrate daily. As shown in Table 3.15, the RMSE deteriorates for BS_{av} by 7.7%, for BS_{NL2} by 34.5%, for CS_{av} 19.2%, for CS_{NL2} by 30.9%, for SV by 23.4%, and for SVJ by 43.3%. We see that the RMSE for five days ahead deteriorates so that the enhanced models are far superior to the parametric ones (now they outperform SVJ considerably). RMSE for ten or more days ahead (not shown for brevity) deteriorates even further for the parametric models. These results are consistent with the arguments in Christoffersen and Jacobs (2004), Bollen and Whaley (2004), and Hull and Suo (2002) that implied volatility functions are not persistent.

	BS_{av}	BS_{NL2}	CS_{av}	CS_{NL2}	SV	SVJ
RMSE	3.538	2.700	2.677	2.311	3.070	2.147
Deterioration Ratio	1.077	1.345	1.192	1.309	1.234	1.433

Table 3.15. Robustness analysis - 5-days ahead (out of sample) pricing performance for selected parametric models
Root Mean Square Error (RMSE) regarding the out of sample pricing performance for selective parametric models for the aggregate period March 3, 2003 to August 31, 2004. The models are used to price call options on day *t* with implied parameters computed on day *t* - 5. The second row reports the Deterioration Ratio which is the RMSE of each model in this table divided by the RMSE obtained for out of sample performance one day ahead.

3.7. Single instrument hedging analysis

We now investigate the hedging performance of the best (with respect to out of sample pricing) models. We follow a single instrument hedging strategy similar to the one conducted by Bakshi et al. (1997). Based on previous research (i.e. Hutchinson et al., 1994, Bakshi et al., 1997, Garcia and Gencay, 2000, Chernov and Ghyssels, 2000), the best model (parametric or nonparametric) in terms of out of sample pricing accuracy does not always prove to be the best performing one with respect to hedging performance. Bollen and Whaley (2004) find that the slope of the daily implied volatility functions in terms of moneyness is very erratic which may explain the poor performance of pricing models when used for hedging. As suggested by Christoffersen and Jacobs (2004) the above ambiguity may be due to the inappropriate choice of the loss function. They suggest that "*the best possible parameter estimates for a hedging exercise will likely be obtained using a hedging based loss function*" (p. 316). For the enhanced models, in order to align the estimation and evaluation loss functions we employ the following methodology (see also Garcia and Gencay, 2000): enhanced models parameters are estimated by minimizing the pricing RMSE but monitoring the hedging RMSE in the validation sample so that for each period the model with the lowest hedging error is chosen. The results are compared to hedging results obtained by the parametric models and to those obtained by enhanced models optimized and chosen based on a pricing loss function.

The single-instrument hedging mitigates the no-arbitrage strategy followed by Black and Scholes (1973), where a portfolio including a short position in a call with a certain exercise price and time to expiration is hedged via a long position in the underlying asset. For such a hedging strategy and for each model, at time instance t we short the m^{th} call option contract with market value, $c_{m,t}^{mrk}$, go long in $\Delta_{m,t}$ "index shares" and invest the residual, $B_{m,t}$, in the risk-free bond. Next, at time $t+\Delta t$ we liquidate the position by buying the call and selling the index and calculate each hedging error $H(\Delta t)$ as follows:

$$H(\Delta t) = \Delta_{m,t} S_{t+\Delta t} + B_{m,t} e^{r\Delta t} - c_{m,t+\Delta t}^{mrk} \quad (3.26)$$

$$B_{m,t} = c_{m,t}^{mrk} - \Delta_{m,t} S_t \quad (3.27)$$

Each trading day the hedged portfolio is rebalanced. The hedging error is calculated when both prices, $c_{m,t}^{mrk}$ and $c_{m,t+\Delta t}^{mrk}$ are available. For each model we calculate the hedging RMSE for the *AggTs* period. The expression $\Delta_{m,t}^{BS}$ for BS is equal to $\partial c_{m,t}^{BS} / \partial S_t$ = $e^{-d_y T} N(d)$, for SVJ is equal to $\partial c_{m,t}^{SVJ} / \partial S_t$ = $e^{-d_y T} \Pi_1$ (see also Appendix in Bakshi et al., 1997) and for CS are given in section 3.2. The theoretical delta value $\Delta_{m,t}$ for a long call always lies between zero and unity (for a positive dividend yield) but for the CS model, these bounds may be violated. In our sample this occurred in very few instances in which cases the delta values were set equal to their theoretical bound. As can be deduced from the results in Hutchinson et al. (1994) (see their Figure 5 and also discussions in Aït-Sahalia and Lo, 1998, pg. 512) there are cases where standard feedforward artificial neural networks fail to produce theoretically consistent delta values. In contrast, our semi-parametric method has the advantage of being consistent with the parametric model being enhanced. In the following tables we discuss the RMSE measure that was used in estimation but for completeness we also report the MAE and MeAE measures.

Single instrument hedging results are reported in Table 3.16 for the parametric models. The single instrument analysis is most appropriate for the parametric BS and CS and the respective enhanced models and we will focus in the comparison of those models. We see that the BS models are the best performers among the parametric ones with respect to hedging, in contrast to pricing where the CS models are superior, confirming thus Bakshi et al. (1997) who also find that model ranking differs between pricing and hedging. We also confirm Dumas et al. (1998) since the several DVF based BS models are indistinguishable with respect to hedging. Similarly indistinguishable among themselves are the parametric CS based models. Among all parametric models, the BS based ones are the best[42].

	BS_{av}	BS_{L1}	BS_{NL1}	BS_{NLL1}	BS_{L2}	BS_{NL2}	BS_{NLL2}	BS_{L3}	BS_{NL3}	BS_{NLL3}
RMSE	1.180	1.116	1.135	1.132	1.114	1.118	1.115	1.114	1.116	1.113
MAE	0.900	0.835	0.857	0.855	0.829	0.835	0.831	0.828	0.831	0.827
MeAE	0.710	0.635	0.663	0.663	0.621	0.633	0.627	0.620	0.624	0.618

	CS_{av}	CS_{L1}	CS_{NL1}	CS_{NLL1}	CS_{L2}	CS_{NL2}	CS_{NLL2}	CS_{L3}	CS_{NL3}	CS_{NLL3}
RMSE	1.369	1.355	1.363	1.362	1.356	1.354	1.357	1.356	1.352	1.355
MAE	1.038	1.021	1.033	1.032	1.019	1.018	1.023	1.018	1.014	1.017
MeAE	0.805	0.779	0.800	0.799	0.776	0.778	0.783	0.774	0.771	0.773

	SVJ	SV
RMSE	1.359	1.378
MAE	1.010	1.020
MeAE	0.750	0.758

Table 3.16. Out of sample hedging performance of parametric models
Error measures (out of sample) for single instrument hedging performance of all parametric models (aggregate period March 3, 2003 to August 31, 2004). RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error and MeAE is the Median Absolute Error.

The hedging performance of the enhanced models in Table 3.17 is given first for models chosen for pricing and then for models chosen explicitly for

[42] Evidence that parametric models which can handle negative skewness and excess kurtosis can underperform BS for single instrument hedging is also documented in Capelle-Blancard et al. (2001), Vähämaa (2003) and Jurczenko et al., (2004).

hedging. In the first case we see that results compare with those of the parametric models, with only the CS based ones (eCS^2_{av}, eCS^2_{NL2}, eCS^3_{av}, eCS^3_{NL2}) demonstrating an improvement over the respective parametric ones. The improvement is present when skewness (eCS^2) and both skewness and kurtosis (eCS^3) are enhanced, but not when only volatility (eCS^1) is enhanced. When models are estimated besed on a hedging criterion, we see that eCS^2_{av}, eCS^2_{NL2}, eCS^3_{av} and eCS^3_{NL2} improve hedging performance considerably, confirming the conjecture in Christoffersen and Jacobs (2004). Finally we note that the benchmark model SVJ (and SV) in a single instrument hedging analysis underperforms the parametric BS and the enhanced models.

	eBS_{av}	eBS_{NL2}	eCS^1_{av}	eCS^1_{NL2}	eCS^2_{av}	eCS^2_{NL2}	eCS^3_{av}	eCS^3_{NL2}
	Hedging performance for enhanced parametric models selected using a pricing criterion							
RMSE	1.123	1.119	1.353	1.355	1.222	1.212	1.192	1.199
MAE	0.843	0.839	1.020	1.021	0.903	0.899	0.869	0.873
MeAE	0.643	0.638	0.783	0.781	0.671	0.665	0.622	0.627
	Hedging performance for enhanced parametric models selected using a hedging criterion							
RMSE	1.117	1.110	1.301	1.294	1.093	1.117	1.080	1.087
MAE	0.826	0.819	0.960	0.949	0.815	0.832	0.800	0.803
MeAE	0.613	0.598	0.701	0.681	0.615	0.630	0.593	0.591

Table 3.17. Out of sample hedging performance of non-parametrically enhanced models (ePOPMs)
Error measures (out of sample) for single instrument hedging performance of selected enhanced parametric models (ePOPMs) for the aggregate period March 3, 2003 to August 31, 2004. The upper panel of results presents the single instrument hedging performance of enhanced models optimized and selected using a pricing criterion while the lower panel presents the single instrument hedging performance of enhanced models selected using a hedging criterion. RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error and MeAE is the Median Absolute Error.

We remark that Bakshi et al. (1997) (see also Dumas et al., 1998, and Chernov and Ghysels, 2000) find that their models' hedging performance is virtually indistinguishable and that the hedging based rankings of the models are in sharp contrast with their out of sample pricing performance. We reach similar conclusions for the case of the parametric models. As a significant variation from previous literature we see that this is not the case for the enhanced models, especially the CS based ones.

	DOTM	OTM	JOTM	ATM	JITM	ITM	DITM
S/X	<0.90	0.90-0.95	0.95-0.99	0.99-1.01	1.01-1.05	1.05-1.10	≥1.10
Short Term Options - <60 Days							
eBS_{av} **volatility**	0.191	0.162	0.154	0.169	0.191	0.228	0.269
eCS^1_{av} **volatility**	0.189	0.165	0.161	0.176	0.195	0.227	0.272
eCS^2_{av} **volatility**	0.217	0.170	0.163	0.178	0.196	0.224	0.266
eCS^2_{av} **skewness**	-0.287	0.447	0.658	0.581	0.489	0.326	0.119
eCS^3_{av} **volatility**	0.270	0.265	0.295	0.304	0.311	0.320	0.339
eCS^3_{av} **skewness**	-1.047	-0.136	0.204	0.286	0.371	0.442	0.490
eCS^3_{av} **kurtosis**	7.493	8.436	9.739	9.722	9.502	9.020	8.279
Medium Term Options - 60-180 Days							
eBS_{av} **volatility**	0.155	0.141	0.149	0.165	0.186	0.217	0.255
eCS^1_{av} **volatility**	0.167	0.149	0.162	0.178	0.195	0.223	0.265
eCS^2_{av} **volatility**	0.160	0.151	0.167	0.184	0.197	0.219	0.251
eCS^2_{av} **skewness**	0.444	0.717	0.731	0.694	0.638	0.559	0.374
eCS^3_{av} **volatility**	0.225	0.262	0.278	0.293	0.304	0.314	0.318
eCS^3_{av} **skewness**	-0.145	0.092	0.257	0.335	0.382	0.466	0.520
eCS^3_{av} **kurtosis**	7.181	8.771	8.958	9.093	9.043	8.961	8.053
Long Term Options - ≥ 180 Days							
eBS_{av} **volatility**	0.146	0.144	0.149	0.158	0.175	0.198	0.240
eCS^1_{av} **volatility**	0.164	0.159	0.167	0.185	0.205	0.223	0.263
eCS^2_{av} **volatility**	0.153	0.161	0.173	0.185	0.199	0.213	0.240
eCS^2_{av} **skewness**	0.700	0.796	0.795	0.737	0.672	0.569	0.526
eCS^3_{av} **volatility**	0.243	0.268	0.286	0.288	0.293	0.294	0.305
eCS^3_{av} **skewness**	0.159	0.271	0.316	0.380	0.482	0.489	0.671
eCS^3_{av} **kurtosis**	8.370	8.921	9.438	9.102	8.660	8.293	7.930

Table 3.18: Summary statistics regarding the enhanced parameters for models (ePOPMs) optimized on a pricing criterion and selected using a hedging criterion
Moneyness and time to maturity tabulation of enhanced parameters implied by some of the enhanced models for the aggregate period March 3, 2003 to August 31, 2004.

Finally we notice that the enhanced models chosen for hedging had a very poor performance in pricing (results not reported for brevity). We examine their enhanced parameter values as provided by the GPF (Table 3.18), especially for eCS^3_{av} that conveys information for skewness and kurtosis. In contrast to Table 3.13, skewness is now positive for 18 out of 21 moneyness/maturity classes with an average value of 0.25 whilst kurtosis bounces between 7.18 and 9.74. These values differ by far from those in Table 3.13 and fully explain the poor pricing performance. This might be attributed to movements in deltas not directly linked to asset price movements (see Capelle-Blancard et al., 2001). The inappropriate choice of the loss function results in parameter estimation not suitable for such a different use. Under this setting we try to find the best delta values, not the best call price. The obtained enhanced parameters for models chosen on the hedging criterion capture efforts to hedge against exposures to the risks of the underlying asset. The hedged positions seem to have two properties: skewness is positive immunizing in this way the downsize risk and kurtosis is excessively high immunizing against the prospect of extreme returns (fat tails). Effectively both effects help reduce the impact of volatility.

3.8. Summary and conclusions

In this study we extend the Dumas et al. (1998) DVF for option pricing, with a nonparametric approach to estimate generalized parameter functions (GPF). The resulting enhanced parametric models have many desirable properties compared to the standard implementation of artificial neural networks like theory consistent option values and Greek letters. In general, this semi-parametric methodology is proposed as a way to alleviate deficiencies of the modern parametric options models and standard nonparametric approaches. For pricing and hedging performance analysis we use the S&P 500 index call options for the period January 2002 to August 2004. We compare the GPF approach with parametric models using both daily overall average implied parameters (for all parametric models considered) and daily contract specific implied parameters derived by the DVF approach (for the BS and CS models). The SVJ and SV models have also been included in the analysis for comparison.

We discuss the calibration of the parametric models, first for the overall average implied parameters of BS, CS, SV and SVJ, and then for the DVF based BS and CS models. We show that a careful estimation/optimization search is needed to obtain good implied parameters. The results obtained out of sample strongly support the proposed methodology. The first important finding is that the DVF approach when applied to CS provides results superior to CS (with overall average parameter estimates) and also to BS (with either overall average or DVF estimates). The second is that the SVJ model is the best model among the parametric models whilst SV is inferior to DVF based BS and CS models. The third is that the increase in the pricing accuracy of the enhanced BS and CS models over the best performing BS and CS parametric ones is considerable and statistically significant. In general, the best enhanced models (with daily implied parameters but monthly estimation of the GPF) are comparable to the daily estimated SVJ.

In addition, we find that the enhanced methodology is robust to the complexity of the GPF. It is also robust to the pricing of contracts not used during estimation, where eCS_{av}^{3} significantly outperforms SVJ.

Consistently with the recommendation in Christoffersen and Jacobs (2004) we observe that single instrument hedging results using ePOPMs chosen using a hedging criterion outperform all the parametric models and the ePOPMs chosen using a pricing criterion.

The proposed approach can also be used in other studies like Brandt and Wu (2002) where option parameters are estimated from liquid European options and then applied to price less liquid and exotic derivatives. In addition, it allows estimation of term and moneyness structure in skewness and kurtosis which is essential for value-at-risk analysis (see Das and Sundaram, 1991, p. 212).

4. Functional Estimation for Options Pricing Via Support Vector Machines

Abstract

The focus of this essay is to explore the pricing performance of Support Vector Machines for pricing S&P 500 index call options. SVM is a novel nonparametric methodology that has been developed in the context of statistical learning theory and until now it has been practically neglected in financial econometric applications. This new method is compared with feedforward Artificial Neural Networks and also with Parametric Options Pricing Models using standard implied parameters and parameters derived via Deterministic Volatility Functions. The empirical analysis has shown promising results for the SVMs.

4.1. Introduction

The Black and Scholes (BS) (1973) model is considered as the most prominent achievement in the options pricing theory. Empirical research has shown that the formula suffers from systematic biases known as the volatility smile/smirk anomaly, which is the result of the simplistic assumptions that underlie its pricing dynamics (see Black and Scholes, 1975, Rubinstein, 1994, Bates, 2003, Bakshi et al., 1997, Andersen et al., 2002). More elaborate POPMs that allow for stochastic volatility and jumps in their diffusion process have been introduced in an attempt to eliminate most of the BS biases (for a review see Bakshi et al., 1997). Although these models seem to produce more accurate pricing results compared to the BS model, yet they are quite challenging and complex when used for real time applications and none is so flexible enough to provide results fully consistent with the observed market data (Bates, 1996 and 2000, and 2003, Bakshi et al., 1997, Dumas et al., 1998, Hull and Suo, 2002, Eraker, 2004). This is why BS has shown severe time endurance and is still widely used by practitioners. In addition, simplistic extensions of BS like the Corrado and Su (CS) (1996) model and the use of BS in the context of Deterministic Volatility Functions (Dumas et al., 1998) generate quite accurate prices for a wide spectrum of European financial options (see also Hull and Suo, 2002).

Financial markets are complex and characterized by a stochastic (time interchanging) behavior resulting to multivariate and highly nonlinear option pricing functions. There is evidence indicating that market participants change their option pricing attitudes from time to time (i.e. Rubinstein, 1994). POPMs may fail to adjust to such rapidly changing market behavior (see also Cont and Fonseca, 2002, for evidence of noticeable variation in daily implied parameters) since they are relying on static dynamics regarding their diffusion process. There is a great quest for nonparametric techniques that can potentially alleviate the limitations of POPMs. In addition to this, market practitioners have always a need for more accurate option pricing models that can be utilized in real-world applications. Under such cases, nonparametric data driven models like SVM and ANNs are powerful candidates to be applied for options pricing.

ANNs are very popular for applications in financial and economic applications (see Tsay, 2002) at least for four reasons. First, there are theoretical foundations showing that ANNs can be used for multidimensional nonlinear regression since they are universal approximators able to approximate any nonlinear function and its derivatives arbitrarily well (see Cybenko, 1989, and Haykin, 1999). Second, they do not necessarily rely on any financial assumptions and can learn the empirical input/output relationships inductively using historical or implied input variables and transactions data. Third, they rely on fairly simple training algorithms. Fourth, their out of sample generalization performance is adequate as long as a large datasets are being used and nowadays this is feasible due to the abundance of historical transactions data provided by numerous vendors. Under these conditions, unavoidably ANNs have also found extensive applications in the options pricing area. The vast amount of empirical evidence from these applications show that ANNs can outperform the most widely used POPMs (like the BS for example), and that they can be more accurate and computationally more efficient alternatives when the underlying asset's price dynamics are unknown (see Hutchinson et al., 1994, Garcia and Gencay, 2000, refer also to results from the first essay).

Unlike ANNs, SVMs have not gained yet any significant popularity in financial econometric applications although they are widely used in electrical engineering, bioinformatics, pattern recognition, text analysis, computer vision etc (see Smola and Schölkoph, 1998, and references therein). SVMs have evolved in the framework of statistical learning theory (see Vapnik, 1995, for extensive details) and can be utilized for problems involving linear or nonlinear regression. The main advantage of SVMs over other nonparametric techniques is that they encompass statistical properties that enable them to generalize satisfactorily well to unseen data. Another significant characteristic is that under SVMs someone solves a convex optimization problem with a unique global (and sparse) solution while other nonparametric methods usually have non-convex error functions which entail the risk of having multiple local minima solutions. Another one significant feature is that SVMs employ the so called VC theory (see Vapnik and Chervonenkis, 1974), which is defined in a strictly statistical framework, that controls in specific ways the model's estimation and parameterization to preclude overfitting so as to ensure good out of sample (generalization) results. Based on the theory that underlies SVM, their potential superiority over ANNs should be more obvious in datasets of small and moderate size (see Vojislav, 2001). For this reason we employ SVM with training data sets that have short time spans.

In this study our main contribution is to develop SVM for pricing European options and to compare it with other alternative pricing approaches like ANNs and POPMs. The methodological framework can also be beneficial to practitioners for real time trading. We consider the traditional SVM for function approximation as originally developed by Vapnik based on the ε-insensitive loss function (see Vapnik, 1995) which is considered to be more robust when noise is non Gaussian. In addition, we consider the Least Squares Support Vector Machines (LS-SVM) which is a subsequent variant of the original SVM methodology, originally proposed by Suykens and co-workers (see Suykens et al., 2002). Compared to SVMs, LS-SVMs can be more robust when noise is Gaussian, they rely on fewer tuning hyper-parameters that can expedite the estimation process and minimize a least squares loss function which is most common in empirical options pricing studies (see Christoffersen and Jacobs, 2004).

To our knowledge this is the first time that such a comprehensive application is considered[43] for options pricing. In this study we estimate ANNs and SVM using two different target functions (desired output). One that approximates the unknown empirical options pricing function explicitly by modeling the market prices of the call options (called the market target function) and one implicitly by modeling the residual between the actual call market price and the parametric option price estimate (called the hybrid target function). These target functions have been also considered in the first two essays of this thesis. We compare them with the parametric BS and CS models using overall average implied parameters and contract specific implied volatility versions derived by the DVF method. Moreover, as an additional benchmark model we use the Stochastic Volatility and Jump (SVJ) model of Bates (1996) since literature documents that it can be an effective remedy to the BS biases (see Bakshi et al., 1997 and Bates, 1996) and can provide

---

[43] There are some studies that apply the SVM in financial time series. Müller et al. (1999) apply the SVM for approximating the noisy Mackey-Glass system and the Santa Fe Times Series Competition (set D). Gestel et al. (2001) apply LS-SVM for one-step ahead prediction of the weekly 90-day T-bill rate and the daily DAX30 closing prices. Cao and Tay (2003) apply SVM to forecast the five day relative difference in percentage of price for five futures contracts.

significantly better pricing results compared to the stochastic volatility model of Heston (1993).

In the following we first review the parametric models, and the standard and hybrid ANN, SVM and LS-SVM models. Then we discuss the dataset and the methodologies employed to get the implied parameter estimates. Subsequently we review the numerical results.

4.2. The parametric models used

Below we briefly discuss the different POPMs that we employ in this study. The first model examined is the Black and Scholes (1973) since is a benchmark and widely referenced model. The Black Scholes formula for European call options modified for dividend-paying (see also Merton, 1973) underlying asset is:

$$c^{BS} = Se^{d_y T} N(d) - Xe^{-rT} N(d - \sigma\sqrt{T}) \qquad (4.1)$$

$$d = \frac{\ln(S / X) + (r - d_y)T + (\sigma\sqrt{T})^2 / 2}{\sigma\sqrt{\mathrm{T}}} \qquad (4.1.1)$$

where c^{BS} is premium paid for the European call option, S is the spot price of the underlying asset, X is the exercise price of the call option, r is the continuously compounded risk free interest rate, d_y is the continuous dividend yield paid by the underlying asset, T is the time left until the option expiration date, σ^2 is the yearly variance rate of return for the underlying asset and $N(.)$ stands for the standard normal cumulative distribution .

The Corrado and Su (1996) model is an extension of the BS model that accounts for additional skewness and kurtosis in stock returns and is used as a benchmark in this essay. Using a Gram-Charlier series expansion of a normal density function Corrado and Su defined their model as (see also the correction in Brown and Robinson, 2002):

$$c^{CS} = c^{BS} + \mu_3 Q_3 + (\mu_4 - 3)Q_4 \tag{4.2}$$

where c^{BS} is the BS value for the European call option given in Eq. (4.1) and,

$$Q_3 = \frac{1}{3!} S e^{-d_y T} \sigma\sqrt{T}((2\sigma\sqrt{T} - d)n(d) + (\sigma\sqrt{T})^2 N(d)) \tag{4.2.1}$$

$$Q_4 = \frac{1}{4!} S e^{-d_y T} \sigma\sqrt{T}((d^2 - 1 - 3\sigma\sqrt{T}(d - \sigma\sqrt{T}))n(d) + (\sigma\sqrt{T})^3 N(d)) \tag{4.2.2}$$

In Eq. (4.2) Q_3 and Q_4 represent the marginal effect of non-normal skewness and kurtosis, respectively in the option price whereas μ_3 and μ_4 correspond to coefficients of skewness and kurtosis. In the above expressions,

$$n(z) = \frac{1}{\sqrt{2\pi}} \exp(-z^2 / 2) \tag{4.2.3}$$

refers to the standard normal probability density function.

In addition to the above models, we also employ as a benchmark the SVJ model of Bates (1996). In their study Bakshi et al. (1997) found that the SVJ exhibited satisfactory out of sample pricing performance for the S&P 500 index options when compared to other parametric option pricing models since it offers a quite flexible distributional structure able to capture the negative skewness and excess kurtosis implicit in the market returns. Under this model the underlying asset follows geometric jump diffusion with the instantaneous conditional variance V_t to follow a mean-reverting square root process:

$$\frac{dS}{S} = (\mu - \lambda\bar{\kappa})dt + \sqrt{V}dZ + \kappa dq \tag{4.3}$$

$$dV = (\alpha - \beta V)dt + \sigma_v \sqrt{V} dZ_v \tag{4.4}$$

with

$$\text{cov}(dZ, dZ_v) = \rho dt$$

$$\ln(1+\kappa) \sim N(\ln(1+\bar{\kappa}) - 0.5\theta^2, \theta^2)$$

$$prob(dq = 1) = \lambda dt$$

where μ is the instantaneous drift of the underlying asset, λ is the annual frequency of jumps, κ is the random percentage jump conditional on a jump occurring, q is a Poison counter with intensity λ, θ^2 is the jump variance, and ρ is the correlation coefficient between the volatility shocks and the underlying asset movements. Moreover, β is the rate of mean reversion and α / β is the variance steady-state level (long run mean).

The value of a European call option is given as a function of state variables and parameters:

$$c^{SVJ} = e^{-rT}[F\Pi_1 - X\Pi_2] \tag{4.5}$$

with $F = E(S_T) = Se^{(r-d_y)T}$ to be the forward price of the underlying asset, with $E(.)$ to be the expectation with respect to the risk-neutral probability measure and S_T the price of S at option's maturity. Evaluation of Π_1 and Π_2 is done under the distributional assumptions embedded in the risk-neutral probability measures by using the moment generating functions of $\ln(S_T / S)$. The following expressions are needed to compute Π_1 and Π_2:

$$F_j(\Phi \mid V,T) = \exp\{C_j(T;\Phi) + D_j(T;\Phi)V + \lambda T(1+\bar{\kappa})^{\mu_j + 0.5} \times [(1+\bar{\kappa})^{\Phi} e^{\theta^2(\mu_j \Phi + \Phi^2/2)} - 1]\} \qquad j = 1, 2 \tag{4.6}$$

$$C_j(T;\Phi) = (r - d_y - \lambda\bar{\kappa})\Phi T - \frac{\alpha T}{\sigma_v^2}(\rho\sigma_v\Phi - B_j - G_j) - \frac{2\alpha}{\sigma_v^2}\ln\left[1 + 0.5(\rho\sigma_v\Phi - B_j - G_j)\frac{1-e^{G_j T}}{G_j}\right] \tag{4.7}$$

$$D_j(T;\Phi) = -2\frac{\mu_j\Phi + 0.5\Phi^2}{\rho\sigma_v\Phi - B_j + G_j\dfrac{1+e^{G_j T}}{1-e^{G_j T}}} \tag{4.7.1}$$

$$G_j = \sqrt{(\rho\sigma_v\Phi - B_j)^2 - 2\sigma_v^2(\mu_j\Phi + 0.5\Phi^2)} \tag{4.7.2}$$

$$\mu_1 = 0.5, \quad \mu_2 = -0.5, \quad B_1 = \beta - \rho\sigma_v, \quad B_2 = \beta \tag{4.7.3}$$

and the resulting probabilities Π_1 and Π_2 are derived by numerically evaluating the imaginary part of the Fourier inversion:

$$prob(S_T e^{(r-d_y)T} > X \mid F_j) = 0.5 + \frac{1}{\pi}\int_0^{\infty}\frac{imag[F_j(i\Phi)e^{-i\Phi\chi}]}{\Phi}d\Phi \tag{4.8}$$

with $\chi \equiv \ln(X/S)$ and the integrals to be evaluated with an adaptive Lobatto quadrature.

Here we must note that such complicated models are not widely used by traders for pricing options (see Hull and Suo, 2002, p. 300). Traders usually rely on simpler models and more intuitive methodologies that are closer to BS model used under the DVF approach which are able to handle contract specific implied parameters (see also Brandt and Wu, 2002).

In conjunction to the above, the DVF approach which was proposed by Dumas et al. (1998) for deriving per contract volatility for the BS model comprises a practical approach in order to mitigate the volatility smile anomaly of the BS model. Berkowitz (2004) demonstrates theoretically that the DVF constitutes a reduced-form approximation to an unknown structural model which under frequent re-estimation can exhibit exceptional pricing performance. For our analysis we estimate the following DVF specification:

$$\text{DVF:} \quad \sigma = \max(0.01, a_0 + a_1 X + a_2 X^2 + a_3 T + a_4 XT) \qquad (4.9)$$

Empirical results from the third essay have shown that the above specification seems to work well for the data under consideration. We also implement the DVF for deriving a per contract volatility estimate for the CS model since it can produce even more accurate results compared to the BS based DVF version (refer to third essay for empirical results).

4.3. The nonparametric approaches: ANNs, SVM and LS-SVM

ANNs comprise a popular methodology for handling function estimation problems for many reasons. First of all, theoretical proofs exist showing that under certain conditions, ANNs are universal approximators able to approximate any nonlinear function arbitrarily well (Cybenko, 1989). In addition, they perform well in situations where there is lack of knowledge for the relationship that underpins a set of variables and they are robust on the presence of noisy data. Nevertheless, they are potentially prone to some practical merits and limitations. First, there are no theoretical foundations on how to select the network type and structure and on how to implement the optimization procedure. For this reason the model structure (number of neuron layers and number of hidden neurons in each of the layers) should be defined a-priori which is not necessarily the best strategy in choosing the optimal network architecture for the faced problem. This task is rather an art instead of science and can be better tackled by experts with experience on how to apply the ANNs methodology having at the same time considerable knowledge on the problem under investigation. Second, estimating ANNs involves the optimization of a highly non-convex error function and frequently enough optimization algorithms get stuck to local minima solutions resulting to suboptimal solutions for the network free parameters (weights and biases)[44]. Given this peculiarity, regularization techniques that are employed in an attempt to control the capacity of the ANNs, like cross validation strategies[45] and early stopping[46], are only partial remedies potentially resulting to structures that do not maximize their generalization performance to unseen data. Finally ANNs learn the empirical functions inductively from transactions data without embedding any information related to the problem under investigation. Under this setting, the estimated weights do not convey any meaningful interpretation to help understand better the input-output relationship. In a nutshell, significant expertise is needed in order to develop ANNs that can be trusted regarding their out of sample pricing estimates.

In contrast, SVM are not confined by the above issues (Smola and Schölkoph, 1998). First of all, the model complexity does not need to be determined a-priori. It is determined endogenously as part of the optimization problem in such a way that maximizes the generalization capability of the model. More importantly, a unique solution is found after estimation as a solution of a (convex) Quadratic Programming (QP) problem with linear constraints, which depends on the estimating data and the selection of few tuning hyper-parameters. In addition, the solution to the QP problem provides the necessary information for choosing the most important datapoints, known as support vectors, among all the data; based on the SVM formulation,

[44] One way to circumvent this is to estimate a predetermined network structure several times always starting with different initial weights and biases and selecting the model with the least error.

[45] Cross-validation is a classical statistical tool for resolving the trade-off between the performance on training data and the complexity of a model. The basic idea of the cross-validation is founded on the fact that good-results taken from the data used for estimation does not necessarily ensure good performance to a testing set with unseen data. To implement the cross validation, a particular dataset is divided in training and validation subsets of data. A set of alternative models is estimated using the training dataset by exploring a meaningful grid of possible parameter combinations in the case of SVM and LS-SVM or by varying the number of hidden neurons in the case of ANNs. Then the model that produces the least error (based on a predefined norm) in the validation subset is considered as the one that can perform the best out of sample using unseen data.

[46] Early stopping is another regularization technique used to control the capacity of an ANN. During the nonlinear optimization of an ANN the error in the estimating (training) data is generally monotonically decreasing as a function of the number of iterations of the algorithm employed. This does guarantees that the error to the validation dataset will also decrease. What usually happens is that the error in the validation dataset is decreasing at the start and then it starts increasing since the ANN tends to memorize (overfit) the estimating data. Early stopping is employed to stop training an ANN at the point with the lowest error in the validation sample with the hope of maximizing the network generalization ability in the testing dataset.

support vectors uniquely define the estimated regression function so in this manner the estimated coefficients are informative. Furthermore, input data of any arbitrary dimensionality can be treated with only linear cost in the number of input dimensions. This property in conjunction with the good inherent regularization properties allows SVM to work particularly well when data is sparse (see Müller et al., 1999, Smola and Schölkoph, 1998). Yet, the performance of the SVM technique, like ANNs, depends crucially by the choice of the loss function which is inextricably connected with the noise in the data (Gaussian or not) and by other data regularities (e.g. non-stationary financial data). In the Appenidx that appears at the end of this essay, we discuss in detail the theory behind the SVMs and we demonstrate the programming formulations for estimating the SVMs and LS-SVMs (main refernces for this are Vapnik, 1995, Smola and Schölkoph, 1998 and Suykens et al., 2002). In the following section, we briefly demonstrate only the essensial programming formulations for the methodologies used in this essay.

4.3.1. Feedforward artificial neural networks

A feedforward artificial neural network is a collection of interconnected processing elements structured in successive layers and is usually depicted as a network of links (termed as *synapses*) and nodes (termed as *neurons*) between layers. A typical feedforward neural network has an input layer, one or more hidden layers and an output layer. The ANNs used in this study have three layers: an *input* layer with N input variables, a *hidden* layer with H neurons, and an output layer with a single neuron. A particular neuron is composed of: *i)* the vector of *input signals*, *ii)* the *vector weights* and the associated *bias*, *iii)* the *neuron* itself that *sums* the product of the input signal with the corresponding weights and bias, and finally, *iv)* the neuron *transfer function* (commonly known also as *activation function*). Each connection is associated with a *weight*, w_{ki}, and a *bias*, b_k, in the hidden layer and a *weight*, υ_k, and a *bias*, υ_0, for the output layer (k = 1,2,…,H, i = 1,2,…,N). In addition, the outputs of the hidden layer are the inputs for the output layer (thus the term feedforward). The operation carried out for estimating output y, is the following:

$$y = f_0\left(v_0 + \sum_{j=1}^{H} v_j f_H\left(b_j + \sum_{i=1}^{N} w_{ji} x_i\right)\right) \qquad (4.10)$$

The weights and biases are adjusted according to the faced problem via optimization (the training algorithm). Their particularity relies on the fact that the neurons on each layer operate collectively and in a parallel manner on all input data.

For the purpose of this study, the hidden layer always uses the hyperbolic tangent sigmoid transfer function, while the output layer uses a linear transfer function. In addition, ANN architectures with only one hidden layer are considered since they operate as a nonlinear regression tool and can be trained to approximate most functions arbitrarily well (Cybenko, 1989). High accuracy can be obtained by including enough processing nodes in the hidden layer. Moreover, we also utilize the network initialization technique proposed by Nguyen and Windrow (see Hagan et al., 1996) that generates initial weights and bias values for a nonlinear transfer function so that the active regions of the layer's neurons are distributed roughly evenly over the input space.

Estimating the networks free parameters is done by minimizing the following sum of squares loss function:

$$\arg\min_{w} \sum_{k=1}^{P} e_k^2 = \sum_{k=1}^{P}\left(y_q - f_0\left(v_0 + \sum_{j=1}^{H} v_j f_H\left(b_j + \sum_{i=1}^{N} w_{ji} x_i\right)\right)\right)^2 \qquad (4.11)$$

where, w is a ν-dimensional column vector with the weights and biases given by: $w = [b_1, \ldots, b_H, w_{11}, \ldots, w_{HN}, v_0, \ldots, v_H]^T$.

Optimization of the loss function in Eq. (4.11) is done with the Levenberg-Marquardt algorithm (further technical details about the implementation of this algorithm for ANNs can be found in the first three essays of this thesis and also in Hagan et al., 1996).

4.3.2. ε-insensitive support vector machines for function approximation

The application of SVM for regression was initially developed only for performing linear regression. The technique has been extended to handle nonlinear regression applications based on a very intuitive idea (see Vapnik, 1995, 1998). First, apply a mapping $\varphi(x)$ (chosen a-priori) of the input data x into an arbitrarily high dimensional feature space which can be (possibly) infinite dimensional. This transformation is usually called the *kernel trick*. Second, the linear SVM regression can be applied to create an approximate linear function in this arbitrarily high dimensional feature space. In this way, doing linear regression in a high dimensional feature space corresponds to nonlinear regression in the (low dimensional) input space (Müller et al., 1999).

The idea behind the SVM for function approximation (Support Vector Regression) is to estimate the coefficient values w (called the weights) and b (called the bias) that optimize the generalization ability of our regressor by minimizing the following regularized loss function:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{j=1}^{P} L_{\varepsilon}\left(t_j, f(x_j)\right) \tag{4.12}$$

where $f(x)$ is the form of the SVM function approximation and is given by:

$$f(x) = w^T \varphi(x) + b \tag{4.13}$$

and $L_{\varepsilon}(t, f(x)$ is the so-called Vapnik's ε-insensitive loss functions defined as:

$$L_{\varepsilon}(t, f(x)) = |t - f(x)|_{\varepsilon} = \begin{cases} 0 & \text{if } |t - f(x)| \le \varepsilon \\ |t - f(x)| - \varepsilon & \text{otherwise} \end{cases} \tag{4.14}$$

In the above formulations $\varphi(x):\Re^N \rightarrow \Re^{N_h}$ represents a nonlinear mapping (transformation) of the input space to an arbitrarily high-dimensional feature space which can be infinite dimensional (in such case the weights vector w will also become infinite dimensional). The constant $C>0$ determines the trade-off between the amount up to which deviations larger than ε are tolerated and the flatness (complexity) of the estimated model. In the case where ε is chosen to be small and some datapoints do not lay within the tube of ε accuracy the estimation of the w and b is done by formulating the following optimization problem in the primal weight space of the unknown coefficients:

$$\min_{w,b,\xi,\xi^*} \quad L_P(w,\ \xi,\ \xi^*)=\frac{1}{2}w^T w + C\sum_{j=1}^{P}\left(\xi_j+\xi_j^*\right) \tag{4.15}$$

subject to

$$t_j - w^T\varphi(x_j) - b \le \varepsilon + \xi_j\,, \quad j=1,\ldots,P \tag{4.15.1}$$

$$w^T\varphi(x_j) - t_j + b \le \varepsilon + \xi_j^*\,, \quad j=1,\ldots,P \tag{4.15.2}$$

$$\xi_j, \xi_j^* \ge 0, \quad j=1,\ldots,P \tag{4.15.3}$$

where ξ_j and ξ_j^* are defined in the prime space, that need to be introduced in order to make the solution of the optimization of the optimization problem feasible for all datapoints that are outside the ε-tube.

Transforming the above into its dual formulation[47] and after applying the kernel trick results to the following quadratic programming problem that depends only by the dual variables α and α^* (see the Appendix for details):

---

[47] In nonlinear regression problems the primal weights vector w can become infinite dimensional due to the applied transformation $\varphi(x)$. For this reason the solution of the problem is better derived via its dual formulation.

$$\max_{\alpha,\alpha^*} \quad L_D(\alpha,\ \alpha^*) = -\frac{1}{2}\sum_{j,i=1}^{P}\left(a_j - a_j^*\right)\left(a_i - a_i^*\right)K\left(x_j, x_i\right) - \varepsilon\sum_{j=1}^{P}\left(a_j + a_j^*\right) + \sum_{j=1}^{P} t_j\left(a_j - a_j^*\right) \qquad (4.16)$$

subject to

$$\sum_{j=1}^{P}\left(a_j - a_j^*\right) = 0 \qquad (4.16.1)$$

$$0 \le a_j, a_j^* \le C \qquad (4.16.2)$$

with:

$$f(x) = \sum_{j=1}^{P}\left(a_j - a_j^*\right)K\left(x, x_j\right) + b \qquad (4.17)$$

and:

$$b = \frac{1}{N}(t_i - \sum_{j=1}^{N}\left(a_j - a_j^*\right)K\left(x_j, x_i\right) + \varepsilon\,\mathrm{sign}\left(a_i - a_i^*\right) \quad \text{for } \alpha_j, \alpha_j^* \in \left(0, C\right)$$

(4.18)

To successfully apply the SVMs for nonlinear regression problems it is necessary to apply the kernel trick by choosing a proper kernel function:

$$K\left(x_j, x_i\right) = \varphi(x_j)^T \varphi(x_i) \qquad (4.19)$$

A function that is symmetric, continuous and satisfies Mercer's condition (see Vapnik, 1995 for details) is an admissible kernel function that represent a scalar product in the (mapped) featured space as expressed in Eq. (4.19). The

Gaussian kernel is a widespread kernel function that is admissible for use with SVM for function approximation:

$$K(x_j, x_i) = exp\left(-\frac{\|x_j - x_i\|^2}{2\sigma_K^2}\right) \tag{4.20}$$

where $\|x_j - x_i\|^2$ measures the distance between two datapoints and σ_K^2 is called the kernel width parameter and is used as a normalizing factor. It can be shown that when the Gaussian kernel function is considered, the nonlinear mapping $\varphi(x_j)$ is infinite dimensional and also that SVM are universal approximators (see Vapnik, 1995 and 1998 for details), an implication of paramount importance that is contributing to a growing popularity of SVM for regression applications.

The application of SVMs in general preserves some very helpful characteristics compared to other learning techniques (e.g. feedforward artificial neural networks,). The system of equations defined by Eqs. (4.16), (4.16.1) and (4.16.2) given a positive definite kernel translates to the optimization of a convex QP problem subject to linear constraints that results in a global and unique solution. On the contrary, feedforward artificial neural networks suffer from existence of multiple local minima solutions[48] since the optimization function is not convex with respect to the network weights and biases[49]. Second, after selecting the SVM tuning parameters ($C, \varepsilon, \sigma_K^2$), the model complexity is implicitly defined by the number of support vectors as

---

[48] Among others, Cybenko (1989) has shown that ANNs with one hidden layer of neurons can be universal function approximators that provide adequate robustness and convergence with good out of sample generalization abilities. However, this property can be of limited use in practice when the optimization algorithm gets stuck in local minima resulting to a suboptimal solution because of the non-convexity of the optimized error/loss function. In this study ANNs are implemented under certain strategies like early stopping and use of cross-validation techniques that try to eliminate the effect of local minima solutions and overfitting of the data.

[49] For ANNs one would have a convex problem if one would fix a number of hidden layer weights and one would compute the output layer's weights (with linear characteristic at the output) from a sum of square error cost function (Suykens et al., 2002).

part if the solution to the convex problem, whilst for the case of the ANNs the number of hidden neurons should be defined a-priori. Third, the solution to the problem is characterized by a sparse representation of the solution. As explained earlier, the final solution is defined solely by the support vectors which represent only a part of the datapoints used initially for the estimation of the model. Another important issue is that the function's representation is independent of the dimensionality of the input space and depends only on the number of support vectors; in other words the size of the QP problem does not depend on the dimensionality of the input space. This is a significant remedy for the curse of dimensionality issue. On the contrary, ANNs are prone to the effects of the curse of dimensionality. In this case, early stopping and cross validation techniques should be very carefully applied by an expert in an attempt to overcome the curse of dimensionality by preventing the networks from memorizing the data used for estimation and to result to a limited or a poor generalization performance (Vojislav, 2001, Suykens et al., 2002).

4.3.3. Least squares support vector machines

The Least Squares Support Vector Machines method is a variant of the original SVM methodology originally proposed and developed by Suykens and co-workers (see Suykens et al., 2002). According to this approach the model estimated is given by the following optimization problem in the primal weight space:

$$\min_{w,b,e} \quad L_P(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{j=1}^{P} e_j^2 \qquad (4.21)$$

subject to

$$t_j = w^T \varphi(x_j) + b + e_j \,, \quad j = 1, \ldots, P \qquad (4.21.1)$$

The above formulation is nothing else but a ridge regression cost function formulated in the featured space defined by the mapping $\varphi(x)$. Parameter γ determines again the trade-off between the model complexity and goodness of fit to the estimation data. As in the case of SVM (see Suykens et al., 2002, pg. 98), after applying the kernel trick we obtain the following linear KKT system in a and b (see the Appendix for details):

$$\sum_{j=1}^{P}\left(\alpha_j K(x_j, x)\right) + b + \frac{a_j}{\gamma} = t_j, \quad j = 1, \ldots, P \tag{4.22}$$

$$\sum_{j=1}^{P}\alpha_j = 0 \tag{4.23}$$

where the resulting LS-SVM model that characterizes the estimated regression function is given by:

$$f(x) = \sum_{j=1}^{P} a_j K\left(x, x_j\right) + b \tag{4.24}$$

Compared to the SVMs case, LS-SVMs preserve the following characteristics. First, the Gaussian kernel function given by Eq. (4.20) can be used in this case too. Second, the dual problem above corresponds to solving a linear KKT system which is a square system with a unique (global) solution when the matrix has full rank. Third, the error variable e_j is used to control deviations from the regression function instead of the slack variables ξ_j, ξ_j^* and a squared loss function is used for this error variable instead of the ε-insensitive loss function. This has two implications regarding the solution of the problem: *i) lack of sparseness* since every data point will now be a support vector, something that can be considered as a drawback compared to the SVM, *ii)* only two parameters γ and σ_K^2 are needed to be tuned compared to three for SVM which is an advantage since it reduces the possible parameters combinations (2-D grid instead of 3-D) and at the same time reduces the risk of selecting a suboptimal parameter combination. Due to the reasons explained above, optimizing a set of LS-SVM models can be potentially faster compared to standard SVMs.

Regarding the estimation process, for SVMs we use the support vector training algorithm proposed by Vishwanathan et al., (2003) while for the LS-SVMs we use a MATLAB® toolbox prepared by Suykens and co-authros (see Suykens et al., 2002 for details). Finally we note that a *z*-score (mean-standard deviation) scaling was applied to all input and output variables during the estimation of all nonparametric models.

4.4. Data and methodology

4.4.1. Data and filtering rules

Our dataset covers the period January 2002 to August 2004 for a total of 671 trading days. The S&P 500 index call options are used because this option market is extremely liquid. They are the most popular index options traded in the CBOE and the closest to the theoretical setting of the parametric models (see Garcia and Gencay, 2000 and Constantinides et al, 2008). Each trading day we have the last available bid and ask call price, along with the strike price[50], date of expiration[51], volume and open interest. In our analysis we use the midpoint of the call option bid-ask spread since as noted by Dumas et al. (1998), using bid-ask midpoints rather that trade prices reduces noise in the cross sectional estimation of implied parameters. Each day the midpoint of the call option bid ask spread at the close of the market, c^{mrk}, is matched with the closing value of S&P 500 index[52].

---

[50] For the purposes of this study we use the following moneyness categories: *deep out the money* (DOTM) when $S/X \le 0.90$, *out the money* (OTM) when $0.90 < S/X \le 0.95$, *just out the money* (JOTM) when $0.95 < S/X \le 0.99$, *at the money* (ATM) when $0.99 < S/X \le 1.01$, *just in the money* (JITM) when $1.01 < S/X \le 1.05$, *in the money* (ITM) when $1.05 < S/X \le 1.10$, *deep in the money* (DITM) when $S/X > 1.10$.

[51] In terms of time length, an option contract is classified as *short term maturity* (when maturity ≤ 60 calendar days), as *medium term maturity* (when maturity is between 61 and 180 calendar days) and as *long term maturity* (when maturity > 180 calendar days).

[52] Data synchronicity should be minimal issue for this highly active market (see also Garcia and Gencay, 2000). Among others, Christoffersen and Jacobs (2004) and

To create an informative dataset we rely on the following filtering rules (see also Bakshi et al., 1997): We first eliminate all observations that have zero trading volume since they do not represent actual trades. Second, we eliminate observations that violate either the lower or the upper arbitrage options bounds. Third, we eliminate all options with less than six or more than 260 days to expiration to avoid extreme option prices that are observed due to potential illiquidity problems. Similarly, price quotes of less than 1.0 index points are not included. Finally, we demand at least four datapoints per maturity to secure that during the implied parameters extraction process, every maturity period is satisfactorily represented. The final dataset has a total of 37202 which 21644 are used in the testing dataset. The data used in this essay are similar to those used in the third one; thus sample characteristics and other descriptives can be found in Table 3.2.

4.4.2. Splitting the data

We must first consider two issues; one regarding which dataset to use for estimating our nonparametric models and one regarding the use of a cross validation method when selecting our models. Regarding the first issue, until now previous studies that apply ANNs for options pricing use long data periods that result in large datasets. This is imperative to properly estimate and select the best ANN models when high out of sample pricing accuracy is requested (see Hutchinson et al., 1994, Garcia and Gencay, 2000, and references therein). Contrary to this, for reasons explained earlier, SVM are potentially more powerful when used with small datasets. They can be estimated very fast when the dataset is small. Moreover, using shorter time horizons might be beneficial in capturing the fast changing market conditions which are probably missed with long time horizons. Using shorter time horizons makes the estimation of SVM more competitive to POPMs which are usually calibrated on a daily basis. For these reasons we use a chronological data partitioning via a rolling-forward procedure.

Regarding the second issue, using cross validation is almost always needed since this is an effective, yet heuristic, way of controlling the capacity

---

Chernov and Ghysels (2000) use daily closing prices of European call options written on the S&P 500 index.

of ANNs. Model capacity for SVM is part of the optimization problem but cross-validation may be needed so as to properly select the tuning hyper-parameters to ensure high out of sample accuracy. On the other hand when the grid of possible tuning hyper-parameters is selected based on prior knowledge then it may be possible to have good out of sample performance without using a validation dataset.

For the case of SVM and LS-SVM we have conducted a pilot study using data from 2002 in order to determine areas of the tuning parameters values that result to models which performed well out of sample. For SVM we restrict our attention to the following hyper-parameter values resulting in a total of 40 possible combinations per training sample:

$$C \in (10, 50, 100, 200)$$
$$\varepsilon \in (0.025, 0.05)$$
$$\sigma_K \in (1.00, 2.50, 5.00, 7.50, 10.00) \quad (4.25)$$

For LS-SVM we restrict our attention to the following hyper-parameter values resulting to a total of 30 possible combinations per training sample:

$$\gamma \in (10,\ 100, 250, 500, 750, 1000)$$
$$\sigma_K \in (10, 20, 30, 40, 50) \quad (4.26)$$

The above allows us to limit the possible number of hyper-parameter combinations for the (testing) data in 2003-2004. We must note though that the above naïve selection does not guarantee that our SVM and LS-SVMs will be optimized in the best possible way and that will result to the overall best out of sample pricing estimates. There are other more sophisticated and structured methodologies for hyper-parameter selection (Cherkassky and Ma, 2004). Such methodologies will be considered in future work.

For ANNs all networks having two to ten hidden neurons (in total nine) are examined per training case. Moreover, for a certain number of hidden neurons the network is (re)initialized and (re)trained 10 separate times in an attempt to minimize the potential problems of obtaining weights and parameters that result from a local minimum (in total 9x10=90 different model/parameter estimations per input/output variable combination and per period). In addition, for a certain number of hidden neurons an early-stopping strategy is also adopted as a second measure in avoiding overfitting.

Regarding the data splitting we estimate SVM and LS-SVMs, our estimating (training) sample is always by using one month of data (around 23 trading days) and our validation sample is always 5 trading days. After estimating all possible model combinations using the hyper-parameter combinations in Eq. (4.25) and (4.26) the model with the least Root Mean Squared Error (RMSE) in the validation dataset is chosen and used for out of sample pricing for the 5 trading days following the validation sample.

For ANNs we follow two different ways for splitting the data. In the first case we comply with previous studies that use rather long periods for training and validation. As like in the third essay of this thesis we use twelve months of data for training and two months for validation. After estimating all possible model combinations the model with the least RMSE in the validation dataset is chosen and used for out of sample pricing for the following one month of data. In the second data splitting we use much less data points to be compatible with the first splitting used for SVM and LS-SVM, that is one month for training and 5 days for validation.

In this essay the period March 2003 to August 2004 is a period where we can get out of sample pricing estimates from all models. For this period we have 21644 datapoints for which we compute and tabulate: the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE), the Median Absolute Error (MdAE) and the 5th Percentile of Absolute Error (P_5AE) and 95th Percentile of Absolute Error ($P_{95}AE$). The focus of our analysis will be based on the RMSE measure since Bates (2000, p. 202) points out that RMSE is a relatively intuitive error measure and is useful for comparison with other work.

4.4.3. Implied parameters

The methodology employed here for the estimation of the overall average implied parameters is similar to that in previous studies (Bakshi et al., 1997, Christoffersen et al., 2006) that adopt the Whaley (1982) simultaneous equation procedure to minimize a price deviation function with respect to the unobserved parameters. Market option prices (c^{mrk}) are assumed to be the corresponding POPM prices (c^k) plus a random additive disturbance term (ε^k), k = BS, CS, or SVJ:

$$c^{mrk} = c^k + \varepsilon^k \tag{4.27}$$

To find optimal implied parameter values per model k we solve an optimization problem that has the following form:

$$SSE(t) = \min_{\xi^k} \sum_{j=1}^{P_t} (\varepsilon_j^k)^2 \tag{4.28}$$

where P_t refers to the number of different call option transaction datapoints available in day t, ξ^k the unknown parameters associated with a specific POPM (k = BS, CS, and SVJ). The SSE is minimized via a Nonlinear Least Squares optimization based again on the Levenberg-Marquardt algorithm. To minimize the possibility of obtaining implied parameters that correspond to a local minimum of the error surface with each model we use a variety of starting values for the unknown parameters based on daily average values reported in previous literature (for further technical details and numerical results refer to the third essay of this thesis).

In addition to the above daily overall average (*av*) implied parameters, we also estimate the DVF volatility estimates using a similar optimization process. For BS this is straightforward; for CS we first estimate the overall average implied parameters and then we fix skewness and kurtosis to compute the contract specific implied volatility based on the volatility structure given in Eq. (4.9). Finally we note that for pricing at time instant t, the implied structural parameters derived at day t -1 are used together with all other needed information based on day t. Daily recalibration of the implied parameters (DVF and overall average) for POPMs is also adopted by Bakshi et al. (1997) and Christoffersen and Jacobs (2004) (see also Hull and Suo, 2002 and Berkowitz, 2004).

4.4.4. The set of alternative models

With the BS models we use as input S, X, T[53], d_y[54], r[55], and any of the following ten volatility estimates: σ_j^{BS} where j = {av, DVF} with BS_j denoting the alternative BS parametric models. Similarly we denote the parametric CS alternatives (CS_j). Finally note that for the SVJ model we only use the overall average parameter estimates (SVJ). All these POPMs are used as benchmark models.

The dividend adjusted moneyness ratio $(Se^{-d_y T})/X$ and time to maturity (T) are always inputs to the nonparametric models. For nonparametric models we have two different target functions. The market target function which represents actual market prices of call options and the hybrid one represents the residual between the actual call market price and the parametric option price estimate.

The notation here depends on the additional inputs that are used from the parametric models. We use SVM_j^i, with j={av, DVF} to denote the SVM that use as additional input variable the BS volatilities: σ_{av}^{BS} and σ_{DVF}^{BS}. In addition, we use i={M, H} for subscript for denoting the nonparametric models that are estimated based on *Market* and the *Hybrid* target function. In a similar fashion we use $LSSVM_j^i$ and ANN_j^i for the LS-SVM and ANNs models. In total we examine four SVM models and four LS-SVMs. In addition, we have four ANN models that are estimated with the twelve-two-one months data-plitting and another four ANN models that are estimated with the same short data splitting as with the SVMs and LS-SVMs.

[53] Time to maturity is computed assuming 252 days in a year.

[54] We have collected a daily dividend yield provided by Thomson Datastream. Jackwerth (2000) also assumes that the future dividends for the S&P 500 index can be approximated by a dividend yield.

[55] Previous studies have used 90-day T-bill rates as approximation of the interest rate. In this study we use nonlinear cubic spline interpolation for matching each option contract with a continuous interest rate, r, that corresponds to the option's maturity. For this purpose, 1, 3, 6, and 12 months constant maturity T-bills rates (collected from the U.S. Federal Reserve Bank Statistical Releases) were considered.

4.5. Analysis of pricing results

Extensive details and numerical results regarding the calibration of the parametric models in obtaining the best daily overall average (*av*) and DVF implied parameters are given in the third essay of this thesis. Moreover, in the same essay we can find details regarding the in sample implied mean values of their parameters as well as their in sample pricing performance.

We start our analysis with Table 4.1 that exhibits the out of sample performance of the parametric models. We see that the DVF based CS models provide better performance than the corresponding DVF based BS ones with about 14% improvement in the RMSE. Overall though, the SVJ model is the top performer in all metrics.

	BS_{av}	BS_{NL2}	CS_{av}	CS_{NL2}	SVJ
RMSE	3.285	2.008	2.245	1.766	1.498
MAE	2.579	1.186	1.709	1.257	1.071
MeAE	2.172	0.833	1.358	0.929	0.796
P_5AE	0.242	0.078	0.118	0.085	0.065
P_{95}AE	6.396	3.100	4.370	3.422	2.996

Table 4.1. Out of sample pricing performance of the parametric models
Error performance results (out of sample pricing) for all parametric models for the aggregate period March 3, 2003 to August 31, 2004.

We then look at the nonparametric models' results. We first concentrate our attention to Table 4.2 which exhibits the out of sample pricing for ANNs using the short (top panel) and the long (lower panel) data-splitting. The first observation is that the pricing performance with the short data-splitting is much better in the three out of the four models considered. Using so short data splittings is not the mainstream for financial applications of ANNs; thus this observation is rather new evidence indicating that under proper development ANNs can have good pricing performance in small datasets most probably because they can better capture the fast changing market conditions. Another important observation from this table is that always the hybrid models perform better than the ANNs estimated with the market target function; the hybrid ANNs (with the short data splitting) are competitive with the best POPMs. In addition, as expected, the models with σ_{DVF}^{BS} always perform better than the models with σ_{av}^{BS}. Specifically, ANN_{DVF}^{H}

is much better that any of the BS and CS DVF based models and its RMSE is also statistically non different to the RMSE of SVJ (see statitscs of Table 4.5 that tabulates standard two-tail *t*-tests and Johnson's, 1978, modified *t*-tests).

	ANN_{av}^{M}	ANN_{av}^{H}	ANN_{DVF}^{M}	ANN_{DVF}^{H}
	Tr: 1 month, **Vd:** 5 days, **Ts:** 5 days			
RMSE	1.922	1.859	1.784	1.517
MAE	1.425	1.379	1.281	1.071
MeAE	1.088	1.089	0.929	0.799
P_5AE	0.096	0.099	0.083	0.070
P_{95}AE	3.970	3.570	3.634	2.900
	Tr: 12 months, **Vd:** 2 months, **Ts:** 1 month			
RMSE	2.201	1.756	2.036	1.684
MAE	1.626	1.340	1.350	1.056
MeAE	1.271	1.086	1.003	0.745
P_5AE	0.120	0.095	0.091	0.067
P_{95}AE	4.391	3.446	3.581	2.924

Table 4.2: Out of sample pricing performance for ANNs
Error performance results (out of sample pricing) for all Artificial Neural Network models for the aggregate period March 3, 2003 to August 31, 2004. The symbol "*N*" is used to indicate that the ANN is estimated based on the market target function while "*H*" is used to indicate the hybrid target function. Morover the subscript "*av*" indicates that the Black and Scholes overall average implied volatility is used as an extra input to the ANNs while "DVF" indicates that the the Black and Scholes overall average implied volatility is used as an extra input. RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error, MeAE is the Median Absolute Error and P5AE (P95AE) is the 5th (95th) Percentile of Absolute Errors.

The out of sample results for SVMs (Table 4.3) and for LS-SVMs (Table 4.4) follow the same pattern as with ANNs. First, we always observe the hybrid models to perform considerably better than the models estimated with the market target function. Second, the models estimated with σ_{DVF}^{BS} as input perform better than the models estimated with σ_{60}^{BS}. At a first glance the RMSE results of these models are for most of the cases above 2.00 meaning that they are not competitive enough to the POPM results shown in Table 4.1; this happens for six out of eight cases. Only SVM_{DVF}^{H} and $LS-SVM_{DVF}^{H}$ have RMSE substantially lower than 2.00. Specifically the RMSE for SVM_{DVF}^{H} is equal to 1.623 and for $LS-SVM_{DVF}^{H}$ is equal to 1.594; these are statistically lower than the RMSE's of BS_{NL2} and CS_{NL2} but statistically higher than the RMSE's of SVJ.

	SVM_{av}^{M}	SVM_{av}^{H}	SVM_{DVF}^{M}	SVM_{DVF}^{H}
Tr: 1 month, **Vd:** 5 days, **Ts:** 5 days				
RMSE	5.944	2.656	2.361	1.623
MAE	2.519	1.684	1.467	1.119
MeAE	1.358	1.166	1.036	0.802
P_5AE	0.125	0.106	0.090	0.067
P_{95}AE	6.574	4.854	4.219	3.200

Table 4.3: Out of sample pricing performance for SVM
Error performance results (out of sample pricing) for all Support Vector Machines for the aggregate period March 3, 2003 to August 31, 2004. The symbol "*N*" is used to indicate that the ANN is estimated based on the market target function while "*H*" is used to indicate the hybrid target function. Morover the subscript "*av*" indicates that the Black and Scholes overall average implied volatility is used as an extra input to the ANNs while "DVF" indicates the Black and Scholes overall average implied volatility is used as an extra input. RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error, MeAE is the Median Absolute Error and P5AE (P95AE) is the 5th (95th) Percentile of Absolute Errors.

	$LS-SVM_{av}^{M}$	$LS-SVM_{av}^{H}$	$LS-SVM_{DVF}^{M}$	$LS-SVM_{DVF}^{H}$
Tr: 1 month, **Vd:** 5 days, **Ts:** 5 days				
RMSE	4.899	3.756	2.107	1.594
MAE	2.362	1.912	1.429	1.120
MeAE	1.348	1.163	1.010	0.801
P_5AE	0.122	0.104	0.088	0.072
P_{95}AE	6.942	5.158	4.180	3.243

Table 4.4: Out of sample pricing performance for LS-SVM
Error performance results (out of sample pricing) for all Least Squares Support Vector Machines for the aggregate period March 3, 2003 to August 31, 2004. The symbol "*N*" is used to indicate that the ANN is estimated based on the market target function while "*H*" is used to indicate the hybrid target function. Morover the subscript "*av*" indicates that the Black and Scholes overall average implied volatility is used as an extra input to the ANNs while "DVF" indicates the Black and Scholes overall average implied volatility is used as an extra input. RMSE is the Root Mean Square Error, MAE is the Mean Absolute Error, MeAE is the Median Absolute Error and P5AE (P95AE) is the 5th (95th) Percentile of Absolute Errors.

	1	2	3	4	5	6	7	8	9
1		3.176	6.020	2.998	6.090	-2.319	4.812	-1.214	5.257
2	-3.675		6.851	-0.705	8.744	-4.039	4.334	-6.583	6.234
3	-6.256	-7.253		-7.950	-0.491	-5.441	-2.870	-10.242	-2.464
4	-3.042	0.751	8.100		11.051	-3.947	5.441	-6.460	8.126
5	-7.166	-11.210	0.547	-12.383		-5.400	-3.179	-10.914	-2.867
6	2.502	4.088	5.452	4.041	5.455		4.828	1.795	5.008
7	-6.134	-5.308	3.107	-5.869	5.224	-4.973		-8.790	0.900
8	1.797	7.259	10.332	7.008	12.364	-2.098	13.528		9.711
9	-6.077	-8.202	2.809	-9.566	4.912	-5.170	-2.045	-12.569	

Table 4.5. *t*-tests for out of sample model performance comparison
Values in the upper (lower) diagonal report the Student *t*-value (Johnson, 1978, modified *t*-value) regarding the comparison of means of the squared residuals between models in the vertical heading versus models in the horizontal heading. In general, a positive (negative) *t*-value larger (smaller) than 1.96 (-1.96) indicates that the model in the vertical (horizontal) heading has a larger MSE than the model in the horizontal (vertical) heading at 5% significance level (for 1% significance level use 2.325 and -2.325 respectively). The models compared are:

1: BS_{NL2}
2: CS_{NL2}
3: SVJ
4: ANN_{DVF}^{M}
5: ANN_{DVF}^{H}
6: SVM_{DVF}^{M}
7: SVM_{DVF}^{H}
8: $LS-SVM_{DVF}^{M}$
9: $LS-SVM_{DVF}^{H}$

There are two additional observations we can make from these tables. First, we should note that in three out of the four cases, the performance of the LS-SVM models is better compared to the SVM models. The only exception is SVM_{av}^{H} with RMSE equal to 2.656 compared to the 3.756 of $LS-SVM_{av}^{H}$. In addition, it is obvious that the RMSE out of sample results for ANNs are lower compared to the counterpart SVMs and LS-SVMs models. This does not necessarily imply that LS-SVMs are superior to SVMs and that ANNs are superior to the other nonparametric models. One explanation for this regards the naïve hyper-parameter selection process we follow which as noted earlier might not be the best strategy to adopt. Furthermore, someone should notice that ANNs and SVMs (and LS-SVMs) employ different forms to model the problem under investigation and they use different loss functions to measure performance. If the data are contaminated with pure Gaussian noise then may observe ANNs and LS-SVM that are optimized based on a sum of squares loss function to perform better than SVMs; also SVMs with the ε-insensitive loss function can potentially perform better than non-Gaussian noise (Müller et al., 1999). In addition, SVMs that use inappropriate large values for ε may introduce systematic bias to the estimation and considerably underfit the relationship (Müller et al., 1999).

4.6. Conclusions

In this essay we investigate the options pricing performance of ANNs, AVMs and LS-SVMs for the period 2002-2004. With these models we use implied volatility inputs obtained by the Black and Scholes model and we estimated them using the market and the hybrid target functions. All results obtained for the nonparametric models are compared with the Black and Scholes, the Corrado and Su and the Stochastic Volatility and Jumps parametric models. The results suggest that LS-SVMs perform better than SVMs but the ANNs performance is the overall best among the nonparametric models. In addition, there is a hybrid ANN model that has comparable performance with *SVJ* which is the best performing POPM.

In our view, the results obtained for SVMs and LS-SVMs are promising enough for the problem under insvestigation. We feel that the out of sample results for ANNs might look better than the ones obtained with SVMs and LS-SVMs because our level of expertise with ANNs is significantly higher. We can expect that under more careful and sophisticated strategies of hyper-paramete selection SVM and LS-SVM can improve their out of sample performance. Further research is needed here to also investigate the performance of SVMs and LS-SVMs combined with other inputs/outputs from the POPMs. Ideally the most advanced extension of this analysis is to manage to combine the methodologies of the third and fourth essays in order to derive generalized parameter functions with SVMs and LS-SVMs.

4.A. Appendix

Below we present an extensive overview regarding the nonparametric methodologies employed in this essay. A shorter version of the following has been included in the main content of the essay.

4.A.1 The nonparametric approaches: ANNs, SVM and LS-SVM

ANNs comprise a popular methodology for handling function estimation problems for many reasons. First, of all theoretical proofs exist showing that under certain conditions, ANNs are universal approximators able to approximate any nonlinear function arbitrarily well (Cybenko, 1989). In addition, they perform good in situations where there is lack of knowledge for the relationship that underpins a set of variables and they are robust on the presence of noisy data. Nevertheless, they are prone also to some practical merits and limitations. First, there are no theoretical foundations on how to select the network type and structure and on how to implement the optimization procedure. For this reason the model structure (number of neuron layers and number of hidden neurons in each of the layers) should be defined a-priori which is not necessarily the best strategy in choosing the optimal network architecture for the faced problem. This task is rather an art instead of science and can be better tackled by experts with experience on how to apply the ANNs methodology having at the same time considerable knowledge on the problem under investigation. Second, estimating ANNs involves the optimization of a highly non-convex error function and frequently enough optimization algorithms get stuck to local minima solutions resulting to suboptimal solutions for the network free parameters (weights and biases). Given this peculiarity, regularization techniques that are employed in an attempt to control the capacity of the ANNs, like cross validation strategies and early stopping, are only partial remedies potentially resulting to structures that do not maximize their generalization performance to unseen data. Finally ANNs learn the empirical functions inductively from transactions data without embedding any information related to the problem under investigation. Under this setting, the estimated weights do not convey any meaningful interpretation to help understand better the input-output relationship.

In contrast, SVMs are not confined by the above issues (Smola and Schölkoph, 1998). First of all, the model complexity does not need to be determined a-priori. It is determined endogenously as part of the optimization problem in such a way that maximizes the generalization capability of the model. More importantly, a unique solution is found after estimation as a solution of a (convex) Quadratic Programming (QP) problem with linear constraints, which depends on the estimating data and the selection of few tuning hyper-parameters. In addition, the solution to the QP problem provides the necessary information for choosing the most important datapoints, known as support vectors, among all the data; based on the SVM formulation, support vectors uniquely define the estimated regression function so in this manner the estimated coefficients are informative. Furthermore, input data of any arbitrary dimensionality can be treated with only linear cost in the number of input dimensions. This property in conjunction with the good inherent regularization properties allows SVMs to work particularly well when data is sparse (see Müller et al., 1999, Smola and Schölkoph, 1998). Yet, the performance of the SVM technique, like ANNs, depends crucially by the choice of the loss function which is inextricable connected with the noise in the data (Gaussian or not) and by data regularities (e.g. non-stationary financial data).

4.A.2 About the empirical and structural risk minimization

When we apply ANNs our ultimate purpose is for a given finite set of input patterns x to define a set of adjustable parameters ω resulting to the estimating function $f(x,\omega)$ that describes the known target patterns (desired output) t. To achieve this we minimize the following error measure which is called the *empirical risk*:

$$\arg\min_{f \in H_l} R_{emp}[f] = \frac{1}{P}\sum_{j=1}^{P} L\left(x_j, t_j, f(x_j,\omega)\right) \qquad (4.A.1)$$

where $L\left(x_j, t_j, f(x_j,\omega)\right)$ represents a general loss function determining how estimation errors are penalize and H_l represents a general class of continuous function from which the approximating function $f(x,\omega)$ can be selected.

By minimizing the empirical risk with the finite sample data we assume that we will manage to identify the best function $f(x,\omega)$ that also minimizes *generalization error* (also termed as the *expected risk*) of the estimator given by:

$$R_{gen}(f) = \int L\big(x, t, f(x,\omega)\big)p(x,t)dxdt \qquad (4.11)$$

which expresses the level of error obtained by the data generating mechanism under the joint probability distribution $p(x,t)$ that governs all input-output data. The fact is that the joint probability distribution $p(x,t)$ is most of the time unknown for real-life applications and the model must be estimated using only the observed datapoints by minimizing the empirical risk. In the context of statistical learning theory, Vapnik has shown that the following probabilistic bound can be derived for the generalization error:

$$R_{gen}(f) \leq R_{emp}(f) + VC(P,h,\eta) \qquad (4.A.2)$$

where in the above bound the second term is a confidence term, called the *Vapnik – Chervonenkis* confidence, that holds with probability 1-η for a sample size of P. The VC confidence also depends crucially on h which characterizes the model complexity known as the VC dimension (see Vapnik, 1995 for further technical details); in general it is an increasing function in the number of free parameters[56] and decreasing in the number of datapoints. The expression in Eq. (4.A.2) is analogous to the bias-variance trade-off. Simple models that have too few adjustable parameters do not have enough representational power and they typically result in high bias (high empirical

---

[56] There can be cases where a model with more free parameters does not necessarily has a higher VC dimension compared to one with fewer free parameters (Vapnik, 1995).

risk). However, they are rather robust and not so sensitive to the data used for estimation resulting to low variance (low VC bound). On the contrary, a high capacity model with a large number of adjustable parameters will result in higher accuracy on the given training set with low bias (low empirical risk). Large capacity models because of high approximating power are able to overfit (memorize) the data by also modelling the noise inherent in the data. Such models are data sensitive meaning that each particular dataset will give rise to a different model meaning that their variance (VC bound) will be high (Vojislav, 2001, pg. 136). Thus for a given finite sample of data points the best generalization performance will be achieved if the right balance is identified between the accuracy obtained on that particular training set, and the capacity of the model.

Thus the informational content behind Eq. (4.A.2) is that we can define an upper bound for this kind of risk that can eventually help us to select among alternative models. Furthermore, in the context of VC theory it can be shown that for bounded loss functions the empirical risk minimization principle is consistent if and only if empirical risk converges (in probability) uniformly to the expected risk in the following sense:

$$\lim_{P \to \infty} p\left[\sup_{\omega}\left(R(f) - R_{emp}(f)\right) > \varepsilon\right] = 0, \quad \forall \varepsilon > 0 \qquad (4.A.3)$$

The above expression imposes the necessary conditions so that the function that minimizes the empirical risk converges to the best function that minimizes the expected risk. An important insight of the VC theory is that a necessary and sufficient condition for a fast rate of convergence of the empirical risk minimization is that the VC dimension of a set of approximating functions be finite (Vapnik, 1995). In other words, without restricting the set of admissible functions, empirical risk minimization is not consistent when trying to estimate a model with a finite number of datapoints. The so called *structural risk minimization* can be employed to circumvent this peculiarity (see Vapnik, 1995) especially when dealing with small samples (Vojislav, 2001). One considers a set H of approximating functions:

$$\ldots H_{l-1} \subset H_l \subset H_{l+1} \ \ldots \qquad (4.A.4)$$

which consists of nested sets of functions of increasing complexity (increasing VC dimension). The larger the VC dimension the smaller the empirical risk can become but the VC confidence term in Eq. (4.A.2) will grow. Structural risk minimization is an effective way of controlling the VC confidence so as to control the generalization risk. SVM can be a promising alternative compared to ANNs since there are theoretical foundations showing that they actually minimize the generalization error by simultaneously minimizing the VC confidence interval and the empirical risk, resulting in models with potentially superior generalization ability (not asymptotically but when using a finite data sample). The most intriguing feature is that this process is part of the formulated optimization problem. On the contrary, heuristic techniques like early stopping and cross validation are used with ANNs in order to select from a possibly infinite pool of candidate models the best model with the smallest possible generalization error.

Below we explain the mathematical framework that underpins the nonparametric methodologies we employ in this study. For this we assume that we have a given dataset with features points (x_1, t_1), (x_2, t_2), …, (x_P, t_P) were $x_j \in \Re^N$ are the input features, $t_j \in \Re$ are the known target values (desired output), N is the number of input variables and $j = 1, 2, \ldots, P$ with P to represent the sample size.

4.A.3. ε-insensitive support vector machines for function approximation

The application of SVM for regression was initially developed only for performing linear regression. The technique has been extended to handle nonlinear regression applications based on a very intuitive idea (see Vapnik, 1995, 1998). Firstly, to apply a mapping $\varphi(x)$ (chosen a-priori) of the input data x into an arbitrarily high dimensional feature space which can be (possibly) infinite dimensional. This transformation is usually called the *kernel trick*. Secondly, the linear SVM regression can be applied to create an approximate linear function in this arbitrarily high dimensional feature space. In this way, doing linear regression in a high dimensional feature space corresponds to nonlinear regression in the (low dimensional) input space (Müller et al., 1999).

The idea behind the SVM for function approximation (Support Vector Regression) is to estimate the coefficient values w (called the weights) and b (called the bias) that optimize the generalization ability of our repressor by minimize the following regularized loss function:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_{j=1}^{P} L_\varepsilon\left(t_j, f(x_j)\right) \qquad (4.A.5)$$

where $f(x)$ is the form of the SVM function approximation and is given by:

$$f(x) = w^T \varphi(x) + b \qquad (4.A.6)$$

and $L_\varepsilon(t, f(x)$ is the so-called Vapnik's ε-insensitive loss functions defined as:

$$L_\varepsilon(t, f(x)) = |t - f(x)|_\varepsilon = \begin{cases} 0 & \text{if } |t - f(x)| \le \varepsilon \\ |t - f(x)| - \varepsilon & \text{otherwise} \end{cases} \qquad (4.A.7)$$

In the above formulations $\varphi(x) : \Re^N \to \Re^{N_h}$ represents a nonlinear mapping (transformation) of the input space to an arbitrarily high-dimensional feature space which can be infinite dimensional (in such case the weights vector w will also become infinite dimensional). Under this Eq. (4.A.6) can be seen as a set of linear functions that are defined in a high dimensional space, thus allowing us to solve nonlinear regression problems with the use of SVM that perform linear regression. Minimizing the norm of the weights vector, $w^T w$ in Eq. (4.A.5), allows us to control the complexity (called flatness or capacity) of the estimated function. The empirical risk minimization is achieved by minimizing the Vapnik's ε-insensitive loss function, $L_\varepsilon(t, f(x))$, that allows us to control the accuracy of the estimated function that is defined by the value ε (called the tube size). Finally, the constant $C > 0$ determines the trade-off between the amount up to which deviations larger than ε are tolerated and the flatness (complexity) of the estimated model.

In the case where ε is chosen to be small and some datapoints do not lay within the tube of ε accuracy the estimation of the w and b is done by formulating the following optimization problem in the primal weight space of the unknown coefficients:

$$\min_{w, b, \xi, \xi^*} \quad L_P(w,\ \xi,\ \xi^*) = \frac{1}{2} w^T w + C \sum_{j=1}^{P} \left(\xi_j + \xi_j^*\right) \tag{4.A.8}$$

subject to

$$t_j - w^T \varphi(x_j) - b \leq \varepsilon + \xi_j\,, \quad j = 1, \ldots, P \tag{4.A.8.1}$$

$$w^T \varphi(x_j) - t_j + b \leq \varepsilon + \xi_j^*\,, \quad j = 1, \ldots, P \tag{4.A.8.2}$$

$$\xi_j, \xi_j^* \geq 0\,, \quad j = 1, \ldots, P \tag{4.A.8.3}$$

where ξ_j and ξ_j^* are positive slack variables, defined in the prime space, that need to be introduced in order to make the solution of the optimization of the optimization problem feasible for all datapoints that are outside of the ε-tube. The above formulation is better solved by introducing positive Lagrange multipliers $\alpha_j, \alpha_j^*, \eta_j, \eta_j^* \geq 0$ (called the dual variables) [57]:

[57] In nonlinear regression problems the primal weights vector w can become infinite dimensional due to the applied transformation $\varphi(x)$. For this reason the solution of the problem is better derived via its dual formulation.

$$\Lambda\left(w,b,\xi,\xi^{*},\alpha,\alpha^{*},\eta,\eta^{*}\right)=$$
$$\frac{1}{2}w^{T}w+C\sum_{j=1}^{P}\left(\xi_{j}+\xi_{j}^{*}\right)-\sum_{j=1}^{P}\alpha_{j}\left(\varepsilon_{j}+\xi^{*}-t_{j}+w^{T}\varphi(x_{j})+b\right)$$
$$-\sum_{j=1}^{P}\alpha_{j}^{*}\left(\varepsilon_{j}+\xi_{j}^{*}-t_{j}+w^{T}\varphi(x_{j})+b\right)-\sum_{j=1}^{P}\left(\eta_{j}\xi_{j}+\eta_{j}^{*}\xi_{j}^{*}\right)$$

(4.A.9)

The saddle point of the Lagrangian formulation should be maximized with respect to the dual variables and minimized with respect to the primal variables:

$$\max_{\alpha,\alpha^{*},\eta,\eta^{*}}\ \min_{w,b,\xi,\xi^{*}}\ \Lambda\left(w,b,\xi,\xi^{*},\alpha,\alpha^{*},\eta,\eta^{*}\right) \qquad (4.A.10)$$

It follows from the saddle point condition that the partial derivatives of Λ with respect to the primal variables (w,b,ξ,ξ^{*}) have to vanish for optimality:

$$\frac{\partial\Lambda}{\partial w}=0\Rightarrow\quad w=\sum_{j=1}^{P}\left(\alpha_{j}-\alpha_{j}^{*}\right)\varphi(x_{j}) \qquad (4.A.11)$$

$$\frac{\partial\Lambda}{\partial b}=0\Rightarrow\quad\sum_{j=1}^{P}\left(\alpha_{j}-\alpha_{j}^{*}\right)=0 \qquad (4.A.12)$$

$$\frac{\partial\Lambda}{\partial\xi_{j}}=0\Rightarrow\quad\eta_{j}=C-\alpha_{j} \qquad (4.A.13)$$

$$\frac{\partial\Lambda}{\partial\xi_{j}^{*}}=0\Rightarrow\quad\eta_{j}^{*}=C-\alpha_{j}^{*} \qquad (4.A.14)$$

Substituting Eq. (4.A.11) through X(4.A.14) into Eq. (4.A.9) results to a quadratic programming problem that depends only by the dual variables α and α^{*} (called the dual problem):

$$\max_{\alpha,\alpha^*} \quad L_D(\alpha,\ \alpha^*) = -\frac{1}{2}\sum_{j,i=1}^{P}\left(a_j - a_j^*\right)\left(a_i - a_i^*\right)K\left(x_j, x_i\right) - \varepsilon\sum_{j=1}^{P}\left(a_j + a_j^*\right) + \sum_{j=1}^{P} t_j\left(a_j - a_j^*\right) \tag{4.A.15}$$

subject to

$$\sum_{j=1}^{P}\left(a_j - a_j^*\right) = 0 \tag{4.A.15.1}$$

$$0 \le a_j, a_j^* \le C \tag{4.A.15.2}$$

Using Eq. (4.A.11), the dual representation of the model becomes:

$$f(x) = \sum_{j=1}^{P}\left(a_j - a_j^*\right)K\left(x, x_j\right) + b \tag{4.A.16}$$

In the above setting the *kernel trick* has been applied with $K\left(x_j, x_i\right) = \varphi(x_j)^T \varphi(x_i)$. It enables us to work in huge dimensional feature spaces $\varphi(x)$ without actually having to do explicit computations in this space. Also note that in this setting, the optimization problem corresponds to finding the *flattest* function in the *feature* space, not in the input space (Smola and Schölkoph, 1998). It is also important to note that we do not need to compute explicitly the value of w to evaluate the function that results from the estimation of the SVMs since $f(x)$ can be easily evaluated via Eq. (4.A.16) with the use of the kernel trick.

Someone can use the complementary Karush-Kuhn-Tucker (KKT) conditions for the computation of the coefficient b. KKT state that product between dual variables and constraints should be zero at the optimal solution. Based on the aforementioned this results to:

$$\alpha_j\left(\varepsilon + \xi_j - t_j + w^T\varphi(x_j) + b\right) = 0 \qquad (4.A.17)$$

$$\alpha_j^*\left(\varepsilon + \xi_j^* - t_j + w^T\varphi(x_j) + b\right) = 0 \qquad (4.A.18)$$

$$\left(C - \alpha_j\right) = 0 \qquad (4.A.19)$$

$$\left(C - \alpha_j^*\right) = 0 \qquad (4.A.20)$$

The above expressions allow us to make several useful conclusions (see Smola and Schölkoph, 1998). Firstly, only samples (x_j, t_j) with corresponding $\alpha_j, \alpha_j^* = C$ lay outside the ε-insensitive tube around the estimated function. Secondly, for the same data point obviously there can never be a set of dual variables α_j, α_j^* which are simultaneously nonzero ($\alpha_j\alpha_j^* = 0$). Thirdly for $\alpha_j, \alpha_j^* \in (0, C)$ it holds that $\xi_j, \xi_j^* = 0$ thus the second factor in Eq. (4.A.17) and (4.A.18) should equal zero respectively. Hence, b can be computed as follows:

$$b = t_j - w^T\varphi(x_j) - \varepsilon \quad \text{for } \alpha_j \in (0, C) \qquad (4.A.21)$$

$$b = t_j - w^T\varphi(x_j) + \varepsilon \quad \text{for } \alpha_j^* \in (0, C) \qquad (4.A.22)$$

From the above, one data point should be in principle sufficient to compute the bias b but for stability purposes it is recommended to take the average over all points that hold $\alpha_j, \alpha_j^* \in (0, C)$ (Müller et al., 1999). Then the estimation of b is given by:

$$b = \frac{1}{N}(t_i - \sum_{j=1}^{N}\left(a_j - a_j^*\right)K\left(x_j, x_i\right) + \varepsilon\,\mathrm{sign}\left(a_i - a_i^*\right) \quad \text{for } \alpha_j, \alpha_j^* \in (0, C)$$

(4.A.23)

From Eq. (4.A.17) and (4.A.18) it follows that for datapoints that lay inside the ε-insensitive tube, $\alpha_j \alpha_j^* = 0$ so that the KKT conditions are satisfied. With this and in conjunction with Eq. (4.A.11) it is true we have a sparse expansion of w in terms of the datapoints x_j. In other words after estimating the SVM model we do not need all x_j to describe w. The examples that come with no vanishing coefficient values for w are called *support vectors* (see also Vapnik, 1995). Support Vectors can depict the distributional features of all data according to the nature of SVMs, removing some trivial data from the whole training set will not greatly affect the generalization performance but speed the training process effectively (Suykens et al., 2002).

To successfully apply the SVMs for nonlinear regression problems it is necessary to apply the kernel trick by choosing a proper kernel function:

$$K\left(x_j, x_i\right) = \varphi(x_j)^T \varphi(x_i) \qquad (4.A.24)$$

such that we do not need to explicitly define the nonlinear mapping function $\varphi(x)$ since Eq. (4.A.24) is a function in input space. A function that is symmetric, continuous and satisfies Mercer's condition (see Vapnik, 1995 for details) is an admissible kernel function that represent a scalar product in the (mapped) featured space as expressed in Eq. (4.A.24). The Gaussian kernel is a widespread kernel function that is admissible for use with SVM for function approximation:

$$K\left(x_j, x_i\right) = exp\left(-\frac{\left\|x_j - x_i\right\|^2}{2\sigma_K^2}\right) \qquad (4.A.25)$$

where $\left\|x_j - x_i\right\|^2$ measures the distance between two datapoints and σ_K^2 is called the kernel width parameter and is used as a normalizing factor. It can be shown that when the Gaussian kernel function is considered, the nonlinear mapping $\varphi(x_j)$ is infinite dimensional and also that SVM are universal approximators (see Vapnik, 1995 and 1998 for details), an implication of paramount importance that is contributing in a growing popularity of SVM for regression applications. It is notable that although implicitly the mapping corresponds to dot products in an infinite dimensional feature space, the complexity of computing the kernel function can be much smaller resulting to tractable computations.

The application of SVMs in general preserve some very helpful characteristics compared to other learning techniques (e.g. feedforward artificial neural networks, etc). The first important characteristic regards their optimization aspect. The system of equations defined Eqs. (4.A.15), (4.A.15.1) and (4.A.15.2) given a positive definite kernel translates to the optimization of a convex QP problem subject to linear constraints that results to a global and unique solution. On the contrary, feedforward artificial neural networks suffer from existence of multiple local minima solutions[58] since the optimization function is not convex with respect to the network weights and biases. Second, after selecting the SVM tuning parameters ($C, \varepsilon, \sigma_K^2$), the model complexity is implicitly defined by the number of support vectors as part if the solution to the convex problem, whilst for the case of the ANNs the number of hidden neurons should be defined a-priori. Third, the solution to the problem is characterized by a sparse representation of the solution. As explained earlier, the final solution is defined solely by the support vectors which represent only a part of the datapoints used initially for the estimation of the model. Another important issue is that the function's representation is independent of the dimensionality of the input space and depends only on the number of support vectors; in other words the size of the QP problem does not depend on the dimensionality of the input space. This is a significant remedy for the curse of dimensionality issue. On the contrary, ANNs are prone to the

---

[58] Among others, Cybenko (1989) has shown that ANNs with one hidden layer of neurons can be universal function approximators that provide adequate robustness and convergence with good out of sample generalization abilities. However this property can be of limited use in practice when the optimization algorithm gets stuck in local minima resulting to a suboptimal solution because of the non-convexity of the optimized error/loss function. In this study ANNs are implemented under certain strategies like early stopping and use of cross-validation techniques that try to eliminate the effect of local minima solutions and overfitting of the data.

effects of the curse of dimensionality. In this case, early stopping and cross validation techniques should be very carefully applied by an expert in an attempt to overcome the curse of dimensionality by preventing the networks to memorize the data used for estimation and to result to a limited or a poor generalization performance (Vojislav, 2001, Suykens et al., 2002).

4.A.4. Least squares support vector machines

The Least Squares Support Vector Machines (LS-SVM) method is a variant of the SVMs methodology originally proposed and developed by Suykens and co-workers (see Suykens et al., 2002). According to this approach the model estimated is given by the following optimization problem in the primal weight space:

$$\min_{w,b,e} \quad L_P(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{j=1}^{P} e_j^2 \qquad (4.A.26)$$

subject to

$$t_j = w^T \varphi(x_j) + b + e_j \,, \quad j = 1,\ldots,P \qquad (4.A.26.1)$$

The above formulation is nothing else but a ridge regression cost function formulated in the featured space defined by the mapping $\varphi(x)$. Parameter γ determines again the trade-off between the model complexity and goodness of fit to the estimation data. Like in the case of SVM (see Suykens et al., 2002, pg. 98), the resulting Lagrangian formulation is:

$$\Lambda(w,b,e,\ \alpha) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{j=1}^{P} e_j^2 - \sum_{j=1}^{P} \alpha_j \left(w^T \varphi(x_j) + b + e_j - t_j \right)$$

(4.A.27)

where α_j are the Lagrange multipliers which in contrast to the SVM case can be both positive or negative due to the equality constraints. Again, the conditions for optimality are given by:

$$\frac{\partial \Lambda}{\partial w} = 0 \Rightarrow \quad w = \sum_{j=1}^{P} \alpha_j \varphi(x_j) \tag{4.A.28}$$

$$\frac{\partial \Lambda}{\partial b} = 0 \Rightarrow \quad \sum_{j=1}^{P} \alpha_j = 0 \tag{4.A.29}$$

$$\frac{\partial \Lambda}{\partial e_j} = 0 \Rightarrow \quad \alpha_j = \gamma e_j, \quad j = 1,\ldots,P \tag{4.A.30}$$

$$\frac{\partial \Lambda}{\partial \alpha_j} = 0 \Rightarrow \quad w^T \varphi(x_j) + b + e_j - t_j = 0, \quad j = 1,\ldots,P \tag{4.A.31}$$

Substituting the expressions for w and e back to Eq. (4.A.27) result to the following linear KKT system in a and b:

$$\sum_{j=1}^{P} \left(\alpha_j K(x_j, x)\right) + b + \frac{a_j}{\gamma} = t_j, \quad j = 1,\ldots,P \tag{4.A.32}$$

$$\sum_{j=1}^{P} \alpha_j = 0 \tag{4.A.33}$$

where the resulting LS-SVM model that characterizes the estimated regression function is given by:

$$f(x) = \sum_{j=1}^{P} a_j K\left(x, x_j\right) + b \tag{4.A.34}$$

Someone can observe that the kernel trick as given by Eq. (4.A.32) is also applied here and that the size of the KKT system is not influenced by the dimension of the input space but is only determined by the sample size.

Compared to the SVMs case, LS-SVMs preserve the following characteristics. First, kernel functions that are admissible for the SVMs can also be used in the formulation of the LS-SVMs so the use of Gaussian kernel function given by Eq. (4.A.25) can be used in this case too. Second, the dual problem above corresponds to solving a linear KKT system which is a square system with a unique (global) solution when the matrix has full rank. In addition, for moderate sample sizes there are algorithms that can efficiently (in terms of time and computer's memory capacity) solve the above system. For instance the Hestenes - Stiefel conjugate gradient algorithm can be applied to solve the above system of equations after transforming it into a positive definite system (see Suykens et al., 2002, for further details). Third, the error variable e_j is used to control deviations from the regression function instead of the slack variables ξ_j, ξ_j^* and a squared loss function is used for this error variable instead of the ε-insensitive loss function. This has two implications regarding the solution of the problem: *i) lack of sparseness* since Eq. (4.A.30) implies that every data point will be now a support vector since no langrage multiplier α_j will be exactly zero which it can be considered as a drawback compared to the SVM, *ii)* only two parameters γ and σ_K^2 are needed to be tuned compared to three for SVM which is an advantage since it reduces the possible parameters combinations (2-D grid instead of 3-D grid) and at the same time reduces the risk of selecting a suboptimal parameter combination. Due to the reasons explained above, optimizing a set of LS-SVMs model can be faster compared to standard SVMs.

II. Concluding Remarks

The research interest of this thesis is on the empirical performance of alternative, parametric and nonparametric, options pricing models that are more realistic and can result in more accurate option estimates.

First, a lot of consideration is given to parametric formulas that are based on more flexible distributional assumptions able to model negative skewness and excess kurtosis. Specifically, we consider as benchmarks the Stochastic Volatility and Stochastic Volatility and Jump models proposed by Bates (1996) which are developed based on a specific parameterized diffusion process that allows for discontinuities and randomly changing variance of the underlying asset. Although these models are known to be more accurate compared to the Black and Scholes formula, they are less intuitive and sometimes exceedingly complex to be applied in practice. For this reason, in the first three essays we incorporate the use of the Corrado and Su (1996) model in the framework we develop. This model is an extension of the Black and Scholes model that can easily handle nonnormal skewness and kurtosis. In addition, to the above we also consider the Deterministic Volatility Functions approach proposed by Dumas, Fleming and Whaley (1998) which is an intuitive approach that relaxes the Black and Scholes assumption of having a constant volatility per options contract. This methodology is very intriguing since there are theoretical proofs showing that it constitutes a reduced-form approximation to an unknown structural model which under frequent re-estimation can exhibit exceptional pricing performance.

The most significant contributions of this thesis regard the development and applications of nonparametric methodologies. The first essay includes comparisons with respect to pricing and trading performance between parametric models and several alternative artificial neural network specifications by using a large amount of input-output combinations. This essay considers in depth significant issues not examined before and reconciles partial evidence reported previously. The second essay uses only the key results of the first essay in order to show that better options pricing performance can be achieved with robust artificial neural networks.

All nonparametric models examined in the first two essays are able to learn to approximate the empirical options pricing function inductively from transactions data. Yet their estimation algorithms do not embed directly theoretical information related to the specific problem under investigation. The third essay improves this by proposing a novel semi-parametric approach that allows a set of the input variables to a parametric model to be determined by systems of equations that are estimated with the use of artificial neural networks. As a result the proposed semi-parametric approach preserves important features concerning the desire for nonnegative option values, theory consistent Greek letters, rational pricing behavior at the boundary of option pricing areas etc. It also presents a very extensive set of pricing and hedging results testing all models considered for robustness under alternative data choices and model complexity.

Finally the last essay is an attempt to examine the applicability of support vector machines in the empirical options pricing research field. This essay reconsiders robust and least squares optimization techniques and elaborates further on issues examined by the first two essays regarding the application of nonparametric methods by using as benchmark models the most sophisticated ones that have been used in the third essay. The results obtained here indicate that this is a very promising methodology and we believe that there is a lot of room for improvement.

To summarize, the best POPM was the SVJ model (far outperforming the SV model). The hybrid structures we develop are superior to the standard ANNs. Among the ones we develop, we believe that the GPF structure is the most promising one and has a value of its one because it extends the DVF methodology (which is a standard benchmark). There seems to be fertile ground for future research in directions that will combine aspects of the third essay with the support vector machines of the fourth. Specifically, in these essays we find that a number of competing models can fit a particular set of data resulting in a range of alternative option pricing and hedging estimates. Since we always choose the model with the best out of sample pricing performance, all other estimated models are frequently ignored. One way to combine estimates from different models is to rely on the Bayesian Model Averaging (see Rafteri et al., 1997 and Hansen, 2007).

III. References

Ackert, L.F. and Tian, Y.S. (2001) "Efficiency in Index Option Markets and Trading in Stock Baskets", *Journal of Banking and Finance*, vol. 25, 1607-1634.

Aït-Sahalia, Y., and Lo, W. A. (1998) "Nonparametric Estimation of State-Price Densities Implicit in Financial Asset Prices." *Journal of Finance*, vol. 53, no- 2, 499-547.

Andersen, T.G., Benzoni, L., and Lund, J. (2002) "An Empirical Investigation of Continuous-Time Equity Return Models." *Journal of Finance*, vol. 57, no. 3, 1239-1276.

Backus, D., Foresi, S., Li, K., and Wu, L. (1997). "Accounting for Biases in Black-Scholes." *Working Paper*, Stern School of Business.

Bakshi, G., Cao, C., and Chen, Z. (1997) "Empirical Performance of Alternative Options Pricing Models." *Journal of Finance*, vol. 52, no. 5, 2003-2049.

Bakshi, G., Cao, C., Chen, Z. (2000) "Pricing and Hedging Long-term Options", *Journal of Econometrics*, vol. 94, no. 1-2, 277-318.

Bandler, W. J, Biernacki, M. R., Chen, H. S., Grobenly, A., P, and Hemmers, H. R. (1994) "Space Mapping Technique for Electromagnetic Optimization" *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 12, 2536-2544.

Bandler, W. J, Ismail, A. M., Sanchez E.R. J., and Zhang, J. Q (1999). "Neuromodeling of Microwave Circuits Exploiting Space-Mapping Technology." *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 12, 2417-2427.

Bandler, W. J, Ismail, A. M., Sanchez E.R. J., and Zhang, J. Q (1999). "Neuromodeling of Microwave Circuits Exploiting Space-Mapping Technology." *IEEE Transactions on Microwave Theory and Techniques*, vol. 47, no. 12, 2417-2427.

Bandler, W. J., Biernacki, M. R., Chen, H. S., Grobenly, A., P., and Hemmers, H. R. (1994). “Space Mapping Technique for Electromagnetic Optimization.” *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 12, 2536-2544.

Bandler, W.J., Chen, H.S., Biernacki, M.R., Gao, L. and Madsen, K. (1993) “Huber Optimization of Circuits: a Robust Approach”, *IEEE Transactions on Microwave Theory and Techniques*, vol. 41, no. 12, 2279-2287.

Bates, D. S. (1991) “The Crash of ’87: Was it Expected? The Evidence From Options Markets” *Journal of Finance*, vol. 46, no. 3, 1009-1044.

Bates, D. S. (1996) “Jumps and Stochastic Volatility: Exchange Rate Processes Implicit in Deutsche Mark Options.” *The Review of Financial Studies*, vol. 9, no. 1, 69-107.

Bates, D. S. (2000) “Post-'87 Crash Fears in the S&P 500 Futures Option Market.” *Journal of Econometrics*, vol. 94, no. 1-1, 181-238.

Bates, D. S. (2003) “Empirical Options Pricing: A Retrospection.” *Journal of Econometrics* 116, 387-404.

Bates, D.S. (2003) “Empirical Option Pricing: A retrospection”, *Journal of Econometrics*, vol. 116, 387-404.

Bensaid, B., J-P. Lesne, H. Pagés and J. Scheinkman (1992) “Derivative Asset Pricing with Transaction Costs”, *Mathematical Fiannce* 2, 63-86.

Berkowitz, J. (2004) “On Justifications for the Ad Hoc Black-Scholes Method of Option Pricing.” Working Paper, University of Houston.

Bishop, M.C., 1995. *Neural Networks for Pattern Recognition.* Oxford University Press.

Black, F. and Scholes, M. (1972) “The Valuation of Option Contracts and a Test of Market Efficiency”. *The Journal of Finance*, vol. 27, 399-417.

Black, F. and Scholes, M. (1973) “The Pricing of Options and Corporate Liabilities.” *Journal of Political Economy*, vol. 81, 637-654.

Black, F. and Scholes, M. (1975) "Fact and Fantasy in the Use of Options." *The Financial Analysts Journal*, vol. 31, 36-41 and 61-72.

Bollen, N. P. B., and Whaley, R. E. (2004) "Does Net Buying Pressure Affect the Shape of Implied Volatility Functions?" *Journal of Finance*, vol. LIX, no. 2, 711-753.

Brandt M. W. and Wu T. (2002) "Cross-Sectional Test of Deterministic Volatility Functions", *Journal of Banking and Finance*, vol. 9, 525-550.

Breeden D. and Litzenberger R. (1978) "Prices of State-Contingent Claims Implicit in Options Prices", *Journal of Business*, vol. 51, 621-651.

Brown, C. and Robinson, D. (2002) "Skewness and Kurtosis Implied by Option Prices: A Correction." *Journal of Financial Research*, vol. 25, no. 2, 279-282.

Canica, L. and Figlewski, S. (1993) "The Informational Content of Implied Volatility", *The Review of Financial Studies*, vol. 6, no. 3, 659-681.

Cao, L. J. and Tay, F. E. H. (2003) "Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting", *IEEE Transactions on Neural Networks* 14, 1506-1518.

Capelle-Blancard, G., Jurczenko, E., and Maillet, B., (2001) "The Approximate Option Pricing Model: Performances and Dynamic Properties." *Journal of Multinational Financial Management* 11, 427-443.

Carverhill, A., Cheuk, T.H.F. (2003) "Alternative Neural Network Approach for Option Pricing and Hedging", Working paper, School of Business, University of Hong Kong.

Chang, W.X. (2005) "Computation of Huber's M-estimated for a Block-angular Regression Problem", Forthcoming in the *Computational Statistics & Data Analysis*.

Charalambous, C., (1992) "A Conjugate Gradient Algorithm for Efficient Trading of Artificial Neural Networks." *IEE Proceedings – G*, 139, 301-310.

Chen, N., Johnson, H. (1985), “Hedging Options”, *Journal of Financial Economics*, vol. 14, no. 2, 317-321.

Cherkassky, V., and Ma, Y. (2004) “Practical Selection of SVM Parameters and Noise Estimation for SVM Regression”, *Neural Networks,* 17, 113-126.

Chernov, M., and Ghysels, E. (2000) “Towards a Unified Approach to the Joint Estimation Objective and Risk Neutral Measures for the Purpose of Option Valuation”, *Journal of Financial Economics*, 56, 407-458.

Chiras, D.P. and Manaster, S. (1978) “The Informational Content of Option Prices and a Test of Market Efficiency”. *Journal of Financial Economics*, vol. 6, 213-234.

Christoffersen, P., and Jacobs, K. (2004) “The Importance of the Loss Function in Option Valuation.” *Journal of Financial Economics*, vol. 72, 291-318.

Christoffersen, P., Heston, S., and Jacobs, K. (2006) “Option Valuation with Conditional Skewness”, *Journal of Econometrics*, vol. 131, no.1-2, 253-284.

Christoffersen, P., Heston, S., and Jacobs, K. (2007) “The Shape and Term Structure of the Index Option Smirk: Why Multifactor Stochastic Volatility Models Work so Well”, Working Paper.

Constantinides, G. M., Jackwerth, J. C., and Perrakis, S. (2008) “Option Pricing: Real and Risk-Neutral Distributions”, In J. B. Birge and V. Linetsky (Eds), *Handbook in OR & MS*, Vol. 15, Elsevier.

Cont, R., and Fonseca, J. (2002) “Dynamics of Implied Volatility Surfaces.” *Quantitative Finance* 2, 45-60.

Corrado, C. J., and Su, T. (1996) “Skewness and Kurtosis in S&P 500 Index Returns Implied by Option Prices.” *Journal of Financial Research*, vol. 19, no. 2, 175-192.

Corrado, C. J., and Su, T. (1997) “Implied Volatility Skews and and Stock Index Skewness and Kurtosis Implied by S&P 500 Index Options.” *Journal of Derivates*, vol. 4, 8-19.

Cybenko, G. (1989) “Approximation by Superpositions of a Sigmoidal Function.” *Mathematics of Control, Signal and Systems* 2, 303-314.

Das, S. R., and Sundaram, R. K. (1999) “Of Smiles and Smirks: A Term Structure Perspective.” *Journal of Financial and Quantitative Analysis*, 34, 211-239.

Day, T.E. and Lewis, C.M. (1988) “The Behavior of the Volatility Implicit in the Prices of Stock Index Options”, *Journal of Financial Economics*, vol. 22, 103-122.

Devabhaktuni, V., Yagoub, M.C.E., Fang, Y., Xu, J. and Zhang, Q.J. (2001) “Neural Networks for Microwave Modeling: Model Development Issues and Nonlinear Modeling Techniques”. *International Journal of RF and Microwave CAE*, vol. 11, 4-21.

Duda, O. R., Hart, E. P., and Stork, G. D. (2001). *Pattern Classification.* 2nd Edition, John Wiley and Sons, Inc.

Dumas, B., Fleming, J., and Whaley, R. (1998). “Implied Volatility Functions: Empirical Tests.” *Journal of Finance*, 53, 2059-2106.

Ederington, L. and Guan, W. (2005) “The Information Frown in Option Prices”, *Journal of Banking and Finance*, vol. 29, no. 6, 1429-1457.

Eraker, B. (2004). “Do Stock Prices and Volatility Jump? Reconciling Evidence from Spot and Option Prices.” *Journal of Finance* LIX, 1367-1403.

Franses, H.P., Kloek, T. and Lucas, A. (1999) “Outlier robust analysis of long-run marketing effects for weekly scanning data”, *Journal of Econometrics*, 89, 293-315.

Galai, D. (1977) “Tests of market efficiency of the Chicago Board Options Exchange”, *Journal of Business* 50(2), 167-197.

Garcia, R. and Gencay, R. (2000) “Pricing and Hedging Derivative Securities With Neural Networks and a Homogeneity Hint.” *Journal of Econometrics*, vol. 94, 93-115.

Gencay, R., and Qi, M. (2001) “Pricing and Hedging Derivative Securities with Neural Networks: Bayesian Regularization, Early Stopping and Bagging”, *IEEE Transactions on Neural Networks* vol. 12, no. 4, 726-734.

Gestel, T. V., Suykens, J. A. K., Baestaens, D. E., Lambrecthts, A., Lanckriet, G., Vandaele, B., Moor, B. D. and Vandewalle, J. (2001) “Financial Time Series Prediction Using Least Squares Support Vector Machines Within the Evidence Framework”, *IEEE Transactions on Neural Networks* 12, 809-821.

Gultekin, N.B., Rogalski, R.J., Tinic S.M. (1982) “Option Pricing Models Estimates: Some Empirical Results”, *Financial Management*, vol. 11, 58-69.

Hagan, M. T. and Menhaj, M. (1994) “Training Feedforward Networks with the Marquardt Algorithm.” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, 989-993.

Hagan, M., Demuth, H., and Beale, M. (1996) *Neural Network Design,* PWS Publishing Company.

Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.J. and Stahel, P.J. (1986). *Robust Statistics: The Approach Based on Influenced Functions*. Wiley, New York.

Hansen (2007) "Least Squares Model Averaging", *Econometrica*, Vol. 75, 1175-1189.

Haykin, S. (1999). *Neural Netwroks – A Comprehensive Foundation*, 2nd Ed., Prentice Hall.

Herrman, R., and Narr, A. (1997) “Neural Networks and the Valuation of Derivaties – Some Insights Into the Implied Pricing Mechanism of German Stock Index Options”. Discussion Paper, University of Karlsruhe.

Heston, S. L. (1993). “A Closed-Form Solution for Options with Stochastic Volatility with Applications to Bond and Currency Options.” *Review of Financial Studies* 6, 327-343.

Huber, P. (1981). *Robust Statistics*. Wiley, New York.

Hull J. and Suo, W. (2002). "A Methodology for Assessing Model Risk and Its Application to the Implied Volatility Function Model." *Journal of Financial and Quantitative Analysis* vol. 37, 297-318.

Hull, C.J., 1999. Options Futures and Other Derivatives, Prentice-Hall (4th edition).

Hutchinson, J. M., Lo, A. W., and Poggio, T. (1994) "A Nonparametric Approach to Pricing and Hedging Derivative Securities Via Learning Networks." *Journal of Finance*, vol. 49, no. 3, 851-889.

Jabr, R.A. (2004) "Power system Huber M-estimation with equality and inequality constraints", Forthcoming in *Electric Power Systems Research*.

Jackwerth, J. C. (2000) "Recovering Risk Aversion from Option Prices and Realized Returns", *The Review of Financial Studies*, vol. 12, no. 2, 433-451.

Jarrow R. and Rudd, A. (1982) "Approximate Option Valuation for Arbitrary Stochastic Processes" *Journal of Financial Economics*, vol. 10, 347-369.

Johnson, N. J. (1978) "Modified t Test and Confidence Intervals for Asymmetrical Populations", *Journal of the American Statistical Association*, vol. 73, no. 363, 536-544.

Jondeau E., and M. Rockinger, (2001), "Gram-Charlier Densities", *Journal of Economic Dynamics and Control* 25, 1457-1483.

Jurczenko, E., Maillet, B., and Negrea, B. (2002). "Revisited Multi-moment Approximate Option Pricing Models: A General Comparison (Part I)." *Working Paper*, LSE-FM.

Jurczenko, E., Maillet, B., and Negrea, B. (2004). "A Note on Skewness and Kurtosis Adjusted Option Pricing Models Under the Martingale Restriction" *Quantitative Finance,* vol. 5, pg. 479-488.

Kamara, A. and Miller, T.W. (1995) "Daily and intradaily tests of put-call parity", *Journal of Financial and Quantitative Analysis*, 30, 519-539.

Koenker, R. (1982) "Robust Methods in Econometrics", *Econometric Reviews*, 1(2), 213-255.

Krishnakumar, J. and Ronchetti, E. (1997) "Robust estimators for simultaneous equations models", *Journal of Econometrics*, 78, 295-314.

Lajbcygier P., Boek C., Palaniswami M. and Flitman A. (1996) "Comparing conventional and artificial neural network models for the pricing of options on futures", *Neurovest Journali,* 4(5), 16-24.

Lajbcygier, P., Flitman, A., Swan, A. and Hyndman, R. (1997) "The pricing and trading of options using a hybrid neural network model with historical volatility", *Neurovest Journal*, 5(1) 27-41.

Latane, H.A. and Rendleman, R.J.Jr, (1976) "Standard Deviations of Stock Price Ratios Implied in Option Prices", *The Journal of Financei,* vol. 31, no. 2, 369-381.

Lehar, A., Scheicher, M., and Schittenkopf, C. (2002). "GARCH Vs. Stochastic Volatility: Option Pricing and Risk Management." *Journal of Banking and Finance*, vol. 26, 323-345.

Leland, H. (1985) "Option Pricing and Replication with Transaction Costs", *Journal of Finance* 40, 1283-1301.

Lim, G.C., Lye J.N., Martin G.M. and Martin V.L. (1997) "Jump Models and Higher Moments". In Creedy J., Martin V. L. (eds.), *Nonlinear Economic Models. Cross-sectional, Time Series and Neural Networks Applications*. Edward Elgar Publishing, Inc., Lyme NH, US.

Long, D.M. and Officer, D.T. (1997) "The Relation Between Option Mispricing and Volume in the Black-Scholes Option Model", *Journal of Financial Research*, XX (1), 1-12.

Lye, J.N. and Martin, V.L. (1993) "Robust Estimation, Non-normalities and Generalized Exponential Distributions" *Journal of the American Statistical Association*, 88 (421), 253-259.

MacBeth, J.D., Merville, L.J., (1980) "Tests of the Black-Scholes and Cox call option valuation models", *Journal of Finance*, vol. 35, no. 2, 285-301.

Merton, R.C. (1973) "Theory of Rational Option Pricing." *Bell Journal of Economics and Management Science* 4, 141-183.

Morgenthaler, S. (1990). Fitting Redescenting M-estimators in Regression. In Lawrence, K.D. and Arthur J.L. (eds.) *Robust Regression*, 105-128. Dekker, NY.

Müller, K., Smola, A., Ratsch, G., Schölkoph, B., Kohlmorgen, J., and Vapnik, V. (1999) "Using Support Vector Machines for Time Series Prediction", In B. Schölkoph, J. Burges, and Smola, A. (Eds.), *Advances in Kernel Methods: Support Vector Machines*, Cambridge, MA: MIT Press.

Ortelli, C. and Trojani, F. (2005) "Robust efficient method of moments", *The Journal of Econometrics*, 128(1), 69-97.

Pan, J. (2002) "The Jump-Risk Premia Implicit in Options: Evidence from Integrated Time-Series Study", *Journal of Financial Economics*, vol. 63, 3-50.

Qi, M. and Maddala, G. S. (1996) "Option Pricing Using Artificial Neural Networks: The Case of S&P 500 Index Call Options", *Neural Networks in Financial Engineering, Proceedings of 3rd International Conference on Neural Networks in the Capital Markets,* Oct 1995, London, World Scientific, pp. 78-91.

Rafteri A. E., Madigan D. and Hoeting J. (1997) "Bayesian Model Averaging for Regression Models," Journal of the American Statistical Association,

Roon, De F., Veld C., Wei, J., (1998 "A study on the efficiency of the market for Dutch long-term call options", *European Journal of Finance* 4 (2) 93-111.

Rousseeuw, P. and Yohai, V.J. (1984) "Robust Regression by Means of S-estimators. Robust and Nonlinear Time Series Analysis". *Lecture Notes in Statistics*, 26, 256-272. Springer, NY.

Rubinstein, M. (1976) "The Valuation of Uncertain Income Streams and the Pricing of Options", *The Bell Journal of Economics* 7, 407-425.

Rubinstein, M. (1994) "Implied Binomial Trees." *The Journal of Finance*, vol. 49, 771-818.

Schittenkopf C. and Dorffner, G. (2001) "Risk-Neutral Density Extraction from Option Prices: Improved Pricing with Mixture Density Networks." *IEEE Transactions on Neural Networks* vol. 12, no. 4, 716-725.

Smola, A. and Schölkoph, B. (1998). "A Tutorial on Support Vector Regression" *NeuroCOLT Technical Report*, NC-TR-98-030, Royal Holloway College, University of London, UK.

Smola, A. J., Schölkopf, B., and Müller, K. R. (1998) "The Connection Between Reguralization Operators and Support Vector Kernels" *Neural Networks*, 11, 637-649.

Suykens, J. A. K., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewaller, J. (2002) *Least Squares Support Vector Machines*, World Scientific Publishing.

Tsay, R. (2002) *Analysis of Financial Time Series*, John Wiley & Sons, New York.

Vähämaa, S. (2003) "Skewness and Kurtosis Adjusted Black-Scholes Model: A Note on Hedging Performance." *Finance Letters* vol. 1, no. 5, 6-12.

Vapnik V., (1995), *The Nature of Statistical Learning Theory*, Springer-Verlag, New York.

Vapnik V., (1998), *Statistical Learning Theory*, John Wiley and Sons, New York.

Vapnik, V. (1998) *Statistical Learning Theory*, New York: Wiley.

Vishwanathan, S. V., Smola, A. J., and Narasimha, M. M. (2003), "SimpleSVM" *Proceedings of the Twentieth International Conference on Machine Learning*.

Vojislav, K. (2001) *Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models*, The MIT Press, London, England.

Watson, P. and Gupta, K.C. (1996) "EM-ANN Models for Microstript Vias and Interconnected in Multilayer Circuits." *IEEE Trans., Microwave Theory and Techniques*, 2495-2503.

Whaley, R. E. (1982) "Valuation of American Call Options on Dividend-Paying Stocks." *Journal of Financial Economics*, vol. 10, 29-58.

Xi, C., Wang, F., Devabhaktuni, V.K. and Zhang, J.Q. (1999) "Huber optimization of neural networks: a robust training method". *International Joint Conference on Neural Networks*, 1639-1642.

Yao, J., Li, Y., Tan, C.L., (2000) "Option Price Forecasting Using Neural Networks", *The International Journal of Management Science*, vol. 28, no. 4, 455-466.

Yohai, V.J. (1987) "High Breakdown-Point and High Efficiency Robust Estimates for Regression". *Annals of Statistics*, 15 (2), 642-656.