Generalized Parameter Functions for Option Pricing

Andreou C. Panayiotis, Charalambous Chris, Martzoukos H. Spiros

University of Cyprus

Department of Public and Business Administration

This version, October, 2006

Abstract

In this paper we combine parametric option pricing (Black and Scholes, 1973, Corrado and Su, 1997, and the Dumas et al., 1998 Deterministic Volatility Functions) models with a nonparametric methodology. Our approach can be seen as a generalization of Dumas et al. and retains the intuition in Christoffersen and Jacobs (2004) that, in deriving implied parameters, optimization should be in terms of the pricing function. The resulting enhanced structure is compared to parametric models with both standard implied parameters and parameters derived via Deterministic Volatility Functions. Empirical results using three years of S&P 500 index call option prices strongly support that our approach significantly improves the performance of parametric option pricing models (Black and Scholes and Corrado and Su).

JEL classification: G13, G14

Keywords: Option pricing, implied volatilities, implied parameters, deterministic volatility functions, artificial neural networks

Acknowledgements: We are thankful to seminar participants at the U. of Cyprus and U. of Manchester, the EFMA Annual Meetings in Madrid, Spain (June 2006) and the Int. Conference on Computing in Economics and Finance in Limassol, Cyprus (June 2006).

1. Introduction

We propose a new approach in the pricing of options by combining nonparametric methodology with several parametric option pricing models (POPMs). This proposed method provides a nonparametric enhancement of the implied parameter values to be used in the POPMs, and the resulting models we call Enhanced Parametric Option Pricing Models (EPOPMs).

The Black and Scholes model (BS) is an options pricing formula (Black and Scholes, 1973, see also Merton, 1973) that is built on a set of unrealistic assumptions and exhibits *systematic biases* like the *volatility smile* (i.e. Black and Scholes, 1975, Rubinstein, 1985, Bakshi et al., 1997, Andresen, 2002). BS has shown severe time endurance and is still widely used by practitioners since it generates quite accurate prices for a wide spectrum of European financial options. The post-BS financial engineering research came up with a variety of POPMs that relax several of the BS fundamental assumptions. Recent POPMs that incorporate stochastic volatility and jump risk factors (e.g. Bakshi et al., 1997, Bates, 1991 and 1996), mitigate much of the bias associated with the original BS. A similar effect is achieved indirectly with the semi-parametric Corrado and Su (1996) model (CS), an important alternative due to its ease of use[1]. Nevertheless, none of these models has managed to generalize *all* of the BS assumptions, and provide results consistent with the observed market data. Important empirical option pricing studies include Bakshi et al. (1997), that examine the cross sectional pricing performance of alternative option pricing models. See also Eraker (2004), Bates (1996 and 2000), Corrado and Su (1996 and 1997), Whaley (1982), Lehar et al. (2002) etc. Our EPOPMs can be considered as a generalization of studies that first employ some kind of methodology to estimate versions of time varying volatility that is subsequently used with a parametric (like the BS) model to price options. For instance, Dumas et al. (1998) estimate arbitrary Deterministic Volatility Functions of quadratic forms and examine how well they predict option prices. Our approach extends Dumas et al. (1998) by retaining the intuition in Christoffersen and Jacobs (2004) that while calculating implied volatilities optimizing should be in respect to the option pricing function. Besides the fact that many complex parametric models seem

[1] 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).

to perform better than BS[2] they are *often too complex to implement, have poor out-of-sample pricing and hedging performance and have implausible and sometimes inconsistent implied parameters* (i.e. Bakshi et al., 1997).

Researchers have also addressed attention to the use of *market-data driven* models such as ANNs that can be used for nonlinear regression. The key power provided by ANNs compared to other statistical techniques (like projection pursuit, generalized additive models, multivariate adaptive regression splines) is that they rely on fairly simple algorithms and the underlying form of the nonlinearity can be learned from training data. The models are very powerful, have nice theoretical properties (with respect to convergence), and apply well to a vast array of real-world applications (see Duda et al., 2001, for further details). Attempts in pricing options with ANNs have shown that these models are promising alternatives in respect to robust pricing accuracy. Contrary to the parametric option pricing models that rely on specific assumptions about the dynamic evolution of some state variables (like the underlying asset, the volatility, the interest rate, etc), ANNs involve no financial theory since option prices are estimated inductively by using options transactions data. ANNs are used to estimate directly the empirical options pricing function (thereinafter termed as the standard ANN approach). Evidence concerning the out-of-sample pricing performance is mixed. Hutchison et al. (1994) apply ANNs on market transactions of the S&P 500 futures call options from 1987 to 1991 to conclude that although the learning networks do not constitute a substitute for the more traditional BS formulas, they are more accurate and computationally more efficient alternatives when the underlying asset's price dynamics are unknown. Anders et al. (1998) as well as Garcia and Gencay (2000), find that the BS with historical volatility underperforms significantly the standard ANNs. Of course, the application of ANNs for pricing of options has also its limitations. First of all, Anders and Korn (1999) indicate that 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

[2] Although the post-BS option-pricing models have managed to eliminate some of the BS biases in practice are very difficult to be implemented due to their complexity. 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*".

information related to the problem under scrutiny. Second, in the absence of any kind of prior information about the problem, ANNs need relative large amounts of training data to ensure an adequate accuracy. As supported by Lajbcygier (2004), the standard ANNs are very sensitive to the nonstationarity of input variables and this problem is exaggerated with the use of large training-validation-testing datasets. Finally, the use of standard ANNs can deliver options prices that violate fundamental financial principles; for instance they can 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) also show that standard ANNs return negative implied state price densities in state regions that available training options data do not contain any information about these regions.

The above discussion demonstrates that we should explore structures that are theoretically consistent with the POPMs. The scope of this paper is to provide a nonparametric enhancement of the implied parameter values used in the POPMs. We thus generalize Dumas et al. (1998) (see also Christoffersen and Jacobs, 2004) Deterministic Volatility Functions (DVF) by using Neuromodeling (ANN) techniques. The DVF approach and our Generalized approach makes an imperfect (parametric) model fit a better one (in this case the market). This methodological framework is conceptually similar to the one developed in Electrical Engineering (see Bandler et al., 1999; and the Space Mapping techniques in Bandler et al., 1994) where the parameter values to an imperfect model are adjusted so as to make the imperfect model approximate the performance of a finer but more expensive one to use. In our case, the nonparametric parameter enhancement will provide the volatility to the BS model, and the parameters to 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 a neural network. Such structures in conjunction with the Black and Scholes model are desirable for a variety of reasons. First, they always return arbitrage-free and nonnegative option values and we thus expect them to exhibit satisfactory pricing performance at the boundary of option pricing areas, in both dense and sparse input areas. Similarly, it is also *certain* that EPOPMs will deliver theory consistent Greek letters. Second, the proposed approach assures nonnegative implied state price densities in all cases. Third, as conjectured by

Wang and Zhang (1997), knowledge enhanced ANN structures should not need large amount of training samples to exhibit a satisfactory performance in out of sample testing as opposed to the case of standard ANNs. In the case the Corrado and Su (or any other) model is used, the first two reasons are true to the extent the theoretical model possesses the relevant properties.

The data for this research come from two dominant world markets, the New York Stock Exchange (NYSE) for S&P 500 and the Chicago Board of Options Exchange (CBOE) for call option contracts, spanning a period from January 2002 to August 2004. Compared to previous literature in empirical options pricing, we examine more explanatory variables including historical and implied ones. Also, instead of constant maturity risk-free interest rate, we use nonlinear interpolation for extracting a continuous risk-free interest rate according to each option's time to maturity.

In this study we build EPOPMs for the BS and the CS model. We compare them with their parametric alternatives using the overall average implied parameters and their DVF versions (which we also extend to the CS model). Moreover, as a benchmark model we include the stochastic volatility and jump model of Bates (1996) since it is an effective remedy to the BS biases (see Bakshi et al., 1997, and Bates, 1996). In the following section we review the parametric models and in the next we explain the implementation of the EPOPM structure. We then discuss the data, filtering and the alternative versions of the models that are compared. Finally we discuss the results and we conclude. It can be seen that the proposed methodology improves significantly pricing performance of parametric option pricing models.

2. The 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 is a benchmark and widely referenced model. The Black Scholes formula for European call options modified for dividend-paying underlying asset is:

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

where,

$$d = \frac{\ln(S/X) + (r - d_y)T + (\sigma\sqrt{T})^2/2}{\sigma\sqrt{\mathrm{T}}} \tag{1.a}$$

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 risk free interest rate*; $d_y \equiv$ *continuous dividend yield paid by the underlying asset*; $T \equiv$ *time left until the option expiration date*; $\sigma^2 \equiv$ *yearly variance rate of return for the underlying asset*; *N(.)* $\equiv$ *the standard normal cumulative distribution* .

The need to include in the analysis another POPM is necessitated by the *smile* behavior of the BS implied volatility for various moneyness (the ratio of the underlying asset to strike price) and time to maturity levels. (see Bakshi et al., 1997). So, we use in addition the Corrado and Su (1996) model, and we include as a benchmark the Stochastic Volatility and Jump model of Bates (1996).

The Corrado and Su model (CS) constitutes an extension of the BS model that accounts for additional skewness and kurtosis in stock returns in a heuristic manner. Corrado and Su, based their extension on a methodology employed earlier in 1982 by Jarrow and Rudd. Using a Gram-Charlier series expansion of a normal density function they defined their model as (see also the correction in Brown and Robinson, 2002):

$$c^{CS} = c^{BS} + \mu_3 Q_3 + (\mu_4 - 3)Q_4 \tag{2}$$

where c^{BS} is the BS value for the European call option adjusted for dividends and,

$$Q_3 = \frac{1}{3!} S e^{-d_y T} \sigma\sqrt{T}((2\sigma\sqrt{T} - d)n(d) + \sigma^2 T \mathrm{N}(d)) \tag{2.a}$$

$$Q_4 = \frac{1}{4!} S e^{-d_y T} \sigma\sqrt{T}((d^2 - 1 - 3\sigma\sqrt{T}(d - \sigma\sqrt{T}))n(d) + \sigma^3 T^{3/2} N(d)) \tag{2.b}$$

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,

$$n(z) = \frac{1}{\sqrt{2\pi}} exp(-z^2 / 2) \tag{2.c}$$

refers to the standard normal probability density function.

Bakshi et al. (1997) found that the Stochastic Volatility and Jump model (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 to mitigate the "smile" anomaly. 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 model the underlying asset follows geometric jump diffusion with the instantaneous conditional variance, V_t, to follow a mean-reverting root process:

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

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

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, a/β translate to a variance steady-state level and β is the rate of mean reversion.

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

$$c^{SV,\,SVJ} = e^{-rT}[FP_1 - XP_2]$$

with $F = E(S_T) = Se^{(r-d_y)\mathrm{T}}$ 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 P_1 and P_2 is done under the distributional assumptions embedded in the *risk-neutral probability* measures by using the moment generating functions of $ln(S_T / S)$ (Bates, 1996). The following expressions are needed to compute P_1 and P_2:

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

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

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

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

$$\mu_1 = 0.5\,,\quad \mu_2 = -0.5\,,\quad B_1 = \beta - \rho\sigma_v\,,\quad \mathrm{B}_2 = \beta$$

and the resulting probabilities are derived by the numerical Fourier inversion:

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

with $\chi \equiv ln(X / S)$ and the integrals to be evaluated with an adaptive Lobatto quadrature.

In this work, we fit the POPMs in daily prices to obtain the implied parameters that minimize an error measure, so these parameters should be perceived as the risk-neutral ones indirectly accounting for the pricing of jump and volatility risk.

Extending the Deterministic Volatility Functions

As mentioned before, the EPOPMs can be thought of as nonparametric/nonlinear generalizations of the Deterministic Volatility Function methodology proposed by Dumas, Fleming and Whaley (1998) and Christoffersen and Jacobs (2004). According to the Dumas et al. (1998), this ad-hoc approach of smoothing the BS implied volatilities across strike prices and maturities exhibits superior in and out of the 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 (and possibly the CS one). They recommend deriving the implied volatility by optimizing in respect to the option pricing function. In addition, Berkowitz (2001) demonstrates based on a theoretical justification 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 three different DVF models as in the study of Dumas, Fleming and Whaley (1998):

DVF#1: $\sigma = \max(0.01, \alpha_0 + \alpha_1 X + \alpha_2 X^2)$

DVF#2: $\sigma = \max(0.01, \alpha_0 + \alpha_1 X + \alpha_2 X^2 + \alpha_3 T + \alpha_4 XT)$

DVF#3: $\sigma = \max(0.01, \alpha_0 + \alpha_1 X + \alpha_2 X^2 + \alpha_3 T + \alpha_4 XT + \alpha_5 T^2)$

The DVF approach as proposed by Dumas et al. (1998) refers to the estimation of the volatility function via simple Ordinary Least Squares by regressing the implied volatilities on different polynomials of T and X. Christoffersen and Jacobs (2004) demonstrate that doing this yields biased estimates of the observed option prices and show how the DVF should be estimated consistently via Nonlinear Least Squares. We estimate the three different DVF models each day using the original (*L*) approach via simple OLS and also with the consistent (*NL*) approach via the nonlinear least squares. For the latter we use several initializations to minimize the risk of estimating coefficients based on a local minimum of the optimization function.

Necessary Greeks

Greek letters are the partial derivatives of a call options with respect to its structural parameters. For the purpose of this study, and in order to have an efficient optimization/training of the EPOPM structure, we need the following Greek letters:

- *BS Vega*:

$$V^{BS} \equiv \frac{\partial c^{BS}}{\partial \sigma} = Se^{-d_y T} \sqrt{T} n(d) \tag{3}$$

- *CS Vega*:

$$\frac{\partial c^{CS}}{\partial \sigma} = \frac{\partial c^{BS}}{\partial \sigma} + \frac{1}{3!} \mu_3 \frac{\partial Q_3}{\partial \sigma} + \frac{1}{4!} (\mu_4 - 3) \frac{\partial Q_4}{\partial \sigma} \tag{4}$$

where,

$$\begin{aligned} \frac{\partial Q_3}{\partial \sigma} = Se^{-d_y T} n(d) \sigma \sqrt{T} \left[3d\sigma\sqrt{T} + 3d^2 + \sigma\sqrt{T} + 3 \right] - \\ S\sqrt{T} d^3 n(d) + 3S\sigma^2 T^{3/2} N(d) \end{aligned} \tag{5}$$

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

- CS *partial derivative of call with respect to skewness*:

$$\frac{\partial c^{CS}}{\partial \mu_3} = Q_3 \tag{7}$$

- CS *partial derivative of call with respect to kurtosis*:

$$\frac{\partial c^{CS}}{\partial \mu 4} = Q_4 \tag{8}$$

3. The EPOPM structure

Multilayer Neural Networks are flexible heuristic techniques for doing statistical pattern recognition and for approximating highly nonlinear functions. A 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 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). 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.

Figure 1 depicts the general idea of the EPOPM structure while Figure 2 depicts the exact network structure developed for the purposes of this study. For our analysis, inputs are set up in feature vectors, $\tilde{x}_q = [x_{1q}, x_{2q} \ldots, x_{Nq}]$ for which there is an associated and known target, t_q, $q \equiv 1,2,\ldots,P$, where P is the number of the available sample feature vectors for a particular training sample. The network's outputs are obtained when the training patterns are presented as inputs at 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 the error between the network output and the desire target values.

The proposed network model under scrutiny has four layers. The first three are typical ANN layers: 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 ($i=1,2,\ldots,H$, $n=1,2,\ldots N$) and a *weight*, $w_{ji}^{(2)}$, and a *bias*, $w_{j0}^{(2)}$, in the hidden layer ($j=1,2,\ldots,M$). 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 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 2 is a typical two-layer ANN with a single output that under proper treatment can be used for nonlinear regression (Hutchison et al., 1994).

The fourth layer, which hereafter will be termed as an *enhanced layer*, makes possible for a chosen POPM to be an *inseparable* part of the network's structure. This is the innovative contribution of the model since under this setting we can hypothesize that our 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 $X_{S1} \supseteq X_S$ should correspond to the *enhanced*[3] variables coming from the network's output layer and $X_{S2} \supset X_S$ those variables that are passed to the parametric model *exogenously*. It is obvious that $X_{S2} = X_S - X_{S1}$ and in the case that we choose to let all parametric model variables to be determined via the network, then $X_{S2} = \varnothing$. The definition of X_{S1} is basically a choice of the researcher and manifests the number of neurons at the output layer and the type of transfer function to be used at the enhanced layer.

According to Figure 2, the operation carried out for computing the final estimated output, *y*, is the following:

$$y = f_{PM}(v, X_{S2}) \tag{9}$$

and,

$$v = [f_{d_1}(d_1), f_{d_2}(d_2), \ldots, f_{d_M}(d_M)] \tag{10}$$

where $f_{PM}(.,.)$ refers to the functional form of the parametric options pricing model, $f_{dj}(.)$ are a smooth monotonically increasing transfer functions and d_j are simply the descaled values of $y_j^{(2)}$, where $j = 1, 2, \ldots, M$.

Computation of $y_1^{(2)}$ follows the functional form of a typical two-layer ANN:

[3] We use the term "enhanced variable" to describe the number of variables that are used as input to the parametric model and come as an output of the network.

$$y_1^{(2)} = f_M[w_{10}^{(2)} y_0 + \sum_{i=1}^{H} w_{1i}^{(2)} f_H(w_{i0}^{(1)} x_{s0} + \sum_{n=1}^{N} w_{in}^{(1)} x_{sn})] \quad (11)$$

where $f_M(.)$ and $f_H(.)$ are smooth monotonically increasing transfer functions associated with the output and hidden layer respectively and x_{sn}, n=1,2,...,N, is just the scaled value of the input x_n. The network's structure employs a scaling scheme for both the inputs and the enhanced variables. This is essential for the training of ANNs since it increases the effectiveness of the optimization algorithm and minimizes the significance of differing dimensions of the input signals (see Haykin, 1999, and Bishop, 1995). We apply a standard *z-score* scaling: $\tilde{z} = (\tilde{x} - m)/s$, where $\tilde{x}$ is the vector of an input/enhanced variable, m is the mean and s the standard deviation of this vector.

[Figures 1 and 2, here]

For our case, the smooth monotonically increasing transfer function is either the hyperbolic tangent sigmoid,

$$f(\eta) = \alpha \left[\frac{e^{b\eta} - e^{-b\eta}}{e^{b\eta} + e^{-b\eta}} \right] \quad (12)$$

the logistic,

$$f(\gamma) = \frac{\alpha}{1 + e^{-b\gamma}} \quad (13)$$

or the linear one,

$$f(\xi) = \xi \quad (14)$$

In the above expressions, with $\alpha, b \in \Re$ where α controls the output range and b the slope of the transfer function. As advised by 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 Bishop (1995) (see also Duda et al., 2001), these transfer functions work well with ANNs. In the hidden layer we always use the standard hyperbolic tangent sigmoid transfer function (with a and b equal unity) for $f_H(.)$, while in the output layer we use a linear transfer function for $f_M(.)$ as this is necessitated by the scaling scheme we apply at the output layer.

The choice of the transfer function at the enhanced layer is dictated by the type of the parametric model we use and the kind of the enhanced variable(s) we choose to map via the network; 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 transfer functions are necessary during the implementation of the method in order to ensure that each of the enhanced variable value is within an acceptable range for use with the parametric model[4]. Table 1 (Panel A) describes the different transfer functions we use at the enhanced level for all cases considered. We use transfer functions that truncate implicitly the enhanced variable value range. For instance in the case of BS we do not allow volatility to be larger than 70%, and for the case of CS, skewness is confined in the $[-15,15]$ range. The choice of the truncation point is not crucial for the implementation of the models as long as we allow the enhanced variables to vary into plausible ranges. 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 (e.g. Corrado and Su, 1997, Bates, 1991).

[Table 1, here]

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

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

$$e_q(w) = y_q(w) - t_q \tag{15}$$

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

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

The traditional backpropagation algorithm which is based on the gradient descent vector is the most popular method for training the ANNs. It is shown in Charalambous (1992) that this training algorithm is often unable to converge *rapidly* to the optimal solution. So, in this paper we rely on the Levenberg-Marquardt algorithm (LM) which is much more efficient training method in terms of training time and convergence rate. 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:

$$F(w) = \sum_{q=1}^{P} e_q^2(w) \equiv \sum_{q=1}^{P} (y_q - t_q)^2 \tag{16}$$

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

$$w_{\tau+1} = w_\tau + \left[J^T(w_\tau)J(w_\tau) + \mu_\tau I\right]^{-1} J^T(w_\tau)e(w_\tau) \tag{17}$$

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:

$$J(w) = \begin{bmatrix} \nabla e_1^T(w) \\ \vdots \\ \nabla e_P^T(w) \end{bmatrix} \tag{18}$$

In the above, I is $\nu \times \nu$ identity matrix, and μ_τ is like a learning parameter that is automatically adjusted in each iteration in order to secure convergence (by assuring that the part in the square brackets of Eq. (17) is always nonsingular). 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 Eq. (20), the weights and biases update takes place in a batch mode and only when all input vectors have been presented to the network. Moreover, 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 transfer function so that the active regions of the layer's neurons are distributed roughly evenly over the input space.

The quantity $\nabla e_q(w)$ is the gradient vector of $e_q(w)$ with respect to the trainable parameter vector w. This quantity is computed in a similar fashion (see Charalambous, 1992) as with the case of the traditional backpropagation algorithm that is commonly used in the context of multilayer perceptron neural networks. Since the error function does not depend explicitly upon the network's weights, $\nabla e_q(w)$ is evaluated via the chain rule. Based on the neural network model depicted in Figure 2, the partial derivative of the error function in Eq. (15) with respect to the weight $w_{ji}^{(2)}$ at the hidden layer is:

$$\frac{\partial e_q}{\partial w_{ji}^{(2)}} = \delta_j^{(2)} y_i^{(1)} \tag{19}$$

and,

$$\delta_j^{(2)} = \frac{\partial f_{PM}}{\partial v_j} f'_{d_j}(d_j) s_j f'_M(\psi_j^{(2)}) \tag{20}$$

where $f'_M(\psi_j^{(2)})$ and $f'_{d_j}(.)$ are the differentials at points $\psi_j^{(2)}$ and d_j respectively, and s_j the standard deviation of the enhanced variable as used during scaling.

Quantity $\frac{\partial f_{PM}}{\partial v_j}$ is the partial derivative of the parametric model with respect to input v_j and makes our network model more dedicated to options pricing. This quantity is very important during training of the ANN because it incorporates knowledge from a parametric model. All necessary Greek letters for the implementation of the alternative models have been previously discussed in the parametric models section.

The partial derivative of the error function in Eq. (15) with respect to the weight $w_{in}^{(1)}$ at the input layer is:

$$\frac{\partial e_q}{\partial w_{in}^{(1)}} = \delta_i^{(1)} x_{sn} \tag{21}$$

where,

$$\delta_i^{(1)} = \varepsilon_i^{(1)} f_H'(\psi_i^{(1)}) \tag{22}$$

$$\varepsilon_i^{(1)} = \sum_{j=1}^{M} w_{ji}^{(2)} \delta_j^{(2)} \tag{23}$$

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. The EPOPM structures with 2 to 6 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, the network is initialized, trained and validated ten separate times. 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* in order to verify the ANN ability to generalize to unseen data.

4. Data and Methodology

The data considered cover the period January 2002 to August 2004. 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 (Garcia and Gencay, 2000). All options data are purchased from CSI. For each trading day we have the available last transaction call price, c^{mrk}, along with the strike price[5], X, date of expiration[6], volume and open interest. Along with the index, we have collected a daily dividend yield, d_y, provided online by Datastream.

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 ten different *overlapping* training (*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. For instance, the first *trn* set covers the period January to December 2002, the first *vld* set covers the period January to February 2002, 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 an aggregate testing period (*agr*) with about 22,000 datapoints by simply pooling together the pricing estimates of all twenty *tst* periods. For period *agr*, 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).

Filtering Rules

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 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 0.5 index points are not included. Finally, we demand at least four datapoints per

---

5 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 $S/X > 1.10$.

6 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.

maturity to secure that during the implied parameters extraction process, every maturity period is satisfactorily represented. The final dataset used is still larger than previous ANN studies. For instance Hutchison et al. (1994) have an average of 6,246 data points per sub-period; Schittenkopf and Dorffner (2001) include a total of 33,633 data points. Sample characteristics for the dataset can be found in Table 2.

[Table 2, here]

Observed Structural Parameters

The moneyness ratio, S/X, is usually the basic input in all network structures since it is highly related with the pricing bias associated with the POPMs (see Hutchison et al., 1994, and Garcia and Gencay, 2000). The dividend adjusted moneyness ratio $(Se^{-d_y T})/\mathrm{X}$ 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, 1, 3, 6, and 12 months constant maturity T-bills rates (collected from the U.S. Federal Reserve Bank Statistical Releases) were considered.

Implied Volatility Measures

The methodology employed in this study for the estimation of the overall average implied parameters is similar to that in previous studies that somehow 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 POPM prices (c^k, k=BS or CS) plus a random additive disturbance term (ε_N^k, k=BS or CS):

$$c_N^{mrk} = c_N^k + \varepsilon_N^k \tag{26}$$

where N refers to the number of different call option transaction datapoints available. To find optimal implied parameter values we solve an optimization problem that has the following form:

$$SSE(t) = \min_{\xi^k} \sum_{j=1}^{N_t} (\varepsilon_j^k)^2 \tag{27}$$

where t represents the time instance, ξ^k the unknown parameters associated with a specific parametric option pricing model (k = BS, CS, and SVJ). The SSE is minimized via a non-linear least squares optimization based again on the Levenberg-Marquardt algorithm. To minimize the possibility to obtain implied parameters that correspond to a local minimum of the error surface with each model we use different starting values for the unknown parameters based on reported average values for the S&P 500 according to Bates (1991), Bakshi et al. (1997), and Corrado and Su (1996 and 1997).

From the above we obtain the following sets of implied parameters:

a. Overall average implied BS volatility estimates $\xi^{BS} = \{\sigma_{av}^{BS}\}$
b. Overall average implied CS estimates $\xi^{CS} = \{\sigma_{av}^{CS},\ \mu_3,\ \mu_4\}$.
c. Overall average implied SVJ estimates $\xi^{SVJ} = \{\sigma_{av}^{SVJ},\ \lambda,\ k,\ \Theta,\ \alpha,\ \beta,\ \sigma_v,\ \rho\}$.

In addition to the above overall average implied parameters we also estimate the three DVF models (DVF#1, DVF#2, and DVF#3) defined earlier. For BS this is straightforward; for CS and SVJ we compute the implied Brownian volatilities after we first estimate and fix the overall average implied parameters. So for CS and SVJ the DVF is a two stage nonlinear estimation and results to three additional volatility measures per model. We differentiate them by using appropriate subscripts: σ_{NL1}^k, σ_{NL2}^k and σ_{NL3}^k for the nonlinear estimation and σ_{L1}^k, σ_{L2}^k and σ_{L3}^k for OLS estimation (k = BS and CS). In addition, the volatility estimates obtained via the nonlinear least squares based on initial values obtained from the ordinary least squares are[7]: σ^{k}_{NLL1}, σ^{k}_{NLL2} and σ^{k}_{NLL3}. For pricing reasons at time instant *t*, the implied structural parameters derived at day *t-1* are used together with all other needed information.

5. Comparison of the Alternative Models

With the BS models we use as input *S, X, T,* d_y, *r*, and any of the following ten volatility forecasts: σ^{BS}_{j} where *j* = {*av, L1, L2, L3, NL1, NL2, NL3, NLL1, NLL2, NLL3*} and we use BS_j to differentiate the alternative BS models. In a similar way and using the proper symbolization, there are ten different CS and SVJ models according to the implied parameters used.

The notation for the models depends on the parametric model considered. We use eBS_j, with *j*={*av, NL2*}, to denote the two enhanced networks that use as an additional input variable the BS volatilities: σ^{BS}_{av} and σ^{BS}_{NL2}; for these models volatility is the only enhanced variable. In the same spirit we use eCS^{sig}_{j}, with *j*={*av,NL2*}, to denote the two EPOPMs that use as additional inputs the CS variables: σ^{CS}_{av} and σ^{CS}_{NL2} respectively, with volatility being the only enhanced variable. Finally, we use eCS^{all}_{j}, with *j*={*av, NL2*}, to denote the two networks that use as additional inputs the CS variables: σ^{CS}_{av}, μ_3, μ_4 and σ^{CS}_{NL2}, μ_3, μ_4 respectively, with volatility, skewness and kurtosis being the enhanced variables. All EPOPM combinations are exhibited in Panel B of Table 1.

Tables 3A (in sample) and 3B (out of sample) exhibit the performance of all models considered in terms of RMSE, MAE and RMeSE, P_5AE, P_{95}AE for period *agr*. Since all types of ANNs are effectively optimized in respect to sums of squares (see Eq. 16), the out of sample pricing performance should be similarly judged on RMSE and in a lesser only degree on other measures.

[Tables 3A, 3B, 4, here]

[7] 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 the ordinary OLS.

We first concentrate our attention to Table 3A (in sample) for the parametric BS, CS and SVJ models. Before the Deterministic Volatility functions are estimated, the more complex models exhibit superior performance, and thus SVJ is the best, followed by CS. The DVF approach improves the pricing performance of the BS and CS models considerably, with the nonlinear version (NL3) being superior. Note that we tried the DVF approach on the SVJ model but it could not improve its performance. We then concentrate on Table 3B (out of the sample). The second nonlinear DVF model (NL2) provides the best out of sample performance for both BS and CS (although for the BS case NL1 was equally good in terms of the RMSE but inferior by far in terms of the other measures). Still, the SVJ model retains the place of the top performer in all metrics.

Finally, we look at Table 4 with the out of sample performance for the EPOPMs. We see here that all enhanced models have very good performance, and some are competitive to the SVJ model too. The best BS version is eBS_{NL2} which is the enhancement of NL2; and the best CS version is eCS_{NL2}^{all} which is the enhancement of NL2 providing as output all three parameters of CS (volatility, skewness and kurtosis). This latter model is also the overall best EPOPM in terms of the RMSE metric, needs to be estimated only once a month and competes with the SVJ model, which is expensive to calibrate daily.

[Table 5 here]

In Table 5 we tabulate statistics for a pairwise comparison of the (statistical significance of) pricing performance in terms of MSE for a selection of models. Since the eighteen testing periods are disjoint and because we have pricing estimates coming from different models we can assume (similarly to Hutchison et al, 1994 and Schittenkopf and Dorffner, 2001) that the pricing errors are independent and a standard two-tail *t*-test can be applied. Similarly to the previous authors we need to report that these tests should be interpreted with caution. We see 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_{NL2}^{all} and is competitive to SVJ (any difference in performance is not statistically significant) and easier to estimate. Our second best EPOPM is eCS_{av}^{all} which is marginally only inferior to the SVJ but much easier to estimate.

6. Summary and Conclusions

In this study we generalize the Dumas et al. (1998) and Christoffersen and Jacobs (2004) DVF approach for option pricing, with a non-parametric approach. Our approach allows a set of the input variables to the parametric model to be jointly determined by a neural network. The enhanced parametric models (EPOPMs) proposed in this study have many desirable properties compared to standard ANNs, like arbitrage-free option values, and theory consistent option values and Greek letters. In general, this methodology is proposed as a way to eliminate some of the deficiencies of the modern parametric options models and standard ANNs.

We compare the proposed methodology with the Black and Scholes, the Corrado and Su and the Stochastic Volatility and Jump models. For pricing performance analysis we use the S&P 500 index call options, with both overall average implied parameters (for all three parametric models) and implied parameters derived from Deterministic Volatility Functions (for the BS and CS models), for the period January 2002 to August 2004.

The results obtained strongly support the proposed methodology. Specifically, we find that the increase in the pricing accuracy of EPOPM-BS over the standard BS models and of the EPOPM-CS over the CS model is considerable and statistically significant in the 1% level.

Literature

Anders, U., Korn, O., and Schmitt, C. (1998). "Improving the Pricing of Options: A Neural Network Approach." *Journal of Forecasting*, 17, 369-388.

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, pp. 1239-1276.

Andreou, P.C., Charalambous, C. and Martzoukos, S.H. (2005). "Pricing and Trading European Options by Combining Artificial Neural Networks and Parametric Models with Implied Parameters." To be published in the *European Journal of Operational Research*.

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, pp. 2003-2049.

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, pg. 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, pg. 2417-2427.

Bates, D. S. (1991) "The Crash of '87: Was it Expected? The Evidence From Options Markets." *Journal of Finance*, vol. 46, no. 3, pp. 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, pp. 69-107.

Berkowitz, J. (2004) "On Justifications for the Ad Hoc Black-Scholes Method of Option Pricing." Working Paper, University of Houston.

Black, F. and Scholes, M. (1973) "The Pricing of Options and Corporate Liabilities." *Journal of Political Economy*, vol. 81, pp. 637-654.

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.

Charalambous, C., (1992) "A Conjugate Gradient Algorithm for Efficient Trading of Artificial Neural Networks." *IEE Proceedings – G*, 139, pp. 301-310.

Christoffersen, P., and Jacobs, K. (2004) "The Importance of the Loss Function in Option Valuation." *Journal of Financial Economics*, vol. 72, pp. 291-318.

Cont, R., and Fonseca, J. (2002) "Dynamics of Implied Volatility Surfaces." *Quantitative Finance*, vol. 2, pp.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, pp. 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, pp. 8-19.

Cybenko, G., (1989) "Approximation by Superpositions of a Sigmoidal Function." *Mathematics of Control, Signal and Systems* 2, pp. 303-314.

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, pg. 2059-2106.

Eraker, B. (2004). "Do Stock Prices and Volatility Jump? Reconciling Evidence from Spot and Option Prices." *Journal of Finance*, vol. LIX, no. 3, pg. 1367-1403.

Garcia, R. and Gencay, R. (2000) "Pricing and Hedging Derivative Securities With Neural Networks and a Homogeneity Hint." *Journal of Econometrics*, vol. 94, pp. 93-115.

Hagan, M. T. and Menhaj, M. (1994) "Training Feedforward Networks with the Marquardt Algorithm." *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989-993.

Hagan, M., Demuth, H. and Beale, M. (1996) *Neural Network Design,* PWS Publishing Company.

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.

Hutchison, 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, pp. 851-889.

Jurczenko, E., Maillet, B., and Negrea, B. (2002). "Revisited Multi-moment Approximate Option Pricing Models: A General Comparison (Part I)." *Working Paper*, LSE-FM.

Lajbcygier, P. (2004). "Improving Option Pricing With the Product constrained Hybrid Neural Network." *IEEE Transactions on Neural Networks*, Vol. 15, No. 2, pg.465-476.

Lehar, A., Scheicher, M., and Schittenkopf., C. (2002). "GARCH Vs. Stochastic Volatility: Option Pricing and Risk Management." *Journal of Banking and Finance*, 26, pg. 323-345.

Merton, R.C. (1973) "Theory of Rational Option Pricing." *Bell Journal of Economics and Management Science,* vol. 4, pp. 141-183.

Rubinstein, M. (1985) "Nonparametric Tests of Alternative Option Pricing Models Using All Reported Trades and Quotes on the 30 Most Active CBOE Option Classes From August 23, 1976 Through August 31, 1978." *The Journal of Finance*, vol. XL, pp. 455-480.

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, pp. 716-725.

Wang, F., and Zhang, J.Q. (1997). "Knowledge-Based Neural Models for Microwave Design." *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 12, pg. 2333-2343.

Watson, P. and Gupta, K.C. (1996) "EM-ANN Models for Microstript Vias and Interconnected in Multilayer Circuits." *IEEE Trans., Microwave Theory and Techniques*, pp. 2495-2503.

Whaley, R.E. (1982) "Valuation of American Call Options on Dividend-Paying Stocks." *Journal of Financial Economics*, vol. 10, pp. 29-58.

Figure 1: Schematic description of the Enhanced structures

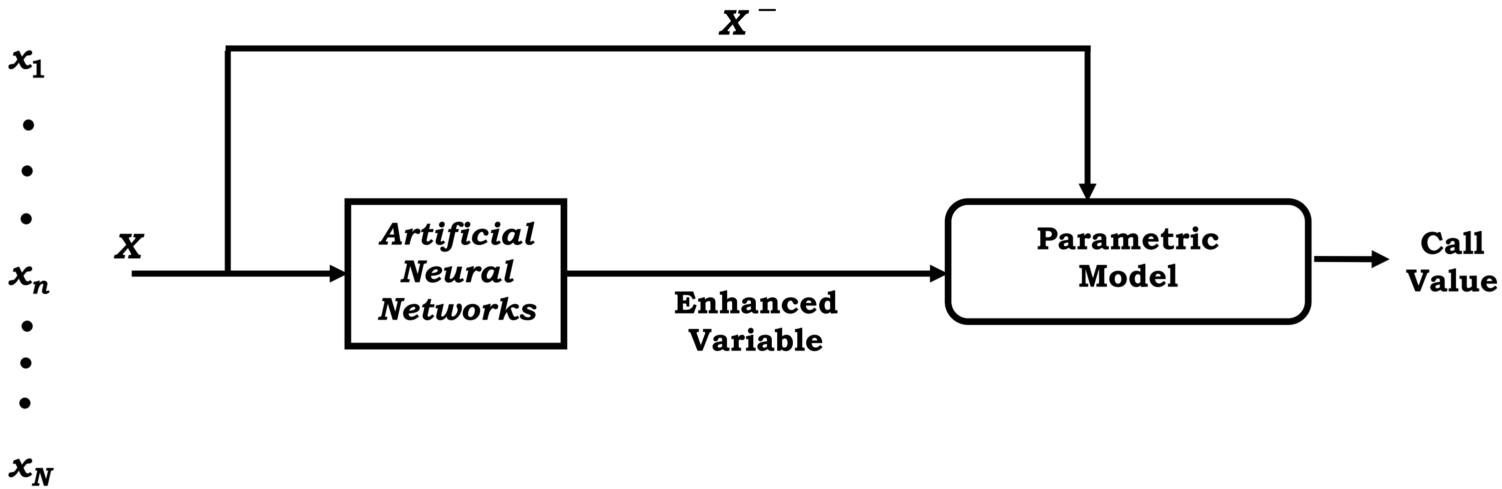
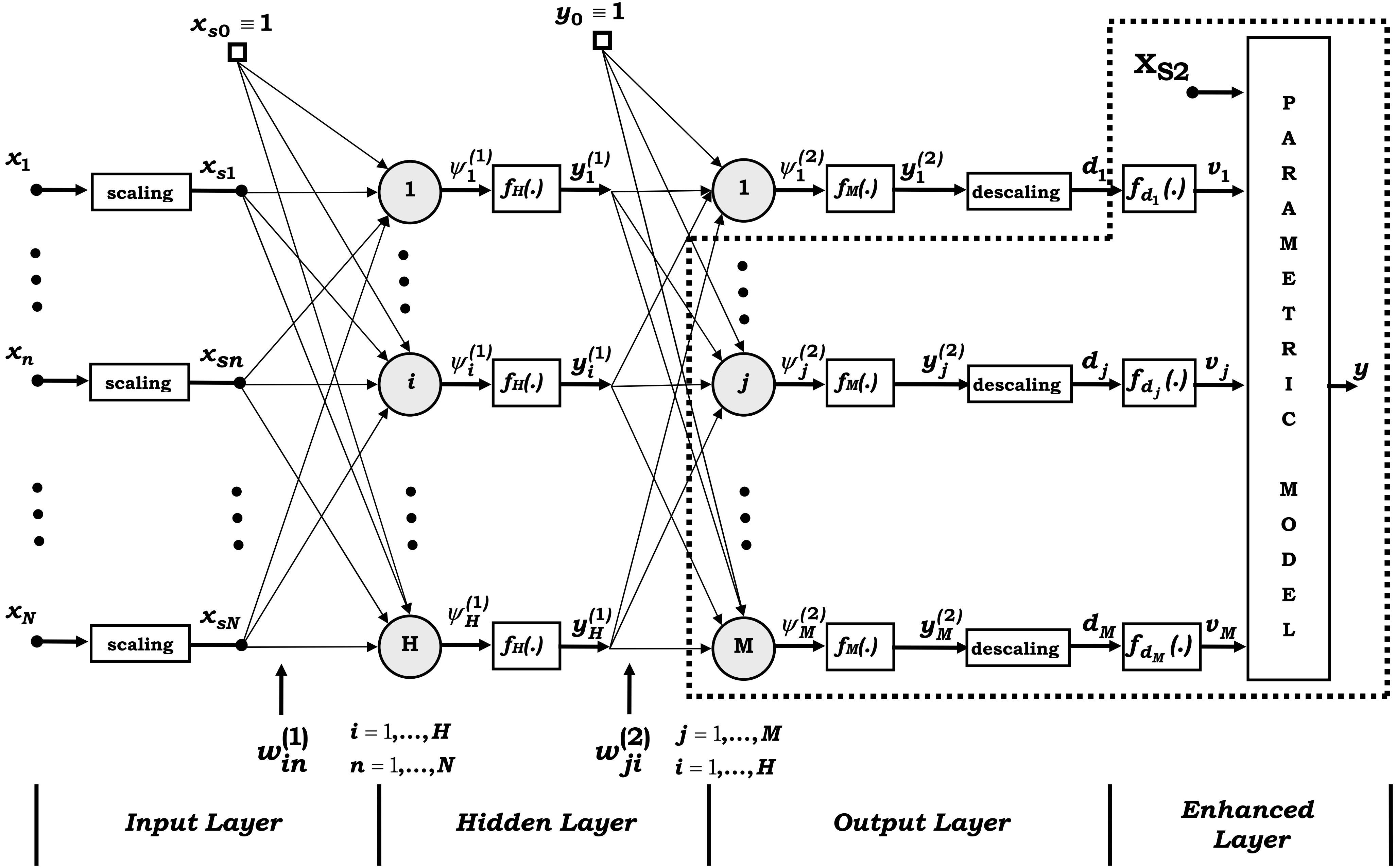


Figure 2: A detailed diagram of the Enhanced structures



Model	Enhanced Variable	Transfer Function	Parameter Values (a, b)
BS	Volatility	Logistic	(1.5,1)
CS	Volatility	Logistic	(1.5,1)
CS	Skewness	Tangent	(15,0.15)
CS	Kurtosis	Logistic	(30,0.20)

Panel A: *Transfer functions used with enhanced variables*

Model	Input Variables	Enhanced Variable(s)
eBS_{av}	$(Se^{-d_y T})/X$, T, r, σ_{av}^{BS}	Volatility
eBS_{NL2}	$(Se^{-d_y T})/X$, T, r, σ_{NL2}^{BS}	Volatility
eCS_{av}^{sig}	$(Se^{-d_y T})/X$, T, r, σ_{av}^{CS}	Volatility
eCS_{NL2}^{sig}	$(Se^{-d_y T})/X$, T, r, σ_{NL2}^{CS}	Volatility
eCS_{av}^{all}	$(Se^{-d_y T})/X$, T, r, σ_{av}^{CS}, μ_3, μ_4	Volatility, skewness, kurtosis
eCS_{NL2}^{all}	$(Se^{-d_y T})/X$, T, r, σ_{NL2}^{CS}, μ_3, μ_4	Volatility, skewness, kurtosis

Panel B: *Description of all EPOPMs*

Table 1: EPOPM structure characteristics

	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	3.585	6.320	12.300	23.960	41.581	75.102	121.855
Implied Volatility	0.242	0.204	0.184	0.190	0.212	0.254	0.327
Volume	444	840	1162	1792	568	257	153
# observations	1461	4009	6861	3980	5023	2718	1454
			Medium Term Options 60-180 Days				
Call	7.283	17.226	33.304	48.498	65.149	92.811	136.617
Implied Volatility	0.195	0.182	0.191	0.198	0.214	0.224	0.246
Volume	343	488	486	717	299	123	102
# observations	1424	1436	1190	625	729	509	390
			Long Term Options ≥ 180 Days				
Call	14.495	31.803	51.041	65.248	81.185	106.598	147.316
Implied Volatility	0.183	0.186	0.194	0.196	0.209	0.217	0.233
Volume	349	372	307	406	202	107	127
# observations	1452	1151	1011	576	567	331	271

Table 2: *Sample characteristics*

We cover the period January 2, 2002 to August 31, 2004. All figures refer to average values (# observations that refer to the number of call option datapoints that are included in certain moneyness and maturity class).

	BS_{av}	BS_{L1}	BS_{NL1}	BS_{NLL1}	BS_{L2}	BS_{NL2}	BS_{NLL2}	BS_{L3}	BS_{NL3}	BS_{NLL3}
RMSE	3.187	2.519	1.511	1.871	1.280	0.779	1.148	1.103	**0.638**	1.028

	CS_{av}	CS_{L1}	CS_{NL1}	CS_{NLL1}	CS_{L2}	CS_{NL2}	CS_{NLL2}	CS_{L3}	CS_{NL3}	CS_{NLL3}
RMSE	1.369	1.861	1.279	1.401	1.027	0.698	0.944	0.882	**0.582**	0.829

	SVJ_{av}
RMSE	**0.437**

Table 3A: In sample performance of parametric models – Jan 2002 to Aug 2004

	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
AE 5th	0.242	0.091	0.115	0.124	0.082	**0.078**	0.085	0.072	0.067	0.073
AE 95th	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
AE 5th	0.118	0.106	0.115	0.112	0.085	**0.085**	0.088	0.085	0.085	0.084
AE 95th	4.370	6.107	4.144	4.470	3.997	**3.422**	3.835	3.879	3.328	3.794

	SVJ_{av}
RMSE	**1.498**
MAE	**1.071**
MeAE	**0.796**
AE 5th	**0.065**
AE 95th	**2.996**

Table 3B: Out of sample performance of parametric models – March 3, 2003 to Aug 31, 2004

	eBS_{av}	eBS_{NL2}	eCS_{av}^{sig}	eCS_{NL2}^{sig}	eCS_{av}^{all}	eCS_{NL2}^{all}
RMSE	1.754	1.732	1.646	1.601	1.568	1.535
MAE	1.327	1.157	1.243	1.176	1.176	1.131
MeAE	1.046	0.834	0.957	0.886	0.908	0.856
P_5AE	0.097	0.069	0.084	0.078	0.079	0.072
P_{95}AE	3.473	3.190	3.345	3.220	3.169	3.075

Table 4: Out of sample performance of the EPOPMs – March 3, 2003 to Aug 31, 2004
Models´ optimization: 1-10 hidden neurons, 20 weight initializations

	BS_{NL2}	CS_{av}	CS_{NL2}	SVJ_{av}	eBS_{av}	eBS_{NL2}	eCS_{av}^{sig}	eCS_{NL2}^{sig}	eCS_{av}^{all}	eCS_{NL2}^{all}
BS_{av}	21.93	38.82	50.73	50.46	55.63	44.93	59.03	59.09	61.02	61.18
BS_{NL2}		-3.53	3.18	6.02	3.40	3.45	4.72	5.23	5.62	5.98
CS_{av}			19.57	22.50	25.35	15.74	31.38	31.56	35.07	35.29
CS_{NL2}				6.85	0.52	0.91	5.11	6.62	8.29	9.32
SVJ_{av}					-7.38	-4.92	-4.21	-2.81	-1.97	-1.01
eBS_{av}						0.66	6.91	8.73	11.83	12.97
eBS_{NL2}							2.47	3.65	4.62	5.45
eCS_{av}^{sig}								2.70	5.28	6.93
eCS_{NL2}^{sig}									1.90	3.62
eCS_{av}^{all}										2.09

Table 5: ***Two-sample matched pair t-tests for comparison of out of sample model performance***
Two sample matched pair t-test comparing the 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 (larger in absolute terms) than 1.96 (or 2.325) indicates that the model in the vertical (horizontal) heading has a larger MSE than the model in the horizontal (vertical) heading at 5% (or 1%) significance level.