

# Option Pricing With Modular Neural Networks

Nikola Gradojevic, Ramazan Gençay, and Dragan Kukulj, *Senior Member, IEEE*

**Abstract**—This paper investigates a nonparametric modular neural network (MNN) model to price the S&P-500 European call options. The modules are based on time to maturity and moneyness of the options. The option price function of interest is homogeneous of degree one with respect to the underlying index price and the strike price. When compared to an array of parametric and nonparametric models, the MNN method consistently exerts superior out-of-sample pricing performance. We conclude that modularity improves the generalization properties of standard feedforward neural network option pricing models (with and without the homogeneity hint).

**Index Terms**—Modular neural networks, nonparametric methods, option pricing.

## I. INTRODUCTION

SINCE the seminal work by Black and Scholes [8], option pricing has become the focus of scholarly attention. Most of the research efforts have attempted to relax some of the restrictive assumptions underlying the Black–Scholes model such as the normality of the log returns or constant volatility, as there is strong empirical evidence suggesting that these assumptions are not appropriate (e.g., [3]). One such example of the misspecifications of the Black–Scholes model is its substantial inaccuracy related to the pricing of the deep out-of-the-money options [27]. For these options, it was found that the Black–Scholes prices overestimate market prices while feedforward neural network (NN) models provide a superior pricing performance. The success of NN option pricing models was also documented in [25], [28], [34], [39], and [43].

A number of alternative approaches to account for the misspecifications of the Black–Scholes model have been proposed.<sup>1</sup> Models specified within the framework of nonconstant (or stochastic) volatility were suggested by Bakshi *et al.* [3], Bates [5], Chernov and Ghysels [13], Das and Sundaram [16], Duffie *et al.* [19], Eraker [23], Hull and White [33], Pan [42], and Scott [46]. It is worth mentioning that stochastic volatility and stochastic interest rate models improve upon the Black–Scholes

model in out-of-sample pricing and hedging exercises [3]. However, the pricing improvements are not robust and exhibit biases for the deepest out-of-the-money call options. Deterministic volatility models have also been considered [22], [45], but Dumas *et al.* [21] reported that the same performance could be achieved by applying an implied volatility smoothing procedure on the Black–Scholes model. Augmenting the underlying price process can produce many different variants of the Black–Scholes model. For example, one of the most popular variants involves fitting a Poisson jump process to the stock price data (e.g., [5] and [42]).

Relaxing the normality assumption of the Black–Scholes model led to parametric [38], [41] and semiparametric [2] option pricing models. Despite the theoretical appeal of parametric models stemming from their stringent parametric assumptions that allow for simpler functional forms, nonparametric models are more flexible in relaxing the distributional assumptions of the Black–Scholes model with a potential for improved out-of-sample pricing performance. Apart from the above mentioned NN models, some notable recent research contributions that follow this approach include a nonparametric American option pricing model by Broadie *et al.* [11] and Broadie *et al.* [12], NN model for the FTSE 100 Index options by Bennell and Sutcliffe [6] and Gottschling *et al.* [29], nonparametric locally polynomial estimator of Ait-Sahalia and Duarte [1], positive convolution estimation of Bondarenko [9], and nonparametric least squares method by Yatchew and Härdle [49].

This paper builds on the option pricing models from [34] and [25]. Instead of relying on the method of hints from [25], we estimate a nonparametric model called modular neural network (MNN) model. *A priori*, training a single NN model on heterogeneous data (options of different moneyness and time to maturity) can be problematic for several reasons. The NN could become trapped in local minima or may generalize poorly [44]. Moreover, the NN model could be susceptible to the “recency effect”: the NN parameters adapted unduly in favor of the most recent training data. Feldkamp and Puskorius [24] approach the recency effect through multistream learning, a concept related to MNNs. During multistream learning, the data set is split into multiple files (streams) and, in each training cycle, the NN parameters in streams are updated independently. This procedure improves the generalization properties of the recurrent NN and is an increasing function of the number of streams. Noteworthy, multistream learning is based on training a single NN whereas an MNN contains many NNs defined across subsets of the data, i.e., modules. Therefore, we find room for improvements upon simple NN option pricing models that fail to account for the recency effect. We use MNN to decompose the data into modules organized with regard to moneyness and maturity and estimate each module independently. In fact, when *a priori* knowledge about the model is unknown, advanced MNNs utilize learning

Manuscript received September 13, 2007; revised July 22, 2008 and October 31, 2008; accepted December 02, 2008. First published March 06, 2009; current version published April 03, 2009.

N. Gradojevic is with Faculty of Business Administration, Lakehead University, Thunder Bay, ON P7B 5E1 Canada (e-mail: ngradoje@lakeheadu.ca).

R. Gençay is with the Department of Economics, Simon Fraser University, Burnaby, BC V5A 1S6 Canada (e-mail: gencay@sfu.ca).

D. Kukulj is with Faculty of Engineering, University of Novi Sad, 21000 Novi Sad, Serbia (e-mail: dragan.kukulj@MicronasNIT.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TNN.2008.2011130

<sup>1</sup>See [26] for a recent review.

algorithms to find the optimal modular architecture [14]. We show that by dividing the data into modules and estimating each module locally it is possible to obtain a more accurate global option pricing function.

To assess the potential gains in pricing accuracy, we work within the framework of [25]. Consequently, for the S&P-500 index European call options data, we estimate the option-price-to-strike-price ratio ( $C_t/K$ ) as a function of two arguments: the stock-price-to-strike-price ratio ( $S_t/K$ ) and the time to maturity ( $\tau$ ). Our estimation results show that MNNs are superior in capturing both cross-sectional and time-series aspects of the fluctuations in the option prices. Compared to the Black–Scholes model, for all years, except 1987, the average out-of-sample accuracy gains are statistically significant and range from 68% to 91% (three modules defined by moneyness) and from 44% to 85% (three modules defined by time to maturity). For the period 1987–1994, the average improvements of the MNN model (modules defined by moneyness) relative to the feedforward NN model with the hint lie between 1.5% in 1989 and 68% in 1987. However, the pricing accuracy of the MNN model based on the time-to-maturity module selection criterion is in most cases inferior to the NN model with the hint. When both of the criteria are applied simultaneously (i.e., nine modules are constructed), the MNN model provides even smaller average out-of-sample pricing errors that are consistently below the ones for the rival models. Therefore, our main empirical result is that we are able to achieve generalization improvements of the option pricing function beyond those provided by the technique of hints. We hope that our results will foster future research on the applications of MNNs in other areas of finance and economics.

The remainder of this paper is organized as follows. Section II introduces the MNN model and explains its advantages over a single feedforward NN model. In Section III, we describe the data. The results of our out-of-sample pricing exercises are presented in Section IV. Section V concludes.

## II. MODULAR NEURAL NETWORK DESIGN

### A. Modular Neural Network Estimator

This paper relies on modularity as a basic model design principle to estimate the option pricing function nonparametrically. Among many other authors, Cofino *et al.* [14], Happel and Murre [31], and Ronco and Gawthrop [44] find that complex problems can be tackled by decomposing them into a number of subtasks. One can understand modularity as the notion of local computation on a specific task domain (or module). Each module is represented by a single feedforward NN model. Generalization is achieved through further interaction (cooperation or competition) with other modules. This concept, also called the “divide-and-conquer” method, can be very useful for function approximation in the presence of heterogeneous data.<sup>2</sup> For instance, an extremely volatile region of data may be followed by a relatively stable data sequence. It can be very

<sup>2</sup>In an innovative and related paper, Keber and Schuster [37] show that the divide-and-conquer-like methods can also be applied to derive the implied volatility from American put options.

difficult for a single NN to extract both processes together and generalize the whole function. Sometimes referred to as “spatial crosstalk,” this problem results in a slow convergence or learning failure [35].

The modular structure of an MNN reduces the complexity of a single NN and enhances its functional approximation capabilities. Due to their considerable flexibility and robustness to specification errors, MNNs can surpass the limitations of single NN models, or, more specifically, be able to control for the spatial crosstalk and recency effects.

To explain the concept of an MNN in the option pricing context, we will assume that a function of two variables  $c_t$  ( $c_t = \phi(x_{1t}, x_{2t}) + \varepsilon_t$ ) is driven by different functions defined over known domains of  $x_{1t}$  and  $x_{2t}$ , where  $t$  denotes the time index. Namely, function  $\phi(x_{1t}, x_{2t})$  can be written as a linear combination of functions  $c_{1t}, c_{2t}, \dots, c_{Mt}$  as follows [36]:

$$\begin{aligned} \phi(x_{1t}, x_{2t}) &= \omega_1 c_{1t}(x_{1t}, x_{2t}) + \dots + \omega_M c_{Mt}(x_{1t}, x_{2t}) \\ &= \sum_{k=1}^M \omega_k c_{kt}(x_{1t}, x_{2t}). \end{aligned} \quad (1)$$

This function can be approximated by an MNN (Fig. 1). A meaningful decomposition of the function  $\phi(x_{1t}, x_{2t})$  would be to approximate it locally by  $M$  modules  $c_{1t}, c_{2t}, \dots, c_{Mt}$  and then to sum the output of each module, while applying the appropriate weights  $\omega_1, \omega_2, \dots, \omega_M$ . In this work, we apply a partitioning approach that sets only one  $\omega_i = 1$ , while all other  $\omega_k = 0$  for  $k \neq i, k = 1, \dots, M$ . This can be viewed as a “crude partitioning” of the  $(x_{1t}, x_{2t})$  space, i.e., the weights are redistributed to only one module at the time.<sup>3</sup>

We specify modules and weights using the moneyness ( $S_t/K$ ) and time-to-maturity ( $\tau$ ) criteria. The modules are feedforward (backpropagation) NNs and are in general estimated as

$$c_{kt} = g_k \left( \beta_{k0} + \sum_{j=1}^{q_k} \beta_{kj} \psi_k \left( \alpha_{kj0} + \sum_{i=1}^s \alpha_{kij} x_{it} \right) \right), \quad k = 1, \dots, M. \quad (2)$$

In our case,  $s = 2$  ( $i = 1, 2$ ) and  $q_k$  is the number of hidden nodes for the  $k$ th module. The single hidden and the output layers of the modules are characterized by two flexible classes of nonlinearities:  $\psi_k$  and  $g_k$ , respectively. The backpropagation learning algorithm requires continuous differentiable nonlinearities. The types used in this paper are the sigmoid logistic or hyperbolic tangent functions in the hidden layer, and the linear function in the output layer.  $\alpha_{kij}$  and  $\beta_{kj}$  denote appropriate connection weights between the adjacent layers for the modules. Subscripts 0 for  $\alpha$  and  $\beta$  stand for NN biases.

We define our option pricing formula as in [25] and [34]

$$C_t = \phi(S_t, K, \tau) \quad (3)$$

where  $C_t$  is the call option price,  $S_t$  is the price of the underlying asset,  $K$  is the strike price, and  $\tau$  is the time to maturity.

<sup>3</sup>The combination weights  $\omega_k$  ( $k = 1, \dots, M$ ) can also be estimated using a “softmax” method [10], [40] or Gaussian mixture models [32].

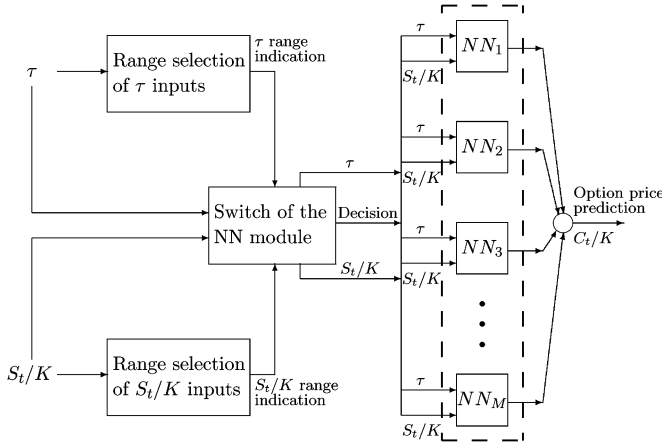


Fig. 1. Modular neural network architecture. The pricing function is decomposed into  $M$  separate nonlinearities ( $NN_1, \dots, NN_M$ ) called modules. The modules are trained independently on the data for the  $M$  option types and during prediction, based on the values of  $\tau$  and  $S_t/K$ , only one NN is active (i.e., when  $\omega_i = 1$ , all  $\omega_k = 0$ ;  $k \neq i$ ,  $k = 1, \dots, M$ ). More specifically, the out-of-sample predictions ( $C_t/K$ ) are generated from nine different NN models that span the whole testing set.

TABLE I  
MODULE SELECTION CRITERIA FOR THE MNN OPTION PRICING  
MODEL WITH NINE MODULES

	$(S_t/K) < 0.97$	$0.97 \leq (S_t/K) \leq 1.05$	$(S_t/K) > 1.05$
$\tau < 0.1$	$c_{1t}$ (module 1)	$c_{2t}$ (module 2)	$c_{3t}$ (module 3)
$0.1 \leq \tau \leq 0.2$	$c_{4t}$ (module 4)	$c_{5t}$ (module 5)	$c_{6t}$ (module 6)
$\tau > 0.2$	$c_{7t}$ (module 7)	$c_{8t}$ (module 8)	$c_{9t}$ (module 9)

Notes: The options are categorized based on maturity (short term, medium term, long term) and moneyness (out-of-the-money, near-the-money, in-the-money). When both criteria are applied at the same time, the out-of-sample predictions are estimated from an MNN with nine modules. Specifically,  $\omega_i$ 's for all nine modules are specified in the following fashion: when  $\omega_i = 1$  ( $i = 1, \dots, 9$ ),  $\omega_k = 0$  ( $k \neq i$ ,  $k = 1, \dots, 9$ ).

Assuming the homogeneity of degree one of the pricing function  $\phi$  with respect to  $S_t$  and  $K$ , one can write the option pricing function as follows:

$$\frac{C_t}{K} = \phi \left( \underbrace{\frac{S_t}{K}}_{x_1}, 1, \underbrace{\tau}_{x_2} \right) = \phi(x_1, x_2). \quad (4)$$

In general, options are often referred to as plain vanilla derivatives because their payoff (or price) is determined by the so-called underlying, which is in our case the S&P-500 stock market index. Call options are more profitable for the buyer when, *ceteris paribus*, the price of the underlying ( $S_t$ ), increases or the strike price ( $K$ ) decreases. Therefore, intuitively, these two variables must be integral parts of the option pricing formula. Further, when time to maturity ( $\tau$ ) increases, call options become more valuable. This is explained by the fact that it is more likely that the option will be in-the-money ( $S_t - K > 0$ ) and, thus, worthwhile exercising at maturity. The preceding explanatory variables can be extended with volatility

(standard deviation of the underlying), risk-free interest rate, and dividends paid on the underlying.<sup>4</sup>

The pricing function  $\phi$  is first approximated by an MNN with three modules determined by the time-to-maturity cutoff points as follows:  $\tau < 0.1$  (short term),  $0.1 \leq \tau \leq 0.2$  (medium term), and  $\tau > 0.2$  (long term).<sup>5</sup> Thus, the pricing function is decomposed into three separate nonlinearities. The modules are trained independently on the data for the three option types and during prediction; based on the value of  $x_2$ , only one is active (i.e., when  $\omega_i = 1$ , all  $\omega_k = 0$ ,  $k \neq i$ ,  $k = 1, 2, 3$ ).

This empirical exercise is followed by selecting modules according to the moneyness criterion as follows:  $(S_t/K) < 0.97$  (out-of-the-money),  $0.97 \leq (S_t/K) \leq 1.05$  (near-the-money), and  $(S_t/K) > 1.05$  (in-the-money). Now,  $x_1$ 's from the prediction part of the sample determine which of the three modules is used to estimate  $c_t$ . Finally, both criteria are applied at the same time and out-of-sample predictions are estimated from an MNN with nine modules. Specifically,  $\omega_i$ 's for all nine modules are specified in the following fashion: when  $\omega_i = 1$  ( $i = 1, \dots, 9$ ),  $\omega_k = 0$  ( $k \neq i$ ,  $k = 1, \dots, 9$ ). Table I describes these modules specified over different ranges of  $\tau$  and  $(S_t/K)$  that are used to estimate functions  $c_{kt}$  ( $k = 1, \dots, 9$ ).

Our primary goal is to improve upon the Garcia and Gencay's model [25] with the hint<sup>6</sup> and also show that MNNs offer significant advantages over simple feedforward neural network models. The choice of the number of modules, first, follows the logic of categorizing options based on maturity (short term, medium term, long term), then, moneyness (out-of-the-money, near-the-money, in-the-money), and, finally, using both criteria. The breakpoints of the categories are from [25, Table 4]. Therefore, we use a "rule of thumb" approach to choose both the number of modules and the cutoff points. In Section IV, we discuss possible methods for selecting the optimal number of modules and breakpoints.

## B. Alternative Non/Semiparametric Estimators

In contrast to parametric models such as the Black-Scholes that assume cumulative normal shapes, nonparametric models such as MNNs make minimal assumptions about the shape or the properties of the underlying price process. In addition, nonparametric pricing functions are convenient for estimating state price densities (SPDs) and the "Greeks." For instance, the MNN estimator can be directly differentiated numerically (or analytically) to extract the SPD or option delta.<sup>7</sup> This section will describe in fairly general terms the relationship between artificial neural networks (NNs) that are building blocks for MNNs and some alternative non- and semiparametric estimators.

<sup>4</sup>As suggested by a referee, we extend the model with the former two variables in Section IV-B.

<sup>5</sup>The units for  $\tau$  are the number of days to maturity divided by the number of days in a year (365 or 366).

<sup>6</sup>The "hint" involves utilizing additional prior information about the properties of an unknown (pricing) function that is used to guide the learning process. In the context of Garcia and Gencay [25], this means breaking up the pricing function into two parts, one controlled by  $S_t/K$  and the other one by a function of time to maturity. Each part contains a cumulative distribution function which is estimated nonparametrically through neural network models.

<sup>7</sup>We formally define the SPD and the delta in Section IV.

Similar to linear regression models, NNs map a set of explanatory variables ( $x_{it}$ , for example,  $i = 1, 2$ ) into a dependent variable ( $c_t$ ). The major difference between NNs and other nonparametric estimators is that NNs are based on a layered structure where each layer transforms  $x_{it}$ 's using special functions known as sigmoid logistic or hyperbolic tangent functions [see (2)]. The number of layers and the structure of each layer have to be determined before estimation, like the bandwidth in kernel regression methodologies. In the case of NNs, this is typically performed through a cross-validation method on the validation part of the data [25]. Then, the parameters of an NN are estimated by minimizing the loss function defined as the sum of squared differences between the observed  $c_t$  and the one predicted by an NN. The goal of this paper is to estimate the parameters for each NN of the MNN option pricing function.

A popular nonparametric alternative to NNs to estimate the pricing function is kernel regression [2]. In contrast to NNs, where observational errors are reduced by averaging the data based on a recursive error minimization procedure, for kernel regression, this is achieved by local averaging (smoothing). The kernel method estimates the price based on the weighted sum of the information from the in-sample data. More specifically, given some vector ( $x_{1t}; x_{2t}$ ), to estimate the price, more weight is assigned to the observations at locations that are closer to the vector. As our option pricing model has two regressors, the appropriate kernel pricing function would be constructed as a product of two univariate kernels. The usual tradeoff between smoothness and goodness-of-fit is achieved by the choice of the bandwidth of the kernel function. As previously suggested, in an NN, this tradeoff is controlled by the structure and the number of the layers, or, more generally, by the number of modules in an MNN. Ait-Sahalia and Lo [2] show that the kernel estimation errors for the SPD and option delta lie within one percent of the theoretical values. The out-of-sample pricing performance of their model is also impressive and superior over the NN model. It is noteworthy that the kernel pricing function from [2] that is used for forecasting is semiparametric. It is informed by dividend yield  $\tau$ ,  $K$ ,  $S_t$ , risk-free interest rate, and nonparametrically estimated volatility. In contrast, the variables in the MNN model are the ratio of the asset price to the strike price and the time to maturity.<sup>8</sup>

Other estimators that can be employed for option pricing include nearest-neighbor estimators, splines, average derivative estimators, local polynomial regression, and orthogonal series expansion. For example, the nearest-neighbor method is based on an assumption that geometric patterns in the past of the time series, similar by some measure (e.g., Euclidean distance) to the currently observed variables, can be used for forecasting (see, e.g., [48]). This set of "related" observations of independent variables is called the nearest neighbors. The smoothing constant analogous to the kernel bandwidth determines the number of nearest neighbors to be considered in the estimation. In general, all of the above methods are concerned with averaging or smoothing the data in a more or less sophisticated fashion while controlling for the tradeoff between bias and variance.

<sup>8</sup>We also compared the forecast performance of the semiparametric model to the MNN model. The results can be found in Section IV (Table III).

### III. THE DATA AND ASSESSMENT OF PREDICTION PERFORMANCE

The data are daily S&P-500 index European call option prices taken from the Chicago Board Options Exchange. For each available complete year, over the period from January 1987 to October 1994, options across different strike prices and maturities are considered. Being one of the deepest and most efficient option markets in the United States, the S&P-500 index option market is an adequate choice for our investigation. This also makes our study directly comparable to [25].

Options with zero volume are not used in the estimation. As in [25], for the NN estimation purposes, the data for each year are divided into three parts: first two quarters (training set), third quarter (validation set), and fourth quarter (out-of-sample, testing set).<sup>9</sup> This produced the following nonoverlapping sub-samples:

- **1987:** training sample: 3610, validation sample: 2010, testing sample: 2239;
- **1988:** training sample: 3434, validation sample: 1642, testing sample: 1479;
- **1989:** training sample: 3052, validation sample: 1565, testing sample: 1515;
- **1990:** training sample: 3605, validation sample: 2075, testing sample: 2166;
- **1991:** training sample: 4481, validation sample: 1922, testing sample: 2061;
- **1992:** training sample: 4374, validation sample: 1922, testing sample: 1848;
- **1993:** training sample: 4214, validation sample: 1973, testing sample: 2030.

The optimal NN architecture for each module is determined from the out-of-sample performance on the validation set with respect to the mean squared prediction error (MSPE). To keep the complexity of the NNs close to [25], the modules are single-hidden-layer NNs with either sigmoid logistic or hyperbolic tangent activation functions. Hence, the search for an optimal NN architecture involves specifying the number of hidden nodes (we choose from 1–15 hidden nodes) and their activation functions. The parameters are estimated using the standard Levenberg–Marquardt algorithm. Overfitting is prevented by early stopping, i.e., stopping the training process when the validation set error starts to increase. To control for possible sensitivity of the NNs to the initial parameter values, the training is performed from ten different random seeds and the average MSPE values are reported. The predictive performance on the testing part of the sample is finally assessed with the MSPE criterion.

The out-of-sample pricing performance of an MNN model is compared to the Black–Scholes model.<sup>10</sup> The Black–Scholes call prices ( $C_t$ ) are computed using the standard formula

$$C_t = S_t N(d) - K e^{-r\tau} N(d - \sigma\sqrt{\tau}),$$

$$\text{where } d = \frac{\ln(S_t/K) + (r + 0.5\sigma^2)\tau}{\sigma\sqrt{\tau}} \quad (5)$$

<sup>9</sup>Initially, volatility is not used as one of the MNN's inputs, i.e., MNN extracts the volatility information through the training procedure using past quarter's data.

<sup>10</sup>This will also shed more light on the magnitude of our model's forecasting improvements, relative to the model with the hint from [25].

where  $N$  is the cumulative normal distribution,  $S_t$  is the price of the underlying asset,  $K$  is the strike price,  $\tau$  is the time to maturity,  $r$  is the risk-free interest rate, and  $\sigma$  is the volatility of the underlying asset's continuously compounded returns estimated from the last 60 days preceding the first day of the last quarter. The risk-free rate is approximated by the monthly yield of the U.S. Treasury bills.

The statistical significance of the prediction performance is assessed with the Diebold–Mariano statistic [18] that is distributed as standard normal in large samples.<sup>11</sup> When the calculated values of the Diebold–Mariano statistic are large and positive, the MNN model is viewed as being able to significantly improve upon the Black–Scholes model.

#### IV. RESULTS

##### A. Basic Model

Table II compares the out-of-sample pricing performance of the NN model with the hint, the MNN model with three modules, and the Black–Scholes model in terms of the MSPE performance measure. The MNN model is estimated ten times from ten different sets of starting values and the average MSPEs are reported along with the average number of hidden-layer units in modules. The third column in Table II reveals that when the modules are selected based on the moneyness criterion, the MNN model outperforms the NN model with the hint for all years. Furthermore, the same architecture of the MNN model consistently outperforms the Black–Scholes model, except in 1987. The pricing improvements are statistically significant according to the Diebold–Mariano statistic. Garcia and Gencay [25] also find 1987 to be problematic, but show that when the third quarter (validation) data are included in the training set, the performance of the NN model with the hint becomes superior to the Black–Scholes model. We will show later that by increasing the complexity of the MNN model we can outperform the Black–Scholes model without using the validation data for training.

The fourth column in Table II reveals that when the modules are selected based on time to maturity, the MNN model does not perform well. In almost all of the years, the MSPEs are greater than the ones for the model with the hint, but still below the ones for the Black–Scholes model at the 5% significance level. Again, this is not the case with 1987. We conjecture that the complexity of the option pricing function is such that it requires more (or different) modules.

To give more flexibility to the MNN model, we employ both module selection criteria over nine modules. Table III shows that this model is more accurate than both the Black–Scholes and the NN with the hint models. This holds for all years and the improvements in the MSPE are statistically significant. The performance of the MNN model is impressive for 1987 as well. Recall that the volatility in the Black–Scholes model is calculated from the information contained in the third quarter. As opposed to the NN model with the hint, for 1987, the MNN model

<sup>11</sup>West [47] argued that this statistic is potentially unreliable. More precisely, forecast errors can be both serially and contemporaneously correlated. We compute the test statistic as the average of the forecast error differences for the ten estimations of the optimal NN architectures.

TABLE II  
PREDICTION PERFORMANCE OF THE MNN OPTION PRICING  
MODEL WITH THREE MODULES

Year	Statistic	ANN with hint	MNN model ( $S_t/K$ )	MNN model ( $\tau$ )	BS model
1987	$\overline{MSPE}$	16.7	5.44 [9]	6.8 [10]	4.38
	$\sigma$	9.51	2.30	4.02	
	DM		-4.23	-7.75	
1988	$\overline{MSPE}$	0.7114	0.6681 [5]	1.1592 [6]	2.07
	$\sigma$	0.0429	0.0134	0.4304	
	DM		8.36	3.30	
1989	$\overline{MSPE}$	0.4138	0.4076 [5]	0.5093 [6]	1.42
	$\sigma$	0.0068	0.0014	0.1152	
	DM		9.61	4.65	
1990	$\overline{MSPE}$	0.6761	0.5759 [9]	0.6639 [6]	2.62
	$\sigma$	0.0763	0.0597	0.1742	
	DM		17.23	14.06	
1991	$\overline{MSPE}$	0.3498	0.3401 [5]	0.3727 [6]	1.73
	$\sigma$	0.0148	0.0226	0.0244	
	DM		7.17	6.38	
1992	$\overline{MSPE}$	0.1511	0.1326 [8]	0.2044 [10]	1.36
	$\sigma$	0.0115	0.0123	0.0534	
	DM		6.60	3.97	
1993	$\overline{MSPE}$	0.1054	0.0665 [10]	0.1442 [11]	0.74
	$\sigma$	0.0222	0.0068	0.0637	
	DM		10.01	2.65	

Notes: The first row of this table contains the out-of-sample average mean squared prediction errors ( $\overline{MSPE}$ ) of the following option pricing models: Garcia and Gencay's feedforward neural network model with the hint (ANN with hint) [25], modular neural network (MNN) model with the moneyness ( $S_t/K$ ) module selection criterion, MNN model with the time-to-maturity ( $\tau$ ) module selection criterion and the Black–Scholes model (BS model). The average number of hidden-layer nodes in MNN modules is reported in the square brackets. The average MSPEs for the MNN models have been obtained as averages across ten different random training seeds.  $\sigma$  is the standard deviation of the ten MSPEs estimated from ten different random seeds. DM denotes the Diebold and Mariano [18] test statistic. This test is used to assess the statistical significance of the MNN model's forecast gains relative to the BS model. All  $\overline{MSPE}$  and  $\sigma$  figures have been multiplied by  $10^4$ .

does not require any additional training data to ensure superior pricing performance.<sup>12</sup> To illustrate the accuracy of the MNN model, we plot out-of-sample predictions of  $C_t/K$  and the actual data across the moneyness dimension for 1990 and 1993 [Fig. 2(a) and (b)]. As can be seen in Fig. 2(a) and (b), the estimates follow the actual prices very closely, especially in 1993, when the average MSPE is much smaller than the one in 1990.

We also compare the MNN model with other alternative approaches such as the kernel regression [2] and the parametric mixture model [41]. The third column of Table III indicates that, for various choices of bandwidth, the kernel model could not improve upon the MNN model in any of the years, except in

<sup>12</sup>Nevertheless, the pricing error in 1987 is unusually large (we thank the referee for pointing this out). By plotting the pricing error against maturity and moneyness, we observe that the inaccuracy in 1987 originates in (mis)pricing out-of-the-money calls, which is not the case in any of the "normal" years. Bates [4] finds that out-of-the-money put options on S&P-500 Index futures were unusually expensive relative to out-of-the-money calls before the October 1987 crash (it can be understood as an expectation of a downward movement). We do not find that call options in the first two quarters (training data) of 1987 were "cheap," but we find that the average call option price in the last quarter (testing data) of 1987 is the lowest of all years. This potentially "confused" the model and deteriorated its out-of-sample performance.

TABLE III  
PREDICTION PERFORMANCE OF THE MNN OPTION PRICING  
MODEL WITH NINE MODULES

	ANN with hint	MNN model	AL	BS model	DM	$\sigma$
$\overline{MSPE}$						
1987	16.7	4.1204 [4]	3.3665	4.38	3.04	1.0376
1988	0.7114	0.6107 [5]	2.3784	2.07	10.88	0.0372
1989	0.4138	0.4008 [5]	1.5854	1.42	9.99	0.0038
1990	0.6761	0.5579 [5]	1.3428	2.62	26.89	0.0334
1991	0.3498	0.3293 [4]	0.8351	1.73	8.30	0.0023
1992	0.1511	0.1285 [4]	0.2756	1.36	6.85	0.0073
1993	0.1054	0.0546 [3]	0.4263	0.74	11.42	0.0062

Notes: This table reports the out-of-sample average mean squared prediction errors (MSPE) of the following option pricing models: Garcia and Gencay's feedforward neural network model with the hint (ANN with hint) [25], modular neural network (MNN) model with nine modules, Ait-Sahalia and Lo's semiparametric estimator (AL) [2], and the Black–Scholes model (BS model). The average number of hidden-layer nodes in MNN modules is reported in the square brackets. The average MSPEs for the MNN models have been obtained as averages across ten different random training seeds.  $\sigma$  is the standard deviation of the ten MSPEs estimated from ten different random seeds. DM denotes the Diebold and Mariano test statistic [18]. This test is used to assess the statistical significance of the MNN model's forecast gains relative to the BS model. All MSPE and  $\sigma$  figures have been multiplied by  $10^4$ .

1987.<sup>13</sup> To implement the mixture method, we find the European option price as the discounted value of the probability distribution above the strike price. Following [41], we use the mixtures of two and three lognormal distributions over option series consisting of call options with different strike prices but the same maturity. We find that the out-of-sample forecast performance of the mixture model over the years is either inferior to the kernel model or similar to the Black–Scholes model.<sup>14</sup> Ait-Sahalia and Duarte [1] show that by constraining the pricing function, the pricing error can be reduced by about 25%–50% for lower bandwidths (and much smaller sample sizes). However, a 50% reduction in the pricing error of the kernel pricing function would not be sufficient to outperform the MNN model. Furthermore, Daghli [15] demonstrates that the shape-constrained kernel produces very modest pricing accuracy gains for large data sets such as ours. Consequently, we do not include this enhancement of the nonparametric kernel into our set of rival models.

Based on these results, we conclude that the modularity feature of the MNN model plays a vital role in outperforming the rival models. This also accords with some other studies that focused on pricing of certain types of options (i.e., deep out-of-the-money options) and found that such partitioning of the data increases the pricing accuracy of the NN models [6], [27]. However, as shown for the time-to-maturity-constructed modules, one has to be cautious when using available domain knowledge

<sup>13</sup>We followed a semiparametric kernel approach from [2] and, first, estimated implied volatilities based on futures price, the exercise price, and time to maturity using the Nadaraya–Watson kernel estimator. Then, call option prices were computed from the estimated volatility function via the Black–Scholes formula. The differences between our estimation and the original approach are that we do not use the dividend yield and the rolling estimation procedure. Nonetheless, the kernel model depends on more variables than the MNN model, which gives it significant informational advantage.

<sup>14</sup>For brevity, these results have not been reported in Table III, but can be available from the authors upon request.

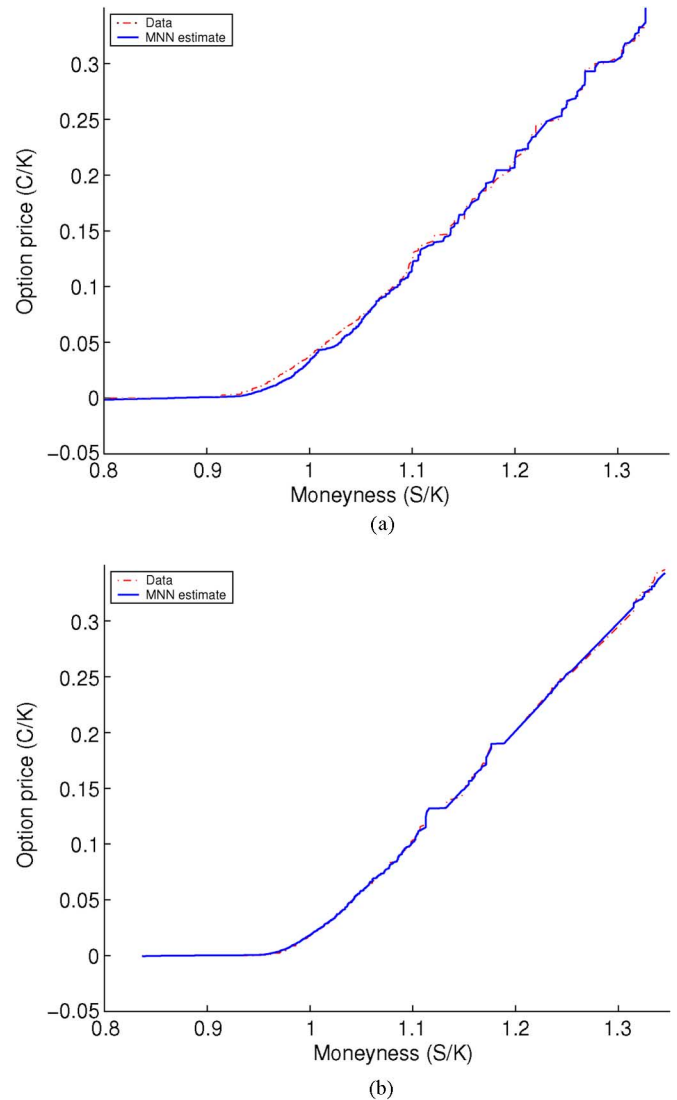


Fig. 2. Data and option prices estimated by the nine-module MNN model. Notes: Out-of-sample predictions of  $C_t/K$  (solid line) and the actual data (dashed line) are plotted against  $S_t/K$  in (a) 1990 and (b) 1993. First, the MNN model is trained using the data from the first two quarters of each year and, then, 2239 (for 1990) and 2030 (for 1993) out-of-sample estimates of  $C_t/K$  are generated. The average MSPEs for 1990 and 1993 are  $0.5579e-04$  and  $0.0546e-04$ , respectively.

to identify modules. Finally, we find the performance improvements closely linked to the number of modules in an MNN.

Increasing the number of modules, however, adds to the complexity of the estimated pricing function. As previously explained, our choice for the number of modules and the break-points for the modules is arbitrary, but sensible. Nevertheless, at this global level, we do not control for the usual tradeoff between smoothness and goodness-of-fit in calibration of nonparametric methods, but only optimize the performance of individual modules. Clearly, moving from three to nine modules improves the accuracy of the MNN model. By observing the average pricing errors of the individual modules of the three- and nine-module MNNs, we first attempt to pin down the cause for these improvements. Tables IV and V contain average pricing errors for the modules for 1993 that are representative for all other years.

TABLE IV  
AVERAGE MSPEs FOR INDIVIDUAL MODULES (1993, THREE MODULES)

	$(S_t/K) < 0.97$ or $\tau < 0.1$	$0.97 \leq (S_t/K) \leq 1.05$ or $0.1 \leq \tau \leq 0.2$	$(S_t/K) > 1.05$ or $\tau > 0.2$
MNN model ( $S_t/K$ )	0.0407	0.0620	0.1884
MNN model ( $\tau$ )	0.2490	0.0731	0.1672

Notes: This table reports the out-of-sample average mean squared prediction errors (MSPE) for each module of a three-module MNN model. The figures for both versions of the model are reported: modules selected by the moneyness criterion (first row) and modules selected by the time-to-maturity criterion (second row). The average MSPEs for the models have been obtained as averages across ten different random training seeds. All  $\overline{\text{MSPE}}$  figures have been multiplied by  $10^4$ .

TABLE V  
AVERAGE MSPEs FOR INDIVIDUAL MODULES (1993, NINE MODULES)

	$(S_t/K) < 0.97$	$0.97 \leq (S_t/K) \leq 1.05$	$(S_t/K) > 1.05$
$\tau < 0.1$	0.0018	0.0407	0.12
$0.1 \leq \tau \leq 0.2$	0.0179	0.0457	0.2094
$\tau > 0.2$	0.0313	0.0595	0.2161

Notes: This table reports the out-of-sample average mean squared prediction errors (MSPE) for each module of a nine-module MNN model. The options are categorized based on two criteria that are applied simultaneously: maturity (short term, medium term, long term) and moneyness (out-of-the-money, near-the-money, in-the-money). The average MSPEs for the models have been obtained as averages across ten different random training seeds. All  $\overline{\text{MSPE}}$  figures have been multiplied by  $10^4$ .

Table IV suggests that when the modules are selected by time to maturity, the pricing performance is poor for two modules (short-term and long-term options). The alternative three-module MNN has problems pricing in-the-money options (i.e., only one module performs poorly) and this seems to be the major reason for its superior performance over the first MNN variant. The nine-module MNN confirms that in-the-money options are relatively difficult to price for all maturities (Table V). Also, it appears that the information that comes from the moneyness criteria compensates for the time horizon effects. In other words, the modules perform roughly equally well for different time-to-maturity ranges.

As recommended by a referee,<sup>15</sup> next, we investigate in more detail the biases of the approximations for all the competing models. We plot the out-of-sample  $(c_t - \hat{c}_t)^2$  ( $Y$ -axis) against either moneyness or maturity ( $X$ -axis) for 1993 ( $t = 1, \dots, 2030$ ). In Fig. 3(a), we do not observe any systematic biases produced by the MNN model. In contrast, the NN model with the hint [Fig. 3(b)] exhibits certain biases with respect to both maturity and moneyness. This model substantially misprices some in-the-money and long-term options. The magnitude of such mispricing pattern substantially increases for the Ait-Sahalia and Lo's model [2] [Fig. 3(c)] as well as for the Black-Scholes [Fig. 3(d)]. The models show similar out-of-sample pricing biases in all other years and they are consistent with the findings in [3].

We further increase the number of modules from nine to 16, i.e., add one more range along each dimension. This is carried

out by dividing the middle moneyness range ( $0.97 - 1.05$ ) into two ranges around 1 ( $0.97 - 1, 1 - 1.05$ ) and the middle maturity range ( $0.1 - 0.2$ ) into  $[0.1 - 0.15]$  and  $(0.15 - 0.2]$ . The average MSPE for 1993 as a result of this exercise slightly increases from  $0.0546e-04$  to  $0.0656e-04$ . For all other years, the average error for the 16-module MNN is about 10%–25% larger. This indicates that the optimal number of modules is greater or equal to nine and less than 16. Because searching for an optimal MNN architecture is beyond the scope of this paper, for future research purposes, we recommend that the optimal number of modules and the cutoff points could be determined either by observing the performance on the validation data (third quarter of each year) or by utilizing some techniques from the statistical learning theory, as suggested in the last section. More precisely, one could select the optimal number of modules ( $M$ ) by using a cross-validation approach on the validation part of the data. Yet, this procedure does not identify the module boundaries and we will offer a possible iteration method for finding them. Starting from some initial boundaries for a given  $M$ , in each iteration, the “goodness” of current partition can be determined using the Davies–Bouldin (DB) index [17]

$$\text{DB} = \frac{1}{M} \sum_{i=1}^M R_i \quad R_i = \max(R_{ij}),$$

$$j = 1, \dots, M, \quad j \neq i \quad (6)$$

where  $R_{ij} = (S_i + S_j)/D_{ij}$  is a similarity measure between modules  $i$  and  $j$ ,  $S_i$  is a dispersion measure of the  $i$ th module calculated as the average Euclidean distance of the data points in module  $i$  to its center, and  $D_{ij}$  is a module dissimilarity measure (distance between the centers of the module pairs). Finally, the partition that produces the minimum DB is considered to be optimal.

Our data set is such that on any given day there are about 25–50 options of different strike price and maturity. Hence, the sample size required for training the MNN model is not large, unlike the sample size for the kernel methodology. Ait-Sahalia and Duarte [1] note that their semiparametric kernel model requires a relatively large data set to be empirically effective. This feature of our model makes it very suitable for practical applications.

One of the relevant practical uses of nonparametric option pricing is the estimation of the SPDs, which allows practitioners to price OTC derivatives consistently with the prices of exchange traded options. To obtain the SPD and the options' delta, we find analytical derivatives of (3) in a standard way

$$\hat{\Delta}_t = \frac{\partial \hat{C}(S_t, K, \tau)}{\partial S_t} \quad (7)$$

$$\hat{f}_t^* = e^{r_t \tau} \left[ \frac{\partial^2 \hat{C}(S_t, K, \tau)}{\partial K^2} \right]_{|K=S_T} \quad (8)$$

Even though our pricing function is very accurate out-of-sample, an unconstrained estimator such as the MNN could produce large errors for the derivatives. Initially, we observed a wiggly and sometimes negative SPD surface. This violates general properties of a density function and, in turn, violates the

<sup>15</sup>We thank the referee for this and other useful suggestions.

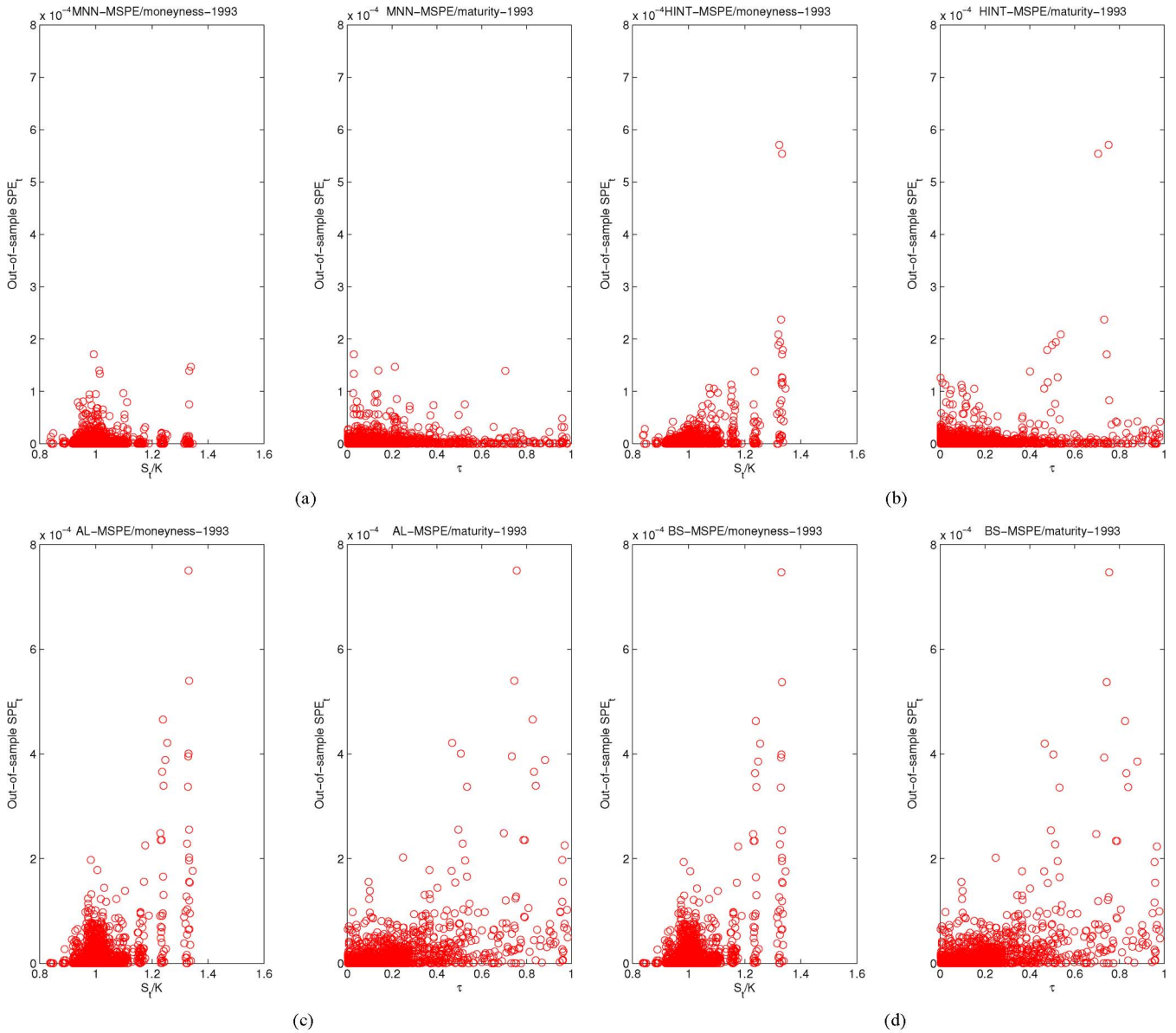


Fig. 3. Error plots of the competing option pricing models. Notes: Out-of-sample squared errors ( $SPE_t = (c_t - \hat{c}_t)^2, t = 1, \dots, 2030$ ) for 1993 are plotted against moneyness ( $S_t/K$ ) and maturity's feedforward neural network model with the hint (a), Ait-Sahalia and Lo's semiparametric estimator [2] (c), and the Black-Scholes model (d).

no-arbitrage principle [1]. In addition, the impact of the violations could be more acute and, for instance, lead to trading on false arbitrage signals. This is also explained in [25] and [26] who observed similar deficiencies in terms of derivatives of a nonparametric pricing function.

To alleviate for this problem, we followed [20] and imposed the *softplus* activation function in the hidden layer of each NN

$$\begin{aligned}
 \psi(Z) &= \log(1 + e^Z) \\
 \psi'(Z) &= \frac{e^Z}{1 + e^Z} \\
 \psi''(Z) &= \frac{e^Z}{(1 + e^Z)^2}.
 \end{aligned} \tag{9}$$

This function and its first two derivatives are always positive. Moreover, its first derivative is the sigmoid logistic function that

is constrained between 0 and 1, like a probability distribution function. The second derivative has the following properties:  $\lim_{Z \rightarrow -\infty} \psi''(Z) = 0$  and  $\lim_{Z \rightarrow +\infty} \psi''(Z) = 0$ . Therefore, it behaves like a probability density function. Dugas *et al.* [20] show that the *softplus* function satisfies the universal approximator property, and thus, can replace the sigmoid logistic function in a backpropagation NN.

In Fig. 4(a) and (b), we plot the average values for deltas and SPDs for 1993 estimated by a constrained MNN model.<sup>16</sup> The averages are received for the sample time to maturity that ranges from 1 to 360 days. The other variables that enter the estimation of deltas and SPDs are set to their average values for 1993. Despite some limitations in terms of smoothness, the nonparametric SPD surface still exhibits excess skewness and

<sup>16</sup>This did not deteriorate the MSPE of the model that remained roughly  $0.05e-04$ .

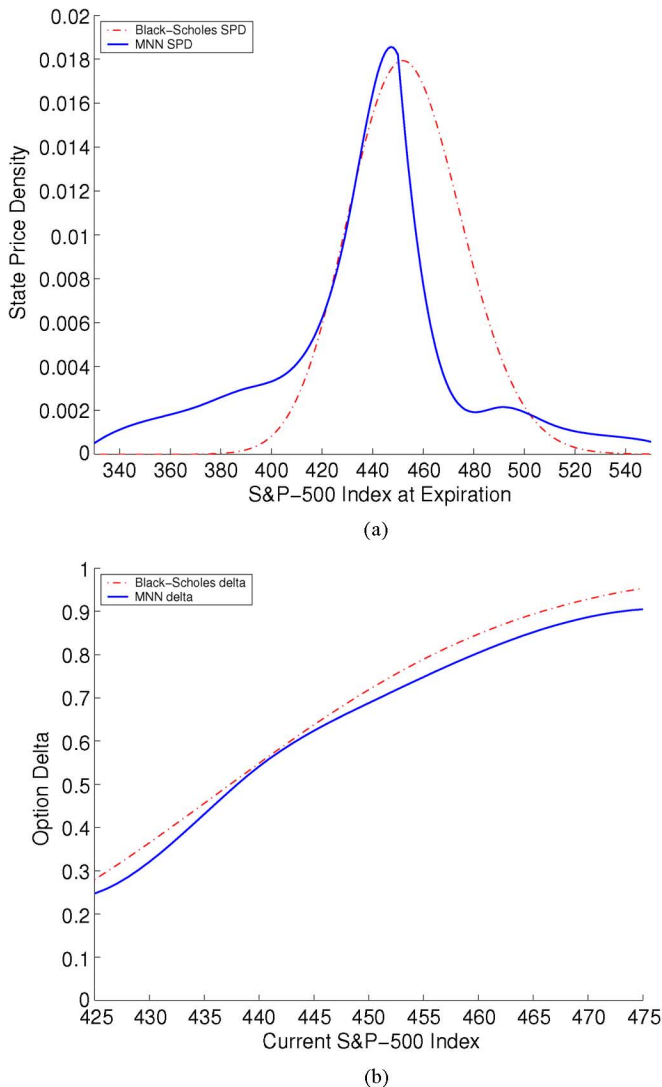


Fig. 4. SPDs and deltas of the constrained nine-module MNN estimator for 1993. (a) SPD estimates from the constrained nine-module MNN model (solid line) and the Black-Scholes SPDs (dashed line). To obtain the SPDs, the second derivative of the MNN estimator is evaluated analytically at  $K = S_T$ . (b) Estimates of an option delta from the constrained nine-module MNN model (solid line) along with the ones from the Black-Scholes model (dashed line). The deltas are estimated as the analytical first derivatives of the MNN pricing function with respect to  $S_t$ . All other relevant variables are fixed at their sample means for 1993. The estimates of SPDs and deltas are averaged across the sample time-to-maturity values that range from 1 to 360 days.

kurtosis and which is also found by Aït-Sahalia and Lo [2]. More importantly, it is bell-shaped and always positive. With regard to the estimated delta, minor discrepancies from the theoretical values are observed in Fig. 4(b), but the shape is reasonable and the values lie between 0 and 1. Note that the wiggleness can be observed mostly near the cutoff points between the modules and could be a direct consequence of the modular complexity of the MNN. For a single NN model, Hutchinson *et al.* [34] also find that deltas estimated by the NN model form do not significantly deviate from their theoretical counterparts.

One way to deal with unconstrained nonparametric estimators is to impose monotonicity and convexity [49] by employing well-behaved functional forms for the NNs in the modules. In

general, the term “well-behaved” refers to a nonnegative function that integrates to unity and we showed that imposing such shape constraints is a potentially promising future research avenue. To specify a valid density function, Gottschling *et al.* [29] propose an alternative method based on an NN approximation and the logarithm of the inverse Box-Cox transformation. The simplest algorithm for imposing constraints on nonparametric techniques is used in [1]: before any estimation is performed, the data are repeatedly transformed until they are monotonic.

### B. Extended Model

To further understand the pricing properties of the MNN model, we extend it with the risk-free interest rate ( $r$ ), historical volatility ( $\sigma_H$ ), and the implied volatility ( $\sigma_I$ ) as follows:<sup>17</sup>

$$\frac{C_t}{K} = \phi \left( \frac{S_t}{K}, \tau, r, \sigma_I \right). \quad (10)$$

An alternative error measure is introduced to gauge the out-of-sample pricing performance—mean absolute percentage error (MAPE):  $\text{MAPE} = 1/n \sum_{t=1}^T |c_t - \hat{c}_t/c_t|$ . Both the basic (two explanatory variables) and the extended (four explanatory variables) MNN models are statistically compared (Diebold-Mariano) to the NN model with the hint. As it can be argued that historical volatility used in the Black-Scholes model puts it at an unfair disadvantage, we also estimate an improvement of the Black-Scholes model by Hanke [30]. This model ( $BS_H$  model) utilizes the Black-Scholes optimal *ex-post* interest rate and volatility to price options. More precisely, on each out-of-sample day, first, the sum of the squared errors between the Black-Scholes price and the actual price is minimized (for all available options) to determine the optimal  $r_t$  and  $\sigma_t$ . Then,  $r_t^{\text{BSopt}}$  and  $\sigma_t^{\text{BSopt}}$  are supplied to the Black-Scholes formula.<sup>18</sup> During the optimization process, we have noticed that the model’s pricing performance has been very sensitive to the choice of the initial parameter values ( $r_{t0}$  and  $\sigma_{t0}$ ). This is particularly pronounced for  $\sigma_{t0}$  where larger starting values tend to deteriorate both the MSPE and the MAPE. To deal with this problem, we initiate the minimization process from ten different random values  $\sigma_{t0}$  within the  $(0, 1]$  interval and report the average values for the MSPE and the MAPE. Because the average pricing errors are also reported for the MNN model, we do not consider this to give advantage to any of the models. However, it is worthwhile to note that the Black-Scholes model is still at an informational advantage by the very construction of the  $BS_H$  model.

Table VI reports the pricing results for all years. We will first analyze the MAPE figures. Both MNN model versions consistently outperform the NN model with the hint whereas the extended MNN model is the most accurate overall. With

<sup>17</sup>We thank the Associate Editor and the referees for the extensions. Implied volatility is the Black-Scholes implied volatility for maturity that is the same as the option’s maturity. The risk-free rate is given by three-month U.S. t-bills rate, and thus, it does not match the maturity of the options. Historical volatility that will be used in the subsequent exercise is the volatility of the underlying asset’s continuously compounded returns estimated from the previous 60 days.

<sup>18</sup>Note that adding *ex-post* information ( $r_t^{\text{BSopt}}$  and  $\sigma_t^{\text{BSopt}}$ ) to the model would favor the extended models relative to the basic two-input MNN and NN with the hint models.

TABLE VI  
PREDICTION PERFORMANCE OF THE COMPETING MODELS

Year	ANN with hint		MNN-basic			MNN-extended			$BS_H$ model	
	$\overline{MAPE}$	$\overline{MSPE}$	$\overline{MAPE}$	$\overline{MSPE}$	DM	$\overline{MAPE}$	$\overline{MSPE}$	DM	$\overline{MAPE}$	$\overline{MSPE}$
1987	36.20	16.7	0.5392	4.12	16.95	0.2767	3.18	15.90	1.08	3.05
1988	3.94	0.7114	0.2338	0.6107	3.18	0.0876	0.5386	2.25	0.7793	1.26
1989	1.02	0.4138	0.2194	0.4008	0.1004	0.1085	0.1912	3.15	0.7985	1.06
1990	1.84	0.6761	0.2904	0.5579	2.00	0.0408	0.2781	7.80	1.07	2.52
1991	1.83	0.3498	0.0932	0.3293	0.7351	0.0254	0.0961	9.32	0.7176	0.7001
1992	1.34	0.1511	0.0810	0.1285	1.73	0.0631	0.0748	6.79	0.6358	0.7268
1993	2.06	0.1054	0.0437	0.0546	3.73	0.0301	0.0215	9.43	0.7930	0.5415

Notes: This table reports the average mean squared prediction errors ( $\overline{MSPE}$ ) and the average mean absolute percentage error ( $\overline{MAPE}$ ) of the following option pricing models: Garcia and Gencay's feedforward neural network model with the hint (ANN with hint) [25], modular neural network model with nine modules and two explanatory variables (MNN-basic), and modular neural network model with nine modules and four explanatory variables (MNN-extended). The last two columns are the MAPE and the MSPE of the improvement of the Black–Scholes model by Hanke [30] ( $BS_H$  model). The average MAPEs and MSPEs for the first three models have been obtained as averages across ten different random training seeds. For the last model, the average MAPEs and MSPEs involve ten different random starting values for estimating  $r_t^{BSopt}$  and  $\sigma_t^{BSopt}$ . DM denotes the Diebold and Mariano [18] test statistic. This test is used to assess the statistical significance of the MNN model's forecast gains relative to the ANN with hint model. All MSPE figures have been multiplied by  $10^4$ .

TABLE VII  
PREDICTION PERFORMANCE OF THE ALTERNATIVE MODEL SPECIFICATIONS

Year	ANN with hint-4		MNN-IR		MNN-H		$BS_H$ – opt model	
	$\overline{MAPE}$	$\overline{MSPE}$	$\overline{MAPE}$	$\overline{MSPE}$	$\overline{MAPE}$	$\overline{MSPE}$	$\overline{MAPE}$	$\overline{MSPE}$
1987	11.91	6.6708	0.2859	13.07	0.4288	2.41	0.8799	6.40
1988	3.88	0.3881	0.0803	0.1873	0.0345	0.1059	1.32	0.0887
1989	1.01	0.2415	0.3917	4.96	0.0464	0.1938	1.44	0.2092
1990	2.43	0.3390	0.2314	0.4707	0.0469	0.1471	1.72	0.1448
1991	3.20	0.2392	0.1467	0.3187	0.0521	0.0942	1.15	0.1108
1992	0.97	0.0714	0.0763	0.1599	0.0300	0.0493	0.9362	0.0451
1993	1.12	0.0690	0.0532	0.0895	0.0597	0.0325	1.87	0.0459

Notes: This table reports the average mean squared prediction errors ( $\overline{MSPE}$ ) and the average mean absolute percentage error ( $\overline{MAPE}$ ) of the following option pricing models: Garcia and Gencay's feedforward neural network model with the hint (ANN with hint-4) [25] and four inputs ( $S_t/K, \tau, r, \sigma_I$ ), modular neural network model with nine modules (MNN-IR) and four inputs ( $S_t/K, \tau, r_t^{BHopt}, \sigma_H$ ), and modular neural network model with nine modules (MNN-H) and four inputs ( $S_t/K, \tau, r_t^{BHopt}, \sigma_t^{BHopt}$ ). The last two columns are the MAPE and the MSPE of the improvement of the Black–Scholes model by Hanke [30] ( $BS_H$  – opt model) that utilize the optimal  $r_{t0}$  and  $\sigma_{t0}$ . The average MAPEs and MSPEs for the first three models have been obtained as averages across ten different random training seeds. All MSPE figures have been multiplied by  $10^4$ .

regard to the MSPE measure, the basic MNN model produces statistically significant forecast improvements over the NN model with the hint, except in 1989 and 1991. With the help of the additional two explanatory variables, the extended MNN model consistently outperforms the NN model with the hint in all years. Reductions in the MSPE can be observed in each year and they range between 12% and 70%, relative to the basic MNN model. The improvement of the Black–Scholes model reduces the MSPE of the original model, but, except in 1987, is not sufficient to improve upon any of the NN models. Noteworthy, with respect to the MAPE, the Black–Scholes improvement outperforms the NN model with the hint.

Some alternative model specifications warrant attention. First, we use the same inputs ( $r_t^{BSopt}$  and  $\sigma_t^{BSopt}$ ) and the same optimizer (the Levenberg–Marquardt algorithm<sup>19</sup>) for both the MNN and the  $BS_H$  model. However, based on the referee's recommendation, we search for  $r_{t0}$  and  $\sigma_{t0}$  that yield the lowest out-of-sample MSPE in the  $BS_H$  model and then feed the resulting optimal inputs into the models. Second, we compare the NN with the hint model with two extra inputs ( $r$  and  $\sigma_I$ ) to the extended MNN model (with the same two inputs). Finally, to understand the role of the implied interest

rate, we estimate the following MNN model:

$$\frac{C_t}{K} = \phi \left( \frac{S_t}{K}, \tau, r^{BSopt}, \sigma_H \right). \quad (11)$$

Table VII contains pricing errors of the additional model specifications across all seven years. The first two columns show that the two extra inputs improve the performance of the NN with the hint model, but it still performs worse than the extended MNN model. Further, the MNN-IR model (11) dominates the basic MNN model in some years, but, overall, implied volatility appears to be more informative than historical volatility. Including the ex-post information to the basic MNN model (i.e., the MNN-H model) results in sizeable improvements over the extended MNN model. The last two columns demonstrate that with respect to the MSPE the best performing  $BS_H$  model can achieve performance similar to the "average" MNN-H model.<sup>20</sup> The  $BS_H$  model is consistently inferior in terms of the MAPE.

<sup>20</sup>The best performing MNN-H model is superior to the best performing  $BS_H$  model in all years. For example, in 1988, the lowest MSPE of the  $BS_H$  model is  $0.0887e-04$  while the lowest MSPE of the MNN-H model is  $0.0685e-04$ .

<sup>19</sup>The results are similar for the Gauss–Newton algorithm.

## V. CONCLUSION

The goal of this paper is to investigate whether a nonlinear MNN model can be a successful option pricing tool. To assess the performance of the MNN model, we study S&P-500 European call options over the 1987–1993 period. Rather than relying on the method of hints, we estimate pricing functions using the modularity feature of the MNNs. This approach yields advantages not only over the Black–Scholes model, but also over the NN model with the hint. When the MNN model utilized three modules selected with regard to the moneyness criterion, the performance of the MNN model is consistently superior to the model with the hint in all years. However, an alternative variant of the MNN model based on the time-to-maturity criterion performs poorly. Moreover, both three-module MNN models are inferior to the Black–Scholes model in 1987. Extending the MNN model to nine modules reduced the pricing error and significantly improved upon the Black–Scholes model and the model with the hint for the whole sample. We contribute the success of the MNN model to its modularity feature that allows MNNs to “specialize” in pricing certain types of options by dividing the search space.

We conclude that the MNN option pricing model, or more generally, the notion of modularity is a promising future research direction in option pricing. Moreover, selection of modules and data partitioning based on more advanced methodologies from the statistical learning theory such as fuzzy clustering [7], [36] or genetic algorithms [14] can complement the current model. This can be of assistance, for instance, in improving the time-to-maturity criterion in the MNN model and also in gaining better insight into its potential pitfalls. More importantly, the success of our approach suggests that MNNs may be used to capture and explain highly volatile and elusive data-generating processes.

## REFERENCES

- [1] Y. Ait-Sahalia and J. Duarte, “Nonparametric option pricing under shape restrictions,” *J. Econometrics*, vol. 116, pp. 9–47, 2003.
- [2] Y. Ait-Sahalia and A. Lo, “Nonparametric estimation of state-price densities implicit in financial asset prices,” *J. Finance*, vol. 53, pp. 499–547, 1998.
- [3] G. Bakshi, C. Cao, and Z. Chen, “Empirical performance of alternative option pricing models,” *J. Finance*, vol. 52, no. 5, pp. 2003–2049, 1997.
- [4] D. S. Bates, “The crash of ’87: Was it expected? The evidence from options markets,” *J. Finance*, vol. 46, no. 3, pp. 1009–1044, 1991.
- [5] D. S. Bates, “Post-’87 crash fears in the S&P 500 futures option market,” *J. Econometrics*, vol. 94, no. 1/2, pp. 181–238, 2000.
- [6] J. Bennell and C. Sutcliffe, “Black–Scholes versus artificial neural networks in pricing FTSE 100 options,” *Int. J. Intell. Syst. Accounting Finance Manag.*, vol. 12, pp. 243–260, 2004.
- [7] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [8] F. Black and M. Scholes, “The pricing of options and corporate liabilities,” *J. Political Economy*, vol. 81, pp. 637–659, 1973.
- [9] O. Bondarenko, “Estimation of risk-neutral densities using positive convolution approximation,” *J. Econometrics*, vol. 116, pp. 85–112, 2003.
- [10] J. S. Bridle, *Probabilistic Interpretation of Feedforward Classification Network Outputs, With Relationships to Statistical Pattern Recognition*. New York: Springer-Verlag, 1989, pp. 227–236.
- [11] M. Broadie, J. Detemple, E. Ghysels, and O. Torr s, “American options with stochastic dividends and volatility: A nonparametric investigation,” *J. Econometrics*, vol. 94, pp. 53–92, 2000.
- [12] M. Broadie, J. Detemple, E. Ghysels, and O. Torr s, “Nonparametric estimation of American options’ exercise boundaries and call prices,” *J. Econom. Dyn. Control*, vol. 24, pp. 1829–1857, 2000.
- [13] M. Chernov and E. Ghysels, “A study towards a unified approach to the joint estimation of objective and risk neutral measures for the purpose of options valuation,” *J. Financial Econom.*, vol. 56, pp. 407–458, 2000.
- [14] A. S. Cofino, J. Guti rrez, and M. Ivanisovich, “Evolving modular networks with genetic algorithms: Application to nonlinear time series,” *Expert Syst.*, vol. 21, no. 4, pp. 208–216, 2004.
- [15] T. Daglish, “A pricing and hedging comparison of parametric and non-parametric approaches for American index options,” *J. Financial Econometrics*, vol. 1, no. 3, pp. 327–364, 2003.
- [16] S. Das and R. Sundaram, “Of smiles and smirks, a term structure perspective,” *J. Financial Quantitative Anal.*, vol. 34, pp. 211–239, 1999.
- [17] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 2, pp. 224–227, Apr. 1979.
- [18] F. X. Diebold and R. S. Mariano, “Comparing predictive accuracy,” *J. Business Econom. Statist.*, vol. 13, pp. 253–263, 1995.
- [19] D. Duffie, J. Pan, and K. Singleton, “Transform analysis and asset pricing for affine jump-diffusions,” *Econometrica*, vol. 68, pp. 1343–1376, 2000.
- [20] C. Dugas, Y. Bengio, F. Belisle, C. Nadeau, and R. Garcia, “Incorporating second-order functional knowledge for better option pricing,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001, vol. 13, pp. 472–478.
- [21] B. Dumas, J. Fleming, and R. E. Whaley, “Implied volatility functions: Empirical tests,” *J. Finance*, vol. 53, pp. 2059–2106, 1998.
- [22] B. Dupire, “Pricing with a smile,” *Risk*, vol. 7, no. 1, pp. 18–20, 1994.
- [23] B. Eraker, “Do stock prices and volatility jump? Reconciling evidence from spot and option prices,” *J. Finance*, vol. 59, pp. 1367–1403, 2004.
- [24] L. Feldkamp and G. Puskorius, “A signal processing framework based on dynamic neural networks with application to problems in adaptation, filtering, and classification,” *Proc. IEEE*, vol. 86, no. 11, pp. 2259–2277, Nov. 1998.
- [25] R. Garcia and R. Gen ay, “Pricing and hedging derivative securities with neural networks and a homogeneity hint,” *J. Econometrics*, vol. 94, no. 1/2, pp. 93–115, 2000.
- [26] R. Garcia, E. Ghysels, and E. Renault, *The Econometrics of Option Pricing*. Amsterdam, The Netherlands: North Holland, 2004.
- [27] R. Gen ay and A. Altay-Salih, “Degree of mispricing with the Black–Scholes model and nonparametric cures,” *Ann. Econom. Finance*, vol. 4, pp. 73–101, 2003.
- [28] R. Gen ay and M. Qi, “Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping and bagging,” *IEEE Trans. Neural Netw.*, vol. 12, no. 4, pp. 726–734, Jul. 2001.
- [29] A. Gottschling, C. Haefke, and H. White, “Closed form integration of artificial neural networks with some applications to finance,” Univ. California, San Diego, CA, Tech. Rep., 2000.
- [30] M. Hanke, “Neural networks vs. Black–Scholes: An empirical comparison of two fundamentally different option pricing methods,” *J. Comput. Intell. Finance*, vol. 7, no. 1, pp. 26–34, 1999.
- [31] B. Happel and J. Murre, “Design and evolution of modular neural network architectures,” *Neural Netw.*, vol. 7, pp. 985–1004, 1994.
- [32] S. Haykin, *Neural Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1999.
- [33] J. Hull and A. White, “The pricing of options on assets with stochastic volatilities,” *J. Finance*, vol. 42, pp. 281–300, 1987.
- [34] J. M. Hutchinson, A. W. Lo, and T. Poggio, “A nonparametric approach to pricing and hedging derivative securities via learning networks,” *J. Finance*, vol. 49, no. 3, pp. 851–889, 1994.
- [35] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts,” *Neural Comput.*, vol. 3, pp. 79–87, 1991.
- [36] J. R. Jang, C. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
- [37] C. Keber and M. G. Schuster, “Generalized ant programming in option pricing: Determining implied volatilities based on American put options,” in *Proc. IEEE Int. Conf. Comput. Intell. Financial Eng.*, 2003, pp. 123–130.
- [38] G. C. Lim, J. N. Lye, G. M. Martin, and V. L. Martin, “The distribution of exchange rate returns and the pricing of currency options,” *J. Int. Econom.*, vol. 45, pp. 351–368, 1998.
- [39] M. Liu, “Option pricing with neural networks,” in *Progress in Neural Information Processing*, S. I. Amari, L. Xu, L. W. Chan, I. King, and K. S. Leung, Eds. New York: Springer-Verlag, 1996, vol. 2, pp. 760–765.

- [40] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. London, U.K.: Chapman & Hall, 1989.
- [41] W. R. Melick and C. P. Thomas, "Recovering an asset's implied pdf from option prices: An application to crude oil during the gulf crisis," *J. Financial Quantitative Anal.*, vol. 32, no. 1, pp. 91–115, 1997.
- [42] J. Pan, "The jump-risk premia implicit in options: Evidence from an integrated time series study," *J. Financial Econom.*, vol. 63, pp. 3–50, 2002.
- [43] M. Qi and G. S. Maddala, "Option pricing using artificial neural networks: The case of S&P 500 index call options," in *Neural Networks in Financial Engineering: Proceedings of the Third International Conference on Neural Networks in the Capital Markets*, A. P. N. Refenes, Y. Abu-Mostafa, J. Moody, and A. Weigend, Eds. New York: World Scientific, 1996, pp. 78–91.
- [44] E. Ronco and P. J. Gawthrop, "Modular neural networks: A state of the art," *Cntr. Syst. Control*, Univ. Glasgow, Glasgow, U.K., Tech. Rep. CSC-95026, 1995.
- [45] M. Rubinstein, "Implied binomial trees," *J. Finance*, vol. 49, no. 3, pp. 771–818, 1994.
- [46] L. Scott, "Pricing stock options in a jump-diffusion model with stochastic volatility and interest rates: Applications of Fourier inversion methods," *Math. Finance*, vol. 7, pp. 413–426, 1997.
- [47] K. D. West, "Asymptotic inference about predictive ability," *Econometrica*, vol. 64, pp. 1067–1084, 1996.
- [48] S. J. Yakowitz, "Nearest neighbour methods for time series analysis," *J. Time Series Anal.*, vol. 8, no. 2, pp. 235–247, 1987.
- [49] A. Yatchew and T. Härdle, "Nonparametric state price density estimation using constrained least squares and the bootstrap," *J. Econometrics*, vol. 133, no. 2, pp. 579–599, 2006.



**Nikola Gradojevic** received the Ph.D. degree in financial economics from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2003.

In August 2003, he joined Faculty of Business Administration, Lakehead University, Thunder Bay, ON, Canada, where he has been an Associate Professor of Finance since 2007. During his career he took positions at UBC (teaching and research), Bank of Canada, Federal Reserve Bank of St. Louis, and in the private sector as a consultant. His research interests include international finance (market microstructure), empirical asset pricing (option pricing), technical trading, asset price volatility and bubbles, and artificial intelligence (neural networks, fuzzy logic, etc.). He is currently also a visiting associate professor at the Department of Industrial Engineering and Management, Faculty of Technical Sciences, Novi Sad, Serbia.



**Ramo Gençay** received the Ph.D. degree in economics from University of Houston, Houston, TX, in 1991.

Currently, he is with the Department of Economics, Simon Fraser University, Burnaby, BC, Canada. His broad specializations are financial econometrics, nonlinear time series, nonparametric econometrics, and chaotic dynamics. His scientific publications have appeared in finance, economics, engineering, statistics, and physics journals. Some of his publications are published in *Journal of the American Statistical Association*, *Journal of Econometrics*, *Journal of International Economics*, *International Economic Review*, *Journal of Nonparametric Statistics*, *Journal of Empirical Finance*, *Journal of Economic Dynamics and Control*, *Journal of Applied Econometrics*, *European Economic Review*, *Journal of Forecasting*, the IEEE TRANSACTIONS ON NEURAL NETWORKS, *Physica A*, *Physica D*, and *Physics Letters A*. He is a coauthor of two books: *An Introduction to High-Frequency Finance* (New York: Academic, 2001) and *Wavelets and Other Filtering Methods in Finance and Economics* (New York: Academic, 2001), and the founding editor of *Finance Research Letters*.



**Dragan Kukulj** (M'97–SM'06) received the Diploma degree in control engineering, the M.Sc. degree in computer engineering, and the Ph.D. degree in control engineering from the University of Novi Sad, Novi Sad, Serbia, in 1982, 1988, and 1993, respectively.

Currently, he is a Professor of Computer-Based Systems at the Department of Computing and Control, Faculty of Engineering, University of Novi Sad. His main research interests include soft computing techniques and their applications in signal processing and process control. He has published over 100 papers in journals and conference proceedings. He is the consultant for Micronas Novi Sad R&D Institute of Information Technology, where he is involved in the areas of soft computing and computer-based systems integration with application in high-tech sensors, audio, and video processing.