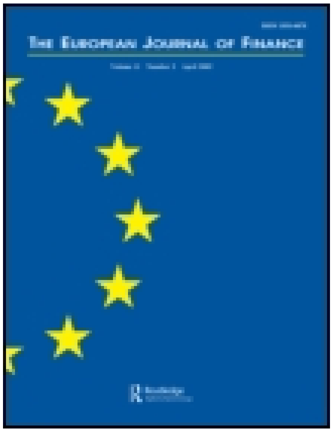

The European Journal of Finance
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/rejf20

Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks

Georgios Sermpinis [a], Jason Laws [b] & Christian L. Dunis [c]

[a] University of Glasgow Business School, University of Glasgow, Gilbert Scott Building, Glasgow, G12 8QQ, UK

[b] University of Liverpool Management School, The University of Liverpool, Chatham Street, Liverpool, L69 7ZH, UK

[c] Liverpool Business School, John Moores University, John Foster Building, 98 Mount Pleasant, Liverpool, L3 5UZ, UK
Published online: 21 Sep 2011.

To cite this article: Georgios Sermpinis, Jason Laws & Christian L. Dunis (2013) Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks, The European Journal of Finance, 19:3, 165-179, DOI: 10.1080/1351847X.2011.606990

To link to this article: http://dx.doi.org/10.1080/1351847X.2011.606990

The European Journal of Finance, 2013
Vol. 19, No. 3, 165–179, http://dx.doi.org/10.1080/1351847X.2011.606990

Modelling and trading the realised volatility of the FTSE100 futures with higher order neural networks

Georgios Sermpinis[a*], Jason Laws[b] and Christian L. Dunis[c]

[a]*University of Glasgow Business School, University of Glasgow, Gilbert Scott Building, Glasgow G12 8QQ, UK;*
[b]*University of Liverpool Management School, The University of Liverpool, Chatham Street Liverpool L69 7ZH, UK;*
[c]*Liverpool Business School, John Moores University, John Foster Building, 98 Mount Pleasant, Liverpool L3 5UZ, UK*

The motivation for this article is the investigation of the use of a promising class of neural network (NN) models, higher order neural networks (HONNs), when applied to the task of forecasting and trading the 21-day-ahead realised volatility of the FTSE 100 futures index. This is done by benchmarking their results with those of two different NN designs, the multi-layer perceptron (MLP) and the recurrent neural network (RNN), along with a traditional technique, RiskMetrics. More specifically, the forecasting and trading performance of all models is examined over the eight FTSE 100 futures maturities of the period 2007–2008 using the realised volatility of the last 21 trading days of each maturity as the out-of-sample target. The statistical evaluation of our models is done by using a series of measures such as the mean absolute error, the mean absolute percentage error, the root-mean-squared error and the Theil *U*-statistic. Then we apply a simple trading strategy to exploit our forecasts based on trading at-the-money call options on FTSE 100 futures. As it turns out, HONNs demonstrate a remarkable performance and outperform all other models not only in terms of statistical accuracy but also in terms of trading efficiency. We also note that both the RNNs and MLPs provide sufficient results in the trading application in terms of cumulative profit and average profit per trade.

Keywords: higher order neural networks; recurrent neural networks; multi-layer perceptron networks; volatility forecast; option trading application

JEL Classification: C45

1. Introduction

Neural networks (NNs) are an emergent technology with an increasing number of real-world applications including finance (Lisboa and Vellido 2000). However, their numerous limitations such as their sensitivity to noise and distortion often create scepticism about their use among practitioners. Although there is a plethora of empirical evidence on their contribution in forecasting and trading equity indexes, there are very few articles that examine their contribution to volatility forecasting and volatility trading applications.

The motivation for this article is the investigation of the use of a promising NN architecture which practically is less sensitive to noise and distortion (Giles and Maxwell 1987), higher order neural networks (HONNs), when applied to the forecasting and trading of the 21-day-ahead annualised volatility of the FTSE 100 futures index. The results are benchmarked against two different NN designs, the multi-layer perceptron (MLP) and the recurrent neural network (RNN), along with a traditional benchmark of RiskMetrics.

*Corresponding author. Email: georgios.sermpinis@glasgow.ac.uk

In order to statistically evaluate our models, we computed the mean absolute error (MAE), the mean absolute percentage error (MAPE), the root-mean-squared error (RMSE) and the Theil U-statistic. We then compared our forecasts with the relevant Black and Scholes (1973) market derived implied volatilities and apply a simple trading strategy based on trading at-the-money (ATM) call options on FTSE 100 futures. We observed that HONNs demonstrate a remarkable performance and outperform all other models not only in terms of statistical accuracy but also in terms of trading performance. We also observed that both the RNNs and MLPs provide satisfactory results in the trading application in terms of cumulative profit and average profit per trade.

The rest of the article is organised as follows. Section 2 presents the literature relevant to HONNs and the applications of NNs in a real trading and option pricing model context. Section 3 describes the data set used for this research and its characteristics. Section 4 gives an overview of the different NN models, the RiskMetrics volatility and the GJR model. Section 5 gives the empirical results of all the models in terms of their statistical accuracy and trading efficiency, while Section 6 provides some concluding remarks.

2. Literature review

HONNs were first introduced by Giles and Maxwell (1987) as a fast learning network with increased learning capabilities. Although their function approximation superiority over the more traditional architectures is well documented in the literature (see, among others, Redding, Kowalczyk, and Downs 1993; Kosmatopoulos et al. 1995; Psaltis, Park, and Hong 1988), their use in finance so far has been limited. This changed when scientists started to investigate not only the benefits of NNs against more traditional statistical techniques but also the differences between the different NN model architectures. Practical applications have now verified the theoretical advantages of HONNs by demonstrating their superior forecasting ability and put them in the front line of research in financial forecasting. For example, Fulcher, Zhang, and Xu (2006) applied HONNs to forecast the AUD/USD exchange rate for the year 2004, achieving 90% accuracy. On the other hand, Dunis, Laws, and Sermpinis (2011) studied the EUR/USD series for a period of 10 years and demonstrated that when multivariate series are used as inputs, HONNs, RNNs and MLPs have a similar forecasting power, while Dunis, Laws, and Sermpinis (2010a), who examined the EUR/USD ECB (European Central Bank) fixing series from the day of its creation until the end of 2007, showed that HONNs have a superior performance to MLPs and RNNs when autoregressive series are used as inputs. Furthermore, more recently, Dunis, Laws, and Sermpinis (2010b) made accurate forecasts of the value at risk of Gold Bullion and Brent Oil using a hybrid HONN–RiskMetrics model.

In the field of exploiting NN forecasts in an option pricing model context, Hutchinson, Lo, and Poggio (1994) used NNs to successfully price the S&P 500 futures options. Malliaris and Salchenberger (1996) accurately forecasted the Black–Scholes derived implied volatility of the S&P 100 ATM call options with an MLP model, while Garcia and Gencay (2000) created an option pricing model with feedforward networks, which provides smaller delta-hedging errors relative to the ones generated by the Black–Scholes model. Yao, Li, and Tan (2000) forecasted the option prices of the Nikkei 225 index futures with backpropagation NNs. They concluded that although NNs outperform the BS model for volatile markets, the BS model is still good for pricing ATM options. Moreover, Meissner and Kawano (2001) used Garch volatility forecasts as inputs to four different NN models and created option pricing models which present significant better pricing performance than the Black–Scholes model. Amilon (2003) studied whether a MLP model could be used to find a call option pricing formula better corresponding to market prices and the properties of the Swedish stock index call options from 1997 to 1999 than the Black–Scholes formula. His results indicate that his MLP outperforms the Black–Scholes formula in both pricing and hedging performances. On the other hand, Hamid and Iqbal (2004) forecasted the volatility of the S&P 500 futures index using an MLP model. He found that his volatility forecasts were not statistically different from the realised volatility and were more accurate than the implied volatility, generated by the Barone-Adesi and Whaley (1987) model, of the S&P 500 index futures options. Moreover, Lin and Yeh (2005) demonstrated the superiority of NNs over the Black–Scholes pricing model in the task of forecasting the option prices of Taiwan stock index options from January 2002 to December 2003. Furthermore, Pires and Marwala (2004) successfully forecasted the prices of American-style call options on the Johannesburg Stock Exchange with Bayesian NNs, while Andreou, Charalambous, and Matzoukos (2008) forecasted with good accuracy the price of European call options on the S&P 500 by combining an MLP model with the Black–Scholes and the Corrado and Su (1996) models. Moreover, Gardovejic, Gencay, and Kukolj (2009), based on the option pricing models of Hutchinson, Lo, and Poggio (1994) and Garcia and Gencay (2000), created a modular NN to price the S&P 500 European call options from January 1987 to October 1994. Their model was more accurate than the BS model in all cases, except in 1987. Gencay and Gibson (2007) forecasted the S&P 500 call options from February 1987 to December 1991 using feedforward NNs, stochastic volatility and stochastic volatility random jump models. Their NN models showed smaller errors than their parametric counterparts.

3. The FTSE100 futures and related financial data

Our benchmark test was to forecast the 21-day-ahead realised volatility of the FTSE 100 futures returns. For the FTSE 100 futures, there are four delivery months: March, June, September and December. Trading ceases on the third Friday of the delivery month of the contract as soon as reasonably practicable after 10:15 (London time) once the expiry value of the index has been determined. In our application, we examined the eight different futures contracts which expire in 2007 and 2008. For each of the eight contracts, we had their closing prices for almost a year before their expiration.

More specifically, we examined the eight different FTSE100 futures contracts presented in Table 1.

In Appendix A.1, we present the FTSE 100 index level and the 21-day realised standard deviations of the FTSE 100 returns from 1 January 2006 to 31 December 2008.

In all cases, we used the last 21 trading days as out-of-sample data set (i.e. 1 month) and the rest of the days as in-sample data set. This 21-day sample also represents roughly 8% of the available

Table 1. FTSE 100 futures contracts.

Delivery month of the contract	Available trading days
March 2007	260
June 2007	265
September 2007	265
December 2007	264
March 2008	264
June 2008	265
September 2008	260
December 2008	264

data for each contract which is in line with the common practice of selecting 8–16% of the data sample for out-of-sample testing (see, among others, Garcia and Gencay 2000; Amilon 2003; Andreou, Charalambous, and Matzoukos 2008). All the eight series are non-normal (Jarque–Bera statistics confirmed this at the 99% confidence interval) having skewness and high kurtosis and are non-stationary. For the purpose of our research, we transformed them into a stationary series of returns using the following formula:

$$R_t = \left(\frac{P_t}{P_{t-1}}\right) - 1, \tag{1}$$

where R_t is the rate of return and P_t is the price level at time t.

The summary statistics of the eight futures returns series revealed that they are slightly skewed and had high kurtosis. The Jarque–Bera statistic confirmed again that all series are non-normal at the 99% confidence interval.

In the absence of any formal theory behind the selection of the inputs of a NN and based on the experiments during the in-sample period, we fed our networks with the first four autoregressive lags of the FTSE 100 returns 21-day annualised realised volatility. The use of autoregressive lags as inputs to NNs is common in the literature (Dunis, Laws, and Sermpinis 2011; Malliaris and Salchenberger 1996; Zhang, Xu, and Fulcher 2002; Fulcher, Zhang, and Xu 2006).

4. Volatility forecasting models

In this article, we benchmarked HONNs with two more traditional NNs designs, an MLP and an RNN model, and statistical techniques such as the RiskMetrics volatility and a GJR model in the task of forecasting the 21-day-ahead annualised volatility of the FTSE 100 futures index. An estimation of the realised 21-day-ahead annualised volatility, which we are interested in forecasting as accurately as possible, can be given by the expression

$$\sigma_{t+1} = \frac{1}{21}\sum_{t-20}^{t}\sqrt{252}|R_t|, \tag{2}$$

where R_t is the realised daily return of the FTSE 100 futures index.

When each of the eight option contracts reached 21 days to expiry, we made a volatility forecast of the FTSE100 futures index with a time horizon corresponding to the remaining life of the option. Therefore, for each model, we had eight different out-of-sample volatility forecasts. Note, however, that although this looks small, in each case, we had to train the models over 20 different networks (Section 4.3). Moreover, research using traded options, rather than over-the-counter options, is restricted by the limited expiry cycle. For financial futures, this is typically four per annum.

4.1 *RiskMetrics volatility*

The RiskMetrics volatility model is treated as a benchmark model owing to its simplicity and popularity in the financial industry. Although it is often considered as too simple a forecast in academia, it is nevertheless popular among quantitative professionals.[1] Derived from the GARCH(1, 1) model, but with fixed coefficients, RiskMetrics volatility is calculated using the standard formula

$$\mathrm{RMVOL}_t^2 = b\sigma_{t-1}^2 + (1-b)R_t^2, \tag{3}$$

where σ_t^2 is the futures index variance at time t, R_t^2 is the futures index squared return at time t and $b = 0.94$ for daily data. In this paper, we used RiskMetrics volatility to forecast 21-day-ahead annualised volatility for the out-of-sample period. The RiskMetrics volatility was calculated from Equation (3), and then we used Equation (4) to calculate the 21-day-ahead annualised volatility forecast:

$$\hat{\sigma}_{t+1} = \mathrm{RMVOL}_t\sqrt{252}. \tag{4}$$

4.2 *GJR model*

We also forecasted the 21-day-ahead volatility with a Glosten, Jagannathan, and Runkle (1993) (GJR) model. One of the primary restrictions of the GARCH models is their symmetric response to positive and negative shocks. However, it has been argued that a negative shock to financial time series is likely to cause volatility to rise by more than a positive shock of the same magnitude (Bekaert and Wu 2000; Brooks 2003). A popular asymmetric formulation of GARCH which has an additional term to account for possible asymmetries is the GJR model, where the conditional variance is given by

$$\sigma_t^2 = w + au_{t-1}^2 + \beta\sigma_{t-1}^2 + \gamma u_{t-1}^2 I_{t-1}, \tag{5}$$

where σ_t^2 and u_t are the conditional variance and the error, respectively, at time t and $I_{t-1} = 1$ if $u_{t-1} \prec 0$ or 0 otherwise.

In this article, we forecasted the daily volatility with Equation (5), and then we used Equation (6) to calculate the 21-step-ahead annualised volatility:

$$\hat{\sigma}_{t+1} = \sigma_{t+1}\sqrt{252}. \tag{6}$$

4.3 *Neural networks*

NNs exist in several forms. The most popular architecture is the MLP. A standard NN has at least three layers. The first layer is called the input layer (the number of nodes of the layer corresponds to the number of explanatory variables). The last layer is called the output layer (the number of nodes of the layer corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input layer from the output layer. The number of nodes of this layer defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layers contain an extra node, called the bias node. This node has a fixed value of one and has the same function as the intercept in traditional regression models. Normally, each node of one layer has connections with all the other nodes of the next layer.

The network processes information as follows: the input nodes contain the value of the explanatory variables. Since each node connection represents a weight factor, the information reaches a single hidden layer node as the weighted sum of its inputs. Each node of the hidden layer passes the information through a nonlinear activation function and passes it on to the output layer if the calculated value is above a threshold.

The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying a learning algorithm called backpropagation of errors[2]

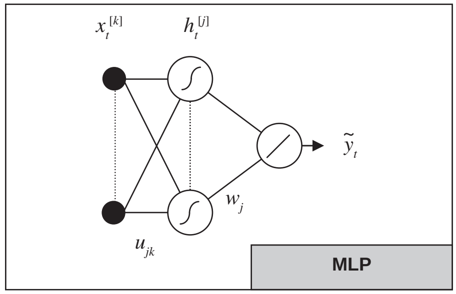

Figure 1. A single-output, fully connected MLP model.

(Shapiro 2000). The learning algorithm simply tries to find those weights which optimise an error function (normally the sum of all squared differences between target and actual values). Since networks with sufficient hidden nodes are able to learn the training data (as well as their outliers and their noise) by heart, it is crucial to protect our forecasts from overfitting. In our application where due to limitations in the available data we had a small data set, we applied the leave-one-out cross-validation method. The superiority of cross-validation from the early stopping procedure in small data sets was well documented by Zhu and Rohwer (1996) and Goutte (1997).[3] In the leave-one-out method, we divided our data set into *n* subsets (where *n* is the in-sample size). Then, we trained the net *n* times, each time leaving out one of the subsets from training but using the omitted subset as validation data. In the end, we averaged out the different validation results from the *n* different rotations of the sample.

The specifications of our NN models (number of iterations, momentum, learning rate and number of hidden layers and nodes) were selected based on trial and error in the in-sample period.

4.3.1 The MLP model

The network architecture of a 'standard' MLP appears as presented in Figure 1, where $x_t^{[n]}(n = 1, 2, \ldots, k + 1)$ are the model inputs (including the input bias node) at time t, $h_t^{[m]}(m = 1, 2, \ldots, j + 1)$ are the hidden node outputs (including the hidden bias node), $\tilde{y}_t$ is the MLP model output, u_{jk} and w_j are the network weights, is the transfer sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{7}$$

is a linear function:

$$F(x) = \sum_i x_i. \tag{8}$$

The error function to be minimised is

$$E(u_{jk}, w_j) = \frac{1}{T} \sum_{t=1}^{T} (y_t - \tilde{y}_t(u_{jk}, w_j))^2, \tag{9}$$

with y_t being the target value and T the number of iterations.

Since the starting point for each network is a set of random weights, forecasts can differ between networks. In order to eliminate any variance between our MLP forecasts, we used the average of a committee of 20 MLPs which provides a better summary of the in-sample statistical performance

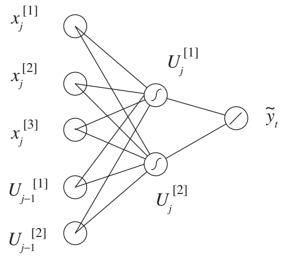

Figure 2. Elman RNN architecture with two nodes on the hidden layer.

(Dunis, Laws, and Sermpinis, 2011, 2010b). The characteristics of the MLPs used in this article are presented in Appendix A.2.

The target value in our networks is the 21-day-ahead estimated annualised volatility of our series as defined from Equation (2).

4.3.2 *The RNN*

A simple RNN has activation feedback, which embodies short-term memory. The advantages of using RNNs over feedforward networks, for modelling nonlinear time series, has been well documented in the past. However, as described in Tenti (1996), 'the main disadvantage of RNNs is that they require substantially more connections, and more memory in simulation, than standard backpropagation networks' (p. 569), thus resulting in a substantial increase in computational time. However, having said this, RNNs can yield better results than simple MLPs due to the additional memory inputs. For an exact specification of the RNN, see Elman (1990).

4.3.2.1. The RNN architecture

A simple illustration of the architecture of an Elman RNN is presented in Figure 2, where $x_t^{[n]}(n = 1, 2, \ldots, k+1)$, $u_t^{[1]}, u_t^{[2]}$ are the model inputs (including the input bias node) at time t, $\tilde{y}_t$ is the recurrent model output, $d_t^{[f]}(f = 1, 2)$ and $w_t^{[n]}$ $(n = 1, 2, \ldots, k+1)$ are the network weights, $U_t^{[f]}(f = 1, 2)$ is the output of the hidden nodes at time t, is the transfer sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{10}$$

is the linear output function:

$$F(x) = \sum_i x_i. \tag{11}$$

The error function to be minimised is

$$E(d_t, w_t) = \frac{1}{T} \sum_{t=1}^{T} (y_t - \tilde{y}_t(d_t, w_t))^2. \tag{12}$$

In short, the RNN architecture can provide more accurate outputs because the inputs are potentially taken from all previous values (see inputs $U_{j-1}^{[1]}$ and $U_{j-1}^{[2]}$ in Figure 2).

Again, in order to eliminate any variance between our RNN forecasts, we used the average of a committee of 20 RNNs. The characteristics of the RNNs used in this article are presented in Appendix A.2.

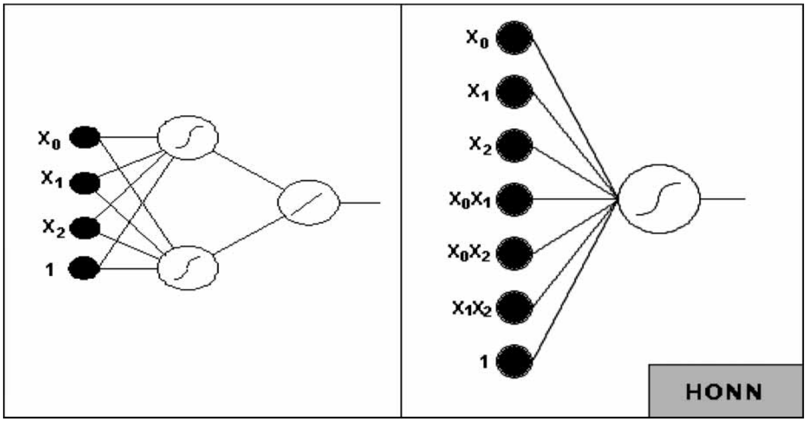

Figure 3. A MLP with three inputs and two hidden nodes (left) and second-order HONN with three inputs (right).

The target value in our networks is the 21-step-ahead estimated annualised volatility of our series as defined from Equation (2).

4.3.3 *Higher order NNs*

HONNs were first introduced by Giles and Maxwell (1987) and were called 'Tensor Networks'. The major advantage of HONNs compared with their NN counterparts is that they are able to provide some rationale for the simulations they produce and thus can be regarded as 'open box' rather than as 'black box' (see Zhang, Xu, and Fulcher 2002). In addition to this, HONNs are able to simulate higher frequency and higher order nonlinear data and consequently provide superior estimations of the underlying pattern than the more classical NN architectures.

While they have already experienced some success in the field of pattern recognition and associative recall,[4] HONNs have not yet been widely used in finance. In Section 2, we have presented a brief review of past studies that applied HONNs in financial forecasting. The architecture of a three-input second-order HONN is shown in Figure 3, where $x_t^{[n]}(n = 1, 2, \ldots, k + 1)$ are the model inputs (including the input bias node) at time t, $\tilde{y}_t$ is the HONN model output, u_{jk} are the network weights, ● are the model inputs, ⌀ is the transfer sigmoid function:

$$S(x) = \frac{1}{1 + e^{-x}}, \tag{13}$$

⌀ is a linear function:

$$F(x) = \sum_i x_i. \tag{14}$$

The error function to be minimised is

$$E(u_{jk}, w_j) = \frac{1}{T} \sum_{t=1}^{T} (y_t - \tilde{y}_t(u_{jk},))^2, \tag{15}$$

with y_t being the target value.

HONNs use joint activation functions; this technique reduces the need to establish the relationships between inputs when training. Furthermore, this reduces the number of free weights and means that HONNs can be faster to train than MLPs. However, because the number of inputs can be very large for higher order architectures, orders of 4 and over are rarely used.

Another advantage of the reduction of free weights is that the problems of overfitting and local optima affecting the results can be largely avoided (see Knowles et al. 2005).

Once again, we used the average of a committee of 20 HONNs. The precise characteristics of these nets are given in Appendix A.2.

Similarly, with the MLPs and the RNNs, the target value in our networks is the 21-day-ahead estimated annualised volatility of our series as defined from Equation (2).

5. Empirical results

5.1 *Statistical performance*

As is standard in the literature, in order to statistically evaluate our forecasts, the RMSE, the MAE, the MAPE and the Theil U-statistics are computed (see, among others, Amilon 2003, Hamid 2004 and Andreou, Charalambous, and Matzoukos 2008). Although we acknowledge that we had only a limited number of observations in order to derive absolutely unambiguous statistical conclusions, we believe that a statistical analysis will provide some information regarding the accuracy of our forecasts. The RMSE and MAE statistics are scale-dependent measures but give a basis to compare volatility forecasts with the realised volatility, while the MAPE and the Theil U-statistics are independent of the scale of the variables. In particular, the Theil U-statistic is constructed in such a way that it necessarily lies between zero and one, with zero indicating a perfect fit. A more detailed description of these measures can be found in Pindyck and Rubinfeld (1998), Theil (1966) and Dunis and Chen (2005), while their mathematical formulas are given in Appendix A.3. For all four of the error statistics retained (RMSE, MAE, MAPE and Theil U), the lower the output, the better the forecasting accuracy of the model concerned. In Tables 2 and 3, we present our results for the in-sample period and out-of-sample period.

As it can be seen from Tables 2 and 3, HONNs outperform all other models and present the most accurate forecasts in statistical terms in both in-sample and out-of-sample periods. It seems like their ability to capture higher order correlations gave them a considerable advantage compared

Table 2. In-sample statistical performance.

	RiskMetrics	GJR	MLPs	RNNs	HONNs
MAE	0.0730	0.0518	0.0512	0.0354	0.0314
MAPE	44.43%	23.47%	22.46%	18.99%	16.49%
RMSE	0.0953	0.0905	0.0891	0.0783	0.0671
Theil U	0.2140	0.2051	0.1402	0.1223	0.0989

Table 3. Out-of-sample statistical performance.

	RiskMetrics	GJR	MLPs	RNNs	HONNs
MAE	0.0663	0.0616	0.0571	0.0512	0.0421
MAPE	26.27%	24.33%	23.91%	20.69%	18.85%
RMSE	0.0937	0.0924	0.0912	0.0829	0.0761
Theil U	0.1607	0.1398	0.1603	0.1391	0.1271

with the other models. Although there is no empirical evidence around the application of HONNs in volatility forecasting, previous studies (Fulcher et al. 2006; Dunis, Laws, and Sermpinis 2010b) suggested that with adequate training, they can be superior in any pattern recognition exercise. On the other hand, RNNs came second and MLPs came third in our statistical evaluation in both periods, while the GJR and the RiskMetrics models presented the less accurate forecasts. Furthermore, we observed that the statistical performance of our NNs was better in sample than out of sample, something that was expected. Moreover, it is worth noting that the time that we needed to train our HONNs was less than half the time needed for the MLPs and the one-tenth needed for RNNs, which is a property attributed to the less connections of an HONN architecture compared with the more complicated MLPs and RNNs.

5.2 *Out-of-sample trading performance*

However, as we were interested in testing our models not only in terms of statistical accuracy but also in terms of trading efficiency, we applied a realistic trading strategy once our volatility forecasts substantially differed from the implied volatilities of the ATM call options on FTSE 100 futures. In Table 4, we present the actual and the derived option premia.

As can be seen from the table, the actual option prices of ATM call options on FTSE 100 futures, 21 days before the expiration of the underlying future contract, considerably differ in most maturities from the Black–Scholes generated option prices if, in the place of the implied volatility, we put our relevant 21-step-ahead volatility forecasts. We also observed that our statistical benchmarks, RiskMetrics and GJR, gave the biggest deviations between the actual and the derived prices. On the other hand, the MLPs gave the smallest difference between the derived premia based on their forecasts and the actual prices.

Our aim was to exploit the differences between the implied volatility and our forecasted volatility by identifying and trading mispriced call options. More specifically, we compared the Black–Scholes derived implied volatility of ATM call options 21 days before their expiration with our relevant 21-day-ahead annualised volatility forecasts.[5] If the difference in absolute terms of the forecasted volatility from the prevailing implied volatility is bigger than a given threshold and the forecasted volatility is bigger than the implied, we will buy the ATM call. If the implied is bigger, we will sell the ATM call. In all the other cases where the difference in absolute terms between the forecasted and the implied is smaller than the threshold, we will not take a position in the market. In our application, we considered four thresholds: 0.5%, 1%, 1.5% and 2%. In Table 5, we present the difference between our forecasts and the relevant implied volatilities.

Table 4. Actual and derived option premia in pound index points.

	Actual	RiskMetrics	GJR	MLPs	RNNs	HONNs
March 2007	65	189	113	67	69	66
June 2007	104	73	110	96	89	91
September 2007	169	175	165	173	178	128
December 2007	179	140	188	181	172	163
March 2008	178	183	184	172	168	184
June 2008	128	115	102	113	112	93
September 2008	128	126	141	141	145	147
December 2008	288	223	220	271	242	263

In terms of our exit rules, a position in a call is held for 10 days after it is first initiated. The rationale for this is that the time decay of an option (measured by its theta) is most severe in the final days of an option's life. Inclusion of this period would bias our results downwards. The transaction costs for one call are assumed to be £3.40 per round trip.[6] The trading performance of our models for each of the four different thresholds is presented in Tables 6–9.

Table 5. Difference between forecasted and implied volatilities.

	RiskMetrics (%)	GJR (%)	MLPs (%)	RNNs (%)	HONNs (%)
March 2007	17.36	2.39	0.32	−0.21	0.21
June 2007	−4.35	2.65	−0.49	−1.58	−1.62
September 2007	−5.37	−0.32	0.62	1.37	−5.23
December 2007	−5.63	2.51	−0.66	−0.73	−0.51
March 2008	0.86	1.17	−0.93	−1.33	−2.45
June 2008	−1.84	−1.95	−1.57	−2.21	−3.13
September 2008	−0.26	4.10	1.61	2.61	−2.12
December 2008	−11.79	−6.04	−1.80	−7.59	−6.81

Table 6. Trading performance for 0.5% threshold.

	RiskMetrics	GJR	MLPs	RNNs	HONNs
Cumulative profit (%)	−145.92	−25.15	−27.06	−6.49	−4.23
Trades	7	7	6	7	7
Buys/sells	2/5	5/2	2/4	2/5	1/6
Profitable trades (%)	29	29	42.9	42.9	57.1
Average profit per trade[a] (%)	−20.85	−3.57	−4.51	−0.93	−0.60

[a]The average profit per trade is simply the cumulative profit divided by the total number of trades.

Table 7. Trading performance for 1% threshold.

	RiskMetrics	GJR	MLPs	RNNs	HONNs
Cumulative profit (%)	−113.67	−25.15	35.46	86.03	88.28
Trades	6	7	4	6	6
Buys/sells	2/4	5/2	1/3	2/4	1/5
Profitable trades (%)	33.33	29	50.00	50.00	66.67
Average profit per trade (%)	−18.94	−3.57	8.87	14.34	14.71

Table 8. Trading performance for 1.5% threshold.

	RiskMetrics	GJR	MLPs	RNNs	HONNs
Cumulative profit (%)	−113.67	7.1	35.46	54.91	88.28
Trades	6	6	4	4	6
Buys/sells	2/4	4/2	1/3	3/1	1/5
Profitable trades (%)	33.33	33.33	50	50	66.67
Average profit per trade (%)	−18.94	1.18	8.87	13.71	14.71

Table 9. Trading performance for 2% threshold.

	RiskMetrics	GJR	MLPs	RNNs	HONNs
Cumulative profit (%)	−197.97	7.1	1.13	34.33	67.71
Trades	5	6	1	3	5
Buys/sells	2/3	4/2	0/1	1/2	1/4
Profitable trades (%)	20.00	33.33	100.00	66.66	60.00
Average profit per trade (%)	−39.59	1.18	1.13	11.44	13.54

We found that HONNs outperformed all other models for each of the four thresholds. Moreover, the MLPs and the RNNs also demonstrated sufficient performance with positive cumulative profits for three of the four thresholds, while the GJR model showed a small profit with thresholds of 1.5% and 2%. On the other hand, the trading results of the RiskMetrics were rather disappointing with negative cumulative profit in all cases. In general, this empirical evidence allows us to argue that NNs were able to successfully identify mispriced options and present a satisfactory trading performance.

6. Conclusion

In this article, we applied MLPs, RNNs and HONNs to a 21-day-ahead forecasting and trading task of the FTSE 100 futures returns realised volatility. As a statistical benchmark, we used a GJR model and the RiskMetrics volatility. We evaluated our forecasts not only in terms of statistical accuracy but also in terms of trading efficiency by applying a simple trading application.

Our results indicate that HONNs outperform all other models as they achieve more accurate forecasts and higher trading performance. These results may be attributed to their ability to capture higher order correlations within a data set. It is also important to note that HONNs require less training time than the MLPs and the RNNs models, a highly desirable feature in a real-life quantitative investment and trading environment.

Notes

1. Note that we also used traditional neural architectures as further benchmarks (see Section 4.3).
2. Backpropagation networks are the most common multilayer networks and are the most commonly used type in financial time series forecasting (Kaastra and Boyd 1996).
3. We also applied the early stopping procedure in our NNs. The statistical and trading performance of our forecasts was similar to those obtained with cross-validation. These results are available upon request.
4. Associative recall is the act of associating two seemingly unrelated entities, such as smell and colour. For more information, see Karayiannis and Venetsanopoulos (1994).
5. We are aware that because of the risk premium the implied volatilities will differ from the relevant realised volatilities at any time point. However, it is beyond the scope of this article to identify and quantify the size and the effect that the risk premium has in our application.
6. These costs were obtained from brokers (see, for instance, www.interactive-brokers.com).

References

Amilon, H. 2003. A neural network versus Black–Scholes: a comparison of pricing and Hedging performances. *Journal of Forecasting* 22, no. 4: 317–35.

Andreou, P., C. Charalambous, and S. Matzoukos. 2008. Pricing and trading European options by combining neural networks with parametric models with implied parameters. *European Journal of Operational Research* 185, no. 3: 1415–33.

Barone-Adesi, G., and E. Whaley. 1987. Efficient analytic approximation of American option values. *Journal of Finance* 42, no. 2: 301–20.

Bekaert, G., and G. Wu. 2000. Assymetry volatility and risk in equity markets. *The Review of Financial Studies* 13, no. 1: 1–42.

Black, F., and M. Scholes. 1973. The pricing of options and corporate liabilities. *Journal Political Economy* 81: 637–54.

Brooks, C. 2003. *Introductory econometrics for finance*. Cambridge: Cambridge University Press.

Corrado, J., and T. Su. 1996. Skewness and Kurtosis in S&P 500 index returns implied by option prices. *Journal of Financial Research* 19, no. 2: 175–92.

Dunis, C., and Y. Chen. 2005. Alternative volatility models for risk management and trading: application to the EUR/USD and USD/JPY rates. *Derivatives Use, Trading & Regulation* 11, no. 2: 126–56.

Dunis, C., J. Laws, and G. Sermpinis. 2011. Higher order and recurrent neural architectures for trading the EUR/USD exchange rate. *Quantitative Finance* 11, no. 4: 615–29.

Dunis, C., J. Laws, and G. Sermpinis. 2010a. Modelling and trading the EUR/USD exchange rate at the ECB fixing. *European Journal of Finance* 16, no. 6: 541–60.

Dunis, C., J. Laws, and G. Sermpinis. 2010b. Modelling commodity value at risk with higher order neural networks. *Applied Financial Economics* 20, no. 7: 585–600.

Elman, J.L. 1990. Finding structure in time. *Cognitive Science* 14: 179–211.

Fulcher, J., M. Zhang, and S. Xu. 2006. The application of higher-order neural networks to financial time series. In *Artificial neural networks in finance and manufacturing*, ed. P.A. Hershey, 80–108. London: Idea Group.

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

Gardovejic, N., R. Gencay, and D. Kukolj. 2009. Option pricing with modular neural networks. *IEEE Transactions on Neural Networks*, 20, no. 4: 626–37.

Gencay, R., and R. Gibson. 2007. Model risk for European-style stock index options. *IEEE Transactions on Neural Networks* 18, no. 1: 193–202.

Giles, L., and T. Maxwell. 1987. Learning, invariance and generalization in higher order neural networks. *Applied Optics* 26, no. 23: 4972–8.

Glosten, L., R. Jagannathan, and D. Runkle. 1993. On the relation between the expected value and the volatility of the nominal excess returns on stocks. *The Journal of Finance* 48, no. 5: 1779–801.

Goutte, C. 1997. Note on free lunches and cross-validation. *Neural Computation* 9, no. 6: 1211–15.

Hamid, S., and Z. Iqbal. 2004. Using neural networks for forecasting volatility of S&P 500 Index futures prices. *Journal of Business Research* 57, no. 10: 1116–25.

Hutchinson, M., W. Lo, and T. Poggio. 1994. A nonparametric approach to pricing and hedging derivative securities via learning networks. *The Journal of Finance* 49, no. 3: 851–89.

Kaastra, I., and M. Boyd. 1996. Designing a neural network for forecasting financial and economic time series. *Neurocomputing* 10, no. 3: 215–36.

Karayiannis, N., and A. Venetsanopoulos. 1994. On the training and performance of high-order neural networks. *Mathematical Biosciences* 129, no. 2: 143–68.

Knowles, A., A. Hussein, W. Deredy, P. Lisboa, and C.L. Dunis. 2005. Higher-order neural networks with Bayesian confidence measure for prediction of EUR/USD exchange rate. CIBEF working papers. http://www.cibef.com.

Kosmatopoulos, E., M. Polycarpou, M. Christodoulou, and P. Ioannou. 1995. High-order neural network structures for identification of dynamical systems. *IEEE Transactions on Neural Networks* 6, no. 2: 422–31.

Lin, C, and H. Yeh. 2005. The valuation of Taiwan stock index option price – comparison of performances between Black–Scholes and neural network model. *Journal of Statistics & Management Systems* 8: 355–67.

Lisboa, P., and A. Vellido. 2000. Business applications of neural networks. In *Business applications of neural networks: the state-of-the-art of real-world applications.* ed. P.J.G. Lisboa, B. Edisbury, and A. Vellido, vii–xxii. Singapore: World Scientific.

Malliaris, M., and L. Salchenberger. 1996. Using neural networks to forecast the S&P 100 implied volatility. *Neurocomputing* 10, no. 2: 183–95.

Meissner, G., and N. Kawano. 2001. Capturing the volatility smile of options on high-tech stocks – a combined Garch-neural network approach. *Journal of Economics and Finance* 25, no. 3: 276–91.

Pindyck, R., and D. Rubinfeld. 1998. *Econometric models and economic forecasts*. New York: McGraw-Hill.

Pires, M., and T. Marwala. 2004. Option pricing using Bayesian neural networks. In *Proceedings of the Annual Symposium of the Pattern Recognition Association of South Africa*, Cape Town, 161–6.

Psaltis, D., C. Park, and J. Hong. 1988. Higher order associative memories and their optical implementations. *Neural Networks* 1: 149–63.
Redding, N., A. Kowalczyk, and T. Downs. 1993. Constructive higher-order network algorithm that is polynomial time. *Neural Networks* 6: 997–1010.
Shapiro, A.F. 2000. A hitchhiker's guide to the techniques of adaptive nonlinear models. *Insurance, Mathematics and Economics* 26, no. 2: 119–32.
Tenti, P. 1996. Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence* 10, no. 6: 567–81.
Theil, H. 1966. *Applied economic forecasting*. Amsterdam, The Netherlands: North-Holland.
Yao, J., Y. Li, and C. Tan. 2000. Option price forecasting using neural networks. *Omega* 28: 455–66.
Zhang, M., S. Xu, and J. Fulcher. 2002. Neuron-adaptive higher order neural-network models for automated financial data modelling. *IEEE Transactions on Neural Networks* 13, no. 1: 188–204.
Zhu, H., and R. Rohwer. 1996. No free lunch for cross-validation. *Neural Computation* 8, no. 7: 1421–6.

Appendix

A.1 *21-day FTSE 100 volatilities*

In Figure A1, we present the FTSE 100 index level and the 21-day rolling annualised volatilities of the FTSE 100 returns from 1 January 2006 to 31 December 2008.

We observed that there was a peak in the 21-day rolling volatilities as we approached the end of a FTSE 100 futures maturity month. We also observed that the volatility during the last six months of 2008 was substantially higher than that during the period before. All these phenomena can lead to the misspecifications of our NN estimations as we were forced to use one year's data for each maturity in order to train our models sufficiently. However, as can been seen from Tables 2 and 3, our models seem to be robust to this anomaly and present statistically accurate forecasts.

A.2 *Network characteristics*

In Table A1, we present the characteristics of the networks used in our research.

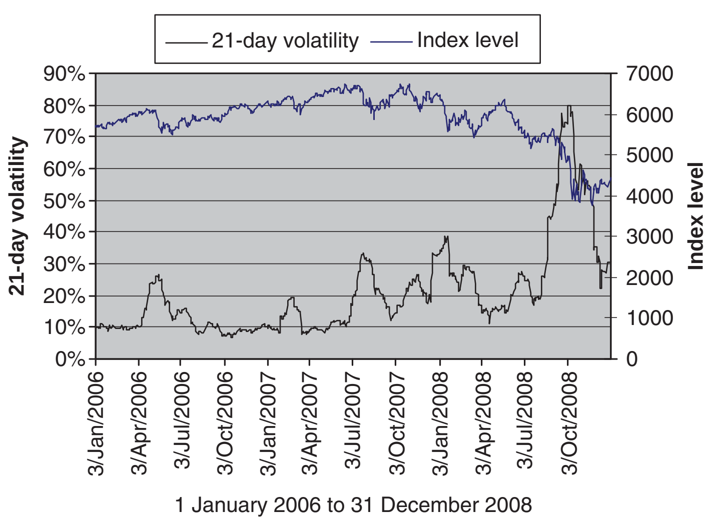

Figure A1. FTSE 100 index level and 21-day 1 January 2006 to 31 December 2008 annualised volatilities.

Table A1. Network characteristics.

Parameters	MLP	RNN	HONN
Learning algorithm	Gradient descent	Gradient descent	Gradient descent
Learning rate	0.001	0.001	0.001
Momentum	0.003	0.003	0.003
Iteration steps	10,000	5000	4000
Initialisation of weights	$N(0,1)$	$N(0,1)$	$N(0,1)$
Input nodes	4	4	4
Hidden nodes (one layer)	3	5	0
Output node	1	1	1

A.3 *Statistical measures*

The statistical measures were calculated as indicated in Table A2.

Table A2. Statistical measures.

Performance measure	Description	
Mean absolute error	$\mathrm{MAE} = \left(\frac{1}{n}\right)\sum_{\tau=t+1}^{t+n} \lvert\hat{\sigma}_\tau - \sigma_\tau\rvert$ with σ_τ being the actual volatility and $\hat{\sigma}_\tau$ the forecasted value	(A1)
Mean absolute percentage error	$\mathrm{MAPE} = \frac{1}{n}\sum_{\tau=t+1}^{t+n} \left\lvert \frac{\sigma_\tau - \hat{\sigma}_\tau}{\sigma_\tau} \right\rvert$	(A2)
RMSE	$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{\tau=t+1}^{t+n} (\hat{\sigma}_\tau - \sigma_\tau)^2}$	(A3)
Theil U	$\mathrm{Theil}U = \frac{\sqrt{(1/n)\sum_{\tau=t+1}^{t+n} (\hat{\sigma}_\tau - \sigma_\tau)^2}}{\sqrt{(1/n)\sum_{\tau=t+1}^{t+n} \hat{\sigma}_\tau^2} + \sqrt{(1/n)\sum_{\tau=t+1}^{t+n} \sigma_\tau^2}}$	(A4)