# Modelling and Trading the English and German Stock Market with Novelty Optimization Techniques.

**Abstract.**

The motivation for this paper is to introduce novel short term models to trade the FTSE100 and DAX30 Exchange traded funds (ETF) indices. There are major contributions in this paper which include the introduction of an input selection criteria when utilising an expansive universe of inputs, a hybrid combination of Partial Swarm Optimizer (PSO) with Radial Basis Functions Neural Networks (RBFNN), the application of a PSO algorithm to a traditional Autoregressive Moving model (ARMA) the application of a PSO algorithm to a Higher Order Neural Network and finally the introduction of a multi-objective algorithm to optimise statistical and trading performance when trading an index. All the machine learning based methodologies and the conventional models are adapted and optimized to model the index. A PSO algorithm is used to optimise the weights in a traditional RBF neural network, in a Higher Order Neural Network (HONN) and the AR and MA terms of an ARMA model. In terms of checking the statistical and empirical accuracy of the novel models we benchmark them with a traditional Higher Order Neural Network, with an autoregressive moving average model (ARMA), with a moving average convergence/divergence model (MACD) and with a naïve strategy. More specifically, the trading and statistical performance of all models is investigated in a forecast simulation of the FTSE100 and DAX30 ETFs time series over the period January 2004 to December 2015 using the last 3 years for out-of-sample testing. Finally the empirical and statistical results indicate that the PSO RBF model outperforms all other examined models in terms of trading accuracy and profitability even with mixed inputs even with only autoregressive inputs.

**Keywords:** Particle Swarm Optimisation, Radial Basis Function, Confirmation Filters, FTSE100, DAX30 Day Trading.

## INTRODUCTION

Modelling and trading financial indices remains a challenging and demanding task for market participants. Forecasting financial time series can be extremely difficult because they are influenced by a large number of variables. Much of the analysed data displays periods of erratic behaviour and as a result drastic declines and spikes in the data series are experienced. Existing linear methods are limited as they only focus on one time series. Some of the older machine learning models also have trouble producing accurate and profitable forecasts due to their rigid architectures. In this paper the proposed models improve on these inefficiencies to make the models more dynamic similar to the time series they are tasked with forecasting. This is particularly important in times of crises as the correlations between different asset classes and time series increase. These inadequacies have been studied in great depth by the scientific community and many methodologies have been proposed to overcome the disadvantages of previous models (Li and Ma, 2010). The main disadvantage of existing non-linear financial forecasting and trading methodologies is that most of them search for global optimal estimators. The problem with this approach is that most of the time global estimators do not exist due to the dynamic nature of financial time-series. Moreover, the algorithms which are used for modelling financial time-series have a lot of algorithms which need to be tuned and if this procedure is performed without careful consideration the accuracy of extracted prediction models will suffer and in some cases result in data-snooping effect. Finally, most of the times the training of a prediction model is performed separately from the construction viable trading signals and thus the overall performance is reduced. In specific, most machine learning algorithms which are designed for forecasting financial time-series deploy only statistical metrics for the optimization steps of their training phase and do not apply any optimization step for improving their trading performances. Here a multi-objective algorithm is employed to optimise both statistical properties and trading performance.

The motivation for this paper is to introduce in a hybrid Neural Network architecture of Particle Swarm Optimization combined with Radial Basis Function (RBF-PSO), which try to overcome some of these limitations. More specifically our proposed architecture is fully adaptive something that decreases the numbers of parameters that the practitioner needs to experiment while on the other hand it increases the forecasting ability of the network. The proposed methodology is superior in comparison to the application of meta-heuristic methods (PSO, Genetic Algorithms, Swarm Fish Algorithm) that have been already presented in the literature (Nekoukar and Beheshti (2010) and Shen *et al.* (2011)) because it eradicates the risk of getting trapped into local optima and the final solution is assured to be optimal for a subset of the training set.

The machine learning model which was used was a hybrid combination of an adaptive version of the Particle Swarm Optimisation (PSO) algorithm (Kennedy and Eberhart, 1995). Numerous existing papers utilize PSO RBF neural networks to model financial time series however many of these are limited in their application as they do not optimize the number of hidden neurons nor do they have a selection criteria for the input series. The Partial Swarm Optimizer applied by Ding *et al.*(2005) was used for selecting the optimal feature subset and optimizing the structure of Radial Basis Function Neural Networks. Moreover, a multi-objective approach was used to account for both statistical and trading performance. In particular two fitness functions are combined to minimize error and maximize annualized returns. This approach was first successfully applied to the modelling and trading of foreign exchange rates (Karathanasopoulos *et al.*, (2012a), (2012c), (2013a), (2013c) and (2015a). Another important limitation of existing methodologies for modelling and trading financial time series is that only a small set of autoregressive inputs and technical indicators are used as inputs. In this investigation, a FTSE100 and DAX30 ETFs specific superset of 50 inputs are evaluated. To the best of our knowledge this is the first time that this adaptive PSO algorithm is combined with an RBF neural network to model and forecast equity indices. Moreover, our proposed machine learning method also applies the PSO algorithm to select the more relevant inputs at each time step. This is different from many other existing non-linear models as most neural networks provide a prediction in the form of a weighted computation of all inputs which are fed into the network during the training process. Therefore, the proposed model has an ability to locate the optimal feature subset which should be used as inputs. This enables the practitioner to introduce a more expansive universe of inputs without having to worry about a noticeable reduction in training times or a redundancy of features. Moreover, the feature selection is a dynamic procedure and not a static one with different feature subsets being selected in different time steps. This also helps remove the risk of survivorship bias when back testing older data as all major equities can be included as inputs. During the back test and for trading the algorithm records the number of times an input is selected which indicates which variables were more influential than others over the examined time period.

The performance of the proposed methodology is compared with numerous linear and non linear methodologies. To allow for a fair comparison we benchmark our proposed algorithm with a Higher Order Neural Network, (HONN) a hybrid HONN-PSO, an autoregressive moving average model (ARMA), an ARMA-PSO, a moving average convergence/divergence model (MACD) plus a naïve strategy in a forecasting and trading simulation of the FTSE100 and DAX30.

The rest of the paper is organized as follows: Section 2 presents a review of literature which is focused on forecasting methodologies and in particular neural networks. Section 3 describes the dataset used for the experiments and the descriptive statistics. Section 4 describes all the models in this paper. Section 5 is the penultimate chapter which presents the empirical results and an overview of the benchmark models. The final chapter presents concluding remarks and future objectives and research.

## LITERATURE REVIEW

Developing high accuracy techniques for predicting time series is a very crucial problem for scientists and decision makers. The traditional statistical methods seem to fail to capture the discontinuities, the nonlinearities and the high complexity of datasets such as financial time series. Complex machine learning techniques like Artificial Neural Networks (NNs) provide enough learning capacity and are more likely to capture the complex non-linear models which are dominant in the financial markets but their parameter tuning remains difficult and generalization problems exist (Donaldson and Kamstra (1996) and Lisboa and Vellido (2000)).

The main objective of this paper is to introduce a novel hybrid method which is able to overcome the difficulties in tuning the parameters of artificial neural networks. For this purpose among the various neural network techniques, we use the Radial Basis Function

Neural Networks (RBF) which has proven experimentally to outperform the more classical NNs architectures (Broomhead and Lowe (1988)). The hybrid method combines the RBF with Particle Swarm Optimization (PSO) algorithm, a state-of-the art heuristic optimization technique (Kennedy and Eberhart (1995)) in a way that optimizes the neural networks parameters, structure and training procedure. Our proposed methodology is an extension of the algorithm proposed by Ding *et. al.* (2005) for forecasting purposes.

The proposed methodology has not been significantly applied in science yet. However, some approaches have been recently proposed for the optimization of RBF Neural Networks and their application in financial time-series forecasting. Nekoukar and Beheshti (2010) propose the application of a modified PSO (using hunter particles to increase diversity) for training Radial Basis Functions. This methodology was applied for the prediction of the price of Iranian stock time-series. Despite the high prediction accuracy of the derived model, this hybrid technique does not provide any method for optimizing the structure of the RBF network. Moreover, the applied PSO algorithm uses constant parameters, which requires an extra time-consuming optimization step. Shen *et al.* (2011) introduce a novel hybrid technique which applies an Artificial Fish Swarm algorithm to train Radial Basis Function Neural Networks for modeling the Shanghai Composite Indices. The prediction results are extremely good, but the artificial fish swarm algorithm is not used for the optimization of the RBF network's structure and it requires some parameters to be tuned via a time consuming trial and error approach. Compared to a simple genetic algorithm and a simple PSO method which are also used to train Radial Basis Function Neural Networks, the Artificial Fish Swarm algorithm produces a slightly higher prediction error but the authors believe that being a new intelligent algorithm it has room for improvement and development. Both of these methods use Mean Square Error as a fitness function and they are not specialized for the prediction of financial time series contrary to our proposed methodology.

More recent research conducted by Lee and Ko (2009) focuses on Radial Basis Function (RBF) NNs. Lee and Ko (2009) proposed a NTVE-PSO method which compares existing PSO methods, in terms of prediction the different practical load types of Taiwan power system (Taipower) in terms of predicting one-day ahead and five-days ahead. Yan *et al.* (2005) contributes to the applications of RBF NN by experiments with real-world data sets. Experimental results reveal that the prediction performance of RBF NN is significantly better than a traditional back propagation neural network models. Marcek *et al.* (2009) estimate and apply ARCH-GARCH models for the forecasting bond price series provided by VUB bank. Following the estimation of these models Marcek *et al.* (2009) then forecast the price of the bond using an RBF NN. Cao and Tay (2003) compare a support vector machine model with an RBF and a generic Back Propagation Neural Network model. In their methodology Cao and Tay (2003) analyse five futures contracts which are trade on the CME. Empirical results from this analysis conclude that the RBF NN outperforms the BP NN while producing similar results to the SVR NN. As an overall summary the predictive ability of an RBF is significantly stronger when compared to any of the aforementioned benchmark models. In some cases the performance is almost double that of other comparable models.

With the emergence of newer technology and faster processing power finance has seen numerous advancements in the area of artificial intelligence. As a result, the accuracy and practicality of such models has led to AI being applied to different asset classes and trading strategies. Enke and Thawornwong (2005) suggest that machine learning methodologies provide higher returns when compared to a buy and hold strategy. De Freitas *et al.* (2000) propose a novel strategy for training NNs using sequential Monte Carlo algorithms with a new hybrid gradient descent / sampling importance resampling algorithm (HySIR). The effectiveness of this model was validated following an application to forecasting FTSE100 closing prices. The HySIR model outperformed all the other benchmarks in terms of trading performance. Their novel technique was fixed from values with weights that generate a 200 input-output data test. The input test data was then used to train the model using the weights estimated at the 200th time step. Tino P., *et al.* (2001), Jasic and Wood (2004), karathanasopoulos et al (2012b) and Bennell and Sutcliffe (2005) , show results which indicate that for all markets the improvement in the forecast by non-linear models is significant and highly accurate. Moreover, Edelman (2008) presented a hybrid Calman filter - Radial Basis Function model used in forecasting one day ahead the FTSE100 and ISEQ. This study used lagged returns from previous days as inputs. The results produced by Eldeman are favourable towards the RBF model as it outperformed the buy and hold strategy, a moving average model and even traditional recurrent neural network.

The last few years of AI research has been continued with Ling Bing Tang *et al.* (2009) which analyses the application and validity of wavelet support vector machine for volatility forecasting. Results from their computer simulations and experiments on stock data reveal that kernel functions in support vector machines are unable to accurately predict the cluster feature of volatility. Miazhynskaia *et al.* (2006) attempt to forecast volatility with numerous models. Their conclusion shows that statistical models account for non-normality and explain most of the fat tails in the conditional distribution. As a result, they believe that there is less of a need for complex non-linear models. In their empirical analysis, the return series of the Dow Jones Industrial Average index, FTSE 100 and NIKKEI 225 indices over a period of 16 years are studied. The results are varied across each of the markets.

More recently Nair *et al.* (2011) propose a hybrid GA neural network which, when compared with benchmark models, outperforms displaying superior accuracy and overall performance. Nair *et al.* (2011) forecasts one day ahead and uses closing prices from the FTSE100, BSE Sensex, Nikkei 225, NSE-Nifty and DJIA as inputs for his models. Karathanasopoulos et al (2012a) and (2014) have used for first time another genetic algorithm named gene expression programming. Gene expression programming comparing to other artificial intelligence models gave better performance in forecasting the Greek main stock index and the EURO/USD exchange rate. Karathansopoulos *et al*. (2013b), (2013d) and (2015b) have forecasted successfully a huge range of time series with combination of support vector machines. In their analysis nonlinear models outperform all the others. Lastly, Karathanasopoulos *et al*. (201) have used a sliding window approach which combines adaptive differential evolution and support vector regression for forecasting and trading the ftse100.

## RELATED FINANCIAL DATA

A robust back test was conducted taking the largest stocks by market capitalization to be included in the training of the networks as a representation of the FTSE100's and DAX30s most heavily weighted stocks over the examined time period. The FTSE 100 and DAX30 are weighted indices according to market capitalization which currently comprise of 100 and 30 large cap constituents listed on the London Stock Exchange and German Stock Exchange. For the purpose of the trading simulation, the FTSE100 and DAX30 exchange traded fund are traded to capture daily movements of the FTSE100 and DAX30 main index accordingly. Trading signals are generated based on the forecast produced by each of the models. When the model forecasts a negative return then a short position (sale) is assumed at the close of each day and when the model forecasts a positive return a long position (purchase) is executed.

Profit / loss is determined by daily positions and in circumstances were consecutive negative or positive changes are forecasted the position is held as a trading decision for the following day. In terms of calculating the daily returns of both data series, we convert them to arithmetic returns by estimating equation 1. Given the price level $P_1$, $P_2$,…,$P_t$, the arithmetic return at time t is formed by:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (1)$$

After using the summary statistics of daily returns series we reveal positive skewness and high kurtosis. The Jarque-Bera statistic confirms again that the two return series are non-normal at the 99% confidence level. These two return series will be forecasted from our models Further to that in order to train our neural networks we further divide our dataset as in table 1:

| Name of Period | Trading Days | Beginning | End |
|---|---|---|---|
| ***Total Dataset*** | 2800 | 01/01/ 2004 | 31/12/2015 |
| ***Training Dataset*** | 2050 | 01/01/2004 | 31/12/2012 |
| ***Validation Set*** | 750 | 01/01/2012 | 31/12/2015 |

**Table 1***: Full Dataset for FTSE100 and DAX30*

In the absence of any formal theory behind the selection of the inputs of a neural network, we conduct neural networks experiments and a sensitivity analysis on a pool of potential inputs in the training dataset in order to help our decision. Based on these experiments and the sensitivity analysis we select as inputs the sets of variables that provide the higher trading performance for each network in the in-sample period. Some inputs in our algorithms are combination of autoregressive returns, moving averages, fixed income returns, commodity returns, equity returns, equity index returns and a volatility time. In details the approach to selecting credible inputs is that for all the stochastic models we use 50 autoregressive returns of the main forecasting index with lags from 1-50 and for the RBF-PSO model for first time we create a pool of 50 mixed inputs which allows the model to select the best inputs for each run. Hence the RBF-PSO runs twice with only autoregressive returns of the main forecasted index and secondly with mixed inputs. The inputs for all the models are presented in table….(All of the data was extracted from FactSet (2015)).

## BENCHMARK MODELS

**Naïve strategy**

The naïve strategy simply takes the most recent period change as the best prediction of the future change. The model is defined by:

$$\hat{Y}_{t+1} = Y_t \qquad (2)$$

Where $Y_t$ is the actual rate of return at period $t$

$\hat{Y}_{t+1}$ is the forecast rate of return for the next period

The performance of the strategy is evaluated in terms of trading performance via a simulated trading strategy.

**Moving Average**

The moving average model is defined as:

$$M_t = \frac{(Y_t + Y_{t-1} + Y_{t-2} + ... + Y_{t-n+1})}{n} \qquad (3)$$

Where $M_t$ is the moving average at time $t$

$n$ is the number of terms in the moving average

$Y_t$ is the actual rate of return at period $t$

The MACD strategy used is quite simple. Two moving average series are created with different moving average lengths. The decision rule for taking positions in the market is straightforward. Positions are taken if the moving averages intersect. If the short-term moving average intersects the long-term moving average from below a 'long' position is taken. Conversely, if the long-term moving average is intersected from above a 'short' position is taken[1].

The forecaster must use judgement when determining the number of periods $n$ on which to

[1]A 'long' position means buying the index at the current price, while a 'short' position means selling the index at the current price.

base the moving averages. The combination that performed best over the in-sample sub-period was retained for out-of-sample evaluation. The models selected for FTSE100 and DAX30 ETFs are FTSE100 (1,8) and DAX30 (1,9) moving averages.

**ARMA Model**

Autoregressive moving average models (ARMA) assume that the value of a time series depends on its previous values (the autoregressive component) and on previous residual values (the moving average component)[2].

The ARMA model takes the form:

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \varepsilon_t - w_1 \varepsilon_{t-1} - w_2 \varepsilon_{t-2} - \ldots - w_q \varepsilon_{t-q} \quad (4)$$

where $Y_t$ is the dependent variable at time $t$

$Y_{t-1}$, $Y_{t-2}$, and $Y_{t-p}$ are the lagged dependent variable

$\phi_0$, $\phi_1$, $\phi_2$, and $\phi_p$ are regression coefficients

$\varepsilon_t$ is the residual term

$\varepsilon_{t-1}$, $\varepsilon_{t-2}$, and $\varepsilon_{t-p}$ are previous values of the residual

$w_1$, $w_2$, and $w_q$ are weights.

Using as a guide the correlogram in the training and the test sub periods we have chosen a restricted ARMA (1,8) model for FTSE100 and ARMA (1,6) for DAX30. All of its coefficients are significant at the 99% confidence interval. The null hypothesis that all coefficients (except the constant) are not significantly different from zero is rejected at the 99% confidence interval

The model selected was retained for out-of-sample estimation. The performance of the strategy

[2] For a full discussion on the procedure, refer to Box et al. (1994)

is evaluated in terms of traditional forecasting accuracy and in terms of trading performance[3].

**ARMA PSO**

The PSO ARMA is optimized by a PSO algorithm to find the optimal combination of AR and MA terms. More explanation on PSO can be found in the next chapter.

**Higher Order Neural Networks**

Higher Order Neural Networks (HONNs) were first introduced by Giles and Maxwell (1987) and were called "Tensor Networks". For Zhang *et al.* (2002), a significant advantage of HONNs is that "HONN models are able to provide some rationale for the simulations they produce and thus can be regarded as "open box" rather then "black box". Moreover, HONNs are able to simulate higher frequency, higher order non-linear data, and consequently provide superior simulations compared to those produced by ANNs (Artificial Neural Networks)" (p. 188). 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. The architecture of a three input second order HONN is shown below:

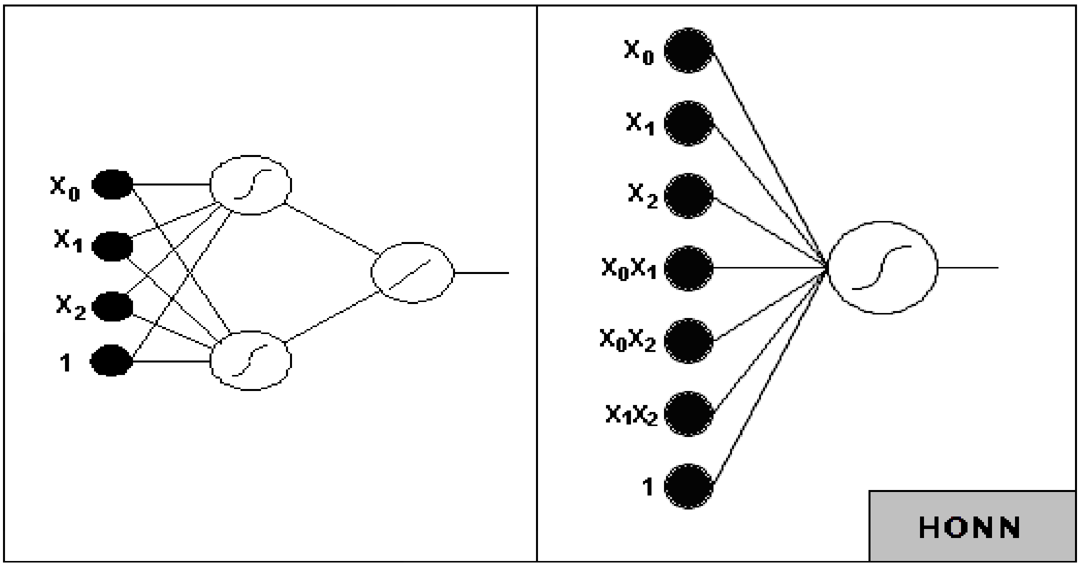



*Fig. 3: Left, MLP with three inputs and two hidden nodes; right, second order HONN with three inputs*

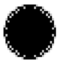
where:

[3] Statistical measures are given in section 4.3 below.

$x_t^{[n]}\ (n = 1,2,\cdots,k+1)$ are the model inputs (including the input bias node) at time $t$

$\tilde{y}_t$ is the HONNs model output

$u_{jk}$ are the network weights

 are the model inputs.

 is the transfer sigmoid function: $S(x) = \frac{1}{1+e^{-x}}$, [5]

 is a linear function: $F(x) = \sum_i x_i$ [6]

The error function to be minimised is:

$$E(u_{jk}, w_j) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \tilde{y}_t(u_{jk},)\right)^2,$$ with $y_t$ being the target value [7]

HONNs use joint activation functions; this technique reduces the need to establish the relationship 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 means that the problems of overfitting and local optima affecting the results can be largely avoided, Knowles *et. al.* (2009). For a complete description of HONNs see Giles and Maxwell (1987).

**HONN-PSO.**

The HONN model is estimated using a traditional back propagation algorithm to adjust the weights when forecasting next day returns. The HONN-PSO model uses the PSO to optimized weights while also maximizing the returns of the model through the equation …..

**Radial Basis Function Neural Networks (RBFNN)**

A radial basis function neural network (RBFNN) is a feedforward neural network where hidden units do not implement an activation function, but a radial basis function. An RBFNN approximates a desired function by superposition of nonorthogonal, radially symmetric functions. They have been proposed by Broomhead and Lowe (1988) as an approach to improve accuracy of artificial neural networks while decreasing training time complexity. Their architecture is depicted in Figure 2.

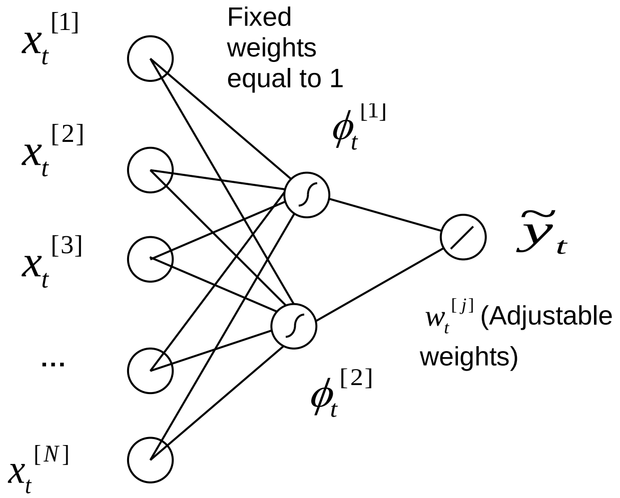



*Fig. 2:* A RBF Neural Network with N inputs and 2 hidden nodes

$x_t \quad (n = 1,2,\cdots,N+1)$ are the model inputs (including the input bias node)

$\tilde{y}_t$ is the RBFNN output

$w_t^{[j]}$ (j=1,2) are the adjustable weights

is the Gaussian function: $\phi^{[i]}(x_t) = e^{-\frac{\|x_t - C_i\|^2}{2\sigma_i^2}}$ [8]

where $C_i$ is a vector indicating the centre of the Gaussian Function and $\sigma_i$ is a value indicating its width. $C_i$, $\sigma_i$ and the weights $w_i$ are parameters which should be optimized through a learning phase in order to train the RBFNN.

is the linear output function: $F(\phi) = \sum_i \phi^{[i]}$ [9]

The error function to be minimised is:

$$E(C,\sigma,w_t) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \tilde{y}_t(w_t,C,\sigma)\right)^2 \qquad [10]$$

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

**Proposed Method RBF-PSO**

In this algorithm the adaptive Particle Swarm Optimization (PSO) methodology was used to locate the parameters $C_i$ of the RBF NN while in parallel locating the optimal number for the hidden layers of the network. The selected candidates are then used as inputs in the proposed model with the adaptive PSO methodology and to reduce the algorithms complexity by using a standard simple neural network topology which is able to improve the generalization properties of the model. The PSO algorithm, proposed by Kennedy and Eberhart (1995), is a population based heuristic search algorithm based on the simulation of the social behaviour of birds within a flock. In PSO, individuals which are referred to as particles are placed initially randomly within the hyper dimensional search space. Changes to the position of particles within the search space are based on the social-psychological tendency of individuals to emulate the success of other individuals. The outcome of modelling this social behaviour is that the search process is such that particles stochastically return towards previously successful regions in the search space. The performance of an RBF NN highly depends on its structure and on the effective calculation of the RBF function's centres $C_i$ and widths σ and the network's weights. If the centres of the RBF are properly estimated then their widths and the networks weights can be computed accurately with existing heuristic and analytical methodologies which are described below in this paper. In this approach the PSO searches only for optimal values of the parameters $C_i$ and the optimal feature subset which should be used as inputs.. For the number of hidden neurons (the RBF NN structure) no further optimization procedure was followed but simple 10 node architecture was selected. This simple topology enables us to alleviate the computational cost of the optimization procedure and to maintain the simplicity in the derived models to achieve better generalization performance.

Each particle i is initialized randomly to have 10 hidden neurons (within a predefined interval starting from the number of inputs until 100 which is the maximum hidden layer size that we applied) and is represented as shown in equation [11]:

$$C^i = \begin{matrix} C^i_{1,1} & C^i_{1,2} & \dots & C^i_{1,d} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ C^i_{10,1} & C^i_{10,2} & \dots & C^i_{10,d} \end{matrix} \quad Input1 \quad Input2 \quad \dots \quad Inputd \qquad [11]$$

Where: N is a large number to point that it does not represent an RBF centre. The variables Input1 to Inputd takes values from -1 to 1 with values larger than o indicating that this feature should be utilized as input.In our PSO variation, initially we create a random population of particles, with candidate solutions represented as showed in equation [2], each one having an initially random velocity matrix to move within the search space. It is this velocity matrix that drives the optimization process, and reflects both the experiential knowledge of the particle and socially exchanged information from the particles neighbourhood. The form of the velocity matrix for every particle is described in the equation below: From the centres of its particle described in equation [11] using the Moody-Darken (1989) approach we compute the RBF widths using equation [12].

$$\sigma^i_j = \left\| c^i_j - c^i_k \right\| \qquad [12]$$

where $c^i_k$ is the nearest neighbour of the centres $c^i_j$. For the estimation of the nearest neighbours we apply the Euclidean distance which is computed for every pair of centres.

At this point of the algorithm the centres and the widths of the RBFNN have been computed. The computation of its optimal weights $w^i$ is accomplished by solving equation [13].

$w^i = (H_i^T \cdot H_i)^{-1} \cdot H_i^T \cdot Y$ [13]

where

$$H_i = \begin{bmatrix} \phi_1^i(x_1) & \phi_2^i(x_1) & \dots & \phi_{10}^i(x_1) \\ \phi_1^i(x_2) & \phi_2^i(x_2) & \dots & \phi_{10}^i(x_2) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ \phi_1^i(x_{n_1}) & \phi_2^i(x_{n_1}) & \dots & \phi_{10}^i(x_{n_1}) \end{bmatrix}$$

where $n_1$ is the number of training samples.

The calculation of $(H_i^T \cdot H_i)^{-1}$ is computationally intensive when the rows of $H_i$ are highly dependent. In order to solve this problem the in-sample dataset is filtered and when the mean absolute distance of two training samples is less than $10^{-3}$ (from the mean values of their input values) then one of them is selected at random to be included in the final training set. As a result, the algorithm becomes faster while maintaining its accuracy. This analytical approach for the estimation of the RBFNN weights is superior in comparison with the application of meta-heuristic methods (PSO, Genetic Algorithms, Swarm Fish algorithm) that have been already presented in the literature because it eradicates the risk of getting trapped into local optima and the final solution is assured to be optimal for a subset of the training set. The algorithm is a multi-objective algorithm which addresses two main elements. The first is an error minimisation algorithm as displayed in equation [14]. The second is employed to optimise and improve the trading performance. Equation [15] optimises annualised returns as first introduced by Karathanasopoulos *et al.* (2013).

$$E(C,\sigma,w_t) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \tilde{y}_t(w_t,C,\sigma)\right)^2 \qquad [14]$$

with $y_t$ being the target value and T the number of trading days.

$R^A$ - MSE – ($n*10^{-2}$) [15]

where: RA = annualised return

MSE = mean square error.

n = number of inputs

Iteratively, the position of each particle is changed by adding in it its velocity vector and the velocity matrix for each particle is changed using the equation below:

$V^{i+1} = w * V^i + c1 * r1 * (C^i_{pbest} - C^i) + c2 * r2 * (C^i_{gbest} - C^i)$ [16]

where w is a positive-valued parameter showing the ability of each particle to maintain its own velocity, $C^i_{pbest}$ is the best solution found by this specific particle so far, $C^i_{gbest}$ is the best solution found by every particle so far, c1 and c2 are used to balance the impact of the best solution found so far for a specific particle and the best solution found by every particle so far in the velocity of a particle. Finally, r1, r2 are random values in the range of [0,1] sampled from a uniform distribution.

Ideally, PSO should explore the search space thoroughly in the first iterations and so the values for the variables w and c1 should be kept high. For the final iterations the swarm should converge to an optimal solution and the area around the best solution should be explored thoroughly. Thus, c2 should be valued with a relatively high value and w, c1 with low values. In order to achieve the described behaviour for our PSO implementation and to avoid getting trapped in local optima when being in an early stage of the algorithm's execution we developed a PSO implementation using adaptive values for the parameters w, c1 and c2. Equations [15], [16] and [17] mathematically describe how the values for these parameters are changed through PSO's iterations helping us to endow the desired behaviour in our methodology.

$w(t)= (0.4/n^2) * (t-n)^2 + 0.4$ [7]

$c1(t)= -2 * t/n +2.5$ [16]

$c2(t)= 2 * t/n + 0.5$ [17]

where t is the present iteration and n is the total number of iterations.

For the initial population of particles a small value of 30 particles (number of articles found with back testing experiments) is used and the number of iterations used was 200 combined with a convergence criterion. Using this termination criterion the algorithm stops when the population of the particles is deemed as converged. The population of the particles is deemed as converged when the average fitness across the current population is less than 5% away from the best fitness of the current population. Specifically, when the average fitness across the current population is less than 5% away from the best fitness of the population, the diversity of the population is very low and evolving it for more generations is unlikely to produce different and better individuals than the existing ones or the ones already examined by the algorithm in previous generations. In summary, the novelty of the algorithm lies in the following points. First of all the feature selection optimizations step allows the utilization of a large number of candidate inputs and enable the final model to only use the most significant variables in order to model an trade the FTSE100 and DAX30 ETFS. Moreover, the adaptive estimation of the models parameters with a single run helps traders to avoid over fitting and data snooping effects. Finally the problem specific fitness function allows for the extraction of models which present high statistical and trading performance

## TRADING PERFORMANCE

### Statistical Performance

The statistical and trading performance for all the models is presented in tables 1-3. The trading strategy for all of the models is to trade based on the forecast produced by each of the models. If the model forecasts a positive return then the trader buys the FTSE100 ETF or DAX30 ETF and if the model predicts a negative return then the trader sells the FTSE100 ETF or DAX30 ETF. For consecutive positive or negative signals the trader holds the previous day's trade to minimise transaction costs. As the proposed model is trained using a multi-objective algorithm the second objective focuses on optimising annualised returns. As a result, table 3 displays results from a filtered trading simulation. These models only trade when the strength of each model's forecast is greater than 30 basis points. This enables the trader to capitalise on more significant moves in the index while avoiding trading during less significant periods. The confirmation filter restricts the model for trading when the forecasted value is less than the optimal confirmation threshold for its out of sample period. Finally, as the non-linear methodologies are stochastic by nature a single forecast is not sufficient enough to represent a credible forecast. For this reason, an average of 1000 estimations where executed to minimise variance. Furthermore in tables 6,7, and 8 the statistical performance in the out-of-sample period of all models is presented. For the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE) and the mean absolute percentage error (MAPE). Interpretation of results is such that, the lower the output, the better the forecasting accuracy of the model concerned. The Pesaran-Timmermann (1992) (PT) test examines whether the directional movements of the real and forecast values are in step with one another. Moreover, it checks how well rises and falls in the forecasted value follow the actual rises and falls of the time series. The null hypothesis is such that the model under study has 'no predictive power' when forecasting the ETF return series. The Diebold-Mariano (1995) DM statistic for predictive accuracy statistic tests the null hypothesis of equal predictive accuracy. Both the DM and the PT tests follow the standard normal distribution.

| Forecast | NAIVE | MACD | ARMA | ARMA-PSO | HONN | HONN-PSO | RBF-PSO Autoregressive inputs | RBF-PSO mixed inputs |
|---|---|---|---|---|---|---|---|---|
| MAE | 0.0145 | 0.0143 | 0.0134 | 0.0110 | 0.0121 | 0.0108 | 0.0104 | 0.0100 |
| MAPE | 421.55% | 387.45% | 321.33% | 300.32% | 250.67% | 245.32% | 236.12% | 150.33% |
| RMSE | 0.0241 | 0.0230 | 0.0210 | 0.0209 | 0.0201 | 0.0195 | 0.0187 | 0.0167 |
| PT-Statisticks | 10.78 | 11.08 | 12.76 | 12.89 | 10.76 | 12.65 | 13.89 | 14.78 |
| DM | -5.78 | -5.89 | -5.45 | -4.78 | -5.05 | -4.08 | -3.56 | -3.12 |
| Correct Directional Change | 50.89% | 51.98% | 54.87% | 56.98% | 53.87% | 60.87% | 62.98% | 64.00% |

**Table 2***: Statistical performance of FTSE100 (out of sample period)*

| Forecast | NAIVE | MACD | ARMA | ARMA-PSO | HONN | HONN-PSO | RBF-PSO Autoregressive inputs | RBF-PSO mixed inputs |
|---|---|---|---|---|---|---|---|---|
| MAE | 0.0140 | 0.0139 | 0.0130 | 0.0109 | 0.0120 | 0.0106 | 0.0100 | 0.0090 |
| MAPE | 332.55% | 300.45% | 231.21% | 210.22% | 200.00% | 189.23% | 180.88% | 123.67% |
| RMSE | 0.0321 | 0.0290 | 0.0223 | 0.0200 | 0.0245 | 0.0189 | 0.0156 | 0.0110 |
| PT-Statisticks | 11.92 | 11.09 | 11.56 | 12.55 | 11.45 | 12.89 | 13.87 | 15.78 |
| DM | -4.45 | -5.89 | -5.04 | -4.89 | -5.98 | -4.01 | -3.98 | -3.78 |
| Corerct Directional Change | 50.07% | 50.67% | 51.54% | 52.23% | 50.23% | 54.17% | 55.98% | 59.87% |

**Table 3**: *Statistical performance of DAX30 (out of sample period)*

By observation, it can be seen that the proposed mixed input PSO RBF model is the strongest statistically. It also predicts the highest number of correct directional changes.

**Empirical Trading Results**

In this section we present the results of all the methodologies applied to trading the DAX30 ETFs and the FTSE100 ETFs. These results are compared with the results of the retained benchmark models. The trading performance of all the models considered in the out-of-sample subset is presented in the table below. Our trading strategy as we mentioned before is go or stay long if the forecasts have a positive movement and go or stay short when a negative direction is forecast.

**unfiltered strategy results:**

| | NAIVE | MACD | ARMA | ARMA-PSO | HONN | HONN-PSO | RBF-PSO Autoregressive inputs | RBF-PSO mixed inputs |
|---|---|---|---|---|---|---|---|---|
| ***Information Ratio*** | 0.59 | 0.53 | 0.90 | 1.27 | 1.11 | 2.09 | 2.58 | 2.70 |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| ***(including costs)*** | | | | | | | | |
| ***Annualised Volatility (including costs)*** | 12.98% | 14.98% | 10.87% | 11.65% | 12.54% | 10.54% | 10.00% | 10.23% |
| ***Annualised Return (including costs)*** | 7.65% | 7.98% | 9.75% | 14.78% | 13.98% | 21.98% | 25.78% | 27.65% |
| ***Maximum Drawdown (including costs)*** | -14.78% | -14.98% | -12.89% | -16.78% | -15.87% | -12.89% | -13.76% | -13.24% |

**Table 4 :** *Out of sample trading performance results for the ftse100 Unfiltered*

| | **NAIVE** | **MACD** | **ARMA** | **ARMA-PSO** | **HONN** | **HONN-PSO** | **RBF-PSO Autoregres sive inputs** | **RBF-PSO mixed inputs** |
|---|---|---|---|---|---|---|---|---|
| ***Information Ratio (including costs)*** | 0.42 | 0.41 | 0.82 | 0.96 | 0.70 | 0.90 | 1.62 | 2.04 |
| ***Annualised Volatility (including costs)*** | 11.53% | 11.75% | 12.98% | 12.37% | 13.89% | 14.45% | 12.87% | 11.52% |
| ***Annualised Return (including costs)*** | 4.89% | 4.87% | 10.65% | 11.87% | 9.76% | 12.98% | 20.87% | 23.53% |
| ***Maximum Drawdown (including costs)*** | -14.89% | -15.87% | -12.98% | -14.66% | -14.35% | -15.87% | -14.73% | -15.83% |

**Table 5 :** *Out of sample trading performance results for the DAX30 Unfiltered*

**filtered strategy results:**

| | **NAIVE** | **MACD** | **ARMA** | **ARMA-PSO** | **HONN** | **HONN-PSO** | **RBF-PSO Autoregres sive inputs** | **RBF-PSO mixed inputs** |
|---|---|---|---|---|---|---|---|---|
| ***BASE POINTS*** | 34 | 40 | 35 | 43 | 40 | 46 | 60 | 49 |
| ***Information Ratio (including costs)*** | 1.22 | 1.39 | 1.46 | 1.72 | 1.53 | 2.04 | 2.50 | 2.57 |
| ***Annualised Volatility (including costs)*** | 10.67% | 10.07% | 10.23% | 9.87% | 9.71% | 9.89% | 9.12% | 9.25% |
| ***Annualised Return (including costs)*** | 12.98% | 13.98% | 14.89% | 16.98% | 14.89% | 20.22% | 22.78% | 23.76% |
| ***Maximum Drawdown (including costs)*** | -10.67% | -10.78% | -11.87% | -12.87% | 12.34% | --12.90% | -11.93% | -10.12% |

**Table 6 :** *Out of sample trading performance results for the ftse100 Filtered*

| | NAIVE | MACD | ARMA | ARMA-PSO | HONN | HONN-PSO | RBF-PSO Autoregressive inputs | RBF-PSO mixed inputs |
|---|---|---|---|---|---|---|---|---|
| ***BASE POINTS*** | 31 | 35 | 40 | 43 | 44 | 50 | 55 | 50 |
| ***Information Ratio (including costs)*** | 0.59 | 0.70 | 0.76 | 204.29 | 1.76 | 2.13 | 1.92 | 2.86 |
| ***Annualised Volatility (including costs)*** | 9.54% | 8.76% | 8.45% | 8.98% | 9.45% | 9.25% | 10.78% | 9.99% |
| ***Annualised Return (including costs)*** | 5.67% | 6.12% | 6.45% | 18.345 | 16.66% | 19.67% | 20.67% | 28.54% |
| ***Maximum Drawdown (including costs)*** | -10.56% | -11.03% | -10.99% | -11.01% | -11.23% | -11.24% | -11.67% | -10.23% |

**Table 7:** *Out of sample trading performance results for the DAX30 Filtered*

As it was expected the proposed methodology clearly outperformed the existing models with leading results across all the examined metrics. Another interesting observation is made when comparing the proposed PSO RBF model with autoregressive inputs with the PSO RBF model with mixed inputs. It's clear the performance of the RBF PSO with mixed inputs. Furthermore, we are quite impressed from the filtered strategy which improves volatility and maximum drawdowns. Lastly the example of HONN PSO and ARMA PSO give us a clear message that partial swarm optimizer when combined with linear or linear models can improve the results

## CONCLUSION

This paper introduces a novel methodology for acquiring profitable and accurate trading results when modelling and trading the FTSE100 and DAX30 etfs indices. The proposed PSO RBF methodology is a combination of a PSO algorithm with a RBF neural network. It not only addresses the limitations of existing non-linear models but it also displays the benefits of using an adaptive hybrid approach to utilizing two algorithms. Furthermore, this investigation also fills a gap in current financial forecasting and trading literature by imposing input selection criteria as a pre-selection system before training each of the neural networks. At the same time we have a novel application of a PSO algorithm to a traditional ARMA model and to a HONN model. Lastly, we use for first time a multi-objective approach to optimising statistical and trading performance.

Experimental results proved that the proposed technique clearly outperformed the examined linear and machine learning techniques in terms of an information ratio and net annualized return. This technique is now a proven and profitable technique when applied to forecasting major ETF indices. Future applications will focus on other equity indices to test the robustness of the PSO RBF model as well as other asset classes. In addition, the lag structure of the inputs will be of more focus in future applications as traders could also benefit from the 'optimisation' of such parameters. The universe of explanatory variables could be enriched further to include more technical time series such as the VWAP (Volume Weighted Average Price), High, Low and Opening prices. Other models outputs could also be included in the input dataset to benefit from the informational content of both existing conventional models and other non-linear methodologies.

## REFERENCES


Bennell, J., and Sutcliffe C., (2005), Black and Scholes Versus Artificial Neural Networks in Pricing FTSE100 Options. Intelligent Systems in Accounting, Finance and Management., *An International Journal*. 12 (4) pp.243-260.

Box, G., Jenkins, G. and Gregory, G. (1994), *Time Series Analysis: Forecasting and Control*, Prentice-Hall, New Jersey.

Broomhead, S. and Lowe, D. (1988), 'Multivariate Functional Interpolation and Adaptive Networks', *Complex Systems*, 2, 321-355.

Cao, L. J., and Tay, F.E.H., (2003) Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting, IEEE transactions on Neural Networks. 14 (6) pp 1506-1518

Chang, C., and Lin, C., (2002) Training Nu-support Vector Regression: Theory and Algorithms. *Neural Computation* 14(8), pp. 1959-77

Diebold, F. X. and Mariano, R. S. (1995), 'Comparing Predictive Accuracy', *Journal of Business and Economic Statistics*, 13, 253-263.

Ding, H., Xiao, Y. and Yue, J. (2005), 'Adaptive Training of Radial Basis Function Networks Using Particle Swarm Optimization Algorithm', *Lecture Notes in Computer Science*, 3610, 119-128.

Donaldson, R. G. and Kamstra, M. (1996), 'Forecast Combining with Neural Networks', *Journal of Forecasting*, 15, 49–61.

De Freitas J, F, G., Niranjan M., Gee A, H, and Doucet A.,(2000) Sequential Monte Carlo Methods to Train Neural Networks Models, *Neural Computation*, 12(4) pp. 995-993.

Ding, H., Xiao, Y., and Yue, J. (2005)Adaptive Training of Radial Basis Function Networks Using Particle Swarm Optimization Algorithm. *Lecture Notes in Computer Science*. 3610, pp. 119-128

Edelman D., (2008) Using Kalman-Filtered Radial Basis Function Networks for index Arbitrage in the Financial Markets. *Natural Computing in Computing in Computational Finance Studies in Computational Intelligenece*, 100, pp 187-195.

Enke D., and Thawornwong S., (2005) The Use Old Data Mining and Neural Networks for forecasting Stock Markets Returns , Expert Systems with Applications 29 (4) 927-940.

Giles, L. and Maxwell, T. (1987) 'Learning, Invariance and Generalization in Higher Order Neural Networks', *Applied Optics*, 26, 4972-4978.

Jasic T., and Wood, Douglas., (2004) The Profitability of Daily Stock Indices Trades Based on Neural Network Predictions: Case Study for the S&P 500, the Dax, the Topix, and the FTSE100 in the Period 1965-1999. *Applied Financial Economics* 14(4), pp. 285-297.

Karathanasopoulos, A., Laws, J., Sermpinis, G., and Dunis, C., (2012a) Forecasting and trading the EUR/USD exchange rate with Gene Expression and Psi Sigma Neural Networks , Expert systems with applications , 39 (10) pp. 8865–8877.

Karathanasopoulos, A., Theofilatos k., Sermpinis, G., Amorgianiotis, T., Georgopoulos, E., and Likothanassis S., (2012b) Modelling and Trading the DJIA Financial Index Using Neural Networks Optimized with Adaptive Evolutionary Algorithms, In Engineering Applications of Neural Networks (Springer Berlin Heidelberg publisher), pp. 453-462.

Karathanasopoulos, A., Sermpinis, G., Theofilatos, K., Georgopoulos, E., and Dunis. C., (2012c)A Hybrid Radial Basis Function and Particle Swarm Optimization Neural Network Approach in Forecasting the EUR/GBP Exchange Rates Returns, Engineering Applications of Neural Networks, 311, pp. 413-422.

Karathanasopoulos, A., Sermpinis, G., Theofilatos, K., Georgopoulos, E., Dunis, C. (2013a) "Forecasting Foreign Exchange Rates with Adaptive Neural Networks Using Radial-Basis Functions and Particle Swarm Optimization." European Journal of Operational Research,

225(3), pp. 528-540.
Karathanasopoulos, A., Dunis, C., Likothanassis, S., Sermpinis, G., and Theofilatos, K. (2013b) A Hybrid Genetic Algorithm-Support Vector Machine Approach in the Task of Forecasting and Trading. Journal of Asset Management, 14, pp. 52-71.
Karathanasopoulos, A., Vasilakis, G., Theofilatos, K., Georgopoulos, E., and Likothanassi, S., A genetic programming approach for EUR/USD exchange rate forecasting and trading , (2013c) , Computational economics , 42 (4) pp 415-431.
Karathanasopoulos, A., Midleton, P., Theofilatos, K., Georgopoulos, E. and Likothanasis S., (2013d) Modeling and Trading FTSE100 Index Using a Novel Sliding Window Approach Which Combines Adaptive Differential Evolution and Support Vector Regression, Artificial Intelligence Applications and Innovation, pp.486-496.
Karathanasopoulos, A., Sermpinis, G., Laws, J.and Dunis, C., (2014) Modelling and Trading the Greek Stock Market with Gene Expression and Genetic Programing Algorithms, Journal of forecasting, 33 (8), pp. 596-610.
Karathanasopoulos, A., Stasinakis, C., Sermpinis, G., and Theofilatos, K., (2015a) Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms—Support vector regression forecast combinations, European Journal of operation research, 247 (3) pp. 831-846.
Karathanasopoulos, A., Sermpinis, G., Stasinakis, C., and Theofilatos, K., (2015b)Forecasting US Unemployment with Radial Basis Neural Networks, Kalman Filters and Support Vector Regressions, Computational economics, pp1-19.
Kennedy J., and R. C. Eberhart, "Particle Swarm Optimization," Proceedings of International Conference on Neural Networks, Orlando, 1997, pp. 1942-1948.
Knowles, A., Hussein, A., Deredy, W., Lisboa, P. and Dunis, C. L. (2009), 'Higher-Order Neural Networks with Bayesian Confidence Measure for Prediction of EUR/USD Exchange Rate', Artificial Higher Order Neural networks for Economic and Business, 1, 48-59,
Lee, C., M. and Ko, C., N., (2009) Time Prediction using RBF Neural Netwoks with nonlinear time-varying evolution PSO algorithm. Neurocomputing, 73 (1-3) pp 449-460.
Li, Y., and Ma, W. (2010) Applications of Artificial Neural Networks in Financial Economics: A Survey. In Proceedings of International Symposium on Computational Intelligence and Design (ISCID), 1, pp. 211-214
Ling-Bing Tang, Ling-Xiao Tang, Huan-Ye Sheng (2009) Forecasting Volatility based on Wavelet Support Vector Machines. *Expert Syst. Appl*. 36(2), pp. 2901-2909
Lisboa, P. and Vellido, A. (2000), 'Business Applications of Neural Networks', vii-xxii, in P. Lisboa, B. Edisbury and A. Vellido [eds.] *Business Applications of Neural Networks: The State-of-the-Art of Real-World Applications,* World Scientific, Singapore.
Moody, J. and Darken, C. J. (1989), 'Fast Learning in Networks of Locally Tuned Processing Units', *Neural Computation*, 1, 2, 281-294.
Nekoukar, V. and Beheshti, H. (2010), 'A Local Linear Radial Basis Function Neural Network for Financial Time-Series Forecasting', *Applied Intelligence*, 33, 3, 352-356.
Pesaran, M.H., and Timmermann, A. (1992), 'A Simple Nonparametric test of Predictive Performance', *Journal of Business and Economic Statistics*, 10, 4, 461-465.
Marcek, D., Marcek, M., and Babel, J.,(2009) (Granular RBF NN Approach and statistical Methods Applied to Modelling and Forecasting High Frequency Data. International Journal of Computational Intelligence systems, 2 (4) pp 353-364.
Miazhynskaia T., Fruhwirth-Schnatter, S., and Dorffner, Georg., (2006) Bayesian Testing for Non-linearity in Volatility Modeling, *Computational Statistics and Data Analysis*. 51 (3), pp. 2029-2042
Nair, B., Gnana Sai, S., Naveen, A., N., Lakshimi, A., Venkatesh, G, S., and Mohandas, V, P., (2011) A GA Artificial Neural Network Hybrid System for Financial Time Series Forecasting.

Information Technology and Mobile Communication , 147, pp. 499-506.
Tino, P., Schittenkopf, C., and Dorffner G., (2001) Financial Volatility Trading using Recurrent Neural Networks, Neural Networks, IEEE Transactions on, 12(4) pp.865-874.
Yan,X.,B, Wang, Z, Yu, S, H. and Li, Y. J., (2005) Time Series Forecasting with RBF Neural Network. Machine Learning and Cybernetics, *Proceedings*, 8 pp 4680-4683.
Zhang, M., Xu, S., X. and Fulcher, J. (2002), 'Neuron-Adaptive Higher Order Neural-Network Models for Automated Financial Data Modelling', IEEE Transactions on Neural Networks, 13, 1, 188-204.

## APENDIX

| Explanatory Variable | Lag(s) | Percentage Selected During Backtest<br>PSO RBF |
|---|---|---|
| Autoregressive Returns | 1 | Selected |
| Autoregressive Returns | 2 | Selected |
| Autoregressive Returns | 3 | Selected |
| Autoregressive Returns | 4 | NO Selected |
| Autoregressive Returns | 5 | NO Selected |
| Autoregressive Returns | 6 | NO Selected |
| HSBC Holdings plc | 1 | NO Selected |
| Vodafone Group Plc | 1 | Selected |
| BP p.l.c. | 1 | Selected |
| Royal Dutch Shell Plc Class A | 1 | Selected |
| GlaxoSmithKline plc | 1 | Selected |
| British American Tobacco p.l.c. | 1 | Selected |
| Royal Dutch Shell Plc Class B | 1 | Selected |
| BG Group plc | 1 | NO Selected |
| Diageo plc | 1 | NO Selected |
| BHP Billiton Plc | 1 | NO Selected |
| Rio Tinto plc | 1 | Selected |
| AstraZeneca PLC | 1 | Selected |
| Gold (NYM $/ozt) Continuous | 1 | Selected |
| Silver (NYM $/ozt) Continuous | 1 | Selected |
| British Pound (CME) Continuous | 1 | Selected |
| British Pounds per Euro | 1 | Selected |
| Euro per British Pounds | 1 | Selected |
| British Pounds per Swiss Franc | 1 | Selected |
| Swiss Franc per British Pounds | 1 | Selected |
| Japanese Yen per British Pounds | 1 | Selected |
| British Pounds per Japanese Yen | 1 | Selected |
| U.S. Dollar per British Pounds | 1 | Selected |
| British Pounds per U.S. Dollar | 1 | Selected |
| Euro STOXX 50 | 1 | Selected |
| S&P 500 | 1 | Selected |
| MSCI EAFE | 1 | Selected |

| | | |
|---|---|---|
| MSCI The World Index | 1 | Selected |
| MSCI AC World | 1 | Selected |
| CBOE Market Volatility Index | 1 | NO Selected |
| Crude Oil (NYM $/bbl) Continuous | 1 | NO Selected |
| Brent Crude (ICE $/bbl) Continuous | 1 | NO Selected |
| US Benchmark Bond - 6 Month | 1 | NO Selected |
| US Benchmark Bond - 5 Year | 1 | NO Selected |
| US Benchmark Bond - 30 Year | 1 | NO Selected |
| US Benchmark Bond - 3 Month | 1 | Selected |
| US Benchmark Bond - 2 Year | 1 | Selected |
| US Benchmark Bond - 10 Year | 1 | Selected |
| US Benchmark Bond - 1 Month | 1 | Selected |
| 21 Day MA | 21 | Selected |
| 50 Day MA | 50 | Selected |
| 100 Day MA | 100 | Selected |
| 150 Day MA | 150 | Selected |
| 200 Day MA | 200 | Selected |
| 250 Day MA | 250 | Selected |

**Table 8.** Input Selection PSO for FTSE100 ETFs

| Explanatory Variable | Lag(s) | Percentage Selected During Backtest PSO RBF |
|---|---|---|
| Autoregressive Returns | 1 | Selected |
| Autoregressive Returns | 2 | Selected |
| Autoregressive Returns | 3 | Selected |
| Autoregressive Returns | 4 | NO Selected |
| Autoregressive Returns | 5 | NO Selected |
| Autoregressive Returns | 6 | NO Selected |
| HSBC Holdings plc | 1 | NO Selected |
| Vodafone Group Plc | 1 | NO Selected |
| BP p.l.c. | 1 | NO Selected |
| Royal Dutch Shell Plc Class A | 1 | NO Selected |
| GlaxoSmithKline plc | 1 | NO Selected |
| British American Tobacco p.l.c. | 1 | Selected |
| Royal Dutch Shell Plc Class B | 1 | Selected |
| BG Group plc | 1 | Selected |
| Diageo plc | 1 | Selected |
| BHP Billiton Plc | 1 | Selected |
| Rio Tinto plc | 1 | Selected |
| AstraZeneca PLC | 1 | Selected |
| Gold (NYM $/ozt) Continuous | 1 | Selected |
| Silver (NYM $/ozt) Continuous | 1 | Selected |

| | | |
|---|---|---|
| British Pound (CME) Continuous | 1 | Selected |
| British Pounds per Euro | 1 | Selected |
| Euro per British Pounds | 1 | Selected |
| British Pounds per Swiss Franc | 1 | Selected |
| Swiss Franc per British Pounds | 1 | Selected |
| Japanese Yen per British Pounds | 1 | Selected |
| British Pounds per Japanese Yen | 1 | Selected |
| U.S. Dollar per British Pounds | 1 | Selected |
| British Pounds per U.S. Dollar | 1 | Selected |
| Euro STOXX 50 | 1 | Selected |
| S&P 500 | 1 | Selected |
| MSCI EAFE | 1 | Selected |
| MSCI The World Index | 1 | Selected |
| MSCI AC World | 1 | Selected |
| CBOE Market Volatility Index | 1 | NO Selected |
| Crude Oil (NYM $/bbl) Continuous | 1 | NO Selected |
| Brent Crude (ICE $/bbl) Continuous | 1 | NO Selected |
| US Benchmark Bond - 6 Month | 1 | NO Selected |
| US Benchmark Bond - 5 Year | 1 | NO Selected |
| US Benchmark Bond - 30 Year | 1 | NO Selected |
| US Benchmark Bond - 3 Month | 1 | NO Selected |
| US Benchmark Bond - 2 Year | 1 | NO Selected |
| US Benchmark Bond - 10 Year | 1 | Selected |
| US Benchmark Bond - 1 Month | 1 | Selected |
| 21 Day MA | 21 | Selected |
| 50 Day MA | 50 | Selected |
| 100 Day MA | 100 | Selected |
| 150 Day MA | 150 | Selected |
| 200 Day MA | 200 | Selected |
| 250 Day MA | 250 | Selected |

**Table 9.** Input Selection PSO for DAX30 ETFs