Comput Econ (2016) 47:569–587
DOI 10.1007/s10614-014-9479-y

Forecasting US Unemployment with Radial Basis Neural Networks, Kalman Filters and Support Vector Regressions

Charalampos Stasinakis · Georgios Sermpinis · Konstantinos Theofilatos · Andreas Karathanasopoulos

Accepted: 3 November 2014 / Published online: 20 November 2014

Abstract This study investigates the efficiency of the radial basis function neural networks in forecasting the US unemployment and explores the utility of Kalman filter and support vector regression as forecast combination techniques. On one hand, an autoregressive moving average model, a smooth transition autoregressive model and three different neural networks architectures, namely a multi-layer perceptron, recurrent neural network and a psi sigma network are used as benchmarks for our radial basis function neural network. On the other hand, our forecast combination methods are benchmarked with a simple average and a least absolute shrinkage and selection operator. The statistical performance of our models is estimated throughout the period of 1972–2012, using the last 7 years for out-of-sample testing. The results show that the radial basis function neural network statistically outperforms all models' individual performances. The forecast combinations are successful, since both Kalman filter and support vector regression techniques improve the statistical accuracy. Finally, support vector regression is found to be the superior model of the forecasting competition. The

C. Stasinakis (✉) · G. Sermpinis
University of Glasgow, Business School, Glasgow G12 8QQ, UK
e-mail: charalampos.stasinakis@glasgow.ac.uk

G. Sermpinis
e-mail: georgios.sermpinis@glasgow.ac.uk

K. Theofilatos
Pattern Recognition Laboratory, Department of Computer Engineering and Informatics, University of Patras, 26500 Patras, Greece
e-mail: theofilk@ceid.upatras.gr

A. Karathanasopoulos
Royal Docks Business School, University of East London, University Way, London E16 2RD, UK
e-mail: a.karathanasopoulos@uel.ac.uk

empirical evidence of this application are further validated by the use of the modified Diebold–Mariano test.

Keywords Forecast combinations · Kalman filter · Support vector regression · Unemployment

1 Introduction

The voluminous macroeconomic literature includes a variety of forecasting competitions of linear and non-linear architectures. Through these studies researchers attempt to shed light on time series, such as inflation or unemployment, that are relevant to monetary and policy decisions worldwide. Several techniques have been applied to such forecasting tasks with ambiguous results. Therefore, statisticians and econometricians turn to highly computational, time-varying and adaptive in nature techniques. Neural networks (NNs) are one such class of models that can assist their quest for improved forecast accuracy. Especially in periods of extreme structural instabilities, NNs' data-adaptive learning and clustering ability can prove to be very useful in forecasting applications (Zhang et al. 1998). It is, thus, not surprising that NNs continue to receive a great deal of attention in the literature (Huang et al. 2013; Özkan 2013; Fernandes et al. 2014; Olmedo 2014).

Forecasting unemployment rates, especially, is a very well documented case study (Szpiro 1997; Montgomery et al. 1998; Rothman 1998; Koop and Potter 1999). Skalin and Teräsvirta (2002) use multivariate STAR models to forecast unemployment rates. Moshiri and Brown (2004) apply a back-propagation model and a generalized regression NN model to estimate post-war aggregate unemployment rates in the USA, Canada, UK, France and Japan. The out-of-sample results confirm the forecasting superiority of the NN approaches against traditional linear and non-linear autoregressive models. Bayesian NNs are applied in the case study of forecasting unemployment in West Germany by Liang (2005). The empirical evidence indicate that the NNs present significantly better forecasts than traditional autoregressive models. Milas and Rothman (2008) use smooth transition vector error-correction models to predict unemployment rates in the non-Euro G7 countries. The proposed model outperforms the linear autoregressive benchmark and improves significantly the forecasts of the US and UK unemployment rate during business cycle expansions. Olmedo (2014) performs a competition between non-linear models, including NNs and nearest neighbour algorithms, to forecast different European unemployment rate time series. The best results are provided by a vector autoregressive and baricentric predictor. As the forecasting horizon lengthens the performance deteriorates and in some cases NNs.

The idea of combining forecasts to improve forecast accuracy is not new (Bates and Granger 1969; Newbold and Granger 1974; Deutsch et al. 1994). Swanson and Zeng (2001) perform forecast combinations based on a model-selection approach and suggest that a SIC-based approach to combine forecasts can be a useful alternative to combination methods such as simple averaging or mean square error minimization. Teräsvirta et al. (2005) examine the forecast accuracy of linear autoregressive, smooth

transition autoregressive and NN models for 47 monthly macroeconomic variables, including unemployment rates, of the G7 economies. The empirical results prove that their forecasting ability is much improved when they are combined with autoregressive models. Kapetanios et al. (2008) report that combinations of statistical forecasts from several models (random walks, STARs, ARs, VARs etc.) generate good forecasts of inflation and growth. They also note that such forecast combinations can serve as an unbiased benchmark, which could be compared with conditional and judgemental policymaker's expectations. Finally, Vasnev et al. (2013) combine forecasts of models incorporating monthly and quarterly macroeconomic time series to predict the monetary operations of the Reserve Bank of Australia. Their findings confirm the benefits of forecast combination models and present alternative methods of forecasting monetary decisions.

Given the previous framework, the rational of this paper is twofold. Firstly, we investigate the efficiency of the radial basis function neural networks (RBFNNs) in forecasting the US unemployment. Secondly, we explore the utility of Kalman filter and support vector regression (SVR) as forecast combination techniques. On one hand, an autoregressive moving average model (ARMA), a smooth transition autoregressive model (STAR) and three different neural networks architectures, namely a multi-layer perceptron (MLP), recurrent neural network (RNN) and a psi sigma network (PSN) are used as benchmarks for our RBFNN. On the other hand, our forecast combination methods are benchmarked with a simple average and a least absolute shrinkage and selection operator (LASSO). The statistical performance of our models is estimated throughout the period of 1972–2012, using the last 7 years for out-of-sample testing. The empirical evidence of this application is further validated by the use of the modified Diebold–Mariano test.

With this study, we intend to extend the growing literature of using RBFNNs and NNs in general in financial and macroeconomic forecasting task. In addition, the evaluation of the Kalman Filter and SVR adds validity to the evidence of previous studies that report the benefits of combining forecasts. Finally, the performance of those non-linear and time-varying combination methods evaluate if there is a need to experiment beyond traditional linear equivalents.

The rest of the paper is organized as follows. Section 2 presents the description of the dataset used in this application. Sections 3 and 4 give an overview of the forecasting models and the forecast combination methods implemented respectively. The statistical performance of our models is presented in Sect. 5. Finally, some concluding remarks are summarized in Sect. 6.

2 US Unemployment Dataset

In this study, we forecast the monthly change of the US unemployment rate (UNEMP). The data can be found on the online federal reserve economic data (FRED) database

Table 1 The US unemployment dataset–neural networks' training dataset

Periods	Months	Start date	End date
Total dataset	492	01/01/1972	01/12/2012
Training dataset (in-sample)	324	01/01/1972	01/12/1998
Test dataset (in-sample)	84	01/01/1999	01/12/2005
Validation dataset (out-of-sample)	84	01/01/2006	01/12/2012

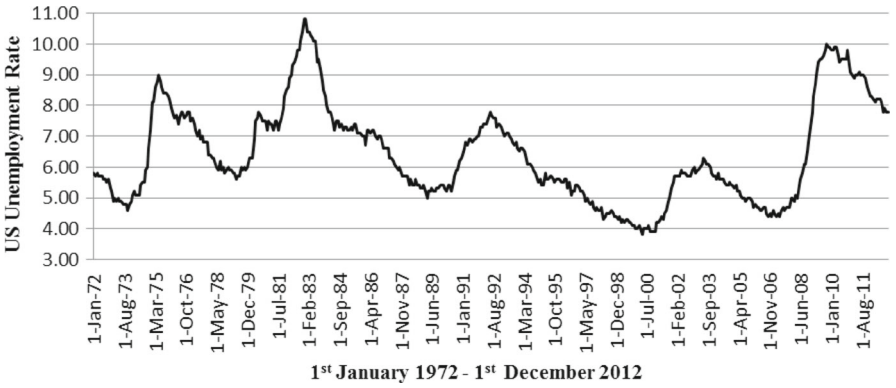

Fig. 1 The US unemployment rate

of the Federal Reserve Bank of St. Louis.[1] This forecasting exercise explores the performance of the models over the period of 1972–2012, using the last 7 years for out-of-sample evaluation. The time series is seasonally adjusted. For training purposes of our NNs, we further divide our in-sample dataset in two sub-periods; the training and test sub-period (see Sect. 3.3). The total dataset is summarized in Table 1.

Figure 1 presents the US unemployment rate for the period under study.

In the literature, there is no formal theory behind the selection of the inputs of a NN. Therefore, we conduct some NN experiments and a sensitivity analysis on a pool of potential inputs in the in-sample dataset in order to help our decision. The set of inputs that provide the best statistical performance for each network in the test sub-period are finally retained. In this case, those sets of inputs are autoregressive terms of UNEMP[2] and are presented in Table 2.

[1] The US unemployment rate or civilian unemployment rate represents the number of unemployed as a percentage of the labour force. Labour force data are restricted to people 16 years of age and older, who currently reside in one of the 50 states or the District of Columbia, who do not reside in institutions (e.g., penal and mental facilities, homes for the aged) and who are not on active duty in the Armed Forces. This is the definition provided by FRED.

[2] We also explored autoregressive terms of other US macroeconomic indicators (e.g. the consumer price index, the industrial production index, M1 money stock) as potential inputs. However, the set of inputs presented in Table 2 gave our NNs the best statistical performance in the test sub-period during our sensitivity analysis.

Table 2 Neural networks' inputs

MLP	RNN	PSN	RBFNN
UNEMP (1)*	UNEMP (1)	UNEMP (1)	UNEMP (2)
UNEMP (2)	UNEMP (3)	UNEMP (2)	UNEMP (3)
UNEMP (4)	UNEMP (4)	UNEMP (3)	UNEMP (4)
UNEMP (5)	UNEMP (6)	UNEMP (6)	UNEMP (7)
UNEMP (7)	UNEMP (7)	UNEMP (8)	UNEMP (8)
UNEMP (10)	UNEMP (9)	UNEMP (10)	UNEMP (9)
UNEMP (11)	UNEMP (11)	–	UNEMP (11)
UNEMP (12)	–	–	UNEMP (12)

* UNEMP (1) is the first autoregressive term of the UNEMP series

3 Forecasting Models

3.1 Auto-regressive Moving Average Model (ARMA)

The ARMA model is used to benchmark the efficiency of the NNs' statistical performance. Using as a guide the correlogram and the information criteria in the in-sample subset, we have chosen a restricted ARMA (7, 7) model, where all the coefficients are significant at the 95 % confidence interval. The selected ARMA model is presented in equation (1) below:

$$\begin{aligned}\hat{Y}_t &= 0.03 + 1.025Y_{t-1} - 0.293Y_{t-2} + 0.511Y_{t-4} - 0.321Y_{t-7} - 1.006\varepsilon_{t-1} \\ &\quad +0.463\varepsilon_{t-2} - 0.545\varepsilon_{t-4} - 0.211\varepsilon_{t-7} \end{aligned} \tag{1}$$

where $\hat{Y}_t$ is the forecasted monthly change of the US unemployment rate.

3.2 Smooth Transition Autoregressive Model (STAR)

STARs initially proposed by Chan and Tong (1986) are extensions of the traditional autoregressive models (ARs). The STAR combines two AR models with a function that defines the degree of non-linearity (smooth transition function). The general two-regime STAR specification is the following:

$$\hat{Y}_t = \Phi_1' \mathrm{X}_t (1 - F(z_t, \zeta, \lambda)) + \Phi_2' \mathrm{X}_t F(z_t, \zeta, \lambda) + u_t \tag{2}$$

where:

- $\hat{Y}_t$ the forecasted value at time t
- $\Phi_i = (\tilde{\varphi}_{i,0}, \tilde{\varphi}_{i,1}, \ldots \tilde{\varphi}_{i,p}),\ i = 1, 2$ and $\tilde{\varphi}_{i,0}, \tilde{\varphi}_{i,1}, \ldots \tilde{\varphi}_{i,p}$ the regression coefficients of the two AR models
- $\mathrm{X}_t = (1, \tilde{\chi}_t')'$ with $\tilde{\chi}_t' = (Y_{t-1}, \ldots, Y_{t-p})$
- $0 \leq F(z_t, \zeta, \lambda) \leq 1$ the smooth transition function
- $z_t = Y_{t-d}, d > 0$ the lagged endogenous transition variable

- ζ the parameter that defines the smoothness of the transition between the two regimes
- λ the threshold parameter
- u_t the error term

In this paper we follow the steps of Lin and Teräsvirta (1994) in order to determine when the series is best modeled as a Logistic STAR or an Exponential STAR process. In our case, the series is modeled as an Exponential one.

3.3 Neural Networks (NNs)

3.3.1 NN Benchmarks

The use of NNs in financial and macroeconomic forecasting is not new, since researchers use them to identify patterns and exploit their adaptive nature in relevant time series (Hiemstra 1996; Moshiri et al. 1999; Zhang and Qi 2005). In this study, three NNs architectures, namely the MLP, RNN and the PSN are applied to the task of forecasting US unemployment rate and act as NN benchmarks to the RBFNN.

These three architectures have at least three layers. The first layer is called the input layer (the number of its nodes corresponds to the number of explanatory variables). The last layer is called the output layer (the number of its nodes corresponds to the number of response variables). An intermediary layer of nodes, the hidden layer, separates the input from the output layer. Its number of nodes defines the amount of complexity the model is capable of fitting. In addition, the input and hidden layer 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 to all the other nodes of the next layer. 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 (Shapiro 2000). The iteration length is optimised by maximising a fitness function in the test dataset.

Unlike MLPs, RNNs have an activation feedback which embodies short-term memory. In other words, the RNN architecture can provide more accurate outputs because the inputs are (potentially) taken from all previous values. Tenti (1996) reports that they need more connections and memory than standard back-propagation networks, but they can yield better results in comparison with simple MLPs due to the additional memory inputs. The PSN model was firstly introduced by Shin and Ghosh (1991). They are a class of feed-forward fully connected higher order NNs, which require less number of weights and processing units for their training. Their main advantage is that they combine the fast learning property of single layer networks with the powerful mapping capability of higher order NNs, while avoiding the combinatorial increase in the required number of weights. The order of the network in the context of PSNs is represented by the number of hidden nodes. In a PSN the weights from the hidden to the output layer are fixed to one and only the weights from the input to the hidden layer are adjusted, something that greatly reduces the training time. The activation function of

the nodes in the hidden layer is the summing function, while the activation function of the output layer is a sigmoid one. For more information on MLP, RNN and PSN architectures see Zhang et al. (1998) and Sermpinis et al. (2012). The summary of the structure and the training characteristics of those networks are presented in the Appendix 1.

3.3.2 Radial Basis Function Neural Networks (RBFNN)

Initially proposed by Broomhead and Lowe (1988), the RBFNNs are feed-forward NNs. Unlike MLP, RNN and PSN, the hidden layer of the RBFNN uses a radial basis function. RBFNNs require less training time, but they can achieve higher levels of accuracy than traditional feed-forward NNs. This is achieved through the superposition of non-orthogonal, radially symmetric functions. Figure 2 shows the general structure of a RBFNN.

In order to define the Gaussian function, we need the two parameters C_i and σ_i. The first one corresponds to the vector indicating the center of the function, while the second one its width. These two parameters along with the adjustable weights are optimized through the learning phase of the training of the RBFNN. Given the target value y_t and the number of iterations T, the error function to be minimized is:

$$E\left(C, \sigma, w_t\right) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \tilde{y}_t\left(w_t, C, \sigma\right)\right)^2 \tag{3}$$

The training characteristics of RBFNN are also presented in Appendix 1.

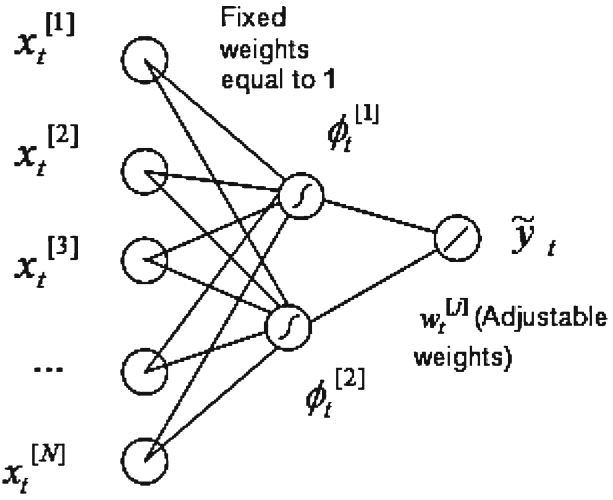

Fig. 2 A RBF neural network with N inputs and two hidden nodes, where $x_t \quad (n = 1, 2, \ldots, N+1)$ are the inputs; $\tilde{y}_t$ is the output; $w_t^{[j]}$ (j$=$ 1, 2) are the adjustable weights; ⊘ is the Gaussian function $\phi^{[i]}(x_t) = e^{-\frac{\|x_t - C_i\|^2}{2\sigma_i^2}}$; ⊘ is the linear output function $F\left(\phi\right) = \sum_i \phi^{[i]}$

4 Forecast Combination Techniques

All the forecast combination techniques implemented in this paper are presented in this section. The traditional models of ARMA and STAR present a considerably worse statistical performance than their NNs' counterparts both in-sample and out-of-sample. Therefore, we decided to exclude them from our forecast combination procedures.

4.1 Simple Average

As a benchmark for the other three, more sophisticated, forecast combination methods, we use a simple average of the four individual forecasts of MLP, RNN, PSN and RBFNN. Thus, given the forecasts f^t_{MLP}, f^t_{RNN}, f^t_{PSN}, f^t_{RBFNN} the combination forecast at time t is calculated as follows:

$$f^t_{c_{NNs}} = \left(f^t_{MLP} + f^t_{RNN} + f^t_{PSN} + f^t_{RBFNN}\right)/4 \tag{4}$$

4.2 Least Absolute Shrinkage and Selection Operator (LASSO)

The LASSO method is a class of shrinkage regressions, which minimizes the residual squared error by adding a coefficient constraint (Sundberg 2006). This is a similar approach to ridge regression (Chan et al. 1999). According to Hastie et al. (2009), though, LASSO should be selected when the used sample consists off few variables with medium/large effect, as in our exercise. Given the following vectors of independent and dependent variables:

$$\begin{pmatrix} X_1^T \\ \vdots \\ X_N^T \end{pmatrix} = \begin{pmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NN} \end{pmatrix}, \quad Y = (y_1, \ldots, y_N)^T \tag{5}$$

and the training data $\{(\boldsymbol{X}_1, y_1), \ldots, (\boldsymbol{X}_N, y_N)\}$, the LASSO coefficients are estimated based on the following argument:

$$\hat{\beta}_{lasso} = \arg\min_{\beta} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{d} \beta_i x_{ij} \right)^2 \right\} \quad \text{subject to} \quad \sum_{j=1}^{d} |\beta_j| \leq k, k > 0 \tag{6}$$

The parameter k is called 'tuning parameter' and controls the amount of shrinkage applied to the coefficients (Tibshirani 2011). For more details on the mathematical specifications of LASSO see Wang et al. (2007).

In this study, a sensitivity analysis is carried out for selecting the optimal value of k based on the in-sample period. Therefore our final constraint is:

$$|\beta_{MLP}| + |\beta_{RNN}| + |\beta_{PSN}| + |\beta_{RBFNN}| \leq 17.2 \tag{7}$$

Subject to the above coefficient constraint, the final LASSO forecast combinations are given by the equation:

$$f^t_{c_{NNs}} = 0.86 f^t_{MLP} + 2.68 f^t_{RNN} + 4.44 f^t_{PSN} + 7.67 f^t_{RBFNN} + \varepsilon_t \tag{8}$$

The use of the constraint creates a penalization balance on each estimate and leads some coefficients to zero or close to zero. In that way, the result is more adaptive than a simple regression.

4.3 Kalman Filter

Kalman filter is an efficient recursive filter that estimates the state of a dynamic system from a series of incomplete and noisy measurements (Wells 1996). In this application, we suggest the use of Kalman Filter as a time-varying coefficient combination forecast. In order to define the recursive algorithm, we need a measurement equation to combine the forecasts and a state equation to update the weights of the combination at each step. Those equations are given below.

Measurement equation:

$$f^t_{c_{NNs}} = \sum_{i=1}^{4} a^t_i f^t_i + \varepsilon_t, \quad \varepsilon_t \sim NID\left(0, \sigma^2_\varepsilon\right) \tag{9}$$

State equation:

$$a^t_i = a^{t-1}_i + n_t, \quad n_t \sim NID(0, \sigma^2_n) \tag{10}$$

where:

- $f^t_{c_{NNs}}$ is the dependent variable (combination forecast) at time t
- f^t_i $(i = 1, 2, 3, 4)$ are the independent variables (individual forecasts) at time t
- a^t_i $(i = 1, 2, 3, 4)$ are the time-varying coefficients at time t for each NN
- ε_t, n_t are the uncorrelated error terms (noise)

The alphas are calculated by a simple random walk and we initialized $\varepsilon_1 = 0$. Following Hatemi-J and Roca (2006), our Kalman filter model has as a final state the following:

$$f^t_{c_{NNs}} = 10.32 f^t_{MLP} + 12.18 f^t_{RNN} + 31.34 f^t_{PSN} + 52.21 f^t_{RBFNN} + \varepsilon_t \tag{11}$$

From the above equation, it is obvious that the Kalman filtering process favors the RBFNN model, which is the model that performs best individually.

4.4 Support Vector Regression (SVR)

Vapnik (1995) established support vector regression (SVR) as a robust technique for constructing data-driven and non-linear empirical regression models. SVRs are

commonly used in financial and macroeconomic applications (Ince and Trafalis 2008; Reboredo et al. 2012; Xu et al. 2013). Their advantages, such as providing global and unique solutions, not suffering from local minima and balancing model accuracy and model complexity are well documented in literature (Suykens et al. 2002; Lu et al. 2009).

A simple SVR function can be specified as:

$$f(x) = w^T \varphi(x) + b \tag{12}$$

where w and b are the regression parameter vectors of the function and $\varphi(x)$ is the non-linear function that maps the input data vector x into a feature space where the training data exhibit linearity (see Fig. 3c).

The ε-sensitive loss L_ε function finds the predicted points that lie within the tube created by two slack variables ξ_i, ξ_i^* (see Fig. 3a, b):

$$L_\varepsilon(x_i) = \begin{cases} 0 & \text{if } |y_i - f(x_i)| \le \varepsilon \\ |y_i - f(x_i)| - \varepsilon & \text{if other} \end{cases}, \varepsilon \ge 0 \tag{13}$$

L_ε finds the predicted values that have at most ε deviations from the actual obtained values y_i. Therefore, ε quantifies the degree of model noise insensitivity. The goal is to solve the following argument:

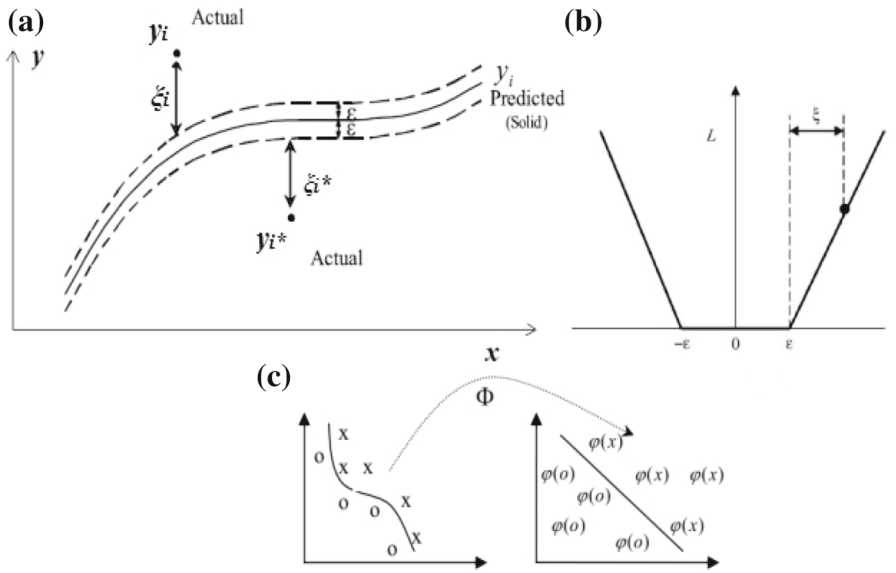

Fig. 3 **a** The $f(x)$ curve of SVR and the ε-tube, **b** plot of the ε-sensitive loss function and **c** mapping procedure by φ(x)

$$\text{Minimize } C\sum_{i=1}^{n}\left(\xi_i+\xi_i^*\right)$$
$$+\frac{1}{2}\|w\|^2 \text{ subject to } \left\{\begin{array}{l}\xi_i\geq 0\\ \xi_i^*\geq 0\\ C>0\end{array}\right\} \text{ and } \left\{\begin{array}{l}y_i-w^T\varphi(x_i)-b\leq+\varepsilon+\xi_i\\ w^T\varphi(x_i)+b-y_i\leq+\varepsilon+\xi_i^*\end{array}\right\} \quad (14)$$

Equation (14) attempts to minimize the sum of the norm term $\|w\|^2$ and the term $\left\{\sum_{i=1}^{n}\left(\xi_i+\xi_i^*\right)\right\}$. The first term characterizes the complexity of the model, while the second is the training error, as specified by the slack variables. The parameter C satisfies the need to trade model complexity for training error and vice versa (Cherkassky and Ma 2004). The above solution is based on the introduction of two Lagrange multipliers a_i, a_i^* and mapping with a kernel function $K(x_i, x)$:

$$f(x)=\sum_{\iota=1}^{n}(a_i-a_i^*)K(x_i,x)+b \text{ where } 0\leq a_i, a_i^*\leq C \quad (15)$$

The application of the kernel function transforms the original input space into one with more dimensions, where a linear decision border can be identified. The original input space consists of four vectors. These vectors correspond to the individual forecasts of MLP, RNN, PSN and RBFNN derived from the empirical simulation of Sect. 3.3. The extended mathematical explanation of this solution can be found in Vapnik (1995).

Choosing the ε parameter is indeed a challenging task, because it depends on the noise of the training datasets. In practice, there are no optimal solutions to this problem. The majority of the researchers adopt the cross-validation approach (Cao et al. 2003; Duan et al. 2003). Hence, we apply the same procedure to our study. Another challenge is the selection of the kernel function. RBF kernels are popular in similar SVR applications, because they efficiently overcome overfitting and seem to excel in directional accuracy (Kim and Sohn 2010; Yu and Yao 2013). The four NN forecasts are used as inputs for a RBF ε-SVR simulation. The RBF kernel is specified as:

$$K(x_i,x)=\exp(-\gamma\|x_i-x\|^2),\ \gamma>0 \quad (16)$$

From Eqs. (14) and (16) it is obvious that we need to determine two kernel-independent parameters (ε and C) and the RBF parameter (γ). This is achieved by a five-fold cross validation in our in-sample dataset, following Duan et al. (2003). The final single SVR forecast combination is calculated with the following optimized set of parameters $\varepsilon=0.15$, $\gamma=4.18$ and $C=94.8$. The out-of-sample observations of UNEMP time series are not used at all for tuning our SVR model.

5 Empirical Results

As it is standard in literature, in order to evaluate statistically our forecasts, the RMSE, the MAE, the MAPE and the Theil-U statistics are computed. For all four of the error statistics retained the lower the output, the better the forecasting accuracy of the model

Table 3 Summary of the in-sample statistical performance

	ARMA	STAR	MLP	RNN	PSN	RBFNN	Simple average	LASSO	Kalman filter	SVR
MAE	1.9941	0.0094	0.0078	0.0077	0.0073	0.0068	0.0066	0.0065	0.0062	0.0058
MAPE (%)	65.25	60.27	52.78	50.17	47.73	44.38	43.02	41.58	40.78	38.52
RMSE	2.5903	1.2105	1.0671	0.9572	0.9045	0.8714	0.8625	0.8556	0.8434	0.8216
Theil-U	0.6717	0.6447	0.6142	0.5827	0.5325	0.5114	0.5021	0.4903	0.4717	0.4479

Table 4 Summary of the out-of-sample statistical performance

	ARMA	STAR	MLP	RNN	PSN	RBFNN	Simple average	LASSO	Kalman filter	SVR
MAE	0.0332	0.0099	0.0082	0.0081	0.0079	0.0075	0.0072	0.0071	0.0068	0.0061
MAPE (%)	67.45	64.27	53.17	51.97	49.38	47.41	47.02	45.14	44.32	40.12
RMSE	2.4043	1.2412	1.1657	0.9954	0.9527	0.9114	0.8915	0.8706	0.8519	0.8327
Theil-U	0.5922	0.6773	0.5954	0.5891	0.5618	0.5331	0.5241	0.5196	0.5023	0.4713

concerned. The mathematical formulas of these statistics are given in Appendix 2. In Table 3 we present the statistical performance of all our models in the in-sample period.

From Table 3 it is obvious that from our individual forecasts, the RBFNN statistically outperforms all other models. All forecast combination techniques improve the forecasting accuracy. SVR is the superior model regarding all four statistical criteria. It would be interesting to see if the in-sample performance coincides with the out-of-sample one. Table 4 summarizes the statistical performance of our models in the out-of-sample period.

The results of Table 4 suggest that the statistical performance of the models in the out-of-sample period is consistent with the in-sample one and their ranking remains the same. All NN models outperform the traditional ARMA and STAR models. In addition, the RBFNN outperforms significantly the MLP and RNN in terms of statistical accuracy. The second best individual performance is presented by PSN, which remains less accurate than RBFNN. This means that the RBFNN manages to overcome the statistical performance of the traditional MLP and RNN, but also of PSN which in general has fast learning and powerful mapping abilities. The forecast combination techniques are all improving the accuracy of the individual performances. Even the least sophisticated simple average presents lower values in all four statistics in comparison with the best individual model, the RBFNN. The LASSO achieves higher forecast accuracy than simple average, but it does not perform better from the Kalman filter and SVR. In this forecasting competition, SVR remains the superior model 'beating' Kalman Filter in every statistic in the out-of-sample period.

The statistical superiority of our best proposed architecture, namely the SVR, is confirmed by the modified Diebold–Mariano (MDM) statistic as proposed by

Table 5 Summary results of modified Diebold–Mariano statistics for MSE and MAE loss functions

	ARMA	STAR	MLP	RNN	PSN	RBFNN	Simple average	LASSO	Kalman filter
MD_1	−10.18	−9.91	−9.31	−9.17	−8.13	−7.83	−6.23	−5.78	−5.53
MD_2	−14.06	−12.42	−10.58	−9.96	−9.81	−8.63	−7.11	−6.92	−6.83
MDM_1	−10.24	−9.97	−9.37	−9.23	−8.18	−7.88	−6.27	−5.81	−5.56
MDM_2	−14.14	−12.49	−10.64	−10.02	−9.87	−8.68	−7.15	−6.96	−6.87

Note MD_1, $MDM1$, and MD_2, MDM_2 are the statistics computed for the MSE and MAE loss function respectively

Harvey et al. (1997). The null hypothesis of the test is the equivalence in forecasting accuracy between couples of forecasting models. The MDM test[3] is an extension of the Diebold–Mariano (Diebold and Mariano 1995) test and its statistic (DM) is presented below:

$$MDM = T^{-1/2}\left[T + 1 - 2k + T^{-1}k\,(k-1)\right]^{1/2} DM \tag{17}$$

where T the number of the out-of-sample observations and k the number of the step-ahead forecasts. In our case we apply the MDM test to couples of forecasts (SVR vs. another forecasting model). A negative realization of the MDM test statistic indicates that the first forecast (SVR) is more accurate than the second forecast. The lower the negative value, the more accurate are the SVR forecasts. The use of MDM test is common practice, because it assesses the significance of observed differences between the performances of two forecasts (Barhoumi et al. 2010). The statistic is measured in the out-of-sample period for the MSE and MAE loss functions. Table 5 presents the values of the DM and MDM statistics for all the cases, comparing the SVR with its benchmarks.

The table shows that the MDM null hypothesis of equal forecasting accuracy is rejected for all comparisons and for both loss functions at the 1 % confidence interval. The statistical superiority of the SVR forecasts is confirmed as the realizations of the MDM statistic are negative for both loss functions.

The results of this section support the idea of combining NN unemployment forecasts, since the Simple Average, LASSO, Kalman Filter and SVR present improve the statistical accuracy both in the in-sample and out-of-sample period. The fact that the in-sample statistical ranking of our NNs is consistent with the out-of-sample one proves that the training of our models is done effectively. The coefficient adaptivity of LASSO does not provide with such forecasting power to outperform the time-varying Kalman Filter process. Nonetheless, it is superior from all NNs and the less sophisticated Simple Average. SVR also confirms its forecasting superiority over all individual architectures and combining techniques. Finally, the fact that SVR is found always more accurate than Kalman Filters suggests that the adaptive trade-off between model complexity and training error of this technique seems more effective than the recursive ability of Kalman Filter to estimate the state of our process.

[3] The MDM test follows the student distribution with T-1 degrees of freedom.

In general, the growing literature of NNs and more specifically of the utility of RBFNNs in similar forecasting exercises is extended. The improved statistical results of the Kalman Filter and SVR are supporting the evidence of previous studies that report the benefits of combining forecasts. In summary, the success of the non-linear and time-varying combination methods of this study indicates a need to experiment with more complex combination techniques and beyond traditional linear equivalents.

6 Concluding Remarks

This motivation of this study is to investigate the efficiency of the RBFNN in forecasting the US unemployment and explores the utility of Kalman Filter and SVR as forecast combination techniques. In terms of our RBFNN, an ARMA, a STAR and three different NNs, namely a MLP, RNN and PSN are used as benchmarks. Our forecast combination methods are benchmarked with a Simple Average and a LASSO. The statistical performance of our models is estimated throughout the period of 1972–2012, using the last 7 years for out-of-sample testing.

The results show that the RBFNN statistically outperforms all models' individual performances. Even PSN which embodies fast learning abilities and powerful mapping capabilities cannot reach the RBFNN's levels of accuracy. The forecast combinations are successful, since both Kalman Filter and SVR techniques improve the statistical accuracy in comparison to the Simple Average and LASSO benchmarks. The Simple Average presents better results than all individual models, but it cannot outperform any of the more sophisticated combination methods. Finally, SVR is found to be the superior model of the forecasting competition, which is further confirmed by the modified Diebold–Mariano test.

The idea of combining NN unemployment forecasts is promising, since the Simple Average, LASSO, Kalman Filter and SVR present improved statistical accuracy both in the in-sample and out-of-sample period. SVR is found always more accurate than Kalman Filter. This indicates that the adaptive trade-off between model complexity and training error of SVR is more effective from the recursive ability of Kalman Filter to estimate the state of our process. The general statistical performance of SVR allows us to conclude that it can be considered as an optimal forecast combination for the models and time series under study. The results are in line with the relevant literature which suggests that adaptive, time-varying, nonlinear models can be used to model macroeconomic series. Finally, The SVR and Kalman Filter forecast combinations could be further extended. A potential extension could be the use of individual forecasts from a larger pool of relevant models or the use of several macroeconomic indicators and different forecast horizons.

Appendix 1: NNs' Structure and Training Characteristics

This appendix section briefly describes the structure of the three traditional NNs used to benchmark the RBFNN. It also includes a summary of the training characteristics of all four NNs.

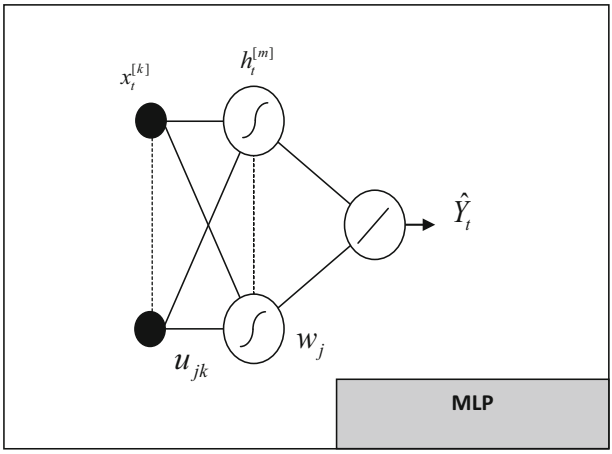

Fig. 4 A single output, fully connected MLP model (bias nodes are not shown for simplicity) where $x_t^{[n]}$ $(n = 1, 2, \ldots, k+1)$ are the inputs; (including the input bias node) at time t; $h_t^{[m]}$ $(m = 1, 2, \ldots, j+1)$ are the hidden nodes outputs; $\hat{Y}_t$ is the MLP output (target value); u_{jk}, w_j are the network weights; $\oslash$ is the transfer sigmoid function $S(x) = \frac{1}{1+e^{-x}}$; $\oslash$ is a linear function $F(x) = \sum_i x_i$

Firstly, a typical MLP model is shown in Fig. 4.

The error function to be minimized is:

$$E\left(u_{jk}, w_j\right) = \frac{1}{T}\sum_{t=1}^{T}\left(Y_t - \hat{Y}_t\left(u_{jk}, w_j\right)\right)^2 \tag{18}$$

Secondly, the simple architecture of an RNN is presented in Fig. 5.

The error function to be minimized is:

$$E\left(d_t, w_t\right) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \tilde{y}_t\left(d_t, w_t\right)\right)^2 \tag{19}$$

Thirdly, Fig. 6 describes the PSN architecture.

The error function minimized in this case:

$$E\left(c, w_j\right) = \frac{1}{T}\sum_{t=1}^{T}\left(y_t - \tilde{y}_t\left(w_k, c\right)\right)^2 \tag{20}$$

Finally, Table 6 summarizes the training characteristics of the four NN architectures used in this forecasting competition.

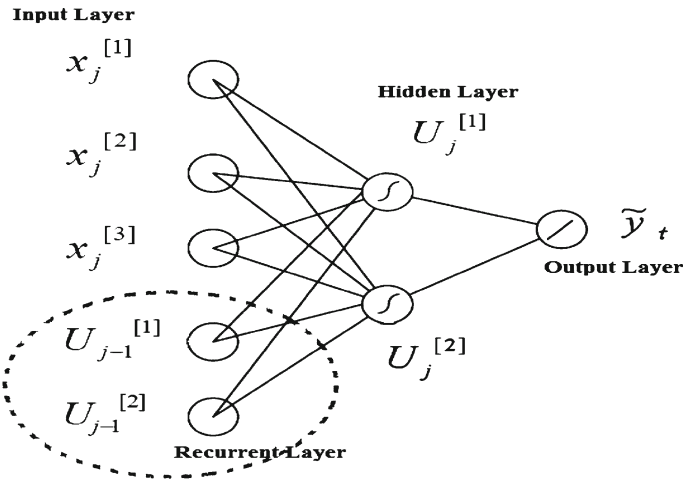

Fig. 5 RNN with two nodes in the hidden layer where $x_t^{[n]}(n = 1, 2, \ldots, k+1), u_t^{[1]}, u_t^{[2]}$ are the RNN inputs at time t (including bias node); $\tilde{y}_t$is the output of the RNN; $d_t^{[f]}(f = 1, 2)$ and $w_t^{[n]}(n = 1, 2, \ldots, k+1)$ are the weights of the network; $U_t^{[f]}, f = (1, 2)$ is the output of the hidden nodes at time t; ∅ is the transfer sigmoid function $S(x) = \frac{1}{1+e^{-x}}$; ∅ is a linear function $F(x) = \sum_i x_i$

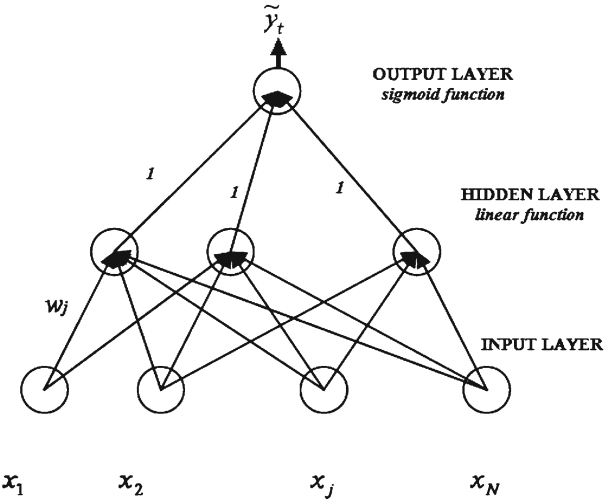

Fig. 6 A PSN with one output layer, where $x_t(n = 1, 2, \ldots, k+1)$ are the model inputs; y_t, $\tilde{y}_t$ are the PSN input and output respectively; $w_j(j = 1, 2 \ldots, k)$ are the adjustable weights (k is the desired order of the network); the hidden layer activation function: $h(x) = \sum_i x_i$; the output sigmoid activation function (c the adjustable term) $\sigma(x) = \frac{1}{1+e^{-xc}}$

Table 6 The NNs training characteristics

Parameters	MLP	RNN	PSN	RBFNN
Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent
Learning rate	0.005	0.003	0.002	0.003
Momentum	0.007	0.005	0.006	0.005
Iteration steps	60000	50000	75000	45000
Initialisation of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)
Input nodes	8	7	6	8
Hidden nodes	6	6	5	4
Output node	1	1	1	1

Appendix 2: Statistical Performance Measures

The statistical performance measures are calculated as shown in Table 7.

Table 7 Statistical Performance Measures and Calculation

Statistical Performance Measures	Description
Mean absolute error	$MAE = (\frac{1}{n}) \sum_{\tau=t+1}^{t+n} \left\| \hat{Y}_\tau - Y_\tau \right\|$ *with* Y_τ *being the actual value and* $\hat{Y}_\tau$ *the forecasted value*
Mean absolute percentage error	$MAPE = \frac{1}{n} \sum_{\tau=t+1}^{t+n} \left\| \frac{Y_\tau - \hat{Y}_\tau}{Y_\tau} \right\|$
Root mean squared error	$RMSE = \sqrt{\frac{1}{n} \sum_{\tau=t+1}^{t+n} (\hat{Y}_\tau - Y_\tau)^2}$
Theil-*U*	$Theil\text{-}U = \frac{\sqrt{\left(\frac{1}{n} \sum_{\tau=t+1}^{t+n} (\hat{Y}_\tau - Y_\tau)^2\right)}}{\sqrt{\frac{1}{n} \sum_{\tau=t+1}^{t+n} \hat{Y}_\tau^2} + \sqrt{\frac{1}{n} \sum_{\tau=t+1}^{t+n} Y_\tau^2}}$

References

Barhoumi, K., Darné, O., Ferrara, L., et al. (2010). Are disaggregate data useful for factor analysis in forecasting French GDP? *Journal of Forecasting*, *29*(1–2), 132–144.

Bates, J. M., & Granger, C. W. J. (1969). The combination of forecasts. *Operational Research Society*, *20*(4), 451–468.

Broomhead, D. S., & Lowe, D. (1988). Multivariable functional interpolation and adaptive networks. *Complex Systems*, *2*, 321–355.

Cao, L. J., Chua, K. S., Guan, L. K., et al. (2003). C-ascending support vector machines for financial time series forecasting. In: Computational Intelligence for Financial Engineering Proceedings (pp. 317–323).

Chan, K. S., & Tong, H. (1986). On estimating thresholds in autoregressive models. *Journal of Time Series Analysis*, *7*(3), 178–190.
Chan, Y. L., Stock, J. H., & Watson, M. W. (1999). A dynamic factor model framework for forecast combination. *Spanish Economic Review*, *1*(2), 91–121.
Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126.
Deutsch, M., Granger, C. W. J., Teräsvirta, T., et al. (1994). The combination of forecasts using changing weights. *International Journal of Forecasting*, *10*(1), 47–57.
Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, *13*(1), 253–263.
Duan, K., Keerthi, S. S., Poo, A. N., et al. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, *51*, 41–59.
Fernandes, M., Medeiros, M. C., & Scharth, M. (2014). Modeling and predicting the CBOE market volatility index. *Journal of Banking & Finance*, *40*, 1–10.
Harvey, D., Leybourne, S., Newbold, P., et al. (1997). Testing the equality of prediction mean squared errors. *International Journal of Forecasting*, *13*(2), 281–291.
Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (2nd ed.). New York: Springer.
Hatemi-J, A., & Roca, E. (2006). Calculating the optimal hedge ratio: constant, time varying and the Kalman filter approach. *Applied Economics Letters*, *13*(5), 293–299.
Hiemstra, Y. (1996). Linear regression versus back propagation networks to predict quarterly stock market excess returns. *Computational Economics*, *9*(1), 67–76.
Huang, S. C., Wang, N. Y., Li, T. Y., Lee, Y. C., Chang, L. F., & Pan, T. H. (2013). Financial forecasting by modified Kalman filters and Kernel machines. *Journal of Statistics and Management Systems*, *16*(2–03), 163–176.
Ince, H., & Trafalis, T. B. (2008). Short term forecasting with support vector machines and application to stock price prediction. *International Journal of General Systems*, *37*(6), 677–687.
Kapetanios, G., Labhard, V., & Price, S. (2008). Forecast combination and the Bank of England's suite of statistical forecasting models. *Economic Modelling*, *25*(4), 772–792.
Kim, H. S., & Sohn, S. Y. (2010). Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*, *201*(3), 838–846.
Koop, G., & Potter, S. M. (1999). Dynamic asymmetries in U.S. unemployment. *Journal of Business & Economic Statistics*, *17*(3), 298–312.
Liang, F. (2005). Bayesian neural networks for nonlinear time series forecasting. *Statistics and Computing*, *15*(1), 13–29.
Lin, C. J., & Teräsvirta, T. (1994). Testing the constancy of regression parameters against continuous structural changes. *Journal of Econometrics*, *62*(2), 211–228.
Lu, C. J., Lee, T. S., Chiu, C. C., et al. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*(2), 115–125.
Milas, C., & Rothman, P. (2008). Out-of-sample forecasting of unemployment rates with pooled STVECM forecasts. *International Journal of Forecasting*, *24*(1), 101–121.
Montgomery, A. L., Zarnowitz, V., Tsay, R. S., Tiao, G. C., et al. (1998). Forecasting the U.S. unemployment rate. *Journal of the American Statistical Association*, *93*(442), 478–493.
Moshiri, S., Cameron, N. E., Scuse, D., et al. (1999). Static, dynamic, and hybrid neural networks in forecasting inflation. *Computational Economics*, *14*(3), 219–235.
Moshiri, S., & Brown, L. (2004). Unemployment variation over the business cycles: A comparison of forecasting models. *Journal of Forecasting*, *23*(7), 497–511.
Newbold, P., & Granger, C. W. J. (1974). Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society*, *137*(2), 131–165.
Olmedo, E. (2014). Forecasting Spanish unemployment using near neighbor and neural net techniques. *Computational Economics*, *43*(2), 183–197.
Özkan, F. (2013). Comparing the forecasting performance of neural network and purchasing power parity: The case of Turkey. *Economic Modelling*, *31*, 752–758.
Reboredo, J. C., Matías, J. M., Garcia-Rubio, R., et al. (2012). Nonlinearity in forecasting of high-frequency stock returns. *Computational Economics*, *40*(3), 245–264.
Rothman, P. (1998). Forecasting asymmetric unemployment rates. *The Review of Economics and Statistics*, *80*(1), 164–168.

Sermpinis, G., Dunis, C., Laws, J., Stasinakis, C., et al. (2012). Forecasting and trading the EUR/USD exchange rate with stochastic neural network combination and time-varying leverage. *Decision Support Systems*, *54*(1), 316–329.

Shapiro, A. F. (2000). A Hitchhiker's guide to the techniques of adaptive nonlinear models. *Insurance: Mathematics and Economics*, *26*(2–3), 119–132.

Shin, Y., & Ghosh, J. (1991). The pi–sigma network: An efficient higher-order neural networks for pattern classification and function approximation. *Proceedings of International Joint Conference of Neural Networks*, *1*, 13–18.

Skalin, J., & Teräsvirta, T. (2002). Modeling asymmetries and moving equilibria in unemployment rates. *Macroeconomic Dynamics*, *6*(2), 202–241.

Sundberg, R. (2006). Shrinkage regression. In A. H. El-Shaarawi & W. W. Piergosh (Eds.), *Encyclopedia of environmetrics* (Vol. 4, pp. 1994–1998). New York: Wiley.

Suykens, J. A. K., Brabanter, J. D., Lukas, L., Vandewalle, L., et al. (2002). Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing*, *48*(1–4), 85–105.

Swanson, N. R., & Zeng, T. (2001). Choosing among competing econometric forecasts: Regression-based forecast combination using model selection. *Journal of Forecasting*, *20*(6), 425–440.

Szpiro, G. G. (1997). A search for hidden relationships: Data mining with genetic algorithms. *Computational Economics*, *10*(3), 267–277.

Tenti, P. (1996). Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, *10*(6), 567–581.

Teräsvirta, T., Dijk, K. V., & Medeiros, M. C. (2005). Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting*, *21*(4), 755–774.

Tibshirani, R. (2011). Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *73*(3), 273–282.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin: Springer.

Vasnev, A., Skirtun, M., Pauwels, L., et al. (2013). Forecasting monetary policy decisions in Australia: A forecast combinations approach. *Journal of Forecasting*, *32*(2), 151–166.

Wang, H., Li, G., & Jiang, G. (2007). Robust regression shrinkage and consistent variable selection through the LAD-Lasso. *Journal of Business and Economic Statistics*, *25*(3), 347–355.

Wells, C. (1996). *The Kalman filter in finance*. Dordrecht: Kluwer Academic.

Xu, W., Li, Z., Cheng, C., Zheng, T., et al. (2013). Data mining for unemployment rate prediction using search engine query data. *Service Oriented Computing and Applications*, *7*(1), 33–42.

Yu, L., & Yao, X. (2013). A total least squares proximal support vector classifier for credit risk evaluation. *Soft Computing*, *17*(4), 643–650.

Zhang, G., Patuwo, B. E., Hu, M. Y., et al. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, *14*(1), 35–62.

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, *160*(2), 501–514.