

Stochastics and Statistics

Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms, Support vector regression forecast combinations

CrossMark

Georgios Sermpinis^a, Charalampos Stasinakis^a, Konstantinos Theo“latos^b,
Andreas Karathanasopoulos^c

^a University of Glasgow Business School, University of Glasgow, Adam Smith Building, Glasgow G12 8QQ, UK

^b Pattern Recognition Laboratory, Department of Computer Engineering & Informatics, University of Patras, 26500 Patras, Greece

^c Andreas Karathanasopoulos Royal Docklands Business School, University of East London, UK

article info

Article history:

Received 19 May 2014

Accepted 19 June 2015

Available online 27 June 2015

Keywords:

Genetic algorithms

Support vector regression

Feature subset

Forecast combinations

Exchange rates

abstract

The motivation of this paper is to introduce a hybrid Rolling Genetic Algorithm-Support Vector Regression (RG-SVR) model for optimal parameter selection and feature subset combination. The algorithm is applied to the task of forecasting and trading the EUR/USD, EUR/GBP and EUR/JPY exchange rates. The proposed methodology genetically searches over a feature space (pool of individual forecasts) and then combines the optimal feature subsets (SVR forecast combinations) for each exchange rate. This is achieved by applying a “tness function specialized for “nancial purposes and adopting a sliding window approach. The individual forecasts are derived from several linear and non-linear models. RG-SVR is benchmarked against genetically and non-genetically optimized SVRs and SVMs models that are dominating the relevant literature, along with the robust ARBF-PSO neural network. The statistical and trading performance of all models is investigated during the period of 1999...2012. As it turns out, RG-SVR presents the best performance in terms of statistical accuracy and trading efficiency for all the exchange rates under study. This superiority confirms the success of the implemented “tness function and training procedure, while it validates the benefits of the proposed algorithm.

© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURQ) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

1. Introduction

Forecasting “nancial time series appears to be a challenging task for the scientific community because of its non-linear and non-stationary structural nature. On one hand, traditional statistical methods fail to capture this complexity, while on the other hand non-linear techniques present promising empirical evidence. However, their practical limitations and the expertise required to optimize their parameters are creating skepticism on their utility.

This study introduces a hybrid Rolling Genetic Algorithm-Support Vector Regression (RG-SVR) algorithm for optimal parameter selection and features subset combination when applied to the task of forecasting and trading the EUR/USD, EUR/GBP and EUR/JPY exchange rates. The proposed model genetically searches over a feature space (pool of individual forecasts) and then combines the optimal feature

subsets for each exchange rate. A novel “tness function specialized for “nancial purposes is used to simultaneously minimize the error of the obtained forecasts, increase the profitability of the “nal forecast combination and reduce the complexity of the algorithm. This is crucial in “nancial applications, where statistical accuracy is not always synonymous with the “nancial profitability of the deriving forecasts. The reduced complexity of the algorithm decreases the computational cost of the proposed methodology and makes it ideal for trading applications, where time efficiency is important. At the same time, it acts as a protection against overfitting and impeded generalization abilities. The model employs a sliding window training approach and is capable of capturing the time-varying relationship that dominates the “nancial trading series.

RG-SVR is benchmarked against seven models. Their statistical and trading performance is compared during the period of 1999...2012. The rationale behind the selection of the benchmarks is twofold. Firstly, Support Vector Machine (SVM) and Support Vector Regression (SVR) architectures with genetically and non-genetically optimized parameters are dominant in the relevant literature. The six most popular and promising variants are identified and included in the study (see Section 4.3). Secondly, it would be unfair to

Corresponding author. Tel.: + 447914053349.

E-mail addresses: georgios.sermpinis@glasgow.ac.uk, sermpinis@hotmail.com (G. Sermpinis), charalampos.stasinakis@glasgow.ac.uk (C. Stasinakis), theo“lk@ceid.upatras.gr (K. Theo“latos), a.karathanasopoulos@uel.ac.uk (A. Karathanasopoulos).

<http://dx.doi.org/10.1016/j.ejor.2015.06.052>

0377-2217/© 2015 Elsevier B.V. and Association of European Operational Research Societies (EURQ) within the International Federation of Operational Research Societies (IFORS). All rights reserved.

compare the hybrid RG-SVR only with hybrids of the same methodology class, especially when other proposed methodologies have shown statistical and trading superiority in a similar task. One such example is the hybrid Neural Network (NN) that combines Adaptive Radial Basis Functions with Particle Swarm Optimization (ARBF-PSO), as introduced recently by Sermpinis, Theo"latos, Karathanasopoulos, Georgopoulos, and Dunis (2013). The authors apply ARBF-PSO to the task of forecasting and trading the same exchange rates that the present study is investigating. Their results show that ARBF-PSO is outperforming several linear and non-linear models, while its structural complexity is not high. Consequently, ARBF-PSO's selection as a benchmark to this application is required and justified.

To the best of our knowledge, the proposed RG-SVR methodology has not been presented in the literature. Similar hybrid applications exist but they are either limited in classification problems (Dunis, Likothanassis, Karathanasopoulos, Sermpinis & Theo"latos, 2013; Huang & Wang, 2006; Min, Lee, & Han, 2006; Wu, Tzeng, Goo, & Fang, 2007) or the Genetic Algorithm (GA) does not extend to optimal feature subset selection (Chen & Wang, 2007; Pai, Lin, Hong, & Chen, 2006; Yuan, 2012). In addition to this fact, the RG-SVR hybrid is the first GA hybrid SVR algorithm that deploys ν -SVR models, minimizes the number of support vectors and applies a specialized loss function and a sliding window training approach. A detail comparison of the algorithm and the previous two algorithms is presented in the next section. The genetically optimized SVM model (GA-SVM) proposed by Huang and Wang (2006), Min et al. (2006), Wu et al. (2007) and Dunis et al. (2013) and the genetically optimized ε -SVR model (GA- ε -SVR) proposed by Chen and Wang (2007), Pai et al. (2006) and Yuan (2012) will act as benchmarks to the RG-SVR algorithm. Compared to non-adaptive algorithms presented in the literature, the proposed model does not require from the practitioner to follow any time-consuming optimization approach (such as cross validation or grid search) and is free from the data snooping bias (all parameters of RG-SVR are optimized in a single optimization procedure).

From the results of this analysis, it emerges that RG-SVR presents the best performance in terms of statistical accuracy and trading efficiency for all exchange rates under study. RG-SVR's trading performance and forecasting superiority not only confirms the success of the implemented fitness function, but also validates that applying GAs in this hybrid model to optimize the SVR parameters is more efficient compared to the optimization approaches (cross validation and grid search algorithms), that dominate the relevant literature.

The rest of the paper is organized as follows. Section 2 is a literature review of previous relevant research on SVMs and SVRs in forecasting. A detailed description of the study's dataset, the EUR/USD, EUR/GBP and EUR/JPY European Central Bank (ECB) fixing series, is presented in Section 3. Section 4 includes the complete description of the hybrid RG-SVR model, while the statistical and trading performance of the implemented models is presented in Sections 5 and 6 respectively. The concluding remarks are provided in Section 7. The essential theoretical background for the complete understanding of the proposed methodology is given in Appendices A...D, along with the technical characteristics of the models used in this study.

2. Literature review

SVMs were originally developed for solving classification problems in pattern recognition frameworks. The introduction of Vapnik's (1995) insensitive loss function has extended their use in non-linear regression estimation problems. SVRs's main advantage is that they provide global and unique solutions and do not suffer from multiple local minima, while they present a remarkable ability of balancing model accuracy and model complexity (Kwon & Moon, 2007; Suykens, De Brabanter, Lukas, & Vandewalle, 2002).

The literature of SVM and SVR applications is voluminous, especially when they are applied in financial tasks. This study aims

to delve deeper into their hybrid structures that are already very popular (Lo, 2000). Lee, Lin, and Wahba (2004) propose the multi-category SVM as an extension of the traditional binary SVM and apply it in two different case studies with promising results. They note that their proposed methodology can be a useful addition to the class of nonparametric multi-category classification methods. Liu and Shen (2006) advance the previous mentioned approach by presenting the multi-category ψ -learning methodology. The main advantage of their method is that the convex SVM loss function is replaced by a non-convex ψ -loss function, which leads to smaller number of support vectors and sparser solution. Martens, Baesens, van Gestel, and Vanthienen (2007) introduce two extraction techniques for SVMs and prove their utility in a series of tests. Hsu, Hsieh, Chih, and Hsu (2009) integrate SVR in a two-stage architecture for stock price prediction and present empirical evidence that shows that its forecasting performance can be significantly enhanced compared to a single SVR model. Lu, Lee, and Chiu (2009) and Yeh, Huang, and Lee (2011) propose also hybrid SVR methodologies for forecasting the TAIEX index and conclude that they perform better than simple SVR approaches and other autoregressive models. Wu and Liu (2007) introduce the robust truncated hinge loss SVM and claim that their method can overcome drawbacks of traditional SVM models, such as the outliers's sensitivity in the training sample and the large number of support vectors. Huang, Chuang, Wu, and Lai (2010) forecast the EUR/USD, GBP/USD, NZD/USD, AUD/USD, JPY/USD and RUB/USD exchange rates with a hybrid chaos-based SVR algorithm. In their application, they confirm the forecasting superiority of their proposed technique compared to chaos-based NNs and several traditional non-linear models. Lin and Pai (2010) introduce a fuzzy SVR model for forecasting indices of business cycles, Kim and Sohn (2010) forecast the credit score of small and medium enterprises with SVM, while Wu and Akbarov (2011) apply successfully weighted SVRs to the task of forecasting warranty claims. Moreover, Jiang and He (2012) propose a hybrid SVR that incorporates the Grey relational grade weighting function. When applied to financial time series forecasting, the local Grey SVR outperforms locally weighted counterparts in terms of computational speed and prediction accuracy. A hybrid architecture for computer products's sales forecasting is also introduced by Lu (2014) based on SVR and multivariate adaptive regression splines.

Most recently, Yao, Crook, and Andreeva (2015) use SVRs in the credit risk modeling framework. Specifically, the authors evaluate the predictive ability of SVR over recovery rates of defaulted corporate instruments between 1985 and 2012. The results show the superiority of the SVR techniques in forecasting these rates compared to other commonly used methods, such as linear regression, fractional response regression and the two-stage methodology. Finally, Geng, Bose, and Chen (2015) present a forecast competition of methodologies, such as NNs, SVM, decision trees and majority voting classifiers, to the task of predicting financial distress of listed Chinese companies. The empirical evidence of that study shows that NNs outperform the SVM, but they acknowledge that this is contradicting previous literature.

Similar applications to the proposed hybrid approach of this study can be found in the literature. For example Min et al. (2006) and Wu et al. (2007) use hybrid GA-SVM models in order to forecast the bankruptcy risk. In both applications, the GAs optimize the parameters of the SVM and select the financial ratios that most affect bankruptcy. Dunis et al. (2013) developed a GA-SVM algorithm and applied it to the task of trading the daily and weekly returns of the FTSE 100 and ASE 20 indices. This approach deals with financial forecasting as a classification problem and has limited applicability. In financial forecasting, though, it is crucial to obtain forecasts that predict not only the sign but also the size of the examined financial indices.

Pai et al. (2006) apply epsilon SVR with genetically optimized parameters (GA- ε -SVR) in forecasting exchange rates, while

Table 1

Theoretical comparison of RG-SVR with other existing hybrid combinations of GAs with SVM/SVR based models.

Method	SVM type	Sliding window	Parameters optimization using GA	Input subset optimization using GAs	Minimization of the support vectors
RG-SVR	v-SVR	Yes	Yes	Yes	Yes
Chen and Wang (2007) , Pai et al. (2006) and Yuan (2012)	ϵ -SVR	No	Yes	No	No
Min et al. (2006) , Wu et al. (2007) and Dunis et al. (2013)	RBF-SVM	No	Yes	Yes	No

Chen and Wang (2007) forecast the tourist demand in China with a similar model. Yuan (2012) suggests that a GA- ϵ -SVR model is more efficient than traditional SVR and NN models, when applied to the task of forecasting sales volume. All these GA- ϵ -SVR applications do not deploy the GA to locate the optimal feature subset but restrict it in optimizing the parameters of the ϵ -SVR models and select the model's inputs empirically.

As described in the previous paragraph, a variety of hybrid methodologies combining SVM/SVR models with GAs have been proposed during the last decade. However, the proposed RG-SVR theoretically outperforms them as it deploys v-SVR models, which are more suitable to this problem than SVMs and ϵ -SVR. SVMs are constrained to classification problems and their applicability is limited. The ϵ -SVR models require the desired accuracy of the approximation to be specified beforehand and are extremely sensitive to the selection of the ϵ parameter compared to the sensitivity of v-SVRs to the ν parameter (Schölkopf, Smola, Williamson, & Bartlett, 2000). Moreover, RG-SVR optimizes on parallel the SVR's parameters and the input's subset, it deploys a sliding window approach to capture the dynamic nature of the examined financial time series, it employs a specialized loss function and tries to minimize the number of support vectors of the model in order to improve its generalization abilities. The advantages of RG-SVR over other published methodologies are outlined in Table 1.

3. The EUR/USD, EUR/GBP and EUR/JPY exchange rates and related financial data

The ECB publishes a daily fixing for selected EUR exchange rates. These reference mid-rates are based on a daily concentration procedure between central banks within and outside the European System of Central Banks, which normally takes place at 2.15 p.m. ECT time. The reference exchange rates are published both by electronic market information providers and on the ECB's website shortly after the concentration procedure has been completed.

Although only a reference rate, many financial institutions are ready to trade at the EUR fixing and it is therefore possible to leave orders with a bank for business to be transacted at this level. Thus, the ECB daily fixings of the EUR exchange rate are tradable levels and using them is a more realistic alternative to, say, London closing prices. This superiority for financial institutions to transact at the EUR fixing is a well-known fact by FX markets' participants. Financial institutions and institutional investors, and therefore through them, High Net Worth individuals (HNW) would definitely be able to transact at the ECB fixing through their investments with hedge funds, private banks, etc. Smaller private investors could also benefit from reasonably attractive cost conditions (see www.interactivebrokers.com), although they would not be able to transact at the ECB fixing as considered in this study.

In this paper, the ECB daily fixings of EUR/USD, EUR/GBP and EUR/JPY exchange rates are examined over the period of 01/02/1999...30/04/2012. The range of these observations (3395 trading days) is used in four consecutive forecasting exercises. In order to train the RG-SVR, it is necessary to divide the in-sample dataset to training

and test subsets (see Section 4.1). The test dataset constitutes approximately the 38 percent¹ of the in-sample dataset in each forecasting exercise. The total dataset and the length of each forecasting exercise are presented in Table 2.

The graph in Fig. 1 shows the total dataset for the three exchange rates under study.

The three observed time series are non-normal (Jarque...Bera statistics, 1980 confirm their non-normality at the 99 percent confidence interval) containing slight skewness and high kurtosis. They are also non-stationary and hence they are transformed them into three daily series of rate returns² using the following formula:

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

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

The summary statistics of the EUR/USD, EUR/GBP and EUR/JPY return series reveal that the slight skewness and high kurtosis remain. In addition, the Jarque...Bera statistic confirms again their non-normality at the 99 percent confidence interval.

The aim of these forecasting exercises is to forecast and trade the one day ahead EUR/USD, EUR/GBP and EUR/JPY exchange rate return ($E(R_t)$). As a first step, we estimate the three return series with several linear and non-linear models. Then, these estimations are used as potential inputs to the RG-SVR algorithm.

4. Hybrid RG-SVR model

The hybrid Rolling Genetic-Support Vector Regression (RG-SVR) model for optimal parameter selection and feature subset combination is presented in this section. Initially, the generic architecture of the proposed methodology is described. Then the feature space, in which the model will search for the optimal subsets and combinations, is identified along with the models that are going to be used as benchmarks. The SVR models are more suitable than the classical SVMs ones for this daily forecasting task, as they provide an exact prediction instead of a binary output. The exact prediction enables the effective application of confirmation filters to improve its performance and provide a hint about the strength of the trading signal.

4.1. Architecture

The proposed model genetically searches over the feature space (pool of individual forecasts) and then combines the optimal feature subsets (SVR forecast combinations) for each exchange rate. In order to achieve this, a simple GA is used where each chromosome comprises feature genes that encode the feature subsets and parameter genes that encode the choice of parameters.

¹ This choice is based on in-sample experimentation. We obtain similar trading and statistical in-sample RG-SVR performance for test datasets that constitute the 29 percent up to the 47 percent of the in-sample dataset.

² Confirmation of their stationary property is obtained at the 1 percent significance level by both the Augmented Dickey Fuller (ADF) and Phillips...Perron (PP) test statistics.

Table 2
The total dataset,neural networks* training datasets.

Forecasting exercise	Periods	Trading days	Start date	End date
1	Total dataset	1857	01/02/1999	28/04/2006
	Training dataset (Ntr) ^a	831	01/02/1999	30/04/2002
	Test dataset	511	02/05/2002	30/04/2004
	Out-of-sample dataset	515	03/05/2004	28/04/2006
2	Total dataset	1852	01/02/2001	30/04/2008
	Training dataset (Ntr)	826	01/02/2001	30/04/2004
	Test dataset	515	03/05/2004	28/04/2006
	Out-of-sample dataset	511	02/05/2006	30/04/2008
3	Total dataset	1854	03/02/2003	30/04/2010
	Training dataset (Ntr)	832	03/02/2003	28/04/2006
	Test dataset	511	02/05/2006	30/04/2008
	Out-of-sample dataset	511	02/05/2008	30/04/2010
4	Total dataset	1857	01/02/2005	30/04/2012
	Training dataset (Ntr)	830	01/02/2005	30/04/2008
	Test dataset	511	02/05/2008	30/04/2010
	Out-of-sample dataset	516	03/05/2010	30/04/2012

^aThe in-sample dataset is the sum of the training and test dataset.

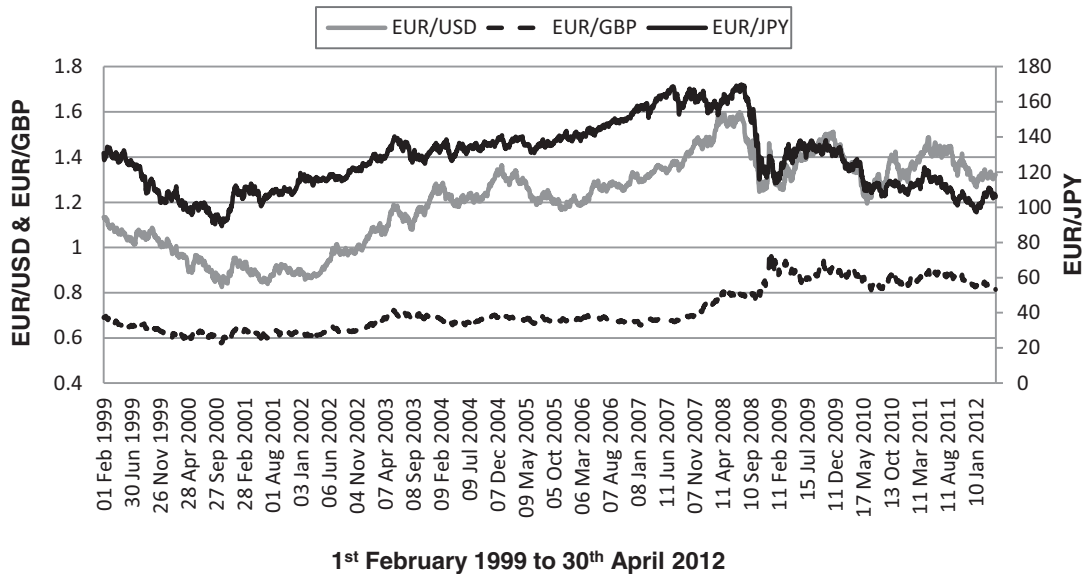


Fig. 1. The EUR/USD, EUR/GBP and EUR/JPY total dataset.

The lack of information about the noise's nature and parameters of the training datasets makes the a priori ϵ -margin setting of ϵ -SVR a difficult task. In order to overcome this and decrease the computational demands of the methodology, the RBF ν -SVR approach is applied in this hybrid RG-SVR model (see Appendix A.2). The impact of ϵ parameter in ϵ -SVR is much more crucial than the impact of ν in ν -SVR. The ν -SVR method is limiting with an upper and lower bound of the number of SVs. In this way, the optimization becomes more stable and the algorithm needs less iterations to find an optimal ν value, which would provide accurate results and good generalization properties. The use of the RBF kernel is justified by its extensive use in the literature and its superiority to other types of kernels, when used in financial time series forecasting (Ince & Trafalis, 2008; Lu et al., 2009).

A RBF kernel is in general specified as:

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

where γ represents the variance of the kernel function. Consequently, the parameters which should be optimized by the GA are C, ν and γ .

Binary representation is used to model the chromosome. Every feature gene is represented with a binary digit. If that gene takes value 1 (0), then this indicates that the relevant feature should be

used (not be used) as input. For the parameter genes 50 bits are used, which are specified as follows:

- 10 bits to represent the integer part of parameter C of SVRs (range [0...1024])
- 10 bits to represent the decimal part of parameter C of SVRs (0.001 precision)
- 10 bits to represent the integer part of the γ parameter of RBF functions (range [0...1024])
- 10 bits to represent the decimal part of the γ parameter of RBF functions (0.001 precision)
- 10 bits to represent the ν parameter of ν -SVR (range [0...1] with 0.001 precision)

During the initialization step all genes are randomly set with values 0 or 1 with equal probabilities for both of them.

The GA uses the one-point crossover and the mutation operators. The one-point crossover creates two offspring from every two parents. The parents and a crossover point c_x are selected at random. The two offsprings are made by both concatenating the genes that precede c_x in the first parent with those that follow (and include) c_x in the second parent. The probability for selecting an individual as a

parent for the crossover operator is called crossover probability and in this application is set to 0.90. Having a high crossover probability enables the model to keep some population for the next generation, hoping to create better new chromosomes from good parts of the old chromosomes. The offspring produced by the crossover operator replaces their parents in the population. The one-point crossover is used for simplicity and to allow bigger blocks of genes to be exchanged during the crossovers. The uniform crossover is usually better only when small mutation probabilities (for example 0.001) are used in order to keep the diversity of chromosomes in the population. In this study, the applied mutation probability is quite big (0.1) and thus the one-point crossover is adequate. The crossover probability is not set to 1 to leave a space for very good solutions of a population to pass through the next generation's population without being altered. The other variation operator which is deployed is the mutation one. The mutation operator places random values in randomly selected genes with a certain probability named as mutation probability. This operator is very important for avoiding local optima and exploring a larger surface of the search space. This probability is set to 0.1 in order to prevent the algorithm from performing a random search.

For the selection step of the GA, the roulette wheel selection process is used (Holland, 1975). In roulette wheel selection chromosomes are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. In the proposed approach, elitism is used to raise the evolutionary pressure in better solutions and to accelerate the evolution. In that way, it is assured that the best solution is copied without changes to the new population, so the best solution found can survive at the end of every generation.

More specifically, the selection step is used to resemble the survivor of the fittest principle. In that way better solutions have higher probabilities to provide offspring in the next generation. After the application of selection probability, an intermediate population of solutions is formed with each one of them being identical to the solutions from the previous population. Then, crossover and mutation probabilities are applied to provide offsprings combining solutions from this intermediate population. Before the application of the crossover operator, the solutions from this intermediate population are used to make pairs of solutions. Some of these pairs are combined using the crossover operator (with the crossover probability) to produce new solutions, while the others are passed to the next intermediate population without being altered. The offspring is derived from the crossover operator and the pairs of solutions that are not recombined form the next intermediate population. After this process, the intermediate population is altered using the mutation operator. In the case of this study, binary mutation is used. If a gene is selected for mutation, then a new random binary value (0 or 1) is constructed and replaces the previous value in this gene. In order to re-assure that no good solutions are missed, elitism is applied. In that way, the best solution is allowed to pass to the next population. The crossover probability is not set to 1 to leave space also for some solutions that are close to optimal not to be recombined, but just change only with the mutation operator (which is applied only in a few genes of the whole chromosome).

For example, assuming population size is N and elitism of one member, the use of the selection operator leads to an intermediate population of N solutions, which form $N/2$ pairs. Some of these pairs are selected for crossover using the crossover probability and others are passing on for mutation, as they are. When the mutation process is completed, a complete set of N solutions is derived. The next iteration starts once one solution is randomly replaced with the best solution of the previous generation.

The RG-SVR hybrid performs a rolling window forecasting exercise. The window size is always ten days (two trading weeks). Else the parameters and the inputs of the algorithm are re-estimated every two weeks. This fact allows the model to capture any structural break in the dataset. Trading series are exhibiting a highly non-linear

nature and are affected by a wide range of factors. Mathematical models with "fixed parameters are impossible to model them perfectly. Fund managers and professionals apply a range of different models and re-estimate their parameters frequently. The nature of the financial markets and their erratic behavior make impossible a perfect ex ante configuration of the re-estimation period. Frequent re-estimations dramatically increase the computational time and might induce noise in the estimations.³ Infrequent re-estimations will lead to a low trading performance. The ten trading days sliding window approach is selected based on a trade-off between the in-sample trading performances of RG-SVR and the computational time needed for its estimation with the used dataset.

The population of chromosomes is initialized in the training sub-period. The optimal selection of chromosomes is achieved, when their forecasts maximized the proposed loss function (see Eq. (3)) in the test-sub period. Then, the optimized parameters and selected predictors of the best solution are used to train the SVR and produce the final optimized forecast for the next observation. This procedure is repeated in each of the four forecasting exercises. For each of these forecasting exercises, the algorithm stores its optimized C, γ and ν parameters and set of optimal predictors.

In order to achieve the optimal selection of the feature subsets (individual forecasts) a three-objective fitness function is applied to the hybrid approach. Firstly the annualized return of the SVR forecast combinations should be maximized and secondly the Root Mean Square Error (RMSE) of the output should be minimized in the test sub-period. The presence of the SVs term in the proposed fitness function enables RG-SVR to extract the minimum prediction models which present good prediction accuracy improving its generalization abilities. Based on the above, the fitness function takes the form of Eq. (3):

$$\text{Fitness} = \text{annualized return} - \frac{1}{10} \times \text{RMSE} - 0.001 \times (\text{SVs}/\text{Ntr}) \quad (3)$$

where Ntr is the size of the training sample.

The aim is to maximize Eq. (3). The proposed fitness function aims to bring a balance between trading profitability (first factor of Eq. (3)) and statistical accuracy (second factor) while retaining the complexity of the algorithm to a minimum. This is very important in trading applications as statistical accuracy does not always imply financial profitability. Additionally, complex models in trading applications are not always applicable due to the increased computation time and the dangers of overfitting and lack of generalization. There are two computational time metrics related to the task of building forecasting models and applying them to new data. The first one is the Computational Time required for Training a model (CTT). The second is the Computational Time for the Application of the trained model to new data (CTA).⁴ The training phase is an offline procedure. Thus, the CTT is of minor importance. It is not required to be applied in real time and can be applied in scheduled times (for example once per week). The critical phase in terms of time complexity is the application to new data phase, as this is probably applied in real time. In the proposed approach, the term SVs/Ntr is introduced in the fitness function. This term does not affect the CTT, but dramatically decreases the CTA by using simplest models. This approach has also a significant effect in the model's generalization properties reducing the danger of overfitting. The RMSE is multiplied by 10 to make the first two factors in Eq. (3) more or less equal in levels. SVs is the number of Support

³ Financial markets are vulnerable to behavioral (Froot, Schafstein, & Stein, 1992) and exogenous factors such as political decisions (Fisman, 2001). These factors are impossible to capture with mathematical models and include noise to time series estimations.

⁴ For our experiments the CTT time using a simple personal computer with Intel I7 processor ranged from 3 to 4 hours while the CTA time for 10 trading days was a less than 1 second.

Table 3
GA characteristics and parameters.

Population size	40
Maximum generations	200
Selection type	Roulette wheel selection
Elitism	Best member of every population is maintained in the next generation.
Crossover probability	0.9
Mutation probability	0.1
Fitness function	Fitness = annualized return $\hat{S} \times 10 \times \text{RMSE}\hat{S}$ $0.001 \times (\text{SVs}/\text{Ntr})$

Table 4
The summary description of the linear models.

Linear models	Description	Total individual forecasts
RW	$E(R_t) = \mu + \epsilon_t$, $N(0, 1)$ where: • μ the in-sample mean	1
AR (p)	$E(R_t) = \beta_0 + \sum_{i=1}^p \beta_i R_{t-i}$ where: • $p = 1, f, 20$ • β_0, β_i the regression coefficients	20
MA (q)	$E(R_t) = (R_{t-1} + \dots + R_{t-q})/q$ where: • $q = 3f25$	23
ARMA (m*, n*)	$E(R_t) = \hat{\varphi}_0 + \sum_{j=1}^m \hat{\varphi}_j R_{t-j} + \hat{a}_0 + \sum_{k=1}^n \hat{w}_k \hat{a}_{t-k}$ where: • $m^*, n^* = 1, f, 15$ • $\hat{\varphi}_0, \hat{\varphi}_j$ the regression coefficients • \hat{a}_0, \hat{a}_{t-k} the residual terms • \hat{w}_k the weights of the residual terms	210

Vectors of the trained SVR model. This number is "rst divided to Ntr to normalize its values from 0 to 1 and then it is further divided with 1000 to decrease its impact to the "nal "tness function. Reducing model complexity is a secondary task compared to forecasting accuracy and the trading profitability.

The size of the initial population is set to 40 chromosomes while the maximum number of generations is set to 200. The algorithm though terminates when the number of generations is 60 on average. This number must be reached in combination with a termination method that stops the evolution, when the population is deemed as converged. The population is deemed as converged when the average "tness across the current population is less than 5 percent away from the best "tness of the current population. More specifically, when it is less than 5 percent 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.

The summary of the GA's characteristics is presented in Table 3.

The flowchart of the proposed methodology is depicted in detail in Fig. 2.

4.2. Feature space and feature subset selection

The forecasting ability of the proposed methodology is evaluated over a feature space that is synthesized by individual linear and non-linear forecasts of each exchange rate over the periods outlined in Table 2. More specifically, the pool is consisted by a Random Walk (RW), a series of Autoregressive (AR), Moving Average (MA), Autoregressive Moving Average (ARMA) linear models and "ve non-linear algorithms, namely a Nearest Neighbors Algorithm (k-NN) a Multi-Layer Perceptron (MLP), a Recurrent Neural Network (RNN), a Higher Order Neural Network (HONN) and a Psi-Sigma Neural Network (PSN). A summary of the used linear models is presented in Table 4, while the applied non-linear models are explained in Appendix B.

These models create a pool of 259 individual forecasts in total for each forecasting exercise. The algorithm genetically searches the above feature space and selects the optimal feature subsets. In all cases, GA selects as inputs for the proposed model among the MLP, HONN, RNN, PSN and k-NN predictors. In other words, the model discards as inputs the linear predictors in favor of the non-linear models (see Table 5). This was expected to some extent due to the non-linear nature of the "nancial time series.

In the "nal stage of the algorithm, RG-SVR applies a GA to select the SVR parameters. As mentioned before this process is repeated every ten trading days, in order the proposed model captures any possible structural break in the series under study. Although the selection of the inputs seems to be consisted in the majority of the runs for the three exchange rates, this is not the case for the SVR parameters as the C, ν and γ parameters vary between 4.10 and 9.20, 0.35 and 0.84 and 8.12 and 19.77 respectively. These variations of the parameters through the different sliding window iterations enable it to adjust to the continuously changing real "nancial model. It is worth noting that the proposed RG-SVR methodology is fully adaptive. The practitioner does not need to experiment with the parameters of the algorithm in order to optimize the forecasts. RG-SVR structure and its parameters are generated in a single optimization procedure, which prevents the data snooping effect.

4.3. Benchmark models

The statistical and trading efficiency of the hybrid model is initially evaluated by benchmarking it with traditional genetic and non-genetically optimized SVRs and SVMs. For the non-genetically optimized SVRs, the selection of the inputs and the SVR parameters is optimized with the statistical methods that dominate the relevant literature (grid search or 5-cross validation over the in-sample period).⁵ The genetically optimized SVR and SVM benchmarks are constructed following the instructions of those introducing them in the literature (see Table 1). A summary of the SVR-SVM benchmarks is presented below:

- An ϵ -SVR model that implements a 5-fold cross-validation and a simple data-driven calculation on the in-sample dataset to calculate parameters ϵ , γ and C respectively (ϵ -SVR₁).
- A ν -SVR model that calculates its parameters ν , γ and C as the ϵ -SVR₁ (ν -SVR₁).
- An ϵ -SVR and ν -SVR model that all parameters are selected based on a grid-search algorithm in the in-sample dataset (ϵ -SVR₂ and ν -SVR₂).
- A GA-SVM model as proposed by Min et al. (2006), Wu et al. (2007) and Dunis et al. (2013).
- A GA- ϵ SVR model as proposed by Chen and Wang (2007), Pai et al. (2006) and Yuan (2012).

For more on the ϵ -SVR₁ and ν -SVR₁ approaches see Duan, Keerthi, and Poo (2003) and Cherkassky and Ma (2004), while for the ϵ -SVR₂ and ν -SVR₂, see Schölkopf and Smola (2002).

In addition, Sermpinis et al. (2013) introduce the ARBF-PSO method in a forecasting and trading competition of several linear and non-linear models. Their hybrid NN structure proves to be superior in statistical and trading terms over the more traditional MLP, RNN and PSN. The authors investigate the ECB daily "xings of EUR/USD, EUR/GBP and EUR/JPY over the period of 04/01/1999...29/04/2011 (3158 trading days). Their statistical and trading results are not directly comparable with the ones of this study, since their in-sample and out-of-sample periods are overlapping with those of Table 2.

⁵ The inputs in all cases of forecasting exercise 1 are selected based on 5-fold cross validation, while in forecasting exercises 2 and 3 with grid search. For the forecasting exercise 4, the inputs selected for EUR/USD and EUR/GBP are based on grid search algorithms, while the ones for EUR/JPY are derived based on 5-fold cross validation.

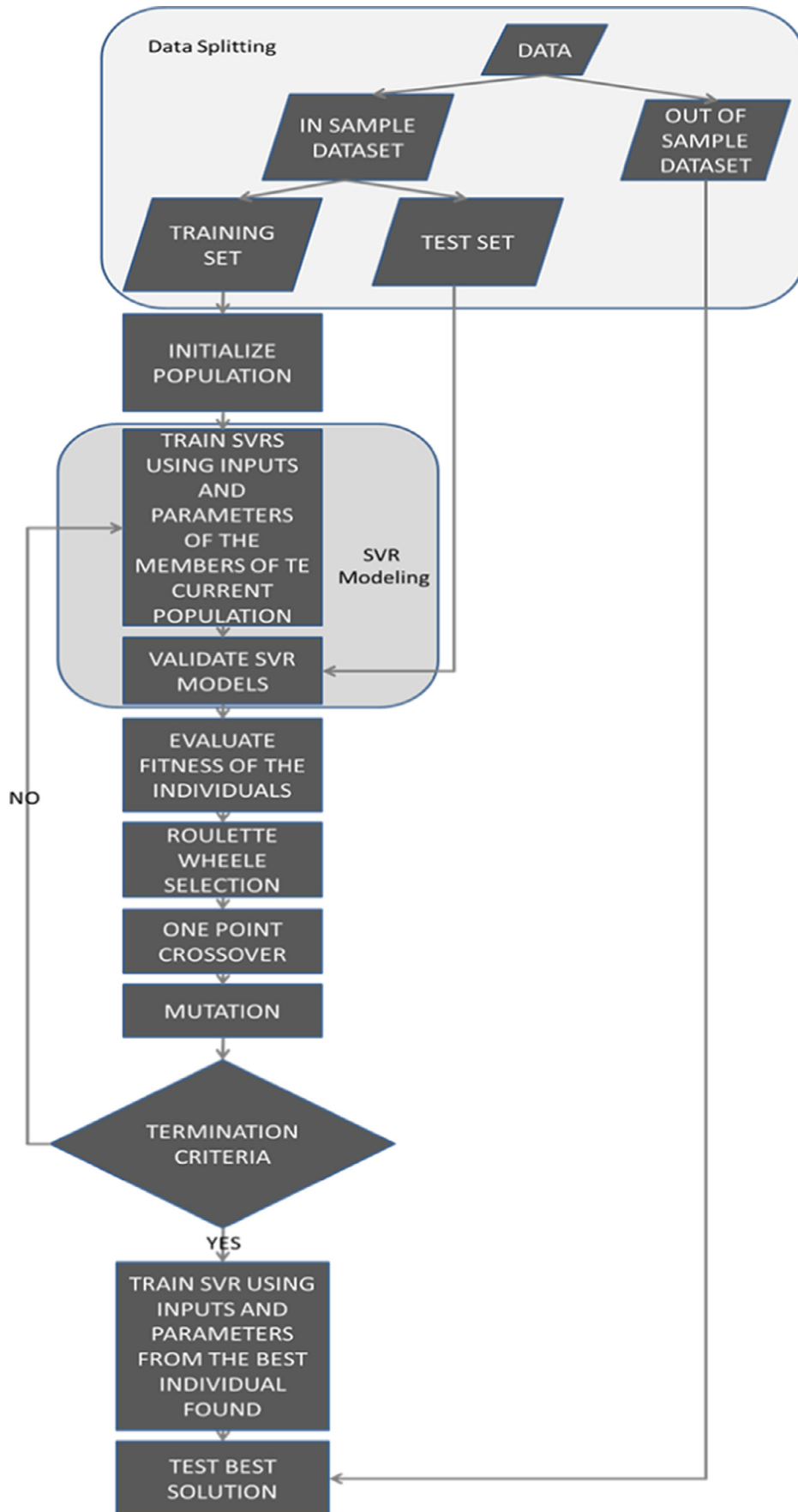


Fig. 2. Hybrid RG-SVR flowchart.

Table 5
Summary of selected inputs of all models.

	2			
	EUR/USD	EUR/GBP	EUR/JPY	EUR/USD
Panel A				
ARBF-PSO	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSN
ε -SVR ₁	MLP, HONN, RNN, PSNk-NN	MLP, RNN, PSNk-NN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSNk-NN
ε -SVR ₂	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSNk-NN
v-SVR ₁	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSN, k-NN	MLP, HONN, RNN, PSNk-NN	HONN, HONN, RNN, PSN
v-SVR ₂	MLP, HONN, RNN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSNk-NN
GA-SVM	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN
GA- ε SVR	MLP, HONN, RNN, PSNk-NN	RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN	HONN, RNN, PSNk-NN
RG-SVR (pop)	MLP, HONN, RNN	MLP, RNNk-NN	MLP, HONN, RNN	MLP, HONN, RNN, PSN
Panel B				
ARBF-PSO	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, PSNk-NN	MLP, HONN, PSNk-NN
ε -SVR ₁	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN	MLP, RNN, PSNk-NN
ε -SVR ₂	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN
v-SVR ₁	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN	HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSN
v-SVR ₂	HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN, PSNk-NN
GA-SVM	MLP, HONN, RNN, PSN	MLP, HONN, RNN, PSN	MLP, HONN, PSNk-NN	MLP, HONN, RNN, PSNk-NN
GA- ε SVR	MLP, HONN, RNNk-NN	MLP, HONN, RNN, PSNk-NN	MLP, HONN, RNN	MLP, HONN, RNN, PSNk-NN
RG-SVR (pop)	MLP, HONN, PSN	HONN, RNNk-NN	MLP, RNN, PSN	HONN, RNN, PSN

a Panels A and B include the input selection per each series under study for forecasting exercises 1, 2 and 3, 4 respectively. RG-SVR re-estimates its inputs every 10 trading days. In other words, in each of the four forecasting exercises RG-SVR algorithm re-estimates its inputs 51 times (the out-of-sample is always between 511 and 516 days). In the table for the sake of space, we present the three most popular inputs of RG-SVR in each forecasting exercise (RG-SVR(pop)). The minimum numbers of inputs (for all forecasting exercises and re-estimations) selected by the GA for the RG-SVR were 3 and the maximum 5.

Nonetheless, ARBF-PSO's performance is impressive over a dataset of the same three exchange rates that this study is investigating. Thus, ARBF-PSO seems a perfect benchmark for RG-SVR. More details on the ARBF-PSO can be found in Appendix B.

The selected inputs for all the benchmark models are presented in Table 5.

It is notable that all models apply non-linear predictors as inputs. The "ve non-linear predictors seem to contain all the information that the 254 linear ones contain. This was expected to some extent by the non-linear behavior of financial trading series such as the ones under study.

5. Statistical performance

As it is standard in the literature, in order to evaluate statistically the obtained forecasts, the Root Mean Squared Error (RMSE) is computed. The mathematical formula of this statistic is presented in Appendix C. The lower the output, the better the forecasting accuracy of the model concerned. The Pesaran...Timmermann (PT) test (1992) examines whether the directional movements of the real and forecast values are in step with one another. In other words, 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 that the model under study has no power on forecasting the relevant exchange rate. The in-sample statistical performance of the models for the four forecasting exercises and the EUR/USD, EUR/GBP and EUR/JPY exchange rates is presented in Table 6.

The results of Table 6 show that RG-SVR presents the best in-sample statistical performance for every exchange rate under study. The PT-statistics rejects the null hypothesis of no forecasting power at the 1 percent confidence interval for all models and series under study. It is also notable that the v-SVR models (v-SVR₁ and v-SVR₂) have lower RMSEs of the statistical measures than the ε -SVR ones (ε -SVR₁ and ε -SVR₂). Additionally the genetically optimized models clearly outperform their statistical optimized SVR benchmarks. ARBF-PSO presents occasionally better RMSE values than some SVR-SVM benchmarks, but it always underperforms in comparison to RG-SVR. Table 7 summarizes the statistical performance of the models under study in the out-of-sample period. As a reference, the overall performance of the best predictor is presented in the "rst column (see Appendix D).

From Table 7 it is suggested that RG-SVR retains its forecasting superiority in the out-of-sample period. The genetically optimized SVRs and SVMs outperform their statistically optimized benchmarks while the PT-statistics indicate that all models continue to forecast accurately the directional change of the three exchange rates. In the out-of-sample statistical evaluation, ARBF-PSO still remains less efficient than the proposed model, although in most cases it outperforms the traditional SVR in terms of RMSE. Tables 5 and 6 show that the grid-search optimized SVRs outperform their counterparts optimized with 5-cross validation while the v-SVR models produce more accurate forecasts than the ε -SVR algorithms. The results support the arguments of Yuan (2012) on the performance of the GA- ε SVR over the traditional SVR models.

In order to further verify the statistical superiority of the proposed algorithm, the Diebold...Mariano (DM) (1995) statistic for predictive accuracy is computed, while the MSE is considered as the loss function. The test is applied in the four consecutive out-of-sample periods. Table 8 below presents the DM statistic comparing the RG-SVR with its benchmarks.

From the above table it is obvious that the null hypothesis of equal predictive accuracy is rejected for all comparisons and for both loss functions at the 1 percent confidence interval (absolute values higher than the critical value of 2.33). Moreover, the statistical superiority of

Table 6
Summary of in-sample statistical performance.

In-Sample Period	Series	Statistic	ARBF-PSO	ε -SVR ₁	ε -SVR ₂	v-SVR ₁	v-SVR ₂	GA-SVM	GA- ε SVR	RG-SVR
01/02/1999...30/04/2004	EUR/USD	RMSE	0.0058	0.0061	0.0057	0.0055	0.0054	0.0049	0.0045	0.0041
		PT-statistic	12.28	8.13	9.63	10.81	11.06	12.92	13.64	15.28
	EUR/GBP	RMSE	0.0041	0.0056	0.0053	0.0052	0.0049	0.0047	0.0045	0.0038
		PT-statistic	11.58	7.79	9.05	10.28	11.38	12.27	13.79	15.38
	EUR/JPY	RMSE	0.0060	0.0063	0.0060	0.0057	0.0053	0.0050	0.0049	0.0045
		PT-statistic	11.52	7.57	9.17	10.79	11.37	12.02	12.98	15.03
01/02/2001...28/04/2006	EUR/USD	RMSE	0.0059	0.0072	0.0064	0.0065	0.0057	0.0053	0.0048	0.0046
		PT-statistic	11.31	7.61	8.82	8.64	10.93	11.95	13.22	14.69
	EUR/GBP	RMSE	0.0046	0.0068	0.0060	0.0058	0.0053	0.0049	0.0047	0.0044
		PT-statistic	12.01	7.93	8.81	9.60	10.54	12.06	12.95	14.38
	EUR/JPY	RMSE	0.0059	0.0062	0.0064	0.0056	0.0051	0.0048	0.0046	0.0043
		PT-statistic	11.05	7.76	7.54	9.30	10.57	11.92	12.53	14.21
03/02/2003...30/04/2008	EUR/USD	RMSE	0.0056	0.0064	0.0058	0.0053	0.0052	0.0048	0.0045	0.0040
		PT-statistic	12.25	7.93	9.22	10.54	12.00	13.12	13.89	14.77
	EUR/GBP	RMSE	0.0046	0.0065	0.0061	0.0056	0.0057	0.0051	0.0048	0.0043
		PT-statistic	10.95	7.88	8.19	10.13	10.84	12.29	13.14	14.81
	EUR/JPY	RMSE	0.0060	0.0060	0.0058	0.0053	0.0052	0.0049	0.0047	0.0042
		PT-statistic	11.48	8.52	9.79	10.18	10.27	12.74	12.68	14.22
01/02/2005...30/04/2010	EUR/USD	RMSE	0.0059	0.0070	0.0071	0.0064	0.0057	0.0053	0.0048	0.0047
		PT-statistic	12.03	7.44	7.40	8.74	10.23	11.58	12.57	12.83
	EUR/GBP	RMSE	0.0056	0.0066	0.0061	0.0055	0.0054	0.0052	0.0050	0.0044
		PT-statistic	10.56	7.48	9.15	10.80	10.91	11.67	12.41	14.69
	EUR/JPY	RMSE	0.0069	0.0067	0.0064	0.0059	0.0055	0.0051	0.0052	0.0048
		PT-statistic	12.15	7.52	7.88	10.40	11.38	12.56	12.51	12.93

Table 7
Out-of-sample statistical performance.

Out-of-sample period	Series	Statistic	Best predictor	ARBF-PSO	ε -SVR ₁	ε -SVR ₂	v-SVR ₁	v-SVR ₂	GA-SVM	GA- ε SVR	RG-SVR
03/05/2004...28/04/2006	EUR/USD	RMSE	0.0080	0.0061	0.0074	0.0069	0.0066	0.0062	0.0059	0.0057	0.0054
		PT-statistic	5.77	8.15	6.23	7.25	7.72	8.03	8.64	9.27	9.62
	EUR/GBP	RMSE	0.0077	0.0055	0.0073	0.0068	0.0061	0.0056	0.0055	0.0053	0.0052
		PT-statistic	5.86	8.99	6.34	7.22	8.18	8.92	9.12	9.60	9.78
	EUR/JPY	RMSE	0.0082	0.0065	0.0076	0.0071	0.0070	0.0067	0.0060	0.0056	0.0053
		PT-statistic	5.79	8.35	6.21	6.86	7.27	7.91	8.57	9.31	9.73
02/05/2006...30/04/2008	EUR/USD	RMSE	0.0075	0.0059	0.0069	0.0066	0.0067	0.0061	0.0058	0.0055	0.0051
		PT-statistic	5.88	8.02	7.01	7.52	7.43	8.02	8.64	9.51	9.90
	EUR/GBP	RMSE	0.0078	0.0058	0.0072	0.0068	0.0063	0.0057	0.0053	0.0053	0.0050
		PT-statistic	5.84	9.12	6.27	7.44	7.95	8.81	9.64	9.57	10.13
	EUR/JPY	RMSE	0.0083	0.0064	0.0073	0.0074	0.0068	0.0066	0.0059	0.0057	0.0055
		PT-statistic	5.74	7.81	6.37	6.31	7.33	7.79	8.48	9.30	9.52
02/05/2008...30/04/2010	EUR/USD	RMSE	0.0077	0.0057	0.0071	0.0068	0.0067	0.0062	0.0058	0.0056	0.0052
		PT-statistic	5.85	7.78	6.63	7.14	7.47	7.83	8.59	9.28	9.71
	EUR/GBP	RMSE	0.0075	0.0055	0.0070	0.0067	0.0064	0.0061	0.0056	0.0054	0.0051
		PT-statistic	5.89	9.12	6.72	7.33	7.98	8.27	9.35	9.66	9.94
	EUR/JPY	RMSE	0.0085	0.0065	0.0075	0.0071	0.0065	0.0063	0.0060	0.0056	0.0053
		PT-statistic	5.69	8.17	6.18	6.92	7.59	7.80	8.41	9.26	9.72
03/05/2010...30/04/2012	EUR/USD	RMSE	0.0084	0.0058	0.0078	0.0077	0.0070	0.0065	0.0063	0.0059	0.0055
		PT-statistic	5.65	8.86	5.87	5.93	6.68	7.61	7.80	8.38	9.41
	EUR/GBP	RMSE	0.0077	0.0054	0.0070	0.0066	0.0064	0.0061	0.0059	0.0058	0.0052
		PT-statistic	5.88	8.54	6.64	7.40	7.57	8.12	8.66	9.05	9.81
	EUR/JPY	RMSE	0.0076	0.0063	0.0069	0.0065	0.0064	0.0063	0.0058	0.0059	0.0055
		PT-statistic	5.82	7.81	6.87	7.54	7.63	7.74	8.90	8.72	9.33

Table 8
Diebold-Mariano statistic for each exchange rate under study.

	Period	Best predictor	ARBF-PSO	ε -SVR ₁	ε -SVR ₂	v-SVR ₁	v-SVR ₂	GA-SVM	GA- ε SVR
EUR/USD	03/05/2004...28/04/2006	Š7.26	Š3.95	Š5.47	Š5.31	Š5.10	Š4.66	Š3.92	Š3.55
	02/05/2006...30/04/2008	Š6.90	Š3.86	Š6.14	Š5.20	Š5.74	Š4.75	Š3.46	Š3.13
	02/05/2008...30/04/2010	Š7.05	Š3.91	Š5.88	Š5.42	Š5.01	Š4.77	Š4.01	Š3.72
EUR/GBP	03/05/2010...30/04/2012	Š7.47	Š3.94	Š5.63	Š5.84	Š5.15	Š4.89	Š4.35	Š4.16
	03/05/2004...28/04/2006	Š6.72	Š3.75	Š5.47	Š5.39	Š5.06	Š4.53	Š3.84	Š3.58
	02/05/2006...30/04/2008	Š7.14	Š3.87	Š5.42	Š5.11	Š4.88	Š4.46	Š3.69	Š3.22
EUR/JPY	02/05/2008...30/04/2010	Š6.81	Š3.96	Š5.79	Š5.62	Š5.35	Š4.92	Š3.57	Š3.42
	03/05/2010...30/04/2012	Š6.92	Š3.99	Š5.74	Š5.23	Š4.84	Š4.56	Š3.88	Š3.60
	03/05/2004...28/04/2006	Š7.23	Š4.25	Š6.49	Š6.10	Š5.71	Š5.39	Š4.18	Š4.05
	02/05/2006...30/04/2008	Š7.56	Š4.63	Š6.12	Š6.45	Š5.22	Š4.75	Š4.51	Š4.10
	02/05/2008...30/04/2010	Š7.61	Š4.07	Š6.57	Š6.19	Š5.70	Š5.15	Š3.73	Š3.51
	03/05/2010...30/04/2012	Š7.18	Š4.12	Š5.88	Š5.51	Š4.95	Š4.67	Š3.75	Š3.94

Table 9
Summary of in-sample trading performance.

In-sample period	Series	Statistic	ARBF-PSO	ε -SVR ₁	ε -SVR ₂	v-SVR ₁	v-SVR ₂	GA-SVM	GA- ε SVR	RG-SVR
01/02/1999... 30/04/2004	EUR/USD	Annualized return (including costs) (percent)	25.95	18.25	19.99	22.13	22.81	25.77	28.20	32.92
		Information ratio (including costs)	2.41	1.69	1.85	2.01	2.09	2.34	2.59	2.99
		Maximum drawdown (percent)	\$20.11	\$11.12	\$12.11	\$11.33	\$12.92	\$13.45	\$12.53	\$12.74
	EUR/GBP	Annualized return (including costs) (percent)	28.44	19.94	20.18	21.70	23.29	25.62	27.91	29.57
		Information ratio (including costs)	2.62	1.81	1.83	1.97	2.12	2.31	2.53	2.69
		Maximum drawdown (percent)	\$14.55	\$13.96	\$13.58	\$12.61	\$12.84	\$12.28	\$12.56	12.16
	EUR/JPY	Annualized return (including costs) (percent)	27.07	21.45	21.62	22.47	22.93	27.12	28.38	30.65
		Information ratio (including costs)	2.32	1.96	1.99	2.07	2.11	2.48	2.64	2.74
		Maximum drawdown (percent)	\$21.57	\$12.84	13.46	\$12.05	\$12.88	\$12.01	\$12.17	\$11.93
01/02/2001... 28/04/2006	EUR/USD	Annualized return (including costs) (percent)	23.14	20.53	22.77	21.84	22.98	25.80	27.64	30.04
		Information ratio (including costs)	2.25	1.88	2.08	1.97	2.11	2.35	2.53	2.77
		Maximum drawdown (percent)	\$21.35	\$13.17	\$12.55	\$12.26	\$14.61	\$12.10	\$12.04	\$12.11
	EUR/GBP	Annualized return (including costs) (percent)	30.55	22.79	24.06	26.56	27.85	29.14	31.02	33.43
		Information ratio (including costs)	2.78	2.05	2.18	2.39	2.53	2.62	2.81	3.04
		Maximum drawdown (percent)	15.39	\$14.23	\$13.79	\$13.32	\$14.47	\$13.01	\$13.94	\$13.29
	EUR/JPY	Annualized return (including costs) (percent)	30.44	24.71	24.98	25.57	27.99	28.61	30.48	32.80
		Information ratio (including costs)	2.76	2.27	2.29	2.34	2.55	2.60	2.79	2.99
		Maximum drawdown (percent)	\$19.05	\$13.47	\$14.91	\$13.03	\$14.52	\$15.11	\$13.18	\$13.48
03/02/2003... 30/04/2008	EUR/USD	Annualized return (including costs) (percent)	30.11	23.57	25.66	27.83	28.90	31.29	32.78	33.16
		Information ratio (including costs)	2.87	2.19	2.40	2.57	2.68	2.92	3.04	3.09
		Maximum drawdown (percent)	\$23.59	\$14.08	\$14.72	\$14.05	\$13.49	\$13.63	\$14.81	\$14.38
	EUR/GBP	Annualized return (including costs) (percent)	28.33	22.06	23.61	24.40	25.77	27.83	27.58	29.45
		Information ratio (including costs)	2.67	2.07	2.21	2.28	2.42	2.61	2.59	2.76
		Maximum drawdown (percent)	\$15.92	\$13.75	\$14.42	\$13.40	\$14.09	\$14.45	\$14.16	\$14.21
	EUR/JPY	Annualized return (including costs) (percent)	30.90	23.92	24.58	26.61	27.84	29.15	31.77	32.36
		Information ratio (including costs)	2.79	2.14	2.20	2.37	2.49	2.61	2.84	2.91
		Maximum drawdown (percent)	\$18.55	\$14.42	\$14.68	\$14.26	\$14.17	\$14.85	\$14.51	\$14.63
01/02/2005... 30/04/2010	EUR/USD	Annualized return (including costs) (percent)	31.05	20.49	21.03	25.37	27.72	30.51	32.00	33.46
		Information ratio (including costs)	2.84	1.88	1.92	2.33	2.53	2.79	2.93	3.07
		Maximum drawdown (percent)	\$24.17	\$17.91	\$18.46	\$17.05	\$17.88	\$17.60	\$18.59	\$17.96
	EUR/GBP	Annualized return (including costs) (percent)	30.87	22.96	24.85	27.54	27.16	29.83	31.48	33.20
		Information ratio (including costs)	2.89	2.14	2.28	2.56	2.53	2.78	2.93	3.10
		Maximum drawdown (percent)	\$17.25	\$17.49	\$17.22	\$17.95	\$17.20	\$17.51	\$17.72	\$17.38
	EUR/JPY	Annualized return (including costs) (percent)	20.36	20.78	22.45	23.39	25.05	28.71	29.99	32.64
		Information ratio (including costs)	1.86	1.92	2.08	2.17	2.34	2.68	2.79	3.04
		Maximum drawdown (percent)	\$20.15	\$17.74	\$18.53	\$18.86	\$18.02	\$18.22	\$18.84	\$18.49

RG-SVR forecasts is confirmed as the realizations of the DM statistic are negative.⁶

6. Trading performance

Section 5 evaluates the forecasts through a series of statistical accuracy measures and tests. However, statistical accuracy is not always synonymous with "financial profitability". In "financial applications, the practitioners's utmost interest is to produce models that can be translated to profitable trades. It is therefore crucial to further examine the proposed model and evaluate its utility through a trading strategy. The trading strategy applied is to go or stay "long" when the forecast return is above zero and go or stay "short" when the forecast return is below zero. The "long" and "short" EUR/USD, EUR/GBP or EUR/JPY position is defined as buying and selling Euros at the current price

⁶ In this study, we apply the DM test to couples of forecasts (RG-SVR vs. another forecasting model). A negative realization of the DM test statistic indicates that the "first forecast (RG-SVR) is more accurate than the second forecast. The lower the negative value, the more accurate are the RG-SVR forecasts.

respectively. Therefore, the trigger for taking a position is the sign of the daily obtained forecast.

In order to calculate transaction costs, trading positions are needed. Transaction costs for a tradable amount, say USD 5...10 million, are about 1 pip per trade (one way) between market makers. But since the EUR/USD, EUR/GBP and EUR/JPY time series are considered as a series of middle rates, the transaction cost is one spread per round trip. For this dataset a cost of 1 pip is equivalent to an average cost of 0.0074 percent, 0.0117 percent and 0.0091 percent per position for the EUR/USD, the EUR/GBP and the EUR/JPY respectively. The annualized return after transaction costs is simply the annualized return minus the relevant annualized transaction cost. The annualized transaction cost is the annualized number of transactions multiplied with their relevant cost. Table 9 presents the summary of the in-sample trading performance of the models for each exchange rate under study, while Appendix C includes the specification of the trading performance measures used in this paper.

From the results of Table 9, RG-SVR demonstrates superior trading performance in terms of annualized return and information ratio for all exchange rates and in-sample periods. The maximum

Table 10
Summary of out-of-sample trading performance.

Out-of-sample period	Series	Statistic	Best predictor	ARBF-PSO	ε -SVR ₁	ε -SVR ₂	v-SVR ₁	v-SVR ₂	GA-SVM	GA- ε SVR	RG-SVR
03/05/2004... 28/04/2006	EUR/USD	Annualized return (including costs) (percent)	8.17	20.05	10.44	11.06	15.87	17.81	19.90	20.65	24.77
		Information ratio (including costs)	0.76	1.85	0.96	1.02	1.46	1.64	1.83	1.90	2.34
		Maximum drawdown (percent)	\$15.26	\$15.03	\$14.09	\$14.42	\$13.01	\$13.65	\$12.41	\$12.69	\$12.84
	EUR/GBP	Annualized return (including costs) (percent)	10.39	19.87	12.21	14.55	15.40	16.92	18.79	20.26	22.04
		Information ratio (including costs)	1.04	1.91	1.22	1.47	1.54	1.69	1.89	2.03	2.27
		Maximum drawdown (percent)	\$14.36	\$10.85	\$14.60	\$13.85	\$13.88	\$13.93	\$13.86	\$13.10	\$13.28
	EUR/JPY	Annualized return (including costs) (percent)	9.08	18.56	10.96	12.47	15.91	16.53	19.39	20.20	23.65
		Information ratio (including costs)	0.83	1.67	0.99	1.13	1.43	1.50	1.75	1.82	2.13
		Maximum drawdown (percent)	\$15.82	\$13.92	\$14.19	\$14.52	\$14.39	\$14.01	\$13.62	\$13.40	\$13.81
02/05/2006... 30/04/2008	EUR/USD	Annualized return (including costs) (percent)	8.83	18.92	10.29	13.04	12.85	14.47	18.37	19.86	22.19
		Information ratio (including costs)	0.81	1.71	0.94	1.19	1.17	1.32	1.67	1.80	2.02
		Maximum drawdown (percent)	\$15.53	\$14.58	\$14.39	\$14.23	\$13.99	\$13.44	\$14.61	\$14.70	\$14.57
	EUR/GBP	Annualized return (including costs) (percent)	10.10	20.38	13.64	15.58	16.75	18.16	20.14	21.08	24.61
		Information ratio (including costs)	1.12	2.24	1.49	1.70	1.83	1.98	2.21	2.28	2.79
		Maximum drawdown (percent)	\$16.82	\$11.03	\$15.61	\$15.29	\$14.58	\$14.33	\$14.89	\$14.48	\$14.84
	EUR/JPY	Annualized return (including costs) (percent)	9.29	19.04	12.22	12.83	14.80	15.32	18.41	19.62	22.47
		Information ratio (including costs)	0.81	1.63	1.06	1.11	1.28	1.33	1.58	1.69	1.93
		Maximum drawdown (percent)	\$15.93	\$13.14	\$15.30	\$15.46	\$15.04	\$15.53	\$15.62	\$15.45	\$15.24
02/05/2008... 30/04/2010	EUR/USD	Annualized return (including costs) (percent)	10.68	23.15	12.25	14.72	16.50	19.00	21.12	22.95	24.71
		Information ratio (including costs)	1.07	2.34	1.23	1.47	1.66	1.89	2.11	2.30	2.49
		Maximum drawdown (percent)	\$15.29	\$10.23	\$14.08	\$14.15	\$14.06	\$13.48	\$13.63	\$14.85	\$14.30
	EUR/GBP	Annualized return (including costs) (percent)	8.59	20.15	10.06	12.61	13.24	14.97	18.34	19.01	20.65
		Information ratio (including costs)	1.05	2.32	1.18	1.49	1.55	1.76	2.16	2.21	2.44
		Maximum drawdown (percent)	\$16.11	\$11.47	\$15.19	\$15.51	\$15.70	\$14.84	\$14.29	\$15.02	\$15.31
	EUR/JPY	Annualized return (including costs) (percent)	8.74	23.19	12.03	13.35	16.42	17.56	20.02	22.87	24.96
		Information ratio (including costs)	0.70	1.89	0.97	1.08	1.33	1.42	1.63	1.86	2.02
		Maximum drawdown (percent)	\$15.18	\$9.17	\$15.58	\$15.32	\$16.00	\$15.82	\$14.73	\$14.92	\$15.02
03/05/2010... 30/04/2012	EUR/USD	Annualized return (including costs) (percent)	9.54	21.56	11.63	12.48	15.12	16.69	19.09	20.60	23.41
		Information ratio (including costs)	0.86	1.95	1.06	1.15	1.39	1.50	1.75	1.90	2.14
		Maximum drawdown (percent)	\$16.77	\$9.32	\$16.20	\$16.27	\$15.84	\$16.49	\$15.35	\$15.58	\$15.64
	EUR/GBP	Annualized return (including costs) (percent)	9.18	22.19	12.16	12.29	14.76	14.71	18.14	20.77	23.52
		Information ratio (including costs)	1.06	2.59	1.40	1.42	1.73	1.69	2.09	2.37	2.71
		Maximum drawdown (percent)	\$18.52	\$12.93	\$18.06	\$17.17	\$17.95	\$17.63	\$17.22	\$17.53	\$17.39
	EUR/JPY	Annualized return (including costs) (percent)	8.37	20.35	10.25	11.82	12.48	14.02	17.43	18.12	22.08
		Information ratio (including costs)	0.68	1.62	0.79	0.96	1.03	1.15	1.42	1.48	1.80
		Maximum drawdown (percent)	\$14.49	\$10.74	\$14.81	\$14.74	\$15.01	\$14.36	\$14.57	\$14.23	\$14.40

drawdowns of the benchmark models are worse than the RG-SVR ones. Their trading performance in the out-of-sample period is presented in Table 10.

RG-SVR continues to outperform all other SVR forecast combination models in terms of trading efficiency. GA- ε SVR is found to be the second best model in forecasting exercises 1 and 2. RG-SVR presents on average 3 percent higher annualized returns and 0.33 higher information ratios compared to GA- ε SVR in these "rst two simulations. In the other two exercises the results of ARBF-PSO are better than GA- ε SVR, which is ranked third. RG-SVR again achieves higher profits and information ratios than ARBF-PSO on an average of 1.46 percent

and 0.15 respectively. Thus, the proposed methodology clearly outperforms its benchmarks in terms of statistical accuracy and "nancial profitability. The non-genetically optimized SVR methodologies remain less efficient in trading terms compared to their counterparts. But it is interesting to outline the profitability divergence between the different SVR models. For instance, between v-SVR₂ and ε -SVR₁ there is an average difference of 4.55 percent in annualized returns after transaction costs for the three exchange rates. Smaller differences are also evident in the other SVR approaches. The SVR's trading performance appears very sensitive to the parameters optimization process.

7. Concluding remarks

The motivation of this paper is to introduce a RG-SVR model for optimal parameter selection and feature subset combination, when applied to the task of forecasting and trading the EUR/USD, EUR/GBP and EUR/JPY exchange rates. The proposed model genetically searches over a pool of individual forecasts, identifies the optimal feature subsets and finally provides a robust single SVR forecast combination for each exchange rate. This is achieved by applying a fitness function specialized for financial purposes and adopting a sliding window approach. RG-SVR is benchmarked not only against genetically and non-genetically optimized SVRs, but also a robust hybrid NN, the ARBF-PSO.

RG-SVR presents the best performance in terms of statistical accuracy and trading efficiency for all the exchange rates under study. RG-SVR's superiority not only confirms the success of the implemented fitness function, but also validates the benefits of applying GAs to v-SVR models. The results also support prior evidence of ARBF-PSO's efficiency in forecasting and trading the three exchange rates. ARBF-PSO outperforms in all simulations for the traditional SVRs. RG-SVR is far more profitable, though. The large differences in the trading performance of the models under study indicate the sensitivity of SVRs to their parameters optimization processes. In summary, the empirical evidence provided by this study should go some way toward convincing statisticians and practitioners to experiment beyond the bounds of traditional SVR optimization techniques.

Acknowledgment

We thank an anonymous referee for his/her valuable comments.

Appendix A. Theoretical background

In this section follows a short theoretical background on SVR, GAs and the issues of parameter and feature subset selection.

A.1. The ϵ -SVR

If we consider the training data $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, where $x_i \in X \subseteq \mathbb{R}^d, y_i \in Y \subseteq \mathbb{R}, i = 1 \dots n$ and n the total number of training samples, then the SVR function can be specified as:

$$f(x) = w^T \varphi(x) + b \tag{A.1}$$

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. A.1c). The ϵ -sensitive loss function L_ϵ is defined as:

$$L_\epsilon(x_i) = \begin{cases} 0 & \text{if } |y_i - f(x_i)| \leq \epsilon \\ \xi_i, \xi_i^* & \text{if other} \end{cases} \tag{A.2}$$

Eq. (A.2) identifies the predicted values that have at most ϵ deviations from the actual obtained values y_i . The ϵ parameter defines the "tube", while the two slack variables, ξ_i and ξ_i^* , show the distance of y_i and y_i^* from the upper and lower bound of the "tube" respectively (see Fig. A.1a and b).

The goal is to solve the following argument:

$$\begin{aligned} & \text{Minimize } C \sum_{i=1}^n (\xi_i + \xi_i^*) + \frac{1}{2} \|w\|^2 \text{ subject to } \begin{cases} \xi_i \geq 0 \\ \xi_i^* \geq 0 \\ C > 0 \end{cases} \\ & \text{and } \begin{cases} y_i - f(x_i) \leq \epsilon + \xi_i \\ f(x_i) - y_i \leq \epsilon + \xi_i^* \end{cases} \end{aligned} \tag{A.3}$$

The above quadratic optimization problem is transformed in a dual problem and its solution is based on the introduction of two Lagrange multipliers a_i, a_i^* and mapping with a kernel function $K(x_i, x)$:

$$f(x) = \sum_{i=1}^n (a_i - a_i^*) K(x_i, x) + b, \text{ where } 0 \leq a_i, a_i^* \leq C \tag{A.4}$$

SVs are called all the x_i that contribute to Eq. (A.4), thus they lie outside the ϵ -tube, whereas non-SVs lie within the ϵ -tube. Increasing ϵ leads to less SVs selection, whereas decreasing it results to more "flat" estimates. The norm term $\|w\|^2$ characterizes the complexity ("fitness") of the model and the term $\{\sum_{i=1}^n (\xi_i + \xi_i^*)\}$ is the training

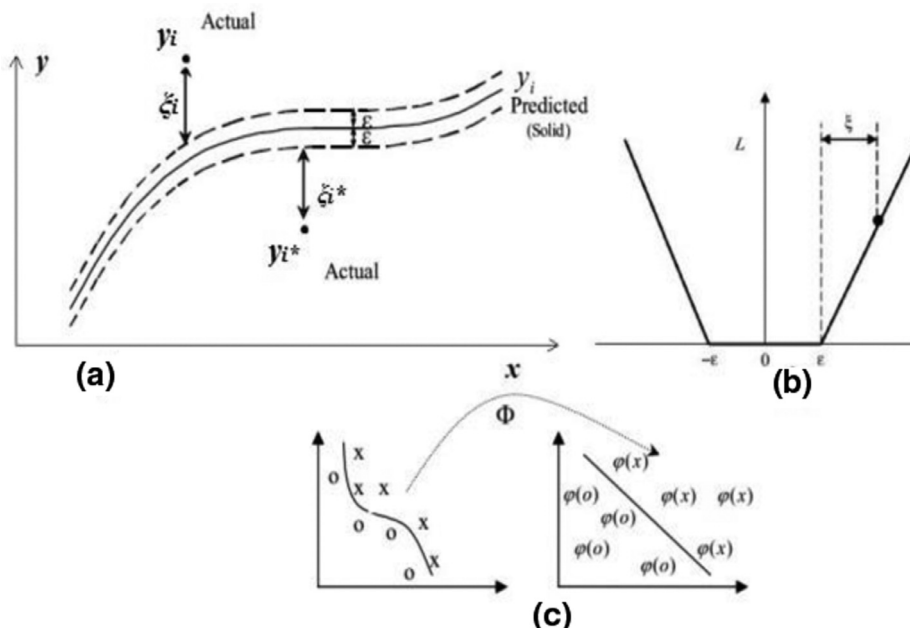


Fig. A.1. (a) The $f(x)$ curve of SVR and the ϵ -tube, (b) plot of the ϵ -sensitive loss function and (c) mapping procedure by $\varphi(x)$.

error, as specified by the slack variables. Consequently the introduction of the parameter ν satisfies the need to trade model complexity for training error and vice versa (Cherkassky & Ma, 2004).

A.2. The ν -SVR

The ν -SVR algorithm encompasses the ε parameter in the optimization process and controls it with a new parameter $\nu \in (0, 1)$ (Basak, Pal, & Patranabis, 2007). In ν -SVR the optimization problem transforms to:

$$\text{Minimize } C \left(\nu \varepsilon + \frac{1}{n} \sum_{i=1}^n (\xi_i + \bar{\xi}_i) \right) + \frac{1}{2} w^2 \text{ subject to } \begin{cases} \xi_i \geq 0 \\ \bar{\xi}_i \geq 0 \\ C > 0 \end{cases} \text{ and } \begin{cases} y_i \tilde{w}^T \varphi(x_i) - \tilde{b} + \varepsilon + \xi_i \\ w^T \varphi(x_i) + b - \tilde{y}_i + \varepsilon + \bar{\xi}_i \end{cases} \quad (\text{A.5})$$

The methodology remains the same as in ε -SVR and the solution takes a similar form:

$$f(x) = \sum_{i=1}^n (a_i \tilde{a}_i) K(x_i, x) + b, \quad \text{where } 0 \leq a_i, \tilde{a}_i \leq \frac{C}{n} \quad (\text{A.6})$$

Based on the ν -trick, as presented by Schölkopf, Bartlett, Smola, and Williamson (1999), increasing ε leads to the proportional increase of the first term of $\left\{ \nu \varepsilon + \frac{1}{n} \sum_{i=1}^n (\xi_i + \bar{\xi}_i) \right\}$, while its second term decreases proportionally to the fraction of points outside the ε -tube. So ν can be considered as the upper bound on the fraction of errors. On the other hand, decreasing ε leads again to a proportional change of the first term, but also the second term's change is proportional to the fraction of SVs. That means that ε will shrink as long as the fraction of SVs is smaller than ν , therefore ν is also the lower bound in the fraction of SVs.

A.3. SVR parameter selection

Although SVR has emerged as a highly effective technique for solving non-linear regression problems, designing such a model can be impeded by the complexity and sensitivity of selecting its parameters. This procedure can be summarized in the following steps:

1. Selection of the kernel function
2. Selection of the regularization parameter C
3. Selection of parameters of the kernel function
4. Selection of the tube size of the ε -sensitive loss function

This selection can be even more complicated and computationally demanding, since individual optimization of the parameters of the above steps is not sufficient. Thus, SVR's performance depends on all parameters being set optimally. Numerous approaches for this optimization have been presented in literature. For example in the ε -SVR, parameter ε can be set simply as a non-negative constant for convenience ($\varepsilon = 0$ or equal to a very small value) (Trafalis & Ince, 2000). This parameter can also be calculated by maximizing the statistical efficiency of a location parameter estimator (Smola, Murata, Schölkopf, & Müller, 1998). Many researchers turn to the ν -SVR approach because it is easier to control parameter ε with parameter ν (Basak et al., 2007; Schölkopf et al., 1999). Cherkassky and Ma (2004) apply RBF kernels in ν -SVR and propose a data-driven choice of parameter C , based on the range of the output values of the training data. But the most popular approach is to use the cross-validation technique (Cao, Chua, & Guan, 2003; Duan et al., 2003) or grid-search algorithms over the dataset (Schölkopf & Smola, 2002).

A.4. Feature selection and GAs

Feature selection is an optimization problem that refers to the search over a space of possible feature subsets in order to find those

that are optimal with respect to specific criteria. Such a problem requires a search strategy that picks the feature subsets and an evaluation method that tests their goodness of fit. Many searching strategies have been proposed in literature, but those who seem to attract more attention are the randomized searches, where probabilistic steps are applied (Sun, Bebis, & Miller, 2004). GAs are commonly used in such cases (Siedlecki & Sklansky, 1989). GAs, formerly introduced by Holland (1975), are search algorithms inspired by the principle of natural selection. They are useful and efficient if the search space is big and complicated or there is not any available mathematical analysis of the problem. A population of candidate solutions, called chromosomes, is optimized via a number of evolutionary cycles and genetic operations, such as crossovers or mutations. Chromosomes consist of genes, which are the optimizing parameters. At each iteration (generation), a fitness function is used to evaluate each chromosome, measuring the quality of the corresponding solution, and the fittest chromosomes are selected to survive. This evolutionary process is continued until some termination criteria are met. In general, GAs can deal with large search spaces and do not get trapped in local optimal solutions like some other search algorithms.

Appendix B. Non-linear models

This appendix section provides a brief description of the k -NN and the NN algorithms applied in this study.

B.1. Nearest Neighbors Algorithm (k -NN)

Nearest Neighbors is a non-linear and non-parametric forecasting method based on the work of Fix and Hodges (1951). It is based on the idea that pieces of time series in the past have patterns which might have resemblance to pieces in the future. Similar patterns of behavior are located in terms of nearest neighbors using a distance called the Euclidean distance and these patterns are used to predict behavior in the immediate future. It only uses local information to forecast and makes no attempt to fit a model to the whole time series at once. The user defines parameters such as the number of neighbors K , the length of the nearest neighbors pattern m and the weighting of final prices in a neighbor α . When α is greater than 1, a greater emphasis is given to similarity between the more recent observations. Huck and Guégan (2005) suggest that a good approximation for choosing the parameters K and m is dependent on the size of the information set. They choose m from the interval:

$$m = [\text{R}(\ln(T)), \text{R}(\ln(T) + 2)] \quad (\text{B.1})$$

where R is the rounding function rounding to the immediate lower figure and T the size of the in-sample dataset. They also suggested that K should be approximately twice the value of m . Thus, for our dataset m lies between 7 and 9 and K lies between 14 and 18.⁷ Based on the above guidelines and Dunis and Nathani (2007) who apply k -NN in financial series, we experiment in the in-sample dataset. The set of parameters selected are those that provide the highest trading performance in the in-sample period.

B.2. Neural Networks (NNs)

The simpler and most popular NN architecture is the Multi-Layer Perceptron (MLP). A standard MLP has 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

⁷ Based on Table 2, the in-sample datasets of forecasting exercises 1, 2, 3 and 4 are 1342, 1341, 1343 and 1341 trading days respectively. The difference between these datasets is very small in terms of observations. Therefore, the rounded range of m (and consequently K) remains the same for all four exercises.

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 network processes information as follows: the input nodes contain the value of the explanatory variables. Since each node connection represents a weight factor, the information reaches a single hidden layer node as the weighted sum of its inputs. Each node of the hidden layer passes the information through a non-linear activation function and passes it on to the output layer if the calculated value is above a threshold. The training of the network (which is the adjustment of its weights in the way that the network maps the input value of the training data to the corresponding output value) starts with randomly chosen weights and proceeds by applying a learning algorithm called back-propagation of errors (Shapiro, 2000).⁸ The maximum number of the allowed back-propagation iterations is optimized by maximizing a fitness function in the test dataset (see Table 2) through a trial and error procedure. More specifically, the learning algorithm tries to find those weights which minimize an error function (normally the sum of all squared differences between target and actual values). Since networks with sufficient hidden nodes are able to learn the training data (as well as their outliers and their noise) by heart, it is crucial to stop the training procedure at the right time to prevent overfitting (this is called "early stopping"). This is achieved by dividing the dataset into 3 subsets respectively called the training and test sets used for simulating the data currently available to fit and tune the model and the validation set used for simulating future values. The network parameters are then estimated by fitting the training data using the backpropagation of errors. The iteration length is optimized by maximizing the forecasting accuracy for the test dataset. Then the predictive value of the model is evaluated applying it to the validation dataset (out-of-sample dataset).

In addition to the classical MLP network, a Recurrent Neural Network is also applied. For an exact specification of recurrent networks, see Elman (1990). A simple recurrent network has 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. Although RNNs require substantially more computational time (see Tenti, 1996), they can yield better results in comparison with simple MLPs due to the additional memory inputs. The third NN model included in the feature space is the Higher Order Neural Network (HONN). HONNs are able to simulate higher frequency, higher order non-linear data, and consequently provide superior simulations. For more information on HONNs see Dunis, Laws, and Sermpinis, (2010, 2011). Psi Sigma Networks (PSNs) are considered as a class of feed-forward fully connected HONNs. First introduced by Ghosh and Shin (1991), the PSN creation was motivated by the need to create a network combining the fast learning property of single layer networks with the powerful mapping capability of HONNs, while avoiding the combinatorial increase in the required number of weights. The order of the network in the context of PSN is represented by the number of hidden nodes. In a PSN the weights from the hidden to the output layer are fixed to 1 and only the weights from the input to the hidden layer are adjusted, something that greatly reduces the training time. More details on the PSN model can be found in Ghosh and Shin (1991).

Table B.1
Neural network inputs.

	MLP	RNN	HONN	PSN
EUR/USD	EUR/USD (1)	EUR/USD (3)	EUR/USD (2)	EUR/USD (1)
	EUR/USD (2)	EUR/USD (5)	EUR/USD (6)	EUR/USD (5)
	EUR/USD (5)	EUR/USD (6)	EUR/USD (7)	EUR/USD (6)
	EUR/USD (6)	EUR/USD (8)	EUR/USD (9)	EUR/USD (8)
	EUR/USD (8)	EUR/USD (10)	EUR/USD (10)	EUR/USD (11)
	EUR/USD (10)	EUR/USD (11)	EUR/USD (12)	EUR/USD (12)
	EUR/USD (11)	EUR/GBP (1)	EUR/GBP (3)	EUR/GBP (1)
	EUR/GBP (1)	EUR/GBP (5)	EUR/GBP (5)	EUR/GBP (3)
	EUR/GBP (7)	...	EUR/JPY (4)	EUR/GBP (4)
	EUR/JPY (2)	...	EUR/JPY (7)	EUR/JPY (4)
EUR/GBP	EUR/USD (2)	EUR/USD (1)	EUR/USD (4)	EUR/USD (3)
	EUR/USD (3)	EUR/USD (2)	EUR/USD (5)	EUR/USD (4)
	EUR/USD (5)	EUR/USD (5)	EUR/USD (7)	EUR/USD (6)
	EUR/USD (7)	EUR/USD (9)	EUR/USD (8)	EUR/USD (7)
	EUR/USD (8)	EUR/USD (10)	EUR/USD (11)	EUR/USD (10)
	EUR/USD (10)	EUR/GBP (5)	EUR/USD (12)	EUR/USD (11)
	EUR/GBP (1)	EUR/GBP (6)	EUR/GBP (1)	EUR/GBP (3)
	EUR/GBP (5)	EUR/JPY (6)	EUR/JPY (2)	EUR/JPY (4)
	EUR/JPY (4)	...	EUR/JPY (4)	...
	EUR/JPY (5)
EUR/JPY	EUR/USD (1)	EUR/USD (2)	EUR/USD (1)	EUR/USD (3)
	EUR/USD (2)	EUR/USD (5)	EUR/USD (2)	EUR/USD (6)
	EUR/USD (5)	EUR/USD (6)	EUR/USD (4)	EUR/USD (7)
	EUR/USD (8)	EUR/USD (7)	EUR/USD (7)	EUR/USD (9)
	EUR/USD (9)	EUR/USD (10)	EUR/USD (8)	EUR/USD (10)
	EUR/USD (12)	EUR/USD (11)	EUR/USD (9)	EUR/GBP (1)
	EUR/GBP (3)	EUR/GBP (2)	EUR/GBP (2)	EUR/GBP (4)
	EUR/GBP (4)	EUR/GBP (3)	EUR/GBP (4)	EUR/JPY (6)
	EUR/JPY (1)	EUR/JPY (5)	...	EUR/JPY (7)
	EUR/JPY (2)

⁸ EUR/USD (1) means that as input is used the EUR/USD exchange rate lagged by one day. Thus, today's closing price is used to forecast the tomorrow's one.

As benchmark to the RG-SVR, this study applies an ARBF-PSO NN model. Its complexity, architecture and characteristics differ from the previous mentioned NNs. Compared to them, in the ARBF-PSO the parameters are optimized through a Particle Swarm Optimization⁹ algorithm. This protects the ARBF-PSO from the dangers of overfitting and data snooping. However, the practitioner still needs to select the network's inputs (in contrast with RG-SVR which is fully adaptive) through a trial and error approach in the in-sample dataset. For a complete description of the ARBF-PSO see Sermpinis et al. (2013). In order the forecasting completion to be fair, the ARBF-PSO has the same pool of inputs as the RG-SVR.

There is no formal theory behind the selection of the NN inputs and their characteristics, such as number of hidden neurons, learning rate, momentum and iterations. For that reason, we conduct NN experiments and a sensitivity analysis on a pool of autoregressive and autoregressive-moving average terms of the series in the in-sample dataset.¹⁰ For example for the number of iterations, our experimentation started from 5.000 iterations and stopped at the 200.000 iterations, increasing in each experiment the number of iterations by 5.000. This is a very common approach in the literature (Tenti, 1996; Zhang, Patuwo, & Hu, 1998). Based on these experiments and the sensitivity analysis, the sets of variables selected are those that provide

⁹ The PSO algorithm is a population based heuristic search algorithm based on the social behavior 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.

¹⁰ We also explored as inputs autoregressive and autoregressive-moving-average terms of other exchange rates (e.g. the USD/JPY and GBP/JPY exchange rates), commodities prices (e.g. Gold Bullion and Brent Oil) and stock market prices (e.g. FTSE100, DJIA, NASDAQ and S&P500). However, the set of inputs presented in Table 2 provide the highest trading performance in the in-sample period.

⁸ Backpropagation networks are the most common multi-layer networks and are the most commonly used type in financial time series forecasting (Kaastra & Boyd, 1996).

Table B.2
Neural network design and training characteristics.

Parameters		MLP	RNN	HONN	PSN	ARBF-PSO
EUR/USD	Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Particle swarm optimization
	Learning rate	0.004	0.002	0.4	0.3	...
	Momentum	0.005	0.003	0.5	0.4	...
	Iteration steps	40 000	30 000	20 000	20 000	...
	Initialization of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)	...
	Input nodes	10	8	10	10	4
	Hidden nodes	8	7	5	6	8
	Output node	1	1	1	1	1
	Hidden node activation function	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \exp(-\frac{z_{\psi} \delta C}{2\sigma^2})$
	Output node activation function	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$
EUR/GBP	Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Particle Swarm Optimization
	Learning rate	0.002	0.003	0.5	0.4	...
	Momentum	0.004	0.005	0.5	0.5	...
	Iteration steps	35 000	30 000	30 000	30 000	...
	Initialisation of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)	...
	Input nodes	10	8	9	8	4
	Hidden nodes	9	7	5	5	7
	Output node	1	1	1	1	1
	Hidden node activation function	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \exp(-\frac{z_{\psi} \delta C}{2\sigma^2})$
	Output node activation function	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$
EUR/JPY	Learning algorithm	Gradient descent	Gradient descent	Gradient descent	Gradient descent	Particle Swarm Optimization
	Learning rate	0.003	0.003	0.5	0.3	...
	Momentum	0.005	0.005	0.5	0.4	...
	Iteration steps	45 000	35 000	30 000	20 000	...
	Initialisation of weights	N(0,1)	N(0,1)	N(0,1)	N(0,1)	...
	Input nodes	10	9	8	9	5
	Hidden nodes	13	11	6	6	5
	Output node	1	1	1	1	1
	Hidden node activation function	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \exp(-\frac{z_{\psi} \delta C}{2\sigma^2})$
	Output node activation function	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$	$F(z_{\psi}) = 1/(1 + e^{\delta z_{\psi}})$	$F(z_{\psi}) = \sum_{\psi=1}^n z_{\psi}$

Note: The input of every node is z_{ψ} , where $\psi = 1, \dots, n$ and n is the number of nodes of the previous layer. The vector indicating the center of the Gaussian function is C and σ is the value indicating its width.

the higher trading performance for each network in the in-sample period. For example, in the "rst forecasting exercise the different sets of inputs of the four NNs for the three series under study are presented in Table B.1.

Table B.2 shows the design and training characteristics of all the above NN architectures for the "rst forecasting exercise.

Appendix C. Statistical and trading performance measures

The statistical and trading performance measures are calculated as shown in Tables C.1 and C.2 respectively.

Table C.2
Trading performance measures and calculation.

Trading performance measures	Description
Annualized return (before transaction costs)	$R^A = 252 \times \frac{1}{N} \times (\sum_{\tau=1}^N R_{\tau})$ where R_{τ} the daily return
Annualized volatility	$\sigma^A = \sqrt{252 \times \frac{1}{N-1} \times \sum_{\tau=1}^N (R_{\tau} - \bar{R})^2}$
Information ratio	$SR = \frac{R^A}{\sigma^A}$
Maximum drawdown	Maximum negative value of $\sum_{\tau=1}^t (R_{\tau} - \bar{R})$ over the period $MD = \min_{i=1, \dots, \tau; \tau=1, \dots, N} (\sum_{j=i}^{\tau} R_j)$

Table C.1
Statistical performance measures.

Statistical performance measures	Description
Mean squared error	$MSE = \frac{1}{N} \sum_{\tau=1}^{t+N} (E(R_{\tau}) - \hat{Y}_{\tau})^2$, with Y_{τ} being the actual value, $E(R_{\tau})$ the forecasted value and N the number of forecasts
Root mean squared error	$RMSE = \sqrt{\frac{1}{N} \sum_{\tau=1}^{t+N} (E(R_{\tau}) - \hat{Y}_{\tau})^2}$

Appendix D. Best predictor

Table D.1 presents the best predictors for the four forecasting exercises and three exchange rates in the in-sample. The overall performance of the best predictor in trading terms is used as benchmark to the SVR and SVM forecast combinations. For example, the best predictor for the EUR/USD exchange rate is composed by the RNN out-of-sample forecasts for the "rst, second and fourth forecasting exercise and the PSN out-of-sample forecasts for the third forecasting exercise.

Table D.1
Best individual forecasting model.

	Forecasting exercise	Best predictor in trading terms	Best predictor in terms of RMSE
EUR/USD	1	RNN	RNN
	2	RNN	RNN
	3	PSN	RNN
	4	RNN	RNN
EUR/GBP	1	PSN	RNN
	2	RNN	RNN
	3	PSN	PSN
	4	RNN	RNN
EUR/JPY	1	RNN	RNN
	2	PSN	PSN
	3	PSN	RNN
	4	PSN	RNN

References

- Basak, D., Pal, S., & Patranabis, D. C. (2007). Support vector regression. *Neural Information Processing, Letters and Reviews*, 11(10), 203...224.
- Cao, L. J., Chua, K. S., & Guan, L. K. (2003). C-ascending support vector machines for financial time series forecasting. In *IEEE international conference on computational intelligence for financial engineering proceedings* (pp. 317...323).
- Cherkassky, V., & Ma, Y. (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1), 113...126.
- Chen, K. Y., & Wang, C. H. (2007). Support Vector Regression with genetic algorithms in forecasting tourism demand. *Tourism Management*, 28(1), 215...226.
- Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics*, 2(1), 134...144.
- Duan, K., Keerthi, S. S., & Poo, A. N. (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51, 41...59.
- Dunis, C. L., Laws, J., & Sermpinis, G. (2010). Modeling and trading the EUR/USD exchange rate at the ECB "xing". *The European Journal of Finance*, 16(6), 541...560.
- Dunis, C. L., Laws, J., & Sermpinis, G. (2011). Higher order and recurrent neural architectures for trading the EUR/USD exchange rate. *Quantitative Finance*, 11(4), 615...629.
- Dunis, C. L., Likothanassis, S. D., Karathanasopoulos, A. S., Sermpinis, G., & Theofilatos, K. A. (2013). A hybrid genetic algorithm...support vector machine approach in the task of forecasting and trading. *Journal of Asset Management*, 14(1), 52...71.
- Dunis, C. L., & Nathani, A. (2007). Quantitative trading of gold and silver using non-linear models. *Neural Network World*, 16(2), 93...111.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179...211.
- Fisman, R. (2001). Estimating the value of political connections. *The American Economic Review*, 91(4), 1095...1102.
- Fix, E., & Hodges, J. L. (1951). *Discriminatory analysis, nonparametric discrimination: Consistency properties: Technical Report 4*. Randolph Field, Texas: USAF School of Aviation Medicine.
- Froot, K. A., Schafarstein, D. S., & Stein, J. C. (1992). Herd on the street: Informational inefficiencies in a market with short-term speculation. *Journal of Finance*, 47(4), 1461...1484.
- Geng, R., Bose, I., & Chen, X. (2015). Prediction of financial distress: An empirical study of listed Chinese companies using data mining. *European Journal of Operational Research*, 241(1), 236...247.
- Ghosh, J., & Shin, Y. (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.
- Holland, J. (1975). *Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence*. Cambridge, MA: MIT Press.
- Huck, N., & Guégan, D. (2005). On the use of nearest neighbors to forecast in finance. *Revue de l'Association française de Finance*, 267...86.
- Hsu, S. H., Hsieh, J. J. P. A., Chih, T. C., & Hsu, K. C. (2009). A two-stage architecture for stock price forecasting by integrating self-organizing map and support vector regression. *Expert Systems with Applications*, 36(4), 7947...7951.
- Huang, S. C., Chuang, P. J., Wu, C. F., & Lai, H. J. (2010). Chaos-based support vector regressions for exchange rate forecasting. *Expert Systems with Applications*, 37(12), 8590...8598.
- Huang, C. L., & Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31(2), 231...240.
- 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.
- Jarque, C. M., & Bera, A. K. (1980). Efficient tests for normality, homoscedasticity and serial independence of regression residuals. *Economics Letters*, 6(3), 255...259.
- Jiang, H., & He, W. (2012). Grey relational grade in local support vector regression for financial time series prediction. *Expert Systems with Applications*, 39(3), 2256...2262.
- Kaasra, I., & Boyd, M. (1996). Designing a neural network for forecasting financial and economic time series. *Neurocomputing*, 10(3), 215...236.
- Kim, H. S., & Sohn, S. Y. (2010). Support vector machines for default prediction of SMEs based on technology credit. *European Journal of Operational Research*, 20(B), 838...846.
- Kwon, Y. K., & Moon, B. R. (2007). A hybrid neurogenetic approach for stock forecasting. *IEEE Transactions on Neural Networks*, 18(3), 851...864.
- Lee, Y., Lin, Y., & Wahba, G. (2004). Multicategory support vector machines. *Journal of the American Statistical Association*, 99(465), 67...81.
- Liu, Y., & Shen, X. (2006). Multicategory ψ -learning. *Journal of the American Statistical Association*, 101(474), 500...509.
- Lo, A. W. (2000). Finance: A selective survey. *Journal of the American Statistical Association*, 95(450), 629...635.
- Lu, C. J. (2014). Sales forecasting of computer products based on variable selection scheme and support vector regression. *Neurocomputing*, 128(27), 491...499.
- Lu, C. J., Lee, T. S., & Chiu, C. C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, 47(2), 115...125.
- Martens, D., Baesens, B., van Gestel, T., & Vanthienen, J. (2007). Comprehensive credit scoring models using rule extraction from support vector machines. *European Journal of Operational Research*, 183(3), 1466...1476.
- Lin, K. P., & Pai, P. F. (2010). A fuzzy support vector regression model for business cycle predictions. *Expert Systems with Applications*, 37(7), 5430...5435.
- Min, S. H., Lee, J., & Han, I. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. *Expert Systems with Applications*, 31(3), 652...660.
- Pai, P. F., Lin, C. S., Hong, W. C., & Chen, C. T. (2006). A hybrid support vector machine regression for exchange rate prediction. *International Journal of Information and Management Sciences*, 17(2), 19...32.
- Pesaran, M. H., & Timmermann, A. (1992). A simple nonparametric test of predictive performance. *Journal of Business and Economic Statistics*, 10(4), 461...465.
- Schölkopf, B., Bartlett, P., Smola, A. J., & Williamson, R. (1999). Shrinking the tube: A new support vector regression algorithm. In M. J. Kearns (Ed.), *Advances in neural information processing systems: 11* (pp. 330...336). Cambridge, MA: MIT Press.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*. Cambridge: MIT Press.
- Schölkopf, B., Smola, A. J., Williamson, R., & Bartlett, P. (2000). New support vector machines. *Neural Computation*, 12(5), 1207...1245.
- Sermpinis, G., Theofilatos, K. A., Karathanasopoulos, A. S., Georgopoulos, E. F., & Dunis, C. L. (2013). Forecasting foreign exchange rates with adaptive neural networks using radial-basis functions and particle swarm optimization. *European Journal of Operational Research*, 223(3), 528...540.
- Shapiro, A. F. (2000). *A Hitchhiker's guide to the techniques of adaptive non-linear models*. Insurance: Mathematics and Economics, 26(2...3), 119...132.
- Siedlecki, W., & Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5), 335...347.
- Smola, A. J., Murata, N., Schölkopf, B., & Muller, K. R. (1998). Asymptotically optimal choice of ϵ -loss for support vector machines. In L. Niklasson, M. Boden, & T. Ziemke (Eds.), *Proceedings of the international conference on artificial neural networks, perspectives in neural computing* (pp. 105...110). Berlin: Springer.
- Sun, Z., Bebis, G., & Miller, R. (2004). Object detection using feature subset selection. *Pattern Recognition*, 37(11), 2165...2176.
- Suykens, J. A. K., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing*, 48(1...4), 85...105.
- Tenti, P. (1996). Forecasting foreign exchange rates using recurrent neural networks. *Applied Artificial Intelligence*, 10(6), 567...582.
- Trafalis, T. B., & Ince, H. (2000). Support vector machine for regression and applications to financial forecasting. *IEEE-INNS-ENNS International Joint Conference on Neural Networks*, 1(6), 348...353.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer.
- Wu, S., & Akbarov, A. (2011). Support vector regression for warranty claim forecasting. *European Journal of Operational Research*, 213(1), 196...204.
- Wu, Y., & Liu, Y. (2007). Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479), 974...983.
- Wu, C. H., Tzeng, G. H., Goo, Y. J., & Fang, W. C. (2007). A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy. *Expert Systems with Applications*, 32(2), 397...408.
- Yao, X., Crook, J., & Andreeva, G. (2015). Support vector regression for loss given default modeling. *European Journal of Operational Research*, 242(2), 528...538.
- Yeh, C. Y., Huang, C. W., & Lee, S. J. (2011). A multiple-kernel support vector regression approach for stock market price forecasting. *Expert Systems with Applications*, 38(3), 2177...2186.
- Yuan, F. C. (2012). Parameters optimization using genetic algorithms in support vector regression for sales volume forecasting. *Applied Mathematics*, 3(10), 1480...1486.
- Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14, 35...62.