



Automatic optimized support vector regression for financial data prediction

Dana Simian¹ · Florin Stoica¹ · Alina Bărbulescu²

Received: 18 July 2017 / Accepted: 25 April 2019 / Published online: 6 May 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

The aim of this article is to introduce a hybrid approach, namely optimal multiple kernel–support vector regression (OMK–SVR) for time series data prediction and to analyze and compare its performances against those of support vector regression with a single RBF kernel (RBF-SVR), gene expression programming (GEP) and extreme learning machine (ELM) on the financial series formed by the monthly and weekly values of Bursa Malaysia KLCI Index, monthly values of Dow Jones Industrial Average Index (DJIA) and New York Stock Exchange. Our method provides an optimal multiple kernel and optimal parameters in Support Vector Regression, improving the accuracy of prediction. The proposed approach is structured on two levels. The macro-level uses a breeder genetic algorithm for choosing the optimal multiple kernel and the SVR optimal parameters. The fitness function of each chromosome is computed in the micro-level using a SVR algorithm. The regression model based on the optimal multiple kernel and optimal parameters is then validated and used for forecasting. The experimental results prove that OMK–SVR performs better than GEP, RBF-SVR and ELM for predicting the future behavior of the study series. A sensitivity study with respect to the number of kernels from the multiple kernel used by OMK–SVR and with respect to the ratio between training and testing data sets was conducted.

Keywords Prediction methods · Support vector regression · Evolutionary computation · Financial forecasting · Genetic algorithms

1 Introduction

Time series modeling and prediction are active topics of research in many areas like meteorology, ecology, finance, signal processing, dynamical systems and statistics. A time series is composed by a finite set of elements observed sequentially over time. The problem of time series prediction consists on finding a function f which predicts future values, x_{t+p} of the data series $\{x_t\}_{t=1}^N$ using past values $X_t = (x_t, x_{t-\tau}, \dots, x_{t-(d-1)\tau})$ where τ is the time delay, d is the embedding dimension or the time window and p is the prediction horizon. Consequently, the predicted

value is given by $x_{t+p} = f(X_t)$. In general, statistical prediction methods [1] cannot capture the nonlinearity of data. Therefore, other nonlinear methods like artificial neural networks (ANNs) [2], support vector regression (SVR) [3–5], gene expression programming (GEP) [6], extreme learning machine (ELM) [7, 8], etc., are being used.

Another problem facing the time series forecast is that the prediction model depends on the data type. Thus, the choice of the methods for forecasting a specific type of data is a problem that is worth to be studied.

Forecasting financial time series is a challenging problem, since the financial environment is continuously changing and the market efficiency strongly influences the predictability. Consequently, different studies regarding the prediction of financial time series are specifically oriented toward different markets: US [9–12], Malaysian [12, 13], Asian [10, 11, 14], Indian [15, 16], European [17, 18], etc. The methods employed evolved from classical statistic ones like exponential smoothing, autoregressive moving average (ARMA) or nonlinear threshold models to modern,

✉ Dana Simian
dana.simian@ulbsibiu.ro
<http://web.ulbsibiu.ro/dana.simian/index.html>

¹ Lucian Blaga University of Sibiu, 10 Victoriei Bd., 550024 Sibiu, Romania

² Ovidius University of Constanta, 124 Mamaia Bd., 900527 Constanța, Romania

heuristic ones based on artificial intelligence (AI) and evolutionary computing (EC) techniques. Artificial neural networks are widely used for financial predictions [13, 19, 20]. The most common ANN algorithm is the back propagation algorithm (BP), but other types of ANN like layer recurrent network (LRN), radial basis network (RBN), generalized regression neural networks were also designed and evaluated in terms of efficiency for financial forecasting [19, 21].

The design of an efficient ANN based on trial basis functions faces the difficulty in selecting a large number of parameters. The unwanted overfitting phenomenon often reduces the generalization capacity of an ANN. Therefore, hybrid solutions appeared in order to circumvent these drawbacks. Hybridization with ARMA-type models [12, 22, 23], evolutionary computation techniques like evolutionary programming [24], genetic algorithms [17, 25, 26] or particle swarm optimization techniques [9], improved the ANN prediction performances. Recently, an algorithm for single-hidden layer feedforward neural network, namely extreme learning machine (ELM), has been proposed in order to overcome the overfitting problems and to increase the generalization performance of back propagation algorithm [6].

A hybrid ARMA–gene expression programming (ARMA–GEP) was used [12] to capture both linear and nonlinear patterns from financial time series. In recent years, many studies focused on designing financial time series forecasting models based on support vector machines (SVMs) [11, 14–16, 18, 27–29]. The comparison between BP and SVM [14] proves that with few exceptions, SVM methods outperform the BP, due to their capability of handling nonlinear data easily. Even better results were obtained using hybrid methods combining SVM with classical statistical methods, ANN or artificial intelligence techniques.

The behavior of forecasting methods depends on the stock market indices chosen for prediction, on the stock markets' characteristics as well as the noise level of the available data.

The aim of this article is to introduce a new approach, namely optimal multiple kernel—support vector regression (OMK–SVR), for time series forecasting and to validate it on financial time series forecasting. The proposed approach is based on multiple SVR kernels built and optimized using hybrid methods. The multiple kernels are able to model both the linear and nonlinear parts of a time series being very suitable for financial time series modeling and prediction. Our hybrid method also allows the automatic choice of the optimal parameters for the SVR model. We test our method against Bursa Malaysia KLCI Index (KLSE), Dow Jones Industrial Average Index (DJIA) and New York Stock Exchange (NYSE). We compare our

method with other forecasting approaches for financial data series (RBF–SVR: SVR with a single RBF kernel, GEP: gene expression programming and ELM: extreme learning machine) in terms of accuracy.

The rest of this paper is organized as follows. Section 2 aims to present the elements of the SVR model necessary to develop our approach. In Sect. 3, we present the proposed OMK–SVR method. In Sect. 4, we report and discuss the experimental results. Comparative studies of performances between the OMK–SVR approach and other methods, based on different performance metrics (mean square error, mean absolute error, correlation actual vs. predicted) showed that the OMK–SVR method outperforms RBF–SVR, GEP and ELM. A sensitivity study of the model parameters is conducted with respect to the value n , the number of single kernels from the multiple kernel and with respect to the ratio between training and testing data sets. Conclusions and further directions of study are formulated in Sect. 5.

2 Support vector regression

SVR is a version of a SVM for regression [3, 4, 22, 30, 31]. SVRs are supervised learning methods. They use a set of training data instances $T = \{(x_i, y_i) | x_i \in X \subseteq R^d, y_i \in R, i = 1, \dots, n\}$ defined by their attributes vector x_i and their target value y_i , in order to produce a model f , which predicts the target values for instances where only features x_j are known. The model is evaluated using a test set of data. The accuracy of the model is defined using an error metric like mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE). For more details on error metrics, see Table 1 from Sect. 4.1.4.

In the linear case, the model is given by $f(x) = \langle w, x \rangle + b$, where $w \in R^d$ is the weight vector, $x \in X$, $b \in R$ is a bias and $\langle \cdot, \cdot \rangle$ denote the inner product in the input space X . The prediction functions produced by SVR are extended on a subset of support vectors, $S \subseteq X$, $|S| = s \leq n$.

SVR minimizes the generalized error, implementing the structural risk minimization principle. The generalized error bound is expressed by means of the regularized risk functional, obtained as a combination between the empirical risk functional and a regularization term that controls the complexity of the hypothesis space [3, 32].

The SVR aims to find a function f that minimizes the regularized risk, $\frac{1}{2} \|w\|^2 + C \sum_{i=1}^n L(y_i, f(x_i))$ where $L(\cdot, \cdot)$ is a ε -insensitive loss function [3, 31], $C > 0$ is regularization constant and $\|\cdot\|$ is the L^2 -norm. The function f has at most ε deviation from the target values y_i for all the training data, being at the same time, as flat as possible.

The problem formulation is given by:

$$\min \left\{ \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \right\}$$

subject to the constraints:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases}$$

The slack variables ξ, ξ^* were introduced to take into account the possibility of an infeasible convex optimization problem. The two parameters, C and ε , control the SVR behavior, and their choice is very important [3, 11, 28]. The parameter ε controls the width of the ε -tube and therefore the number of support vectors. It was proved in [3] that there is a linear dependency between the noise level and the optimal ε parameter for SVR. The constant $C > 0$ determines the tradeoff between the flatness of f and the admitted deviation of the errors from the ε -tube. In many cases, these two parameters are chosen using iterative search grid or improved search grid methods or they are adjusted based on experience and experimental results [33].

In the linear case, the expression of f using the so-called Support Vector Expansion is [3, 4, 31, 32]:

$$f(x) = \sum_{i=1}^s \alpha_i \langle x_i, x \rangle + b \tag{1}$$

The function f is expressed in terms of Lagrange multipliers α_i and the instances $x_i, i = 1, \dots, s$ representing the support vectors. These vectors are characterized by non-zero values of Lagrange multipliers.

The nonlinear regression problems are reduced to linear ones in a higher dimensional feature space by using a mapping Φ . It is not necessary to explicitly know the feature mapping Φ , but it can be implicitly defined by the kernel function $K(u, v) : X \times X \rightarrow R$, having the property that $K(u, v) = \langle \Phi(u), \Phi(v) \rangle$ where $\langle \cdot, \cdot \rangle$ is the inner product in the higher dimensional feature space $\Phi(X)$. This implicit representation is known as “the kernel trick”. Using the kernel function, the expansion of f may be written as:

$$f(x) = \sum_{i=1}^s \alpha_i K(x_i, x) + b \tag{2}$$

A kernel function must satisfy the Mercer’s conditions [3], that is, it has to be a continuous, symmetric, positive and semi-definite function. The most common kernel functions are:

Polynomial of degree $d : K(x_i, x_j) = (\langle x_i, x_j \rangle + r)^d \tag{3}$

Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{4}$

Sigmoid: $K(x_i, x_j) = \tanh(\gamma \langle x_i, x_j \rangle + 1) \tag{5}$

Other functions satisfying Mercer’s theorem can be also used [3, 11, 28, 34, 35]. There are no general criteria to choose a particular kernel. Standard SVRs use a single kernel, and the prediction requires the choice of kernel parameters. The choice of kernel parameters is a difficult task and strongly depends of the field the data come from. In many cases, parameters are tuned by hand based on experience and taking into account experimental results [11, 28]. Recently, there have been many attempts to develop hybrid methods based on genetic algorithms or other EC techniques to automate the SVM kernel parameters choice. Usually, the choice of the kernel is made in an empirical way [11, 28]. The results obtained in the SVM classifiers prove that multiple kernels are able to provide better prediction models for real complex problems [35, 36].

Concluding to achieve satisfactory SVR-based predictions, a good choice of various parameters is crucial: C and ε parameters from the SVR model, kernels’ type and kernels’ parameters. Moreover, single kernels are not appropriate for solving complex real-world prediction problems.

In this article, we introduce a new hybrid method for building optimal multiple SVR kernels and for the automatic selection of optimal SVR model parameters. Our proposed method, namely optimal multiple kernel—support vector regression (OMK–SVR), uses an evolutionary technique based on a breeder genetic algorithm for the choice of an optimal multiple kernel and of the SVR parameters C and ε . The results obtained on financial time series show the superior efficiency of the method compared to existing approaches (e.g., RBF-SVR, GEP and ELM).

3 Optimal multiple kernel–support vector regression (OMK–SVR)

3.1 General presentation

In [35], we developed a general framework for building SVM optimal multiple kernels. The methods derived from this general framework are hybrid methods proving real advantages in classification of nonlinear data and overcoming the difficulties related to the strong dependence of SVM performances on the type of data. In this paper, we extend the principle from [35] for designing optimal multiple SVR kernels for financial time series forecasting. We propose a regression method based on SVR, namely Optimal Multiple Kernel–support vector regression

(OMK–SVR). The main characteristic of the OMK–SVR approach is the use of optimal multiple kernels. Multiple kernels can be obtained by using single kernels and the set of operations (+, *, exp) which preserve Mercer’s conditions [37]. The design of an optimal multiple kernel requires the choice of the single kernels, the operations between the kernels and the parameters defining the single kernels. To optimize the parameters of the multiple kernel and of the SVR, we propose a hybrid method structured on two levels: micro-level and macro-level. In the macro-level, we generate multiple kernels and choose both the optimal kernel and the optimal SVR parameters using a breeder genetic algorithm. In the genetic algorithm, every chromosome encodes the expression of a multiple kernel. The quality of chromosomes (fitness function) is computed in the micro-level, using a SVR algorithm acting on a particular set of data. The fitness function is defined using a precision metric for the SVR prediction accuracy. For more details about the fitness function, see Sect. 3.3.

3.2 Multiple kernel formal representation

A tree structure is used to formally represent the multiple kernel. The terminal nodes contain single kernels while the intermediate ones contain operations from the set of admissible operations (+, ×, exp). An intermediate node containing the operation *exp* will have only one descendent, more precisely, the left one.

In Fig. 1, a multiple kernel composed by four single kernels and three operations from the set of admissible operations is represented. Consequently, the multiple kernel will be of the form:

$$K = (K_1 \text{ op1 } K_2) \text{ op3 } (K_3 \text{ op2 } K_4).$$

We note that it is possible that the optimal multiple kernel be a combination of single kernels of the same type. However, as it can be seen in Sect. 4.2.1, the model generated by the OMK–SVR with all single kernels are of RBF type is different from the model generated by a SVR with only one RBF kernel, even if the parameters of this single kernel are optimized.

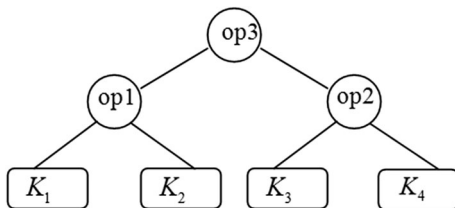


Fig. 1 General representation of multiple kernel

3.3 The macro-level

In the macro-level, the optimal multiple kernel is built using a hybrid procedure. The number *n* of single kernels that compose the multiple kernel is an input data of this level. It is considered arbitrary but fixed, and it is not subject to optimization on the macro-level. The aim of this level is to choose the types of the single kernels, their parameters, the operations used to obtain the multiple kernel and the SVR parameters *C* and ϵ in order to minimize the prediction error. We implemented the macro-level using a breeder genetic algorithm. Genetic algorithms are well-known meta-heuristic search algorithms solving complicated practical optimization problems [38]. In breeder genetic algorithms, the solutions (chromosomes) are represented as vectors of real numbers [39], enabling better modeling of real-world problems and offering advantages in optimization of regression models [40]. In our approach, the aim of the breeder genetic algorithm is to find new values for the parameters of the multiple kernel and also for the parameters used by the SVR algorithm, in order to reach a better prediction. For each single kernel, we will store the operators, the kernel’s type, single kernels parameters and the parameters of the ϵ SVR in the chromosome. The parameters of the single kernel *j* are denoted by *d_j* and *r_j* in the case of a polynomial kernel and γ_j in the case of RBF and sigmoidal kernels. The parameters of the SVR are *C* and *p* where *p* represents the ϵ from the loss function of a SVR. Each parameter is encoded through a gene, using a variable of type real. Therefore, the number of chromosome genes encoding the multiple kernel depends on the number of the single kernels used.

Proposition 1 *The number of the genes from a chromosome encoding a multiple kernel obtained from a complete tree structure composed of $n = 2^k$ single kernels is $5 \times 2^k + 1$, for $k \geq 0$.*

Proof A complete tree with 2^k leaves has *k* intermediary levels, corresponding to $1 + 2 + \dots + 2^{k-1} = 2^k - 1$ intermediary nodes. Therefore, we will have $2^k - 1$ genes for encoding the operations between single kernels. For encoding the kernels type and parameters, we need 4×2^k genes and two genes are necessary for encoding the SVR parameters *C* and *p*. □

In the case of four single kernels, the chromosome which encodes the multiple kernel has 21 genes and its structure is given in Fig. 2.

op	op	op	t1	d1	r1	v1	t2	d2	r2	v2	t3	d3	r3	v3	t4	d4	r4	v4	c	p
1	2	3																		

Fig. 2 Chromosome structure for *n* = 4 single kernels

Both the operators op_i , $i \in \{1, 2, 3\}$, and the single kernels types t_j , $j \in \{1, 2, 3, 4\}$, are mapped to the set $\{1, 2, 3\}$ using, respectively, the one to one functions $\{+, \times, \exp\} \rightarrow \{1, 2, 3\}$ and $\{\text{Pol}, \text{RBF}, \text{SIG}\} \rightarrow \{1, 2, 3\}$, for $j \in 1, \dots, 4$. Thus, a chromosome is an array of real values with length 21.

In the case of eight single kernels, the chromosome has 41 genes.

In the limit case $k = 0$, the multiple kernel reduces to a single kernel and the chromosome has genes encoding the kernel type, the parameters d and r —in the case of a polynomial kernel—and γ —in the case of RBF and sigmoidal kernels, and the SVR model parameters C and p .

In a breeder genetic algorithm, the populations are evolved using specific mutation and crossover operations [39, 40]. A truncated selection is used. The new generation is created using only the $T\%$ best individuals of the initial population of chromosomes. T is a constant of the algorithm. According to [39], T must be chosen between 50 and 10% and typically $T = 40\%$ gives better results. Two individuals from this truncated population are randomly selected and mated using the crossover operator until a new population of individuals is obtained. With a small probability, a mutation operator is then applied to the offspring. The best chromosome (evaluated through the fitness function) remains in the population from one generation to another. The number of the population individuals does not change.

The breeder crossover operator combines two chromosomes $x = \{x_1, \dots, x_n\}$ and $y = \{y_1, \dots, y_n\}$, with $x_i, y_i \in \mathbf{R}$ into a new chromosome $z = \{z_1, \dots, z_n\}$ with $z_i = x_i + \alpha_i(y_i - x_i)$, $i = 1, \dots, n$ and α_i is a random variable uniformly distributed on $[-\delta, 1 + \delta]$. The value of δ depends on the problem to be solved and typically is situated in the interval $[0, 0.5]$.

A value x_i is selected with a small probability p_m for mutation. The probability of mutation is typically chosen as $p_m = 1/n$. The mutation changes the value x_i according to the rule $x_i = x_i + s_i r_i a_i$, with $s_i \in \{-1, 1\}$ randomly uniform. $r_i = r \cdot \|D_{x_i}\|$, $r \in [0.1, 0.5]$, $x_i \in D_{x_i}$, $\|D_{x_i}\|$ is the length of the domain where is situated x_i and $a_i = 2^{-k \cdot \alpha}$ with $\alpha \in [0.1, 0.5]$ randomly uniform and k is the mutation precision (number of bytes used to represent a number in the machine where the breeder algorithm is executed).

3.4 The micro-level

The fitness function for the chromosomes generated in the macro-level is computed in the micro-level. The data is divided into two subsets: the training subset and the test subset. The training subset is used in the micro-level for obtaining the regression models and for computing the values of the fitness function for each chromosome, while

the test subset is used for evaluation of the optimal regression model provided by the breeder genetic algorithm in the macro-level.

In the micro-level, for any chromosome there is a SVR training—prediction session. The fitness function is the Mean Squared Error (MSE) of the prediction provided by the regression model generated through training, using multiple kernel and SVR parameters encoded in each chromosome. Let us consider that the set on which we make the evaluation of MSE consists of n data. Let us denote by $\{p_i\}_{i=1}^n$ the values of the data predicted from the SVR whose kernel structure and parameters are encoded in the chromosome c_k and with $\{a_i\}_{i=1}^n$ the real values of data. Then, the fitness function is given by:

$$f(c_k) = \frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2 \quad (6)$$

The fitness function can be computed using k -fold cross-validation or using the all training subsets (see more details about computation of fitness function in Sect. 4.1.3).

3.5 Model evaluation

At the end of the breeder genetic algorithm, the best chromosome gives the optimal form of the multiple kernel which will be evaluated on the validation subset of data in order to validate the model. After the validation, if the model accuracy is satisfactory, it can be used for forecasting. The model performance is evaluated using different metrics (see Sect. 4.2 for more details about the performance metrics).

3.6 Implementation details

For implementation, we used object oriented programming in the JAVA language. We started from the implementation of the ε -SVR given in the LIBSVM—Library for support vector machines [41], we adapted and enhanced it taking into account the particularities of OMK-SVR approach.

To implement a custom multiple kernel, we started from the JAVA classes implemented in [41] and modified them according to our representation of the multiple kernel. The classes *svm_parameter*, *svm_predict*, *svm_model*, *svm_train* and *Kernel* must be adapted to our particular model. In the class *svm_train*, we added a new version for *set_parameters* method in order to pass additional parameters through the extended *svm_parameter* object to the new version of the *svm_train* method of the class *svm*. The class *svm_predict* was extended with a new *predict* method having as parameters the values extracted from a chromosome in order to build the multiple kernel. The *Kernel* class is modified to accomplish the kernel substitution. A

method for computing the hybrid multiple kernels is necessary. We built a new method, namely $k_function$, for the computation of the single kernels which are then combined using the operations given in the model of the chromosome. In the genetic algorithm, the operations and all parameters assigned to a multiple kernel (type of the single kernels and all other parameters) are obtained from a chromosome, which is then evaluated using the result of the modified *predict* method.

Two other methods of the class *svm_train*, namely *do_cross_validation* and *run*, were adapted to use the error from a training based on tenfold cross-validation as fitness function value in the genetic algorithm.

4 Experimental results and sensitivity study

In this section, we report the results from the experiments conducted for financial time series forecasting using the proposed method, OMK–SVR. Two experiments were conducted. The first one focuses on the evaluation of OMK–SVR performances, and comparison with other forecasting methods (GEP, RBF-SVR, ELM). The goal of the second experiment is to provide a study on parameter sensitivity of the OMK–SVR.

Taking into account the conclusions from [39], in all experiments, in the breeder genetic algorithm from the macro-level of OMK–SVR, we chose the following values for the parameters: $T = 40\%$, $\delta = 0.5$, the probability of mutation $p_m = 0.8$, $r = 0.5$ and $k = 8$.

Our experiments were performed on a Intel(R), Core(TM) i7-5500U CPU @ 2.400 GHz, with 8.00 GB RAM and a 64-bit Operating System.

Financial time series forecasting was performed on the time series of the monthly and weekly KLSE (Bursa Malaysia KLCI) Index, the monthly DJIA (Dow Jones Industrial Average) index and weekly NYSE (New York Stock Exchange). The data for the stock market prediction have been downloaded from *finance.yahoo.com*. For each experiment, we provide a detailed description of the input data in the corresponding subsection.

4.1 Experiment 1

4.1.1 Goal and motivation

In order to evaluate the performance of the proposed OMK–SVR approach, we carried out comparisons with other representative prediction methods: SVR with RBF single kernel (RBF-SVR), gene expression programming (GEP) [12, 42] and extreme learning machine (ELM) [7, 8]. The following reasons led to the choice of these methods for comparisons.

Studies on parameter sensitivity of support vector regression [43] revealed the superior forecasting performance of the RBF kernel against other single kernels. The comparison of OMK–SVR performances with those of RBF-SVR could emphasize the contribution of the multiple kernel in the OMK–SVR model, even if all the single kernels in the OMK–SVR multiple kernel are of RBF type.

The OMK–SVR approach is a hybrid method using EC techniques. GEP also belongs to the field of EC, being automatic model induction techniques.

ELM is a quite recent non-iterative algorithm for single-hidden layer feedforward neural networks. It has good generalization performances and is significantly faster than the BP algorithm. The performances of ELM are dependent on the number of nodes from the hidden layer. Comparison between ELM and SVR with parameters tuned by hand reveals a better behavior of ELM. Therefore, a comparison between OMK–SVR and ELM in term of accuracy is of interest.

4.1.2 Datasets

Experiment 1 is divided into three sub-experiments denoted by Experiment 1a, Experiment 1b and Experiment 1c.

In Experiment 1a, we used KLSE monthly data between December 1993 and March 2015 (256 values) and KLSE weekly data between 03.12.1993 and 16.03.2015 (1106 values). In Experiment 1b, we used DJIA monthly data between January 1985 and March 2015 (368 values). In Experiment 1c, we used NYSE weekly data between December 1965 and March 2018 (2764 values).

In all these experiments, the dataset was split between training and testing sets of data. The reported values from Sect. 4.1.5 use a 95–5% ratio. The influence of the ratio between training and testing data on the prediction performances is studied in Sect. 4.2, proving that the 95–5% ratio gives the best results from the set of ratios {95/5, 80/20, 70/30}.

According to [41, 43, 44], for computational reasons and to speed up the training process, all the data from a given time series were scaled to the interval [0, 1]. High attribute values might lead to numerical problems. Moreover, scaling avoids the domination of the attributes in greater numeric ranges against those in smaller numeric ranges. It is important to use the same scaling method for training and testing data.

4.1.3 Setting for OMK–SVR

The optimal values for the parameters of the SVR multiple kernel, the SVR constant C and the tube size p were obtained using the hybrid method that are presented in Sect. 3. The types of single kernels used in the building

process of the multiple kernel are denoted by $\{1, 2, 3\}$ (i.e. $\{Poly, RBF, Sig\}$). In order to speed up the method, we determined, using a rough grid search, maximal intervals for the parameters C , p and γ (for the RBF and sigmoid single kernels). According to the studies about the SVM with RBF parameters [44, 45], the parameters were forced to lie in the following intervals: $C \in [0.01, 1500]$, $p \in [0.0008, 0.003]$, $\gamma \in [0.01, 500]$.

The characteristics of the macro- and micro-levels of our hybrid method were the following:

- a. At the macro-level, the choice of parameters from the specified intervals was done by the breeder genetic algorithm. The dimension of population was 100, and the number of generations was 300.
- b. At the micro-level, the algorithm computes the fitness function for any chromosome (multiple kernel), on the training set of data. Two different methods were applied:
 - I. Using tenfold cross-validation in the SVR training—prediction session for any chromosome. The fitness function is the error returned by the *run* method from the class *svm_train*.
 - II. Without tenfold cross-validation in the SVR training—prediction session for any chromosome. The fitness function is the error returned by the new *predicts* method from the class *svm_predict*. The *predict* method has as input the parameters of the multiple kernel encoded within the chromosome and makes use of the model constructed in the training session.

We remark that in our experiments, there was no significant difference between the results obtained by applying the OMK–SVR algorithm using tenfold cross-validation in the SVR trainingprediction session for any chromosome or without it. The use of tenfold cross-validation increases the execution time and requires an additional step of generating the prediction model to be used in the validation phase.

We used four single kernels for building the multiple kernel. The sensitivity study made in Sect. 4.2 shows that four single kernels are a reasonable choice to provide enough complexity to the generated model and to avoid overfitting.

For the KLSE monthly series, the automatic choice of optimal multiple kernel and SVR parameters in the OMK–SVR approach have given: operators in multiple kernel $\{2, 2, 2\}$, single kernels types $\{2, 2, 2, 2\}$, single kernels parameters $\gamma \in \{337.428826, 448.9518, 362.14396, 424.6855\}$; $C = 814.5280$ and $p = 0.0021$. The optimal multiple kernel is described by the formula $(RBF1 \times RBF2) \times (RBF3 \times RBF4)$.

For the KLSE weekly series, the automatic choice of optimal multiple kernel and SVR parameters in the OMK–SVR approach have given: operators in multiple kernel $\{2, 1, 2\}$, single kernels types $\{2, 2, 2, 2\}$. The optimal multiple kernel is described by the formula $(RBF1 \times RBF2) \times (RBF3 + RBF4)$, having the parameter γ of the single kernels $\{461.821680, 461854667, 194.0081, 498.8080\}$, $C = 1185.993122$ and $p = 0.001841$.

For DJIA monthly series, the automatic choice of the optimal multiple kernel and SVR parameters in OMK–SVR approach provided: operators in multiple kernel $\{2, 2, 2\}$, single kernels types $\{2, 2, 2, 2\}$, single kernels parameters $\gamma \in \{417.024452, 176.148546, 482.860743, 474.415528\}$; $C = 1454.673004$ and $p = 0.0009$. The optimal multiple kernel is described by the formula $(RBF1 \times RBF2) \times (RBF3 \times RBF4)$.

For the NYSE weekly series, the automatic choice of optimal multiple kernel and SVR parameters in the OMK–SVR approach have given: operators in multiple kernel $\{2, 2, 2\}$, single kernels types $\{2, 2, 2, 2\}$. The optimal multiple kernel is described by the formula $(RBF1 \times RBF2) \times (RBF3 \times RBF4)$, having the parameter γ of the single kernels $\{419.2128, 396.383, 422.5353, 388.0306\}$, $C = 769.124$ and $p = 0.001813$.

The fact that the optimized multiple kernel is composed for all the datasets only by RBF single kernels confirms the results from the empirical sensitive study on the type of single SVR kernels [43]. The empirical sensitivity study on the type of single SVR kernels [43] found that polynomial kernels show consistently an inferior performance. Another drawback of the polynomial kernels is the greater number of hyperparameters compared to RBF and sigmoid kernels. The selection of good parameters for the sigmoid kernel is more difficult than for the RBF kernel, due to the fact that the positive semi-definite condition for the kernel might not be satisfied for some values of the parameter γ [29, 43]. The selection of the RBF kernel by the breeder genetic algorithm confirms the results of previous empirical sensitivity studies on the kernel type in the SVR approach. The breeder algorithm avoids the selection of polynomial and sigmoid single kernels if the stopping criteria (number of generations or time) are large enough.

4.1.4 Settings for the alternative methods used for comparison

The parameters' choice in RBF-SVR is based on a grid search procedure [42, 45]. For the KLSE monthly dataset, the parameters of RBF-SVR are $C = 965.887505$; $\gamma = 0.773866$ and $p = 0.001$. For the KLSE weekly dataset, the RBF-SVR is defined by $C = 2718.74877$, $\gamma = 0.0393$ and $p = 0.001$. For the NYSE weekly dataset, the RBF-SVR is defined by $C = 2928.66286$, $\gamma = 0.24516649$ and $p = 0.001$.

Table 1 Performance metrics for model validation and their formulas

Metrics	Formula	Notations
RMSE (root mean square error)	$\sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2}$	p_i —predicted data
MAE (mean absolute error)	$\frac{1}{n} \sum_{i=1}^n a_i - p_i $	a_i —observed (actual) data
MAPE (mean absolute percentage error)	$\frac{1}{n} \sum_{i=1}^n \frac{ a_i - p_i }{a_i}$	n —number of data
r_{ap} (correlation actual vs. predicted)	$\frac{\sum_{i=1}^n (p_i - m_p)(a_i - m_a)}{\sqrt{\sum_{i=1}^n (p_i - m_p)^2} \sqrt{\sum_{i=1}^n (a_i - m_a)^2}}$	m_a —mean of observed data m_p —mean of predicted data

To employ GEP, the operator rates were used at standard values as proposed in the literature [46–48]. We note that in GEP the number of genes in a chromosome was four, the gene head length was eight, the maximum number of generations was 2000, the number of generations without improvement was 1000, and the linking function was addition. The time delay was $\tau \in \{1, \dots, 5\}$, and the number of independent runs was 100, for each τ . We report the best solution model identified over all runs that was obtained for $\tau = 1$.

ELM was carried out in Java. A number of 30 runs have been conducted, and the average results were reported. We used the sigmoid function as activation function. The number of nodes in the hidden layer was set using a grid search in the interval [100, 11000] such that the root mean square error (RMSE) on the training data set to be minimized. The expression of RMSE is given in Table 1. We used 9000 nodes in the hidden layer for the KLSE monthly series, 1000 nodes for the KLSE weekly series, 10000 for DJIA monthly series and 300 for NYSE weekly data series.

4.1.5 Model validation

The prediction models obtained using OMK–SVR, and the alternative methods are used for forecasting the next values from the financial series. Then, the predicted data are mapped back in the original domain of the data. The model validation is performed using different performance (error) metrics. Four error metrics were computed in this study. These are: the root mean squared error (RMSE), the mean absolute error (MAE), the mean absolute percentage error (MAPE) and the correlation coefficient between actual and predicted data r_{ap} . We note that instead of RMSE, we could use MSE ($MSE = RMSE^2$). Their formulas are given in Table 1. The computation of the metrics values was performed in R. The prediction performance can be influenced by the error metrics [43]. MAE is an easy interpretable error measure, often used in time series prediction. RMSE and MSE are commonly used in forecasting. They give more weight to large errors than MAE. MAPE is a unit-free measure, showing the percentage error. The

correlation coefficient between the real and predicted data measures the quality of fitting between the data predicted from the model and the actual data.

4.1.6 Experiment 1a: results and discussion

The experimental results obtained for KLSE monthly series are presented in Table 2 and those for KLSE weekly series are given in Table 3. The bold values in the next tables represent the best values obtained for a given metric using different techniques.

Analyzing the results for monthly series (Table 2), we see that the OMK–SVR method outperforms all the other approaches taken into account both on training and validation datasets for monthly series. The performances of RBF-SVR and GEP are substantially the same. ELM gave better results on the validation data set than on the training data set. Compared to RBF-SVR and GEP, the accuracy of ELM is worse on the training data series but is much better for the validation data sets. These results show that indeed the ELM can overcome the overfitting drawbacks of neural networks. However, OMK–SVR outperforms ELM both in training and validation datasets for the monthly series. The RMSE ratio between ELM and OMK–SVR is 3.0165 for the training set and 2.7438 for the validation set of data. The results in terms of RMSE, MAE and MAPE are quite similar. The results of all methods are comparable in terms of r_{ap} for the training set of data, but for the validation set the OMK–SVR approach is far superior.

To illustrate the OMK–SVR behavior, in Fig. 3, we present the chart of true and approximated monthly scaled data, in the training and validation process. The method performs well in both cases.

For weekly series (Table 3), the results of all methods for training data are comparable in terms of r_{ap} . For the training set of data, RBF-SVR and GEP gave better results on the training datasets with respect to RMSE and MAE, but on the validation set OMK–SVR gave the best results. With respect to MAPE, the best results on the training and validation sets for the monthly and weekly series have been obtained by using OMK–SVR.

Table 2 Comparative performances results for KLSE monthly

	OMK–SVR	RBF-SVR	GEP	ELM
Training				
r_{ap}	0.992	0.970	0.970	0.928
RMSE	44.084	61.344	61.343	132.979
MAE	29.345	44.868	45.465	99.779
MAPE	0.035	5.341	5.393	0.116
Validation				
r_{ap}	0.916	– 0.832	– 0.919	– 0.193
RMSE	49.309	301.557	388.576	51.599
MAE	39.232	258.542	328.562	41.840
MAPE	0.022	14.695	18.630	0.0232

Table 3 Comparative performances results for KLSE weekly

	OMK–SVR	RBF-SVR	GEP	ELM
Training				
r_{ap}	0.993	0.994	0.994	0.905244
RMSE	42.519	26.193	26.198	200.7746
MAE	29.427	18.676	18.712	162.3963
MAPE	0.033	2.163	2.162	0.162569
Validation				
r_{ap}	0.911	0.877	– 0.906	– 0.23284
RMSE	23.472	208.786	274.046	61.82019
MAE	20.212	172.460	227.528	51.63312
MAPE	0.011	9.800	12.918	0.029452

This aspect is very important, because MAPE is an indicator widely used in statistics for testing the prediction accuracy of a forecasting method. The closer MAPE is to zero, the better the prediction. ELM performs worse on the training set of data, but outperforms RBF-SVR and GEP on validation data. The experiments performed for monthly and weekly data series showed that the OMK–SVR algorithm learns well the data and predicts very well the values for a horizon of 7 months, and respectively, 9 weeks, which is a quite good forecasting period for a financial index. From the point of view of finance, when dealing with the stock market, the prediction of the indices on a short period is very important since trading the futures is done many times in real time.

4.1.7 Experiment 1b: results and discussion

The experimental results obtained for the DJIA monthly series are presented in Table 4. Analyzing the results from Table 4, we get similar conclusions as for Experiment 1a.

The OMK–SVR method outperforms RBF-SVR and GEP both on training and validation datasets for the DJIA monthly series. The performances of RBF-SVR and GEP are essentially the same. Both algorithms perform well enough on the training set, as we can see from the high values of the correlation coefficient, close to 1. They have far worse results on the validation data, showing that their prediction power is low. ELM has better results on the validation data than on the training data. ELM outperforms RBF-SVR and GEP on validation dataset, but has far worse results than OMK–SVR.

To illustrate the OMK–SVR behavior on DJIA monthly series, in Fig. 4, we present the chart of real and predicted scaled data, in the training and validation process. The method performs well in both cases.

4.1.8 Experiment 1c: results and discussion

The experimental results obtained for the NYSE weekly series are presented in Table 5.

Analyzing the results from Table 5, we get quite similar conclusions as for the KLSE weekly data series. The results of all methods, in terms of r_{ap} , are comparable for training data. On the training set of data, GEP gave better results with respect to RMSE and MAE, but on the validation set OMK–SVR gave the best results. With respect to MAPE, the best results on the training and validation sets have been obtained by using OMK–SVR. This is important because MAPE is an important indicator for testing the prediction accuracy of a forecasting method.

As in the previous experiments, ELM performs worse on the training set of data but outperforms RBF-SVR and GEP on validation dataset, with respect to all metrics taken into account. The worse results on the validation data set are given by RBF-SVR. This proves that single kernels are not always able to model complex data series.

4.2 Experiment 2: sensitivity study

The goal of this experiment is to provide a sensitivity analysis of the proposed method, OMK–SVR. We investigate the forecasting performance of alternative parameter setups. The parameters involved in the OMK–SVR approach can be grouped into two categories as they are automatically obtained in the hybrid procedure specific to the method or are independent of it. To the first category belong, the type of single kernels composing the multiple kernel, the hyperparameters of the single kernels (d and r for a polynomial kernel and γ for RBF and sigmoidal kernel) and the parameters C and ε of the SVR. The parameters that are not automatically selected are the number of the single kernels composing the multiple kernel and the ratio between the training and testing data. A

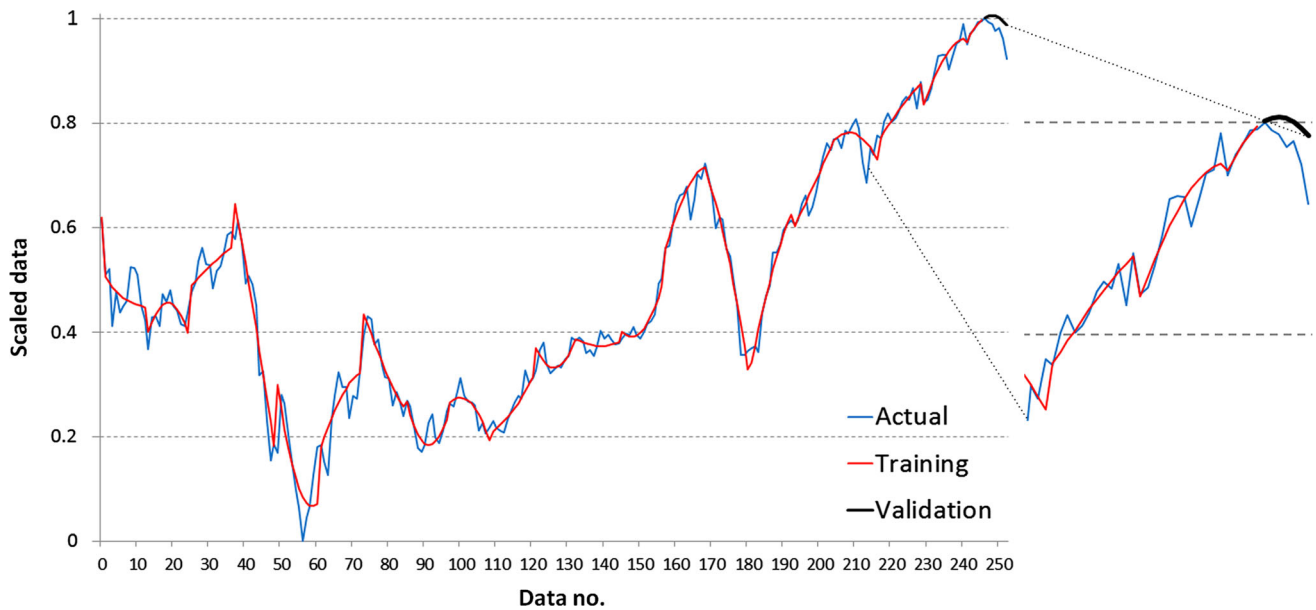


Fig. 3 Behavior of OMK–SVR on training and validation datasets for KLSE monthly series with scaled data

Table 4 Comparative performances results for DJIA monthly

	OMK–SVR	RBF-SVR	GEP	ELM
Training				
r_{ap}	0.997	0.979	0.979	0.981
RMSE	293.930	552.319	550.190	812.662
MAE	178.653	239.821	241.386	599.547
MAPE	0.024	5.380	5.324	0.098
Validation				
r_{ap}	0.937	0.000	0.000	0.735
RMSE	178.067	4181.013	6066.596	1051.338
MAE	132.977	3798.146	5520.890	1001.461
MAPE	0.007	27.577	39.957	0.058

sensitivity study can only be performed on these last parameters. We keep one of these parameters constant and analyze the influence of the other parameter on the prediction performance. The results are reported for the DJIA monthly series between January 1985 and March 2015.

4.2.1 Sensitivity study on the number of single kernels

We keep the ratio between the training and testing data sets constant and equal to 95/5 and modify the number of the single kernels from the multiple kernel. We compared the results obtained for OMK–SVR with multiple kernels composed by four, eight and one kernels. The OMK–SVR with one kernel uses the automatic parameter selection based on the breeder algorithm, whereas the RBF-SVR

uses a grid search procedure in order to determine suitable parameters. We emphasize that in the OMK–SVR with one single kernel, the kernel type is automatically selected in the breeder algorithm. The characteristics of the multiple kernels are given in Table 6.

We use the one to one mappings between the kernel types {Pol, RBF, Sigmoid} and the set {1,2,3}, and respectively, between the set of operations {+, ×, exp} and {1, 2, 3}. Consequently, the optimal multiple kernel in the case of $n = 4$ is given by $(RBF0 \times RBF1) \times (RBF2 \times RBF3)$. In the case of $n = 8$, the optimal multiple kernel is described by the formula $[(K0 \text{ op}0 K1) \text{ op}4 (K2 \text{ op}1 K3)] \text{ op}6 [(K4 \text{ op}2 K5) \text{ op}5 (K6 \text{ op}3 K7)] = [(RBF0 \times RBF1) + (RBF2 \times RBF3)] \times [(RBF4 \times RBF5) \times (RBF6 + RBF7)]$. The results are presented in Table 7.

Analyzing the results from Table 7, one can see that in case of training data, the quality of precision in terms of the correlation coefficient is slightly lower in the case of $n = 1$, showing that more than one kernel is necessary in order to obtain a sufficiently complex model, capable to fit the real-world data. The other error metrics show a better behavior of the multiple kernel with $n = 8$ in the case of training data. The more complex model obtained for $n = 8$ fits better the training data. The results obtained for the testing data reveals better performances for the multiple kernel with $n = 4$ in terms of all error metrics and model fitting. Using only a single kernel does not give enough prediction power to the model, but increasing the number of single kernels too much leads to overfitting. A larger value of n also results in higher computational effort. Therefore, we consider that the value $n = 4$ is a reasonable choice for the number of kernels in the multiple kernel.

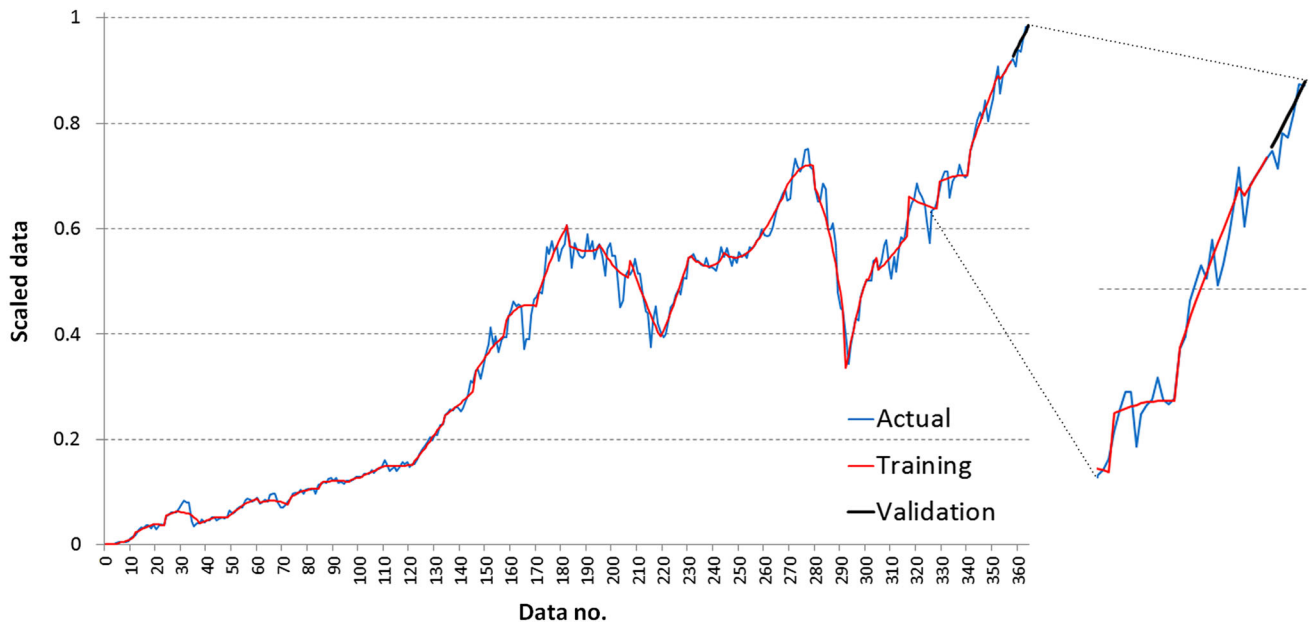


Fig. 4 Behavior of OMK–SVR on training and validation datasets for DJIA monthly series with scaled data

Table 5 Comparative performances results for NYSE weekly

	OMK–SVR	RBF-SVR	GEP	ELM
Training				
r_{ap}	0.998	0.992	0.998	0.983
RMSE	213.278	225.438	115.846	691.83
MAE	118.878	128.76	60.181	458.145
MAPE	0.033	0.136	0.038	0.185
Validation				
r_{ap}	– 0.642	0	0	– 0.0402
RMSE	408.016	2006.44	743.22	408.719
MAE	375.782	1966.66	611.016	381.061
MAPE	0.029	0.155	0.047	0.0318

The last column from Table 7 gives the results obtained using a single RBF kernel with parameters obtained in a grid search selection, as shown in Sect. 4.2. The experimental results show that the hybrid optimization algorithm from OMK–SVR, based on breeder genetic algorithm, gives better performances than RBF-SVR both in training and testing stages. The differences are more significant for the testing dataset proving the better prediction capability of the model with automatic optimization of the kernel and SVR parameters.

All the parameters from the first group of parameters specified in Sect. 4.3 are optimized together in the hybrid optimization algorithm from OMK–SVR. Consequently, a sensitive analysis on these parameters is not necessary.

4.2.2 Sensitivity study on the ratio between training and testing data sets

We keep the number of single kernels from the multiple kernel constant (we set $n = 4$) and consider different values for the ratio between the training and testing datasets: 95/5, 80/20 and 70/30. The results are reported in Table 8.

The correlation between the actual and predicted values is close to 1 for both training and validation, for all scenarios, showing that the models are adequate. The use of r_{ap} as quality index is not enough since it does not provide information about the estimation errors in the models. Therefore, its significance must be interpreted together with other indicators, as RMSE, MAE or MAPE.

MAE is built on the absolute values of the differences between the predicted and recorded values, so it is more clinching point of view of “closeness” of the values to each other. Therefore, smaller the MAE is, better the model is. In our experiments, the smallest MAE corresponds to 95/5 ratio, which is 1.13 times smaller than that for 80/20 ratio and 1.23 times smaller than for 70/30 ratio, for the training set. For the validation set, MAE and 1.48 times smaller than that for 80/20 ratio and 1.90 times smaller than for 70/30 ratio.

Since for computing RMSE, the quadratic estimation errors are considered, the values of RMSE are bigger than those of MAE. Even for the training set, RMSE corresponding to 95/5 ratio is only 1.09 and 1.14 times smaller than those for 80/20 ratio and 70/30 ratio, respectively, the differences increase for the validation sets, being,

Table 6 Characteristics of multiple kernels from OMK–SVR

Number of single kernels n	Types of single kernels	Operations	Kernel no.	Parameter γ of the kernel	C	P
4	(2,2,2,2)	(2,2,2)	K0	417.024452	1454.673004	0.000945
			K1	176.148546		
			K2	482.860743		
			K3	474.415528		
8	(2,2,2,2,2,2,2,2)	(2,2,2,1,1,2,2)	K0	485.486404	1044.728796	0.001239
			K1	111.135712		
			K2	414.700210		
			K3	498.165030		
			K4	492.884978		
			K5	457.300188		
			K6	348.401522		
			K7	271.064743		
1	2	–			1458.811092	0.002165
			499.778996			

Table 7 Comparative performances for different numbers of single kernels

	OMK–SVR four kernels	OMK–SVR eight kernels	OMK–SVR one kernel	RBF-SVR
Training				
r_{ap}	0.997	0.997	0.996	0.979
RMSE	293.930	279.029	348.938	552.319
MAE	178.653	168.082	230.486	239.821
MAPE	0.024	0.023	0.032	5.380
Testing				
r_{ap}	0.937	0.904	0.932	0.000
RMSE	178.067	288.472	237.137	4181.013
MAE	132.977	228.039	185.357	3798.146
MAPE	0.007	0.013	0.010	27.577

respectively, 1.38 (for 80/20 ratio) and 1.71 times bigger (for 70/30 ratio).

By definition, MAPE is built using the absolute values of the ratios between the errors and the recorded values, so for a good model that value should be very close to zero. This is the case for all the experiments that we are analyzing. Even for the training sets, the values of MAPE do not differ significantly, and for the validation sets, MAPE corresponding to 95/5 ratio is 1.57 and two times smaller than that computed for the ratio 80/20 and 70/30, respectively.

Therefore, we can conclude that the ratio 95/5 gives the best results in our experiments.

Table 8 Comparative performances for different ratios between training and testing data sets

	OMK –SVR four kernels 95/5	OMK–SVR four kernels 80/20	OMK–SVR four kernels 70/30
Training			
r_{ap}	0.997	0.996	0.994
RMSE	293.930	319.763	334.205
MAE	178.653	202.008	219.945
MAPE	0.024	0.023	0.023
Validation			
r_{ap}	0.937	0.933	0.930
RMSE	178.067	246.456	305.054
MAE	132.977	197.161	252.168
MAPE	0.007	0.011	0.014

5 Conclusions and future work

A new method for the prediction of financial time series has been introduced in this article. The new approach, OMK–SVR, was tested on many financial time series, providing more accurate predictions than RBF–SVR, GEP and ELM, in terms of many error metrics.

The superior efficiency of the method is given by the use of an optimal multiple kernel. One of the main contributions of this article is the idea of simultaneous optimization of the multiple kernels and the SVR parameters using a breeder genetic algorithm. The breeder genetic algorithm uses the same number of genes to represent each parameter: one gene storing a real-type value. Thus, it overcomes the drawbacks of classical genetic algorithms that require a different number of genes to represent different parameters [35]. The OMK–SVR algorithm simultaneously optimizes the most important parameters of the method: the type and the parameters of the simple kernels composing the multiple kernels, the operations used for building the multiple kernels and the parameters C and ε of the SVR model.

We conducted a sensitivity study with respect to the other two parameters that are not automatically optimized: the number of single kernels that compose the multiple kernel and the ratio between the training and testing data. The experiments demonstrated the superior forecasting performance of multiple kernels composed by $n = 4$ single kernels. Increasing this number leads to overfitting and more computational effort, while decreasing it results in a not sufficiently complex model and low prediction power. The ratio 95/5 between training and testing data has given the best results.

All the experiments were conducted on real data. For validation of our proposed method, we used real financial weekly and monthly data series from markets with different behaviors (Bursa Malaysia KLSE, Jones Industrial Average Index DJIA and New York Stock Exchange NYSE). These are major indices used on the financial markets, whose predictions are important for international financial markets. We point out that even if the series have different behaviors, our algorithm performs better than the competitors. It is important since, in the known literature, the algorithms are generally tested only on the same type of financial series.

For future research, we intend to validate the results of the sensitivity analysis of the OMK–SVR reported in this article on other financial series.

In the training stage (with or without cross-validation), the algorithm uses the mean squared error as fitness function. We shall analyze the influence of others fitness functions on the prediction accuracy.

Acknowledgements The work of the first author, Dana Simian, was supported from the “Lucian Blaga” University of Sibiu research Grant LBUS-IRG-2015-01. The work of the second author, Florin Stoica, was supported from the “Lucian Blaga” University of Sibiu research Grant LBUS-IRG-2015-01.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interests.

References

- Makridakis S, Wheelwright SC, Hyndman RJ (1998) Forecasting methods and applications. Wiley, New York
- Zhang G, Patuwo BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. *Int J Forecast* 14(1):35–62
- Smola AJ, Scholkopf B (2004) A tutorial on support vector regression. *Stat Comput* 14:199–222
- Drucker H, Burges CJC, Kaufman L, Smola A, Vapnik V (1997) Support vector regression machines. *Neural Inf Proc Syst* 9:155–161
- Kang F, Xu Q, Li J (2016) Slope reliability analysis using surrogate models via new support vector machines with swarm intelligence. *Appl Math Model* 40(11–12):6105–6120
- Li Q, Jiang S (2004) Gene expression programming in prediction. In: WCICA 2004. Fifth world congress on control and automation. <https://doi.org/10.1109/wcica.2004.1341971>
- Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. *Neurocomputing* 70:489–501
- Kang F, Liu J, Li J, Li S (2017) Concrete dam deformation prediction model for health monitoring based on extreme learning machine. *Struct Control Health* 24(10):e1997
- Majhi R, Panda G, Sohoo S, Panda A, Choubey A (2008) Prediction of S&P 500 and DJIA stock indices using particle swarm optimization technique. In: Proc IEEE congress on evolutionary computation, pp 1276–1282
- Mehak U, Kamran R, Syed HA, Syed SAA (2016) Stock market prediction using machine techniques. In: Proceedings of the 3rd international conference on communications and information sciences. <https://doi.org/10.1109/iccoins.2016.7783235>
- Allen DE, McAller M, Singh AK (2014) Machine news and volatility: the don jones industrial average and the TRNA sentiment series. E-Prints Complutense. <http://eprints.ucm.es/24356/>. Accessed 1 July 2017
- Bărbulescu A, Băutu E (2012) A Hybrid Approach for Modeling Financial Time Series. *Int Arab J Inf Techn* 9(4):327–335
- Yao J, Poh H-L (1995) Forecasting the KLSE index using neural networks. In: Proceedings of ICNN'95, pp 1012–1017. <https://doi.org/10.1109/icnn.1995.487559>
- Chen W-H, Shih J-Y (2006) Comparison of support-vector machines and back propagation neural networks in forecasting the six major Asian stock markets. *Int J Electron Fin* 1(1):49–67
- Chandwani D, Saluja MS (2014) Stock direction forecasting techniques: an empirical study combining machine learning system with market indicators in the indian context. *Int J Comput Appl* 92(11):8–17
- Choudhry R, Garg K (2008) A Hybrid machine learning system for stock market forecasting. *Eng Technol* 2(3):689–692
- Shazly MRE, Shazly HEE (1999) Forecasting currency prices using genetically evolved neural network architecture. *Int Rev Financ Anal* 8(1):67–82

18. Gestel TV et al (2001) Financial time-series prediction using least squares support vector machines within the evidence framework. *IEEE T Neural Netw* 12(4):809–821
19. Tarsauliya A et al (2010) Analysis of Artificial Neural Network for Financial Time Series Forecasting. *Int J Comput Appl* 9(5):16–22
20. Mizuno H, Kosaka M, Yajima H (1998) Application of neural network to technical analysis of stock market prediction. *Stud Inf Control* 7(2):111–120
21. Wunsch DC, Saad EW, Prokhorov DV (1998) Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE T Neural Netw* 9(6):1456–1460
22. Khandelwal I, Adhikari R, Verma G (2015) Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. *Proc Comput Sci* 48:173–179
23. Merh N, Saxena VP, Pardasani KR (2010) A comparison between hybrid approaches of ANN and ARIMA for Indian stock trend forecasting. *Bus Intel J* 3(2):23–43
24. Yao X, Liu Y (1997) A new evolutionary system for evolving artificial neural networks. *IEEE T Neural Netw* 8(3):694–713
25. Mayer HA, Schwaiger R (1999) Evolutionary and coevolutionary approaches to time series prediction using generalized multi-layer perceptrons. In: *Proceedings of 1999 congress on evolutionary computation—CEC99*, vol 1, pp 275–280. <https://doi.org/10.1109/cec.1999.781936>
26. Flores J et al (2009) Financial time series forecasting using a hybrid neural-evolutionary approach. In: *Proceedings of 15th SIGEF international conference*, Lugo, Spain, pp 547–555
27. Kim KJ (2003) Financial time series forecasting using support vector machines. *Neurocomputing* 55(1–2):307–319
28. Cao L, Tay FEH (2001) Financial forecasting using support vector machines. *Neural Comput Appl* 10:184–192
29. Cao L (2003) Support vector machines experts for time series forecasting. *Neurocomputing* 51:321–339
30. de Oliveira JFL, Ludermitr TB (2014) Iterative ARIMA-Multiple Support Vector Regression models for long term time series prediction. In: *ESANN 2014 proceedings, European symposium on artificial neural networks, computational intelligence and machine learning*. Bruges, Belgium <http://www.i6doc.com/fr/livre/?GCOI=28001100432440>
31. Vapnik V (1995) *The nature of statistical learning theory*. Springer, Berlin
32. Basak D, Pal S, Patranabis DC (2007) Support vector regression. *Neu Inf Pro Lett Rev* 11(10):203–224
33. Huang Q, Mao J, Liu Y (2012) An improved grid search algorithm of SVR parameters optimization. In: *2012 IEEE 14th international conference on communication technology*. <https://doi.org/10.1109/icct.2012.6511415>
34. Diosan L, Oltean M, Rogozan A, Pecuchet JP (2007) Improving SVM performance using a linear combination of kernels. In: *Proceedings of the ICANNGA07, LNCS*, vol 4432, pp 218–227
35. Simian D, Stoica F (2012) A general frame for building optimal multiple SVM kernels. *LNCS* 7116:256–263
36. Gonen M, Alpaydin E (2011) Multiple kernel learning algorithms. *J Mach Learn Res* 12:2211–2268
37. Minh HQ, Niyogi P, Yao Y (2006) Mercer's theorem feature maps, and smoothing learning theory. *Lect Notes Comput Sci* 4005:154–168
38. Stoica F, Cacovean LF (2009) Using genetic algorithms and simulation as decision support in marketing strategies and long-term production planning. In: *Proceedings of 9th international conference on simulation, modelling and optimization (SMO'09)*, pp 435–439
39. Mühlenbein H, Schlierkamp-Voosen D (1994) Predictive models for the breeder genetic algorithm. I Continuous parameter optimization. *Evol Comput* 1(1):25–49
40. Stoica F, Gh Boitor C (2014) Using the Breeder genetic algorithm to optimize a multiple regression analysis model used in prediction of the mesiodistal width of unerupted teeth. *Int J Comput Commun* 9(1):62–70
41. Chang C-C, Lin C-J (2001) LIBSVM: a library for support vector machines. <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
42. Bautu E, Barbulescu A (2013) Forecasting meteorological time series using soft computing methods: an empirical study. *Appl Math Inf Sci* 7(4):1297–1306
43. Crone SF, Lessmann S, Pietsch S (2006) Parameter Sensitivity of Support Vector Regression and Neural Networks for Forecasting. In: *Proceedings of international conference on data mining DMIN'06*, pp 396–402
44. Hsu C-W, Chang C-C, Lin C-J (2016) A practical guide to support vector classification. <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>
45. Han S, Qubo C, Meng H (2012) Parameter selection in SVM with RBF kernel function. In: *Proceedings of the world automation congress 2012, Mexico*
46. Bărbulescu A, Băutu E (2010) Mathematical models of climate evolution in Dobruja. *Theor Appl Climatol* 100(1–2):29–44
47. Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. *Nonl Phen Compl Syst* 13(2):87–129
48. Ferreira C (2006) *Gene expression programming: mathematical modeling by an artificial intelligence*. Springer, Berlin

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.