Applied Soft Computing 12 (2012) 807–818

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A hybrid stock selection model using genetic algorithms and support vector regression

Chien-Feng Huang*

Department of Computer Science and Information Engineering, National University of Kaohsiung, Kaohsiung, Taiwan, ROC

ARTICLE INFO

Article history:
Received 28 January 2011
Received in revised form 1 June 2011
Accepted 23 October 2011
Available online 31 October 2011

Keywords:
Stock selection
Support vector regression
Genetic algorithms
Parameter optimization
Feature selection
Model validation

ABSTRACT

In the areas of investment research and applications, feasible quantitative models include methodologies stemming from soft computing for prediction of financial time series, multi-objective optimization of investment return and risk reduction, as well as selection of investment instruments for portfolio management based on asset ranking using a variety of input variables and historical data, etc. Among all these, stock selection has long been identified as a challenging and important task. This line of research is highly contingent upon reliable stock ranking for successful portfolio construction. Recent advances in machine learning and data mining are leading to significant opportunities to solve these problems more effectively. In this study, we aim at developing a methodology for effective stock selection using support vector regression (SVR) as well as genetic algorithms (GAs). We first employ the SVR method to generate surrogates for actual stock returns that in turn serve to provide reliable rankings of stocks. Top-ranked stocks can thus be selected to form a portfolio. On top of this model, the GA is employed for the optimization of model parameters, and feature selection to acquire optimal subsets of input variables to the SVR model. We will show that the investment returns provided by our proposed methodology significantly outperform the benchmark. Based upon these promising results, we expect this hybrid GA–SVR methodology to advance the research in soft computing for finance and provide an effective solution to stock selection in practice.

1. Introduction

Stock selection has been a challenging and important research area in finance and investment decision-making. This line of research is highly contingent upon reliable prediction of future performance of stocks and successful portfolio construction. Recent advances in computational intelligence and data mining are leading to significant opportunities to solve these problems more effectively. Feasible quantitative models include methodologies stemming from soft computing [1] for prediction of financial time series, multi-objective optimization of expected investment return and risk reduction, and portfolio management – selection of investment instruments based on asset ranking using a variety of input variables and historical data, etc. [2,3]. All these research efforts were in an attempt to facilitate the task of decision-making for investment.

In the research area of stock selection and portfolio optimization, several machine learning methodologies have been developed, including fuzzy systems, artificial neural networks (ANNs), evolutionary algorithms (EAs) as well as support vector machines (SVMs). Earlier work includes several fuzzy approaches; for instance, Chu et al. [4] used fuzzy multiple attribute decision analysis to select stocks for portfolio construction. Analogously, Zargham and Sayeh [5] employed a fuzzy rule-based system to evaluate a set of stocks for the same purpose. Although these fuzzy approaches denote early efforts in employing computational intelligence for financial applications, they usually lack sufficient learning ability.

Quah and Srinivasan [6] studied an ANN stock selection system to choose stocks that are top-ranked performers. They showed their proposed model outperformed the benchmark model in terms of compounded actual returns overtime. Chapados and Bengio [7] also trained neural networks for estimation and prediction of asset behavior in order to facilitate decision-making in asset allocation. Although these models worked in some applications, they often suffer from the overfitting problem and may tend to fall into a local optimum.

For portfolio optimization, Kim and Han [8] proposed a genetic algorithm (GA) approach to feature discretization and the determination of connection weights for ANNs to predict the stock price index. They suggested that their approach was able to reduce the numbers of attributes and the prediction performance was enhanced. In addition, Caplan and Becker [9] employed genetic programming (GP) to develop a stock ranking model for the high

* Tel.: +886 7 5919798; fax: +886 7 5919514.
E-mail address: cfhuang15@nuk.edu.tw

1568-4946/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2011.10.009

technology manufacturing industry in the U.S. More recently, Becker et al. [10] explored various single-objective fitness functions for GP to construct stock selection models for particular investment specifics with respect to risk. In a nutshell, these GP-based models rank stocks from high to low according to a pre-defined objective function.

Because stock market data is highly noisy and complex in dimensionality, it often occurs that most of the aforementioned approaches exhibit inconsistent and unpredictable performance. These challenges arise mainly from the fact that the characteristics and processes of the underlying system that generate time series are generally nonlinear and non-stationary, and for these systems the models solving the relevant applications are usually unknown a priori. An advanced class of novel machine learning algorithms – support vector machines – that improve upon the deficiency of well-known linear techniques for solving these complex applications, was thus developed by Vapnik [11]. As opposed to the traditional empirical risk minimization principle employed by ANNs that minimizes the error on training data, SVMs employ the principle of structural risk minimization that aims to minimize the upper bound of generalization error, and over-fitting is less likely to occur. In general, the optimal solution to SVMs may also be global whereas other neural-network models tend to fall into a local optimal solution. As a result, SVM research thus far has showed that this methodology can outperform other non-linear methods, including neural-network based non-linear prediction, case based reasoning, Linear Discriminant Analysis, Quadratic Discriminant Analysis and Elman Back-propagation Neural Networks [12–15]. In this study, we therefore adopt this methodology for the investment problem investigated here.

Furthermore, even though SVMs have been employed as a popular research methodology in the area of financial applications, most of them focused on the forecast of future direction of either a stock market index or individual stocks [14–18]. Rather than the prediction of financial time series alone, in this study we investigate the task of stock selection using SVMs. This problem is challenging and important in investment, but it is not clear yet how SVMs can be used to advance this research area. Although there exists an earlier attempt using SVMs for this problem by Fan and Palaniswami [19], they solely employed SVMs to classify stocks into winning or losing groups, and this coarse-grained classification procedure usually failed to capture more subtle characteristics of individual stocks. In this study, we will utilize SVMs for regression (support vector regression – SVR) of stock returns, which then serve as surrogates for the actual returns of stocks to imply their quality and relative rankings. Via this improvement, we shall demonstrate SVR as an effective means for stock selection.

However, despite the promising performance of the SVM and SVR in classification and regression, respectively, its success in solving these two problems is highly contingent upon the input variables (features) to the model. Yang and Honavar [20] indicated that several classification issues are determined by the choice of features that describe given patterns presented to a classifier, such as the classification accuracy of the learned classifier, the computational overhead required for learning a classification function, the number of training examples needed for learning, and the cost associated with the features.

The goal of feature selection aims to identify useful, non-redundant subsets of features for a given data mining or machine learning task. By extracting the most essential yet least number of features, one can reduce the computational cost significantly, and construct models that are generalized enough to bring about consistent performance over unseen datasets. Furthermore, since the variables relevant to the SVM/SVR consist of not only the features but also the kernel parameters, it is expected that a successful model along this line of research shall take into consideration these two issues simultaneously.

In the literature, simultaneous optimization on kernel parameters and feature subsets for SVM-based models has been conducted. Fröhlich et al. [21] first presented a study on this problem for SVM by using the GA, in which feature selection was the main research subject. Huang and Wang [22] then presented a different version for this sort of simultaneous optimization and showed that the classification accuracy of their proposed SVM can be improved for several UCI datasets [23]. Due to these promising results, in this stock-selection study, we thus propose to employ a SVR-based model with a hybrid feature selection and parameter optimization methodology by the GA. In our proposed framework, the task of feature selection depends on the learning algorithm that constructs the SVR model, and our scheme shall be categorized as a wrapper approach [24,25], as opposed to a filter approach. The wrapper approach for feature selection is employed in this study because of its improved performance over the filter approach [22–26]. In essence, the optimization method we adopted here is very similar to that proposed by Huang and Wang [22], yet we will demonstrate our main contribution lies in a proper setup that successfully applied this hybrid methodology to stock selection, which is a new SVR application area.

In a nutshell, the methodology we proposed here is to use the SVR to generate reliable surrogates of actual stock returns for stock rankings. Top-ranked stocks are then chosen for portfolio construction. For the simultaneous optimization on model parameters and feature subsets, we employ the GA for this task. We will report the portfolios constructed by our proposed scheme will substantially outperform the benchmark over the long period of time.

This paper is organized into five sections. Section 2 outlines the methods employed in our study. Section 3 describes the research data used in this study. In Section 4, we describe the experimental design and empirical results are reported and discussed. Section 5 presents the conclusions and future research directions.

2. Methodology

This section first reviews the SVM/SVR theory, followed by the description for our proposed stock selection model. Afterwards, model optimization, including parameter optimization and feature selection, will be performed by the GA. The detailed explanations about the SVM and GA theories may be found in the references listed in this paper.

2.1. Support vector machines

The SVM was first proposed by Vapnik [11], which aims to learn a separate function that divides training instances into distinct groups according to their class labels. By this point of view, SVMs form a class of supervised learning models with main applications to solving problems in classification and regression.

2.1.1. Classification

Through mapping input vectors x into a high-dimensional feature space, SVM models constructed in the new space may represent a linear or nonlinear decision boundary in the original space. In the new space, an optimal separation between instances of distinct classes is achieved by the hyperplane that has the maximal distance to the nearest training instances. As a result, SVMs are known as a methodology that generates the maximum margin hyperplane to provide the maximum separation between distinct classes. The maximum margin hyperplane for a given learning problem is uniquely defined by the instances that are closest to it, and these instances are known as *support vectors*. In addition, the separate function can be linear or nonlinear. In the linearly separable case,

the instances can be separated by a linear hyperplane; otherwise, the case is nonlinearly separable.

For the linearly separable case, consider a given set S with n labeled training instances $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Each training instance $x_i \in R^k$, for $i = 1, \ldots, n$, belongs to either of the two classes according to its label $y_i \in \{-1, +1\}$, where k is the input dimension. The maximum margin hyperplane can be represented by the following equation:

$$y = b + \sum w_i y_i x(i) \cdot x, \tag{1}$$

where $\cdot$ represents the dot product; the vector x represents a test example and the vectors $x(i)$s are the support vectors. In this equation, b and w_i are parameters that determine the hyperplane and have to be learned by the SVM.

To obtain an optimal hyperplane, one solves the following convex quadratic programming (QP) problem [11]:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}||w||^2 \\ \text{Subject to} & y_i(w \cdot x_i + b) \geq 1, \quad i = 1, \ldots, n \end{array}. \tag{2}$$

For the nonlinearly separable case, a high-dimensional version of Eq. (1) is represented as follows:

$$y = b + \sum w_i y_i K(x(i), x), \tag{3}$$

where the function $K(x(i), x)$ is defined as the kernel function. There are various kernels for generating the inner products to construct SVMs with different types of nonlinear decision surfaces in the input space. Common choices of the kernel functions include the polynomial kernel $K(x, y) = (xy + 1)^d$, and the Gaussian radial basis function $K(x, y) = \exp(-1/\delta^2(x - y)^2)$, where d is the degree of the polynomial kernel and δ^2 is the bandwidth of the Gaussian radial basis function [14].

2.1.2. Regression

The concept of a maximum margin hyperplane described above only applies to classification. However, the SVM models have been extended for general estimation and prediction problems, including a version of SVM for regression proposed by Drucker et al. [27], which is known as support vector regression (SVR).

The objective of SVR is to find a function that approximates the training instances well by minimizing the prediction error. When minimizing the error, the risk of over-fitting is reduced by simultaneously trying to maximize the flatness of the function. To obtain an optimal hyperplane, one again solves the following quadratic programming problem:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}||w||^2 \\ \text{Subject to} & ||y_i - (w \cdot x_i + b)|| \leq \varepsilon \end{array}, \tag{4}$$

where $\varepsilon \geq 0$ represents the bound for the prediction error.

The above convex optimization problem is feasible in cases where $f = \langle w, x \rangle + b$ actually exists and approximates all pairs (x_i, y_i) with ε precision. To permit some errors in the exchange for model flexibility, one introduces slack variables ξ_i, ξ_i^* to tackle otherwise infeasible constraints of the following optimization problem:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{Subject to} & \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{array}. \tag{5}$$

The constant C determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated. By constructing the Lagrangian function, this optimization problem can be formulated as a dual problem:

$$\begin{aligned} L = {} & \frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) - \sum_{i=1}^{l}\lambda_i(\varepsilon + \xi_i - y_i + \langle w, x_i \rangle + b) \\ & - \sum_{i=1}^{l}\lambda_i^*(\varepsilon + \xi_i^* + y_i - \langle w, x_i \rangle - b) \\ & - \sum_{i=1}^{l}\lambda_i(\eta_i\xi_i + \eta_i^*\xi_i^*), \quad \text{and} \quad \lambda_i, \lambda_i^*, \eta_i, \eta_i^* \geq 0. \end{aligned}$$

Solving the Lagrangian, one obtains the optimal solutions w^* and b^*:

$$w^* = \sum_{i=1}^{l}(\lambda_i - \lambda_i^*)x_i,$$

$$b^* = y_i - \langle w^*, x_i \rangle - \varepsilon, 0 \leq \lambda_i \leq C, \quad i = 1, \ldots, l,$$

$$b^* = y_i - \langle w^*, x_i \rangle + \varepsilon, 0 \leq \lambda_i^* \leq C, \quad i = 1, \ldots, l.$$

Analogous to classification, the inner products can be replaced by proper kernels for nonlinear problems. The tradeoff between minimizing the prediction error and maximizing the flatness of the regression function is controlled by enforcing the upper limit C on the absolute value of the coefficients w_is. The upper limit restricts the influence off the support vectors on the shape of the regression function and is a parameter that the user must specify in addition to ε. The larger C is, the more closely the function can fit the data. In the degenerate case where $\varepsilon = 0$, the algorithm simply performs least-absolute-error regression under the coefficient size constraint, and all training instances become support vectors. Conversely, if ε is large enough, the error approaches zero, and the algorithm outputs the flattest that encloses the data irrespective of value of C.

2.2. Predicted stock returns via SVR as surrogates for stock selection

In this study, we are concerned with the relative quality of stocks that can be described by their predicted returns at a future point of time. The predicted stock returns by the SVR are determined by the model parameters and the set of input features to the SVR, including fundamental variables that describe firms' share price rationality, growth, profitability, liquidity, efficiency, and leverage attributes. The predicted returns are then used to determine the rankings of stocks, by which top-ranked stocks are picked to construct the portfolio.

More specifically, the SVR predicts the return of stock i at time t as $y_{i,t}(\boldsymbol{F}, \boldsymbol{\theta}) \in R$, for $i = 1, \ldots, n$, where $\boldsymbol{F}$ and $\boldsymbol{\theta}$ denote the set of input features and kernel parameters used by the SVR. It is worth mentioning that the prediction for the returns of individual stocks does not have to be perfect. Rather, it is crucial that the predicted returns can serve as reliable surrogates of actual returns to facilitate effective stock rankings. Thus given the predicted returns of all stocks, the ranking of a stock can be defined as:

$$\alpha_{i,t}(\boldsymbol{F}, \boldsymbol{\theta}) = \rho(y_{i,t}(\boldsymbol{F}, \boldsymbol{\theta})), \tag{6}$$

where $\alpha_{i,t} \in N$ is the ranking of stock i at time t; $\alpha_{i,t} \geq \alpha_{j,t}$ iff $y_i, t \geq y_{j,t}$.

The task of stock selection can be achieved using these rankings whereby top-ranked m stocks (stocks corresponding to the top m αs) are selected as components of a portfolio. The performance of

a portfolio can be evaluated by averaging the actual returns of the stocks in the portfolio, which is defined as:

$$\overline{R_t} = \frac{1}{m}\sum_{i=1}^{m} R_t(s_{i,t}), \tag{7}$$

where $s_{i,t}$ is the i-th ranked stock at time t; $R_t(\cdot)$ is the actual return for a stock at time t and $\overline{R_t}$ is the average return over all the m stocks in the portfolio at time t.

In this study we use the cumulative total (compounded) return, R_c, to evaluate the performance of a stock selection model, where R_c is defined by the product of average yearly return, $\overline{R_t}$, of the stocks in a portfolio over n consecutive years as:

$$R_c = \prod_{t=1}^{n} \overline{R_t}. \tag{8}$$

To sum up, the SVR-based stock selection algorithm employed here can be described as follows.

Step 1: $i \leftarrow 1$.
Step 2: Use the input data (i.e., $\boldsymbol{F}$, $\boldsymbol{\theta}$) to the SVR and actual returns of stocks of year i to find the support vectors for the instance sets.
Step 3: Use the input data ($\boldsymbol{F}$, $\boldsymbol{\theta}$) of year $i+1$ to compute the predicted yearly returns of stocks for stock ranking by Eq. (6).
Step 4: Pick top m stocks according to the rankings obtained in Step 3. Compute the average yearly return of these selected stocks by Eq. (7).
Step 5: $i \leftarrow i+1$; repeat Steps 2–4 until $i = n-1$.
Step 6: Compound the average yearly returns by Eq. (8) to obtain the cumulative total return of the portfolio over the n years.

2.3. Model optimization

The performance of the SVR-based stock selection model is determined by the input features $\boldsymbol{F}$ and the parameters $\boldsymbol{\theta}$ to the SVR. Therefore, we expect that the optimization of $\boldsymbol{\theta}$ as well as subsets of $\boldsymbol{F}$ shall enhance the effectiveness of the SVR model. In this study, we propose to use genetic algorithms for simultaneous optimization with respect to these two tasks. In the next subsections we describe the basics of the GA, the motivation of using this technique, and the relevant optimization scheme for our stock selection model.

2.3.1. Genetic algorithms

Genetic algorithms were developed by Holland [28] and have been used as computational models of natural evolutionary systems and as adaptive algorithms for solving optimization problems. GAs operate on an evolving population of artificial organisms, or agents. Each agent is comprised of a genotype (often a binary string) encoding a solution to some problem and a phenotype (the solution itself). GAs regularly start with a population of randomly generated agents within which solution candidates are embedded. In each iteration, a new generation is created by applying variations, such as crossover and mutation, to promising candidates selected according to probabilities biased in favor of the relatively fit agents. As a result, evolution occurs by iterated stochastic variation of genotypes, and selection of the best phenotypes in an environment according to how well the respective solution solves a problem (or problem-specific fitness function). Successive generations are created in the same manner until a well-defined termination criterion is met. The core of this class of algorithms lies in the production of new genetic structures along the course of evolution, thereby providing innovations to solutions for the problem at hand. The steps of a simple GA are shown in the following:

Step 1: Randomly generate an initial population of l agents, each being an n-bit genotype (chromosome).
Step 2: Evaluate each agent's fitness.
Step 3: Repeat until l offspring have been created.
(a) select a pair of parents for mating;
(b) apply variation operators (crossover and mutation);
Step 4: Replace the current population with the new population.
Step 5: Go to Step 2 until terminating condition.

2.3.2. Chromosome encoding and fitness function

Among many paradigms of search algorithms GAs have been proven to have an advantage over traditional optimization methods in problems with many complex, discontinuous constraints in the search space. This methodology contributes for a global, population-based search in the search space, in contrast with the kind of local, greedy search conducted by most rule-induction and decision-tree algorithms. Lower computation cost is a general advantage of local, greedy search algorithms. However, the solution quality achieved by these algorithms can be greatly degraded if there exists a considerable degree of feature interactions, which is usually the case for real-world problems. Since GAs can be designed to perform a global search for various combinations of sets of features that improve given optimization criteria, this class of algorithms are expected to cope better with feature interaction problems.

In addition to several existing results demonstrating the effectiveness of using GAs for feature selection [21,22,29], it is also appealing to use GA's straightforward binary coding scheme to designate allele '1' or '0' to represent a feature being selected or not, respectively. Therefore, in this study, we propose to use the GA to search for optimal subsets of features for the SVR-based stock selection model.

Apart from feature selection, we use the RBF kernel function for the SVR model because it can be used to analyze high dimensional data [22,30,31].[1] In this case, three free parameters, C, δ^2 and ε are to be provided for the SVR model. Notice that for the numerical optimization and tuning on these parameters, other options may be preferable to the GA – for instance, ES (Evolution Strategies, especially Evolution Strategy with Covariance Matrix Adaptation [33]), Sequential Parameter Optimization (SPO) [29,34], and Particle Swarm Optimization (PSO) [35].

It is worth mentioning that by translating features into numerical values, Lin et al. [36] then used the PSO for numerical optimization on feature selection and model parameters, simultaneously. However, with this setup, Lin et al. [36] showed that the performance of the PSO-based SVM is similar to that of the GA-based model in the test problems they studied. Apparently, whether it is beneficial to convert the simultaneous optimization problem to all numerical (convenient for the PSO search) or all combinatorial values (convenient for the GA search) remains an open question.[2] However, as can be seen shortly in our results, feature selection appears to play a more significant role than the parameter optimization alone, we hereby propose to adopt the GA for the overall optimization task due to its straightforward binary encoding scheme and effectiveness for feature selection.

In the encoding design of our proposed scheme, the composition of a chromosome is devised to consist of four portions – the

[1] The other reason we chose the Gaussian kernel is that a Gaussian kernel satisfies Mercer's condition [32], which shall make the system more reliable.

[2] According to the No-Free-Lunch theorems [37], one has no guarantee that an algorithm will perform well on a particular problem without tailoring it to the domain at hand. Therefore, a further study on the characteristics of the stock selection domain shall facilitate the decision for which algorithm to use, and we will leave this as a future research project.

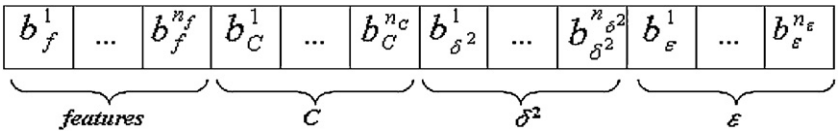

Fig. 1. Chromosome encoding.

candidate feature sets, C, δ^2 and ε. Here we use the binary coding scheme to represent a chromosome in the GA. In Fig. 1, loci b_f^1 through $b_f^{n_f}$ represent candidate features 1 through n, respectively, with allele '1' or '0' indicating the feature being selected or not. The value of parameter C is encoded by loci b_C^1 through $b_C^{n_C}$. The value of δ^2 is encoded by loci $b_{\delta^2}^1$ through $b_{\delta^2}^{n_{\delta^2}}$. As for the value of ε, it is encoded by loci b_ε^1 through $b_\varepsilon^{n_\varepsilon}$.

In our scheme, the chromosome representing the genotypes of parameter C, δ^2 and ε is to be transformed into the phenotype by Eq. (9) for further fitness computation. The precision representing each parameter depends on the number of bits used to encode it in the chromosome, which can be determined as follows:

$$y = \min_y + \frac{d}{2^l - 1} \times (\max_y - \min_y), \tag{9}$$

where y is the corresponding phenotype for the particular parameter; $\min_y$ and $\max_y$ are the minimum and maximum of the parameter; d is the corresponding decimal value, and l is the length of the block used to encode the parameter in the chromosome.

With this encoding scheme, we define the fitness function of a chromosome as the annualized return of the portfolio:

$$\text{fitness} = \sqrt[n]{R_c}, \tag{10}$$

where R_c is the cumulative total return computed by Eq. (8).

Our proposed stock-selection methodology is a multi-stage process, including feature selection and parameter optimization by the GA, support vector regression, stock ranking, selection and performance evaluation. The flowchart of this hybrid algorithm is shown in Fig. 2.

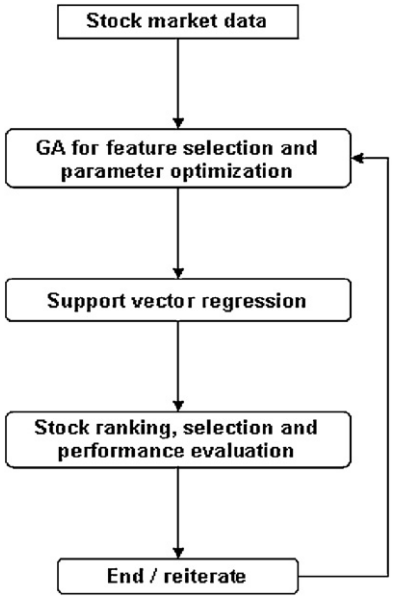

Fig. 2. Flow chart of the hybrid GA–SVR algorithm.

3. Data and fundamental variables

We use the constituent stocks of the 200 largest market capitalizations listed in the Taiwan Stock Exchange as the investment universe. The yearly financial statement data and stock returns used for this research are retrieved from the TEJ (Taiwan Economic Journal Co. Ltd., http://www.tej.com.tw/) database for the period of time from 1996 to 2010. For the choice of fundamental variables, early studies indicated that several financial ratios play key roles in future stock returns. Most of them applied profitability (e.g., ROE, ROA, operating profit margin, and net profit margin), leverage (e.g., DB ratio), liquidity (e.g., current ratio and quick ratio), efficiency (e.g., inventory turnover rate and receivables turnover rate), and growth (e.g., operating rating income growth rate and net income growth rate) related ratios to examine the relationship between fundamentals and stock returns. Mukherji et al. [38], Jensen et al. [39], Danielson and Dowdell [40], Lewellen [41], Fama and French [42], and Hjalmarsson [43] also showed that the ratios relating to share price rationality, e.g., PE, PB, and PS ratios, are likely to influence future stock returns. According to the previous literature, Table 1 provides the aforementioned six attributes that are to be employed for this study, including fourteen financial ratios. For each year, investable stocks are described by these fourteen financial ratios and their historical returns are provided.

4. Empirical results

In this study, standardization was first applied to the research data – every original attribute is scaled into the range of $[-1, 1]$ by subtracting the mean, and dividing the result by the standard deviation. This treatment is to ensure that all the attributes lie in the same parameter range, in order to prevent attributes with large ranges from overwhelming others and prediction errors may be reduced.

For the support vector regression algorithm, stock data of one year is used to train the SVR model, and the trained model is used to predict the next year's return. In the following subsections, four scenarios are examined: (1) SVR using several illustrative values from the parameter ranges suggested by [14,50] and all the features; (2) SVR using parameters optimized by the GA and all the features; (3) SVR using several illustrative values from the parameter ranges suggested by [14,50], but using an optimal subset of features selected by the GA; and (4) SVR using both the optimal parameters and subsets of features optimized by the GA. For the second and third scenarios, the experiments were conducted by disabling the optimization corresponding to the parameters and feature selection, respectively, in the chromosomes. Here we present empirical results on the performance comparison of these stock selection models.

4.1. SVR using non-optimized parameters and all the features

We first compare the cumulative benchmark return (the product of the average yearly returns of the 200 stocks in the investment universe) and the cumulative average return (by Eq. (8)) of longing a number of top-ranked stocks (10, 20 and 30 stocks) using SVR with

Table 1
Attributes used in the stock selection model.

Attribute	Ratios	Description	Refs.
Share price rationality	(1) PE ratio	Price-to-earnings ratio = share price/earnings per share	[38,40,41,43]
	(2) PB ratio	Price-to-book ratio = share price/book value per share	[38–42]
	(3) PS Ratio	Price-to-sales ratio = share price/sales per share	[38]
Profitability	(4) ROE	Return on equity (after tax) = net income after tax/shareholders' equity	[44,45]
	(5) ROA	Return on asset (after tax) = net income after tax/total assets	[44]
	(6) OPM	Operating profit margin = operating income/net sales	[46]
	(7) NPM	Net profit margin = net income after tax/net sales	[45]
Leverage	(8) DE ratio	Debt-to-equity ratio = total liabilities/shareholders' equity	[44]
Liquidity	(9) CR	Current ratio = current assets/current liabilities	[44]
	(10) QR	Quick ratio = quick assets/current liabilities	[44]
Efficiency	(11) ITR	Inventory turnover rate = cost of goods sold/average inventory	[44]
	(12) RTR	Receivables turnover rate = net credit sales/average accounts receivable	[47]
Growth	(13) OIG	Operating income growth rate = (operating income at the current year − operating income at the previous year)/operating income at the previous year	[48]
	(14) NIG	Net income growth rate = (net income after tax at the current year − net income after tax at the previous year)/net income after tax at the previous year	[49]

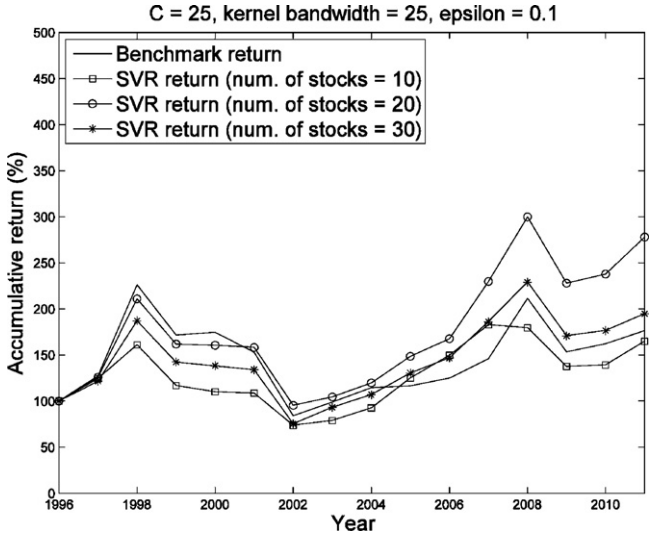

Fig. 3. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR for $(C, \delta^2) = (25, 25)$.

the Gaussian radial basis function.[3] As an illustration, we examine four combinations of $(C, \delta^2) = (25, 25)$, $(25, 100)$, $(100, 25)$, $(100, 100)$ and $\varepsilon = 0.1$ as the model parameters to the SVR. The values of C and δ^2 are chosen from the potentially beneficial range of the SVR parameters suggested by [14,50].[4] Figs. 3 and 4 show the results for the cases of $(C, \delta^2) = (25, 25)$ and $(25, 100)$, respectively. In Fig. 3, as can be seen, longing 20 and 30 stocks outperform the benchmark at the end of year 2010, whereas as shown in Fig. 4, longing 10, 20 and 30 stocks all outperform the benchmark. Figs. 5 and 6 show the results for the cases of $(C, \delta^2) = (100, 25)$ and $(100, 100)$, respectively. In Fig. 5, longing 20 and 30 stocks still outperform the benchmark at the end of year 2010. In Fig. 6, longing 10, 20 and 30 stocks all outperform the benchmark again. These figures thus show that the portfolios constructed by the SVR (without any optimization on the parameters and sets of features yet) can outperform the benchmark.

4.2. SVR using optimized parameters and all the features

The results just presented reveal the necessity of a comprehensive study on the optimization of our stock selection model. We

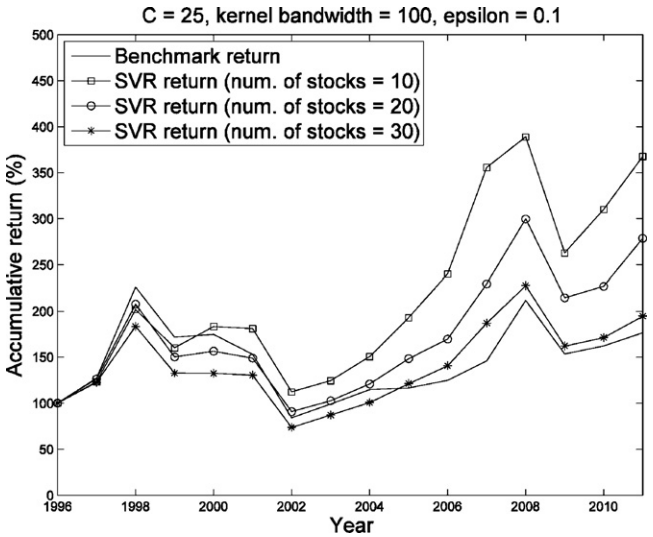

Fig. 4. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR for $(C, \delta^2) = (25, 100)$.

[3] "Longing" is a term used in Finance. In this study, "longing" a stock means buying it and also keeping it for another year.

[4] The value of ε used here is merely for the illustrative purpose. Further results of the optimization on C, δ^2 and ε will be shown in the following subsections.

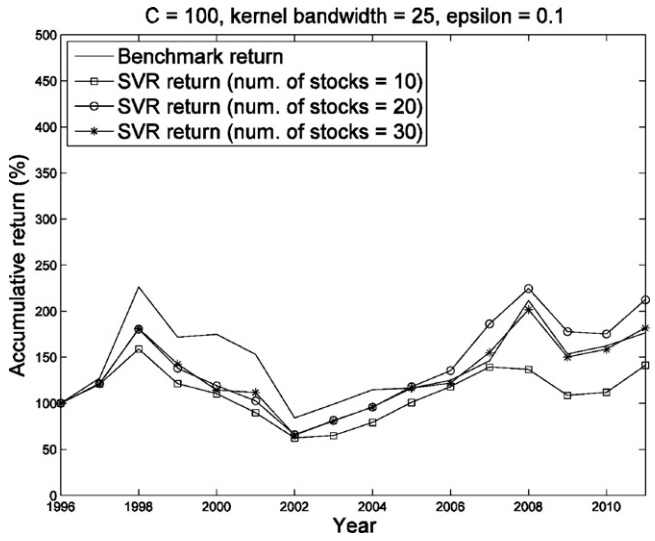

Fig. 5. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR for $(C, \delta^2) = (100, 25)$.

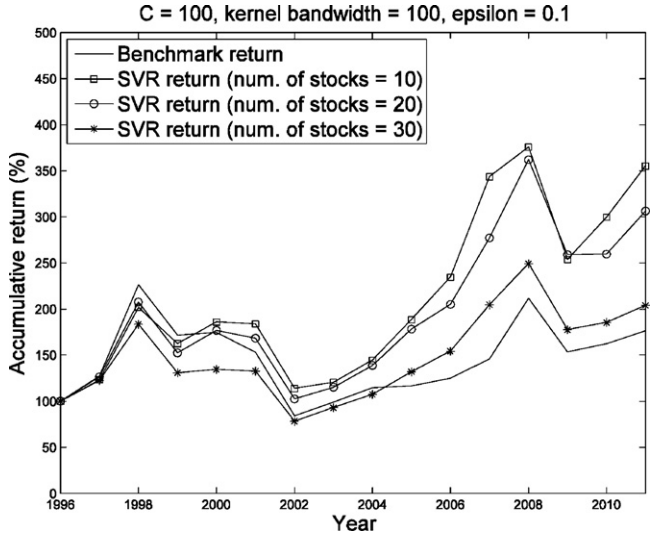

Fig. 6. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR for $(C, \delta^2) = (100, 100)$.

proceed this optimization study step by step. Here we first examine the effects of optimizing the SVR parameters (C, δ^2 and ε) by the GA, while all the features remain to be used by the SVR. Fig. 7 displays the averaged best-so-far values over 50 runs attained by the GA over 70 generations.[5] The averaged best-so-far performance curve is calculated by averaging the best-so-far obtained at each generation for all 50 runs, where the vertical bars overlaying the curves represent the 95% confidence intervals about the means.

Fig. 8 displays an illustration of the cumulative benchmark and model returns of longing a number of top-ranked stocks recommended by the SVR. This figure shows that the portfolios of maintaining 10, 20 and 30 stocks all outperform the benchmark at

[5] In order to study the change of the quality of solutions over time, a traditional performance metric for search algorithms is the "best-so-far" curve that plots the fitness of the best individual that has been seen thus far by generation n for the GA – i.e., a point in the search space that optimizes the objective function thus far. In addition, for the GA experiments here we employed a binary tournament selection [51], one-point crossover and mutation rates of 0.7 and 0.005, respectively. We also used 100 bits to represent each of C, δ^2 and ε.

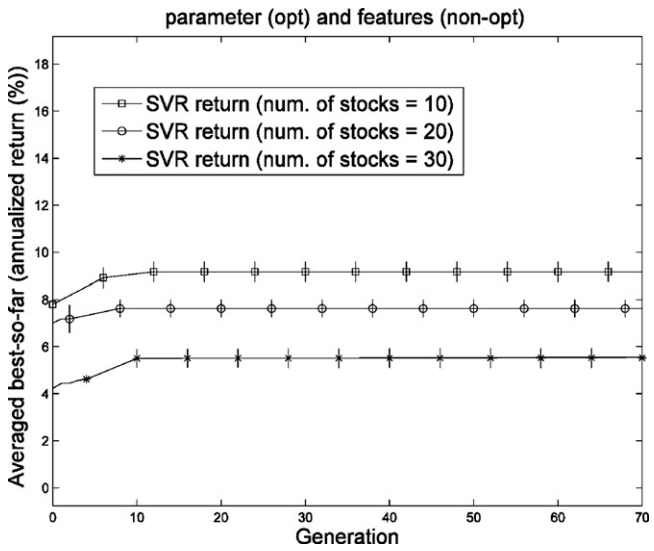

Fig. 7. Best-so-far curves by the GA.

the end of year 2010. As a result, the optimization on SVR's model parameters by the GA can be advantageous to stock selection.

4.3. SVR using non-optimized parameters and optimal subsets of features

Next we examine the models using feature selection by the GA to select (sub-)optimal sets of the input features to the SVR while C, δ^2 and ε are non-optimized. Fig. 9 displays the averaged best-so-far trajectories by the GA over 70 generations. Fig. 10 displays an illustration of the cumulative benchmark and model returns of longing a number of top-ranked stocks recommended by the SVR. As can be seen, the portfolios of maintaining 10, 20 and 30 stocks all significantly outperform the benchmark as time goes by. Therefore, feature selection by the GA can be advantageous to our stock selection model. Comparing with the results in the previous subsection, one can also notice that feature selection is much more advantageous to stock selection than the optimization of model parameters alone.

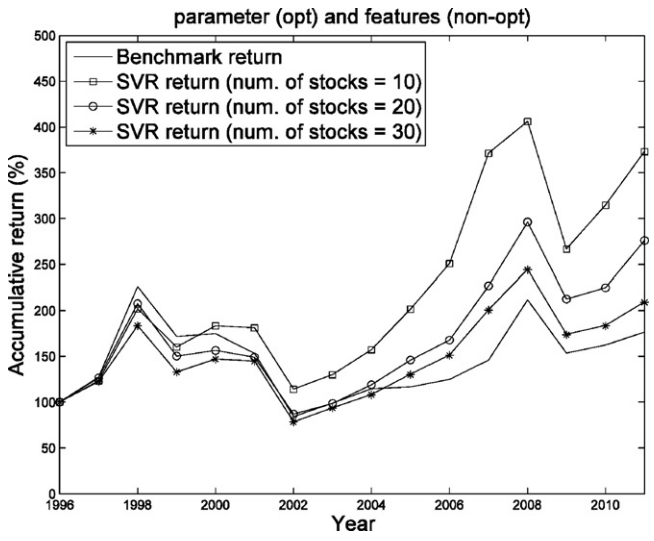

Fig. 8. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR.

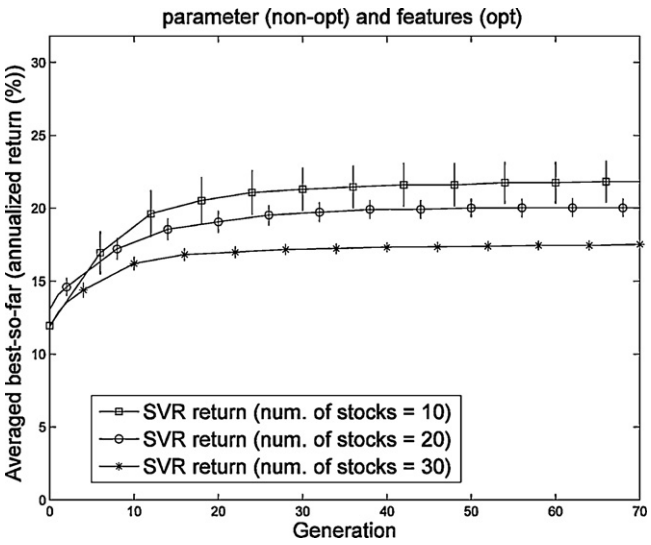

Fig. 9. Best-so-far curves by the GA.

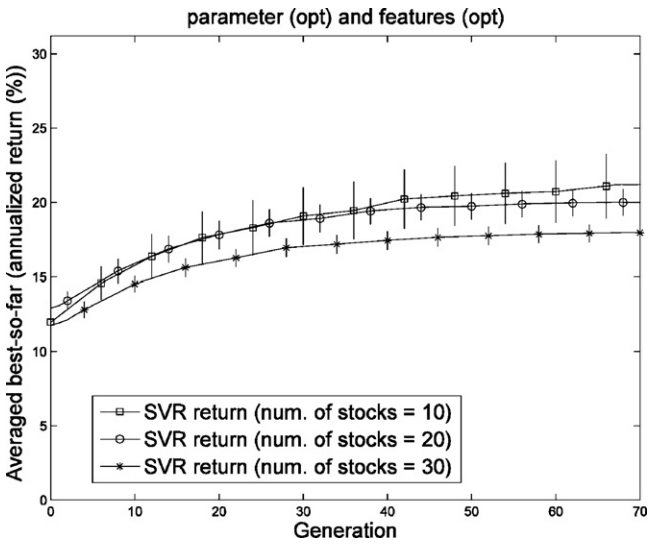

Fig. 11. Best-so-far curves by the GA.

4.4. SVR using optimized parameters and optimal sets of features

Here we examine the effect of optimized C, δ^2, ε as well as subsets of features. Fig. 11 displays the corresponding averaged best-so-far curves. Notice that, for each case in Figs. 9 and 11, the averaged best-so-far curves converged at nearly the same values, thereby indicating the effect of the optimization on model parameters is again not as significant as feature selection. However, Fig. 11 exhibits slightly larger 95% confidence intervals, implying it is likely to generate higher annualized returns by the models with optimized parameters. Fig. 12 displays an illustration for this effect, in which the cumulative returns by the models with optimized parameters are higher than those without optimized parameters (in Fig. 10) at the end of 2010.

In order to examine the validity of our proposed methodology, further statistical tests are presented next.

4.4.1. Model validation

In the previous subsections, the models were obtained using all the financial data available from year 1996 to 2010. In reality, the data is usually divided into two mutually exclusive parts, a training part *Dr* that is used to construct the models, and a testing part *Dv*

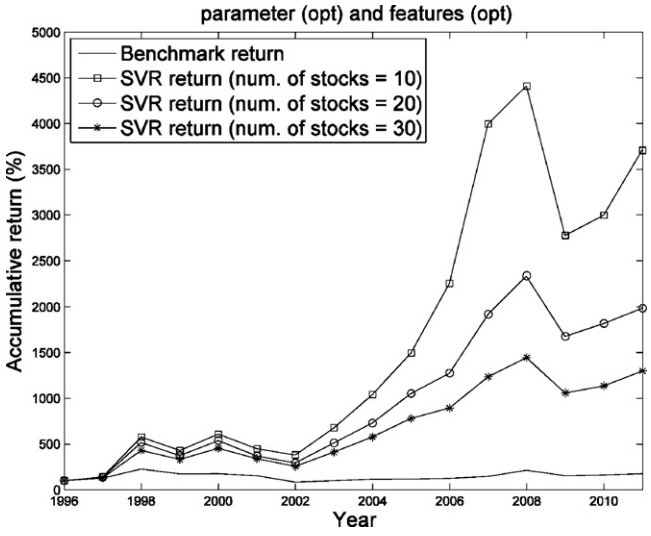

Fig. 12. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR.

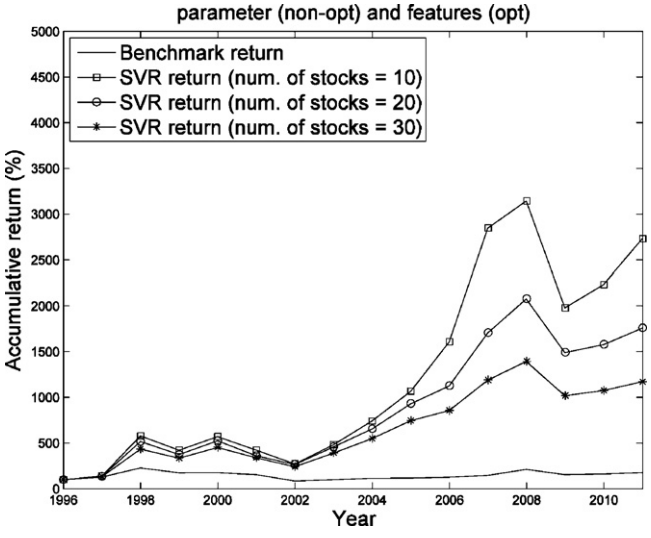

Fig. 10. Cumulative returns of benchmark vs longing top-ranked stocks by the SVR.

that is used to validate the models. The goal of doing so is to examine if the models learned in the training phase are applicable to the hold-out data samples during the testing phase.

In this study, we split the data into two parts: the data for the first n years are used to train the stock selection models and the data for the remaining years are used to validate the models. As shown in Table 2, in the first row the gray area indicates the training data (year 1996) and the model learned from this is examined by the testing data in the black area (year 1997–2010). In the second row,

Table 2
Model validation: training period (gray) and testing period (black).

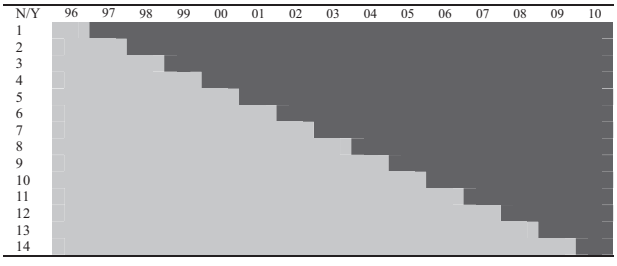

Table 3
Statistics of the benchmark and SVR–GA stock selection models for 10 stocks.

Case index	Training period	Annualized benchmark return (%)	Mean of annualized model return (%)	Variance of annualized model return (%)	Testing period	Annualized benchmark return (%)	Mean of annualized model return (%)	Variance of annualized model return (%)
1	1	27.2615	61.7676	5.5770	2–15	2.3489	5.2850	7.7316
2	1–2	50.4192	107.7513	1768.1	3–15	−1.9077	2.9041	22.9912
3	1–3	19.7369	41.7102	261.3085	4–15	0.2147	4.5146	37.3409
4	1–4	14.9839	39.6706	229.5432	5–15	0.0694	6.0917	25.3487
5	1–5	8.9005	24.9487	67.3695	6–15	1.4078	7.3426	19.7575
6	1–6	−2.8317	14.0150	49.0752	7–15	8.5513	13.6560	28.4809
7	1–7	−0.1497	19.5937	102.3874	8–15	7.4737	16.0384	26.6916
8	1–8	1.7285	24.4311	67.8455	9–15	6.3206	14.4583	18.5463
9	1–9	1.7348	29.2812	53.8386	10–15	7.0959	10.4337	12.8360
10	1–10	2.2505	29.7401	56.8345	11–15	7.1129	5.8753	19.9693
11	1–11	3.4922	33.5983	52.1087	12–15	4.8263	−1.6058	2.7269
12	1–12	6.4443	30.1030	52.3017	13–15	−5.9271	−5.4475	5.5000
13	1–13	3.3426	23.7139	29.7553	14–15	7.1805	14.0516	21.5848
14	1–14	3.5111	21.5515	35.0980	15	8.6541	22.1951	51.9317

Table 4
Statistics of the benchmark and SVR–GA stock selection models for 20 stocks.

Case index	Training period	Annualized benchmark return (%)	Mean of annualized model return (%)	Variance of annualized model return (%)	Testing period	Annualized benchmark return (%)	Mean of annualized model return (%)	Variance of annualized model return (%)
1	1	27.2615	49.2870	9.9148	2–15	2.3489	7.7781	4.7613
2	1–2	50.4192	120.1057	213.0099	3–15	−1.9077	7.8102	6.5107
3	1–3	19.7369	49.2441	78.2888	4–15	0.2147	7.9523	9.4030
4	1–4	14.9839	46.0218	76.8551	5–15	0.0694	8.0555	10.7992
5	1–5	8.9005	26.5578	18.6353	6–15	1.4078	9.0566	19.1845
6	1–6	−2.8317	14.6168	20.4075	7–15	8.5513	17.5293	17.2620
7	1–7	−0.1497	22.6957	22.4438	8–15	7.4737	17.5719	1.6793
8	1–8	1.7285	24.4411	31.5812	9–15	6.3206	14.3132	4.6329
9	1–9	1.7348	26.7380	16.4054	10–15	7.0959	10.9044	3.0459
10	1–10	2.2505	26.4249	11.8420	11–15	7.1129	9.1535	4.5380
11	1–11	3.4922	28.9976	7.4316	12–15	4.8263	1.0441	1.2721
12	1–12	6.4443	27.6245	13.8451	13–15	−5.9271	−4.6934	2.7622
13	1–13	3.3426	22.2604	10.7925	14–15	7.1805	8.7823	9.8464
14	1–14	3.5111	21.2373	6.8446	15	8.6541	12.0599	20.1834

the training data is from year 1996 to 1997, and the testing data is from year 1998 to 2010, and so on.

Notice that this setup is different from the regular cross-validation procedure where the process of data being split into two independent sets is randomly repeated several times without taking into account the data's temporal order. However, in the stock selection study here, temporal order is critical as practically one would like to use all available data so far to train the model and hope to apply the models in the future to gain real profits.

With this setup, Tables 3–5 show the model validation for selecting 10, 20 and 30 stocks by the GA–SVR for 50 runs. In the 10-stock model, an inspection on the means of annualized model returns shows that the model outperformed the benchmark in 12 out of 14 testing cases. For the 20 and 30-stock models, the model further outperformed the benchmark in 13 out of 14 testing cases.

In the mean time, the variances of the annualized model returns decrease as more stocks are being selected into a portfolio. That is,

Table 5
Statistics of the benchmark and SVR–GA stock selection models for 30 stocks.

Case index	Training period	Annualized benchmark return (%)	Mean of annualized model return (%)	Variance of annualized model return (%)	Testing period	Annualized benchmark return (%)	Mean of annualized model return (%)	Variance of annualized model return (%)
1	1	27.2615	44.8906	1.7525	2–15	2.3489	8.3085	2.5501
2	1–2	50.4192	104.7810	40.3210	3–15	−1.9077	4.9268	1.6224
3	1–3	19.7369	50.4839	11.3351	4–15	0.2147	7.2328	1.0980
4	1–4	14.9839	48.0834	19.9948	5–15	0.0694	5.0577	2.2778
5	1–5	8.9005	28.2866	10.9252	6–15	1.4078	7.5587	7.3837
6	1–6	−2.8317	16.4473	16.2906	7–15	8.5513	16.2570	10.0173
7	1–7	−0.1497	21.3169	4.2989	8–15	7.4737	14.4447	0.9508
8	1–8	1.7285	23.5317	8.5510	9–15	6.3206	11.5049	0.4528
9	1–9	1.7348	24.5284	7.0644	10–15	7.0959	8.2811	1.0768
10	1–10	2.2505	23.9266	2.8914	11–15	7.1129	7.3060	1.0417
11	1–11	3.4922	24.9979	1.5967	12–15	4.8263	0.5182	0.7316
12	1–12	6.4443	24.4613	3.8801	13–15	−5.9271	−4.7967	0.8794
12	1–13	3.3426	19.6302	1.6463	14–15	7.1805	9.2482	2.9524
14	1–14	3.5111	18.0476	1.9504	15	8.6541	12.4724	8.8784

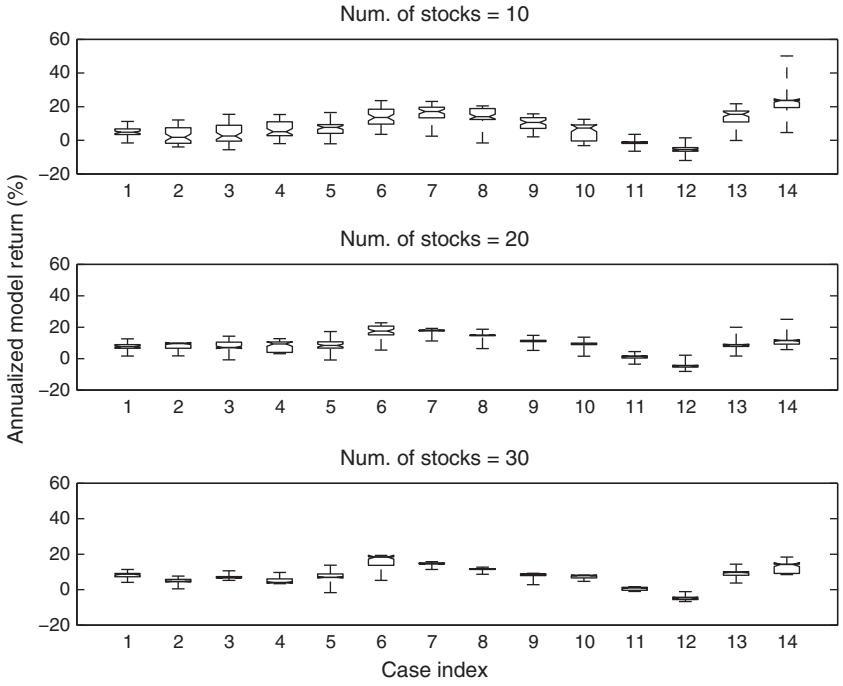

Fig. 13. Box plots for the annualized model returns.

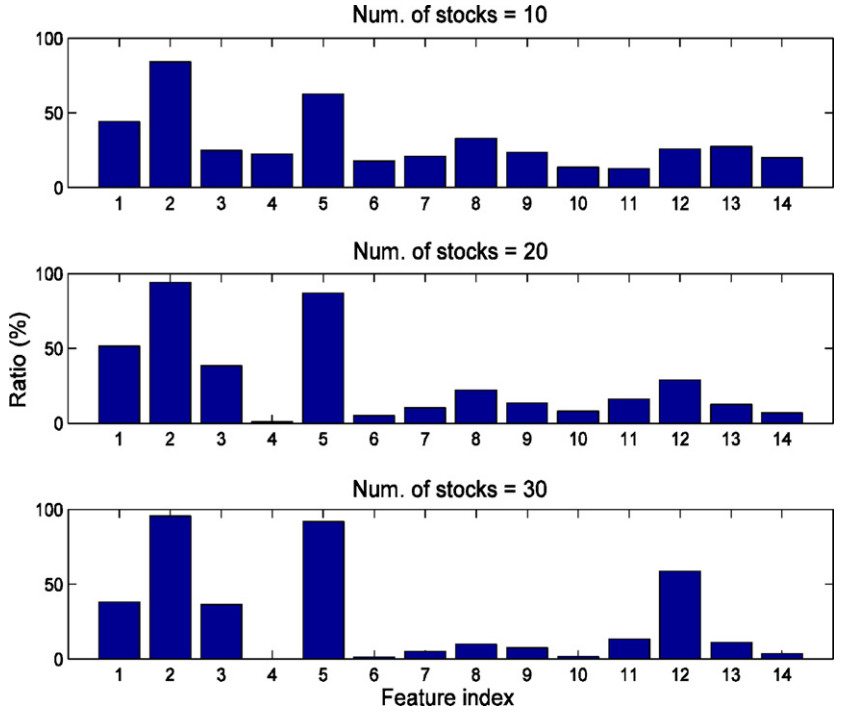

Fig. 14. Ratios of features being selected.

Table 6
Distributions of *TP*, *FP*, *TN*, *FN* for the 30-stock model.

Case index														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
TP	50	50	50	50	49	48	50	50	46	32	0	47	43	48
FP	0	0	0	0	1	2	0	0	4	18	50	3	7	2
TN	50	50	50	50	49	48	50	50	48	43	0	47	43	48
FN	0	0	0	0	1	2	0	0	2	7	50	3	7	2

Table 7
Accuracy and precision for the 10, 20 and 30-stock models.

	10-stock model	20-stock model	30-stock model
Accuracy	0.800	0.8907	0.8850
Precision	0.7671	0.8871	0.8757

selecting 30 stocks apparently yields more consistent performance on the model return than the 10 and 20-stock cases do.

Here we also provide Fig. 13 to visualize the results of Tables 3–5 through three diagrams – each of them shows a series of 14 box plots, where x and y-axis designate the case index and the annualized model return, respectively. Each box plot is generated for the corresponding 50-run GA–SVR model. These plots thus offer a visual gist on the spread of the model returns, and clearly selecting more stocks tends to reduce the variation in model returns.

Furthermore, it is worthwhile to investigate which features have been selected by the GA since such findings shall be important for investment in practice. Here we display in Fig. 14 the ratio of the number of times a feature being selected to the total of 700 models studied.[6] As can be seen, the results are fairly consistent over the three scenarios. Especially in the 20 and 30-stock cases, among all the features, feature 2 (PB ratio) and feature 5 (ROA) appear being selected most times, indicating that the GA is able to consistently find similar subsets of significant features for the construction of the models.

Finally, we examine the accuracy and precision of our proposed model, where the accuracy and precision are defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{respectively.}$$

In these two definitions, TP and TN denote the number of true positives and true negatives, respectively. FP and FN denote the number of false positives and false negatives, respectively. In this study, a true positive occurs when the annualized model return (i.e., the annualized average return of the stocks selected by the model) does outperform the benchmark; otherwise, the model generates a false positive. Analogously, a true negative occurs when the annualized average return of the unselected stocks does underperform the benchmark; otherwise, the model generates a false negative.

As an illustration, Table 6 displays the distributions of TP, FP, TN and FN over the 14 testing cases for the 30-stock model. One can notice that, in case 11, the model generated all false positives and negatives; and there are also 18 false positives in case 10. Our detailed inspection indicates that this is mainly because the support vectors acquired in the training phase failed to capture the seemingly distinct characteristics of the bear-market induced by the financial crisis in 2007–2009; as a consequence, the SVR model was not able to recommend winning stocks successfully.

In Table 7, we summarize the results of the accuracy and precision calculated for the 10, 20 and 30-stock models, which appear to be satisfactory. Thus all the results so far provide promising evidence for the effectiveness of our proposed methodology.

5. Conclusions

In this paper we presented a hybrid GA–SVR model for stock selection. The SVR method was used to generate predicted returns on a collective of stocks, which in turn served as surrogates of the actual returns for stock rankings. Top-ranked stocks are then selected as components in a portfolio. On top of this model, the GA was employed for feature selection and optimization of model parameters. We have evaluated our GA–SVR models statistically and validated the effectiveness of this method by comparing with the benchmark.

In this study, we have shown that feature selection can shed light on which features play more important roles in our proposed model. Interestingly, the results also showed that, in this particular application, the contribution of feature selection to effective stock selection appear to be more significant than that of the optimization on the model parameters alone. This work again highlights the crucial importance of feature selection in complex real-world problems, such as the stock selection problem studied here.

Overall, the empirical results showed that the investment returns provided by our proposed model can significantly outperform the benchmark. Therefore, we expect this hybrid GA–SVR methodology to advance the research in computational finance and provide a promising solution to stock selection in practice. In the future, a plausible research direction is to employ more advanced SVR models to investigate how performance of stock selection can be further improved. In addition, because investment return and risk management appear to be two distinct objectives, in the future work, we expect that a study for simultaneous optimization on these multi-objectives is also a promising research subject to explore. Finally, we intend to conduct a further study on the characteristics of the stock selection domain to determine which algorithms, including ES, PSO or GA, shall be most fruitful for the optimization on our proposed work.

Acknowledgements

This work is fully supported by the National Science Council, Taiwan, Republic of China, under grant number NSC 99-2221-E-390-032. The author would also like to thank Prof. Chih-Hsiang Chang for his generosity in providing the financial data.

References

[1] A. Mochón, D. Quintana, Y. Sáez, Soft computing techniques applied to finance, Applied Intelligence 9 (2) (2008) 111–115.
[2] D. Zhang, L. Zhou, Discovering golden nuggets: data mining in financial application, IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Review 34 (4) (2004) 513–522.
[3] K.K. Lai, L. Yu, S. Wang, C. Zhou, A double-stage genetic optimization algorithm for portfolio selection, in: Proceedings of the 13th International Conference on Neural Information Processing, 2006, pp. 928–937.
[4] T.C. Chu, C.T. Tsao, Y.R. Shiue, Application of fuzzy multiple attribute decision making on company analysis for stock selection, in: Proceedings of Soft Computing on Intelligent Systems and Information Processing, 1996, pp. 509–514.
[5] M.R. Zargham, M.R. Sayeh, A web-based information system for stock selection and evaluation, in: Proceedings of the First International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems, 1999, pp. 81–83.
[6] T.-S. Quah, B. Srinivasan, Improving returns on stock investment through neural network selection, Expert Systems with Applications 17 (1999) 295–301.
[7] N. Chapados, Y. Bengio, Cost functions and model combination for VaR-based asset allocation using neural networks, IEEE Transactions on Neural Networks 12 (2001) 890–906.
[8] K.-J. Kim, I. Han, Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index, Expert Systems with Applications 19 (2000) 125–132.
[9] M. Caplan, Y. Becker, Lessons learned using genetic programming in a stock picking context, in: U.-M. O'Reilly, T. Yu, R. Riolo, B. Worzel (Eds.), Genetic Programming Theory and Practice II, Springer, Ann Arbor, MI, 2004, pp. 87–102, Chapter 6.
[10] Y. Becker, P. Fei, A. Lester, Stock selection – an innovative application of genetic programming methodology, in: R. Riolo, T. Soule, B. Worzel (Eds.), Genetic Programming Theory and Practice IV, Genetic and Evolutionary Computation, vol. 5, Springer, Ann Arbor, MI, 2006, pp. 315–334, Chapter 12.
[11] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.
[12] N.I. Sapankevych, R. Sankar, Time series prediction using support vector machines: a survey, IEEE Computational Intelligence Magazine 4 (2) (2009) 24–38.

[6] We studied 50 GA models for each of the 14 cases; thus there are 700 models in total.

[13] J.H. Min, Y.-C. Lee, Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters, Expert Systems with Applications 28 (4) (2005) 603–614.
[14] K.-J. Kim, Financial time series forecasting using support vector machines, Neurocomputing 55 (2003) 307–319.
[15] W. Huang, Y. Nakamori, S.Y. Wang, Forecasting stock market movement direction with support vector machine, Computers and Operations Research 32 (2005) 2513–2522.
[16] C.-L. Huang, C.-Y. Tsai, A hybrid SOFM-SVR with a filter-based feature selection for stock market forecasting, Expert Systems with Applications 36 (2009) 1529–1539.
[17] M.C. Lee, Using support vector machine with a hybrid feature selection method to the stock trend prediction, Expert Systems with Applications 36 (2009) 10896–10904.
[18] D.Y. Chiu, P.J. Chen, Dynamically exploring internal mechanism of stock market by fuzzy-based support vector machines with high dimension input space and genetic algorithm, Expert Systems with Applications 36 (2009) 1240–1248.
[19] A. Fan, M. Palaniswami, Stock selection using support vector machines, in: Proceedings of the International Joint Conference on Neural Networks 3, 2001, pp. 1793–1798.
[20] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, IEEE Intelligent Systems 13 (2) (1998) 44–49.
[21] H. Fröhlich, O. Chapelle, B. Schölkopf, Feature selection for support vector machines by means of genetic algorithms, in: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 142–148.
[22] C.-L. Huang, C.-J. Wang, A GA-based feature selection and parameters optimization for support vector machines, Expert Systems with Applications 31 (2006) 231–240.
[23] S. Hettich, C.L. Blake, C.J. Merz, UCI Repository of Machine Learning Databases, Department of Information and Computer Science, University of California, Irvine, CA, 1998, http://archive.ics.uci.edu/ml/.
[24] G. John, R. Kohavi, K. Peger, Irrelevant features and the subset selection problem, in: Proceedings of the 11th International Conference on Machine Learning, 1994, pp. 121–129.
[25] R. Kohavi, G. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[26] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, 2nd edition, Morgan Kaufmann, San Francisco, 2005.
[27] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, Advances in Neural Information Processing Systems 9 (1996) 155–161.
[28] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.
[29] W. Konen, P. Koch, O. Flasch, T. Bartz-Beielstein, M. Friese, B. Naujoks, Tuned Data Mining: A Benchmark Study on Different Tuners, Cologne University of Applied Sciences, CIOP Technical Report 03/11 (2011).
[30] C.W. Hsu, C.C. Chang, C.J. Lin, A practical guide to support vector classification, technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2003. Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.
[31] H.T. Lin, C.J. Lin, A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, technical report, Department of Computer Science and Information Engineering, National Taiwan University (2003). Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/tanh.pdf.
[32] S. Qiu, T. Lane, Parallel computation of RBF kernels for support vector classifiers, in: Proceedings of the 5th SIAM International Conference on Data Mining (SDM05), 2005, pp. 334–345.
[33] N. Hansen, S.D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), Evolutionary Computation 11 (1) (2003) 1–18.
[34] T. Bartz-Beielstein, C. Lasarczyk, M. Preuß, Sequential parameter optimization, in: Proceedings of the 2005 Congress on Evolutionary Computation (1), 2005, pp. 773–780.
[35] J. Kennedy, R.C. Eberhart, Particle swarm optimization, in: Proceedings of the IEEE International Conf. on Neural Networks, 1995, pp. 1942–1948.
[36] S.-W. Lin, K.-C. Ying, S.-C. Chen, Z.-J. Lee, Particle swarm optimization for parameter determination and feature selection of support vector machines, Expert Systems with Applications 35 (2008) 1817–1824.
[37] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, Evolutionary Computation 1 (April) (1997) 67–82.
[38] S. Mukherji, M.S. Dhatt, Y.H. Kim, A fundamental analysis of Korean stock returns, Financial Analysts Journal 53 (3) (1997) 75–80.
[39] G.R. Jensen, R.R. Johnson, J.M. Mercer, New evidence on size and price-to-book effects in stock returns, Financial Analysts Journal 53 (6) (1997) 34–42.
[40] M.G. Danielson, T.D. Dowdell, The return-stages valuation model and the expectations within a firm's P/B and P/E ratios, Financial Management 30 (2) (2001) 93–124.
[41] J. Lewellen, Predicting returns with financial ratio, Journal of Financial Economics 74 (2) (2004) 209–235.
[42] E.F. Fama, K.R. French, Average returns, B/M, and share issue, Journal of Finance 63 (6) (2008) 2971–2995.
[43] E. Hjalmarsson, Predicting global stock returns, Journal of Financial and Quantitative Analysis 45 (1) (2010) 49–80.
[44] M. Omran, Linear versus non-linear relationships between financial ratios and stock returns: empirical evidence from Egyptian firms, Review of Accounting and Finance 3 (2) (2004) 84–102.
[45] R. Bauer, N. Guenster, R. Otten, Empirical evidence on corporate governance in Europe: the effect on stock returns, firm value and performance, Journal of Asset Management 5 (2) (2004) 91–104.
[46] M.T. Soliman, The use of DuPont analysis by market participants, Accounting Review 83 (3) (2008) 823–853.
[47] T.A. Carnes, Unexpected changes in quarterly financial-statement line items and their relationship to stock prices, Academy of Accounting and Financial Studies Journal 10 (3) (2006) 99–116.
[48] D. Ikenberry, J. Lakonishok, Corporate governance through the proxy contest: evidence and implications, Journal of Business 66 (3) (1993) 405–435.
[49] G. Sadka, R. Sadka, Predictability and the earnings–returns relation, Journal of Financial Economics 94 (1) (2009) 87–93.
[50] F.E.H. Tay, L. Cao, Application of support vector machines in financial time series forecasting, Omega 29 (2001) 309–317.
[51] D.E. Goldberg, K. Deb, A comparative analysis of selection schemes used in genetic algorithms, Foundation of Genetic Algorithms (1991) 69–93.