DOI:10.3233/JIFS-16209
IOS Press

Fuzzy support vector regression model for forecasting stock market volatility

Jui-Chung Hung*
Department of Computer Science, University of Taipei, Taipei City, Taiwan

Abstract. Stock market volatility exhibits characteristics such as clustering and time-varying fluctuations. This paper proposes a two-stage method for addressing these concerns. The involved procedure is as follows: First, a fuzzy system is used to analyze clustering regimes according to the size of fluctuations. Second, the clustering regimes of Stage I are used to establish a support vector regression (SVR) model, which is used to reduce the time-varying complexity. However, the fuzzy-SVR model combines the parameters of membership functions and SVR models, further complicating the problem. Thus, this paper presents parallel research based on a genetic algorithm (GA) for estimating the parameters of the membership functions and SVR model. Data from four stock markets—the Taiwan Stock Exchange weighted stock index (Taiwan), the NASDAQ Composite index, the Hang Seng index (Hong Kong), and the Shanghai Composite index (Shanghai)—were analyzed in this study to illustrate the performance of the proposed model. According to the simulation results, the forecasting of out-of-sample volatility performance was significantly improved when the model accounted for the behavioral effect of both clustering and time-varying fluctuations.

Keywords: Support vector regression, forecasting volatility, fuzzy system, genetic algorithm, clustering

1. Introduction

Volatility is a measure of risk in financial markets [22, 23, 30, 31, 38] and is known to be a major financial tool for describing stock markets. Although the autoregressive conditional heteroscedasticity (ARCH) model, developed by Engle in 1982 [8], has been proven useful in several economic applications, some problems continue to affect its forecasting accuracy. Therefore, using the ARCH model is limited by the fixed lag structure of the conditional variance. Bollerslev used the autoregressive moving average to expand the ARCH model, creating the generalized autoregressive conditional heteroscedasticity (GARCH) model [4], which uses volatility clustering and the fat tail features of financial time series to estimate volatility.

Volatility clustering is a common feature of various financial series and refers to the phenomenon in which large changes tend to be followed by large changes and small changes tend to be followed by small changes. Based on the weighted average of the past squared residuals, the GARCH model features declining weights that never entirely achieve zero. Therefore, the leverage effect that frequently appears in financial markets decreases the forecasting accuracy of the GARCH model. The leverage effect indicates that volatility responds differently to positive and negative information, and negative returns of financial assets tend to result in greater volatility. Over the past few years, several studies have developed many improved versions of the

*Corresponding author. Jui-Chung Hung, Department of Computer Science, University of Taipei, Taipei City 11048, Taiwan. Tel.: +886 2 23113040; Fax: +886 2 23118508; E-mail: juichung@gmail.com.

GARCH model to resolve this problem; for example, the exponential GARCH (EGARCH) model was introduced by Nelson [34], the fuzzy switch GARCH model has been proposed in [2, 18], and the Glosten–Jagannathan–Runkle GARCH (GJR-GARCH) model was introduced by Glosten et al. [13]. These models can capture the leverage effect in financial markets; however, they do not simulate the time-varying fluctuations according to the volatility of stock markets effectively.

To overcome the time-varying characteristics of volatility, many studies have proposed using robust methods, such as neural networks and support vector regression (SVR), for financial forecasting [12, 15, 17, 19, 20, 37, 40, 42]. Various volatility forecasting methods are combined through simple averaging based on the artificial neural network by Xiao et al. [41]. The volatility of foreign exchange data are forecast using the support vector machine (SVM) proposed by Gavrishchaka and Ganguli [11]. A novel neural network technique with an SVM is used in financial time-series forecasting [37]. The Bayesian framework is combined with least squares SVMs in the nonlinear regression method by Gestel [12]. These methods can efficiently capture volatility under small fluctuations; however, they degrade capture volatility when the fluctuations are large.

SVR, a variation of the SVM proposed by Vapnik et al. [39], is a powerful machine-learning method based on statistical learning theory. The method involves a structural risk minimization principle, instead of the usual empirical risk, which aims at minimizing a bound on the generalization error of a model, rather than minimizing only the mean square error over the data set [16]. Thus, SVR is useful for constructing data-driven nonlinear empirical process models and is effective in forecasting the time-varying characteristics of volatility [42]; however, SVR loses its efficiency by overfitting when volatility fluctuates greatly. The SVR model describes random errors or noise instead of the underlying relationship. Therefore, this paper proposes a fuzzy-SVR model, in which a fuzzy system is used, for analyzing the clustering regimes on the basis of the size of fluctuations; the robust characteristics of SVR are combined to forecast the volatility of stock markets.

In recent decades, fuzzy systems have been extensively applied in a wide variety of industrial systems because of their model-free approach [1, 6, 16, 43] and the excellent approximation ability of fuzzy systems [10, 26]. In generally, fuzzy systems are universal approximators to uncertain nonlinear systems (i.e., they can approximate any behavior related to complexity dynamics within a predefined range of desired accuracy), such as nonlinear discrete-time systems with backlash [27] and multiinput multi-output nonlinear systems [28, 29] by an adaptive fuzzy controller to approach the desired control performance. The fuzzy systems combine the ease of implementation and the convenience of linear models with the ability to capture and approximate a wide range of functions. This paper proposes fuzzy systems as a judicious choice for analyzing the size of fluctuations that feature time-dependent variances. In the proposed method— the fuzzy-SVR model—SVR models and fuzzy systems are combined and applied to forecast the volatility of stock markets. The process of optimizing the parameters of fuzzy systems and SVR models is highly complex and nonlinear. Therefore, a genetic algorithm (GA)-based parameter estimation algorithm is proposed for deriving the optimal solution for the fuzzy-SVR model.

The GA is a parallel search method for obtaining the global optimal solution of complex optimization problems that emulate natural genetic operations such as reproduction, crossover, and mutation [7, 14]. The GA applies operations to a population of binary strings that represent potential solutions. At each generation, the GA explores different areas of potential solutions and then directs the search to the region in which a high probability of determining improved performance exists. Because the GA simultaneously evaluates many points in a parameter space, it can ultimately converge on the global solution. In particular, the algorithm can iterate several times on each data point. Accordingly, it is suitable for addressing the parameter problem of the fuzzy-SVR model.

The remainder of this paper is organized as follows. Section 2 describes the SVR and fuzzy-SVR models. Section 3 presents the GA-based optimization of the fuzzy-SVR model and an adaptive forecasting algorithm. The experimental results that illustrate the effectiveness of the proposed method are explained in Section 4. Finally, Section 5 presents the conclusion.

2. Fuzzy-SVR model

The empirical and theoretical show the fluctuations of finance market volatility is asymmetries, general falling stocks to returns give rise to higher volatility than do equivalent rising stocks to returns [13]. Moreover, the volatility clustering characteristic that large changes tend to follow large changes and small changes tend the follow small changes [13]. In this study, we used the SVR model to capture the clustering characteristic of volatility. The SVR model involves obtaining widespread acceptance in data-driven nonlinear modeling applications [15]. The SVR model is an extraction of the training data that serves as a support vector and therefore represents a stable characteristic of the data. Moreover, it entails a structural risk minimization principle—instead of the usual empirical risk—that aims at minimizing a bound on the generalization error of a model, rather than minimizing only the mean square error over the data set [30]. Thus, SVR is useful for forecasting time-varying clustering data [37]. To consider the differential effects of the propagation of volatility caused by a rising or falling stock market, we applied fuzzy systems to address the fluctuation asymmetries by falling/rising stock to returns. The general expert information can represent the fuzzy terms (e.g., low, high); this representation may be used for convenience because of a lack of more precise knowledge. In this study, we used the IF-THEN rules of fuzzy systems to appropriately simulate the fluctuation asymmetries of the stock market. The following subsections describe the proposed method in detail.

2.1. SVR model

Consider an in-sample data set $\{\mathbf{x}(1), \mathrm{y}(1), \ldots, \mathbf{x}(N), \mathrm{y}(N)\}$, where $\mathbf{x}(i) \in R^D$, $y(i) \in R$, $\mathbf{x}(i)$ is the input attribute value of the ith in-sample data in D dimensional space, $y(i)$ represents the stock market volatility ith in-sample data [36], and N is the size of the in-sample data. The linear regression function can be expressed as $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$, where $\mathbf{w}$ is normal to the hyperplane and b is a constant. Methods involving the SVR [3, 5] model have been developed for determining a function $f(\mathbf{x})$ with a maximal ϵ deviation from the actually obtained targets $y(i)$ for all the in-sample data while being as flat as possible. In general, the input in-sample data are delivered to a nonlinear map from the input space to the output space, and the input data are mapped on a higher dimensional feature space through this mapping. The estimate function $f(\mathbf{x})$ is formulated as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{w}^T \phi(\mathbf{x}(i)) + b, \tag{1}$$

where $f(\mathbf{x})$ is the flatness of the function (i.e., one seeks a small weight vector $\mathbf{w}$ and $\phi(\bullet)$ is the nonlinear mapping function). The SVR involves minimizing the following risk function:

$$\min_{\mathbf{w},b,\xi_i,\xi_i^*} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

$$\text{subject to} \begin{cases} y(i) - \left(\mathbf{w}^T\phi(\mathbf{x}(i)) + b\right) \leqq \varepsilon + \xi_i \\ \left(\mathbf{w}^T\phi(\mathbf{x}(i)) + b\right) - y(i) \leqq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geqq 0, \ \forall i = 1, 2, \ldots, N \end{cases}, \tag{2}$$

where ε is the error tolerance, ξ_i and ξ_i^* are the slack variables that respectively specify the upper and lower training errors subject to an error tolerance ε, and C is the cost function of the trade-off between the flatness of $f(\mathbf{x})$ and the training error. The constraints in Equation (2) indicate that most of the input data $\mathbf{x}(i)$ are placed inside the tube ε. If $\mathbf{x}(i)$ is outside the tube named the support vector, then a ξ_i or ξ_i^* error exists (in this study, ξ_i^* was primarily minimized in the objective function). The characteristics of the slack variables ξ_i and ξ_i^*, error tolerance ε, and support vectors are shown in Fig. 1.

Equation (2) is a quadratic optimization problem with inequality constraints; according to the Karush–Kuhn–Tucker optimality conditions [39], the SVR training procedure amounts can be used to solve the convex quadratic problem by using the Lagrange multiplier. Thus, Equation (2) can be translated into a dual problem as follows:

$$\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i=1}^{N}\sum_{\mathrm{j}=1}^{N}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right)K(\mathbf{x}(i), \mathbf{x}(j))$$

$$-\varepsilon\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{N}\mathbf{y}(i)(\alpha_i - \alpha_i^*)$$

$$\text{subject to} \begin{cases} \sum_{i=1}^{N}\left(\alpha_i - \alpha_i^*\right) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C \end{cases} \tag{3}$$

where α, α_i^* are Lagrange multipliers, samples with $\alpha_i - \alpha_i^* \neq 0$ are support vectors, and $K(\mathbf{x}(i), \mathbf{x}(j))$ is the kernel function of the dot product $\langle\phi(\mathbf{x}(i)) \cdot \phi(\mathbf{x}(j))\rangle$ that facilitates inputting training data implicitly into a feature space and training the SVR in that space without representing the feature vectors explicitly [3, 5]. All kernel functions must satisfy Mercer's condition [33]. Typically, the commonly used kernel functions include linear kernels, polynomial kernels, and radial basis function (RBF)

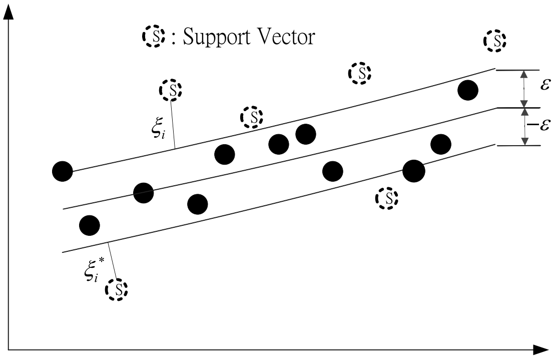


Fig. 1. The slack variables, error tolerance, and support vectors of SVR.

kernels [25]. In this study, the RBF was adopted and formulated as

$$K(\mathbf{x}(i),\mathbf{x}) = exp\left(-\frac{\|\mathbf{x}(i)-\mathbf{x}\|^2}{\sigma^2}\right), \quad (4)$$

where σ is the kernel bandwidth parameter. By solving the dual problem expressed in Equation (3), the optimal SVR model depends on a setting that is mainly determined by C, ε, and σ. In general, the optimization problem is a highly nonlinear function, and many local extreme values may exist. Therefore, the GA was used in this study to specify C, ε, and σ and solve the SVR optimization problem.

2.2. Fuzzy-SVR model

Fuzzy systems are universal approximators [26, 43] that can approximate the behavior of a system in which analytic functions or numerical relations do not exist. The empirical and theoretical appeal of the SVR model is due to it minimizing the regression error based on the structural risk minimization principle and capturing the small fluctuations of time-varying data. However, the model uses the insensitive error tolerance that contains a fixed and symmetrical margin, thus failing to accommodate the large fluctuations and sign asymmetries: Negative shocks to returns give rise to higher volatility than do equivalent positive shocks to returns. Ignoring this fact can lead to poor prognostic characteristics. According to Fama [9], stock market volatility exhibits the property that large changes tend to follow large changes and small changes tend to follow small changes. To account for the differential effects of the fluctuation size of stock market volatility, this study incorporated fuzzy systems into the proposed SVR model. The resulting fuzzy model is described using the IF-THEN rules and is used to ensure that the SVR model appropriately addresses the problem of forecasting stock market volatility. The basic configuration of fuzzy systems consists of three components: a fuzzy rule base, a fuzzy inference engine, and a defuzzifier. The fuzzy rule base consists of a collection of IF-THEN rules [18, 43]. The lth rule of the fuzzy system for SVR is described as follows:

$$\begin{aligned}
&Rule^{(l)}:\\
&\text{IF } z_1(i) \text{ is } F_{l1} \text{ and } \cdots \text{and } z_n(i) \text{ is } F_{ln}, \ \text{ THEN}\\
&f(\mathbf{x}) = \sum_{i=1}^{N}\left(\alpha_{i,l}-\alpha_{i,l}^*\right) K_l(\mathbf{x}(i),\mathbf{x}) + b_l,\\
&\text{for } l = 1,2,\ldots,L, \qquad (5)
\end{aligned}$$

where $f(\mathbf{x})$ is the system output; F_{lj} for $j = 1,\ldots,n$ and $l = 1,\ldots,L$ is the fuzzy set; L is the number of IF-THEN rules; n is the number of premise variables; and $z_1(i), z_2(i),\ldots,z_n(i)$ are the premise variables. This study used the stock market volatility property of large changes tending to follow large changes and small changes tending to follow small changes. The GARCH model proposed by Bollerslev [4] entails using prior conditional variances to estimate the degree of volatility transmission. The ability of the GARCH model to explain the transmission of volatility is a crucial advantage. Therefore, in this paper the premise variables, namely the previous values of the realized volatility, are defined as

$$z_j(i) = y(i-j), \ \text{for } j = 1,2,\ldots, \ n \qquad (6)$$

For the functional fuzzy system, defuzzification [18] can be written as

$$\begin{aligned}
f(\mathbf{x}) &= \frac{\sum_{l=1}^{L} u_l(\mathbf{z}(i))\left[\sum_{i=1}^{N}\left(\alpha_{i,l}-\alpha_{i,l}^*\right) K_l(\mathbf{x}(i),\mathbf{x}) + b_l\right]}{\sum_{l=1}^{L} u_l(\mathbf{z}(i))}\\
&= \sum_{l=1}^{L} g_l(\mathbf{z}(i))\left[\sum_{i=1}^{N}\left(\alpha_{i,l}-\alpha_{i,l}^*\right) K_l(\mathbf{x}(i),\mathbf{x}) + b_l\right]
\end{aligned} \qquad (7)$$

and

$$u_l(\mathbf{z}(i)) = \prod\nolimits_{j=1}^{n} F_{lj}(z_j(i))$$

$$g_l(\mathbf{z}(i)) = \frac{u_l(z(i))}{\sum\limits_{k=1}^{L} u_k(z(i))}$$

$$\mathbf{z}(i) = [z_1(i), z_2(i), \ldots, z_n(i)]^t, \tag{8}$$

where $F_{lj}(z_j(i))$ is the grade of the membership of $z_j(i)$ in F_{lj}. A Gaussian membership function is used, and $u_l(\mathbf{z}(i))$ can be formulated as

$$u_l(\mathbf{z}(i)) = \prod\nolimits_{j=1}^{n} F_{lj}(z_j(i)) = \prod\nolimits_{j=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{z_j(i) - m_{lj}}{\delta_{lj}}\right)^2\right), \tag{9}$$

where m_{lj}, δ_{lj} are respectively the center and spread of the lth rule membership function corresponding to the jth premise variable. This formula clearly shows the use of the product to represent the premise.

$$\sum_{l=1}^{L} u_l(\mathbf{z}(i)) > 0 \text{ for } l = 1, 2, \ldots, L \tag{10}$$

Thus, the following expression [18] is obtained:

$$g_l(\mathbf{z}(i)) \geq 0 \text{ for } l = 1, 2, \ldots, L \text{ and } \sum_{l=1}^{L} g_l(\mathbf{z}(i)) = 1 \tag{11}$$

According to Equations (5)–(11), the system output can be expressed as

$$f(\mathbf{x}) = \frac{\sum\limits_{l=1}^{L} u_l(\mathbf{z}(i)) \left[\sum\limits_{i=1}^{N} \left(\alpha_{i,l} - \alpha_{i,l}^*\right) K_l(\mathbf{x}(i), \mathbf{x}) + b_l\right]}{\sum\limits_{l=1}^{L} u_l(\mathbf{z}(i))} = \frac{\sum\limits_{l=1}^{L} \prod_{j=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{z_j(i)-m_{lj}}{\delta_{lj}}\right)^2\right) \left[\sum\limits_{i=1}^{N} \left(\alpha_{i,l} - \alpha_{i,l}^*\right) K_l(\mathbf{x}(i), \mathbf{x}) + b_l\right]}{\sum\limits_{l=1}^{L} \prod_{j=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{z_j(i)-m_{lj}}{\delta_{lj}}\right)^2\right)} \tag{12}$$

A reasonable success criterion is the minimization of the mean square error, the objective function of which is defined as

$$E = \sum_{i=1}^{N} \left[y(i) - f(\mathbf{x}(i))\right]^2, \tag{13}$$

where $y(i)$ for $i = 1, 2, \ldots, N$ represents target values of the stock market volatility, N is the number of in-sample data, and $f(\mathbf{x})$ is the output of the fuzzy-GARCH model in Equation (12). Apparently, E is a highly nonlinear function of the fuzzy system that determines the center and spread of m_{lj} and δ_{lj} for $l = 1, \ldots, L$ and $j = 1, \ldots,$ n, respectively, and the SVR primarily determines C_l, ε_l, and σ_l for $l = 1, \ldots, L$. This function may have several local minima. Locating the global minimum of E in Equation (13) by using conventional methods is typically extremely difficult. Therefore, in this study, the GA was used to specify the center and spread of m_{lj} and δ_{lj}, and the SVR primarily determined parameters C_l, ε_l, and σ_l to solve the fuzzy-SVR problem in Equation (13).

3. GA-based fuzzy-SVR model and volatility forecasting

3.1. GA-based fuzzy-SVR model

The GA is composed of probabilistic heuristic search processes based on natural genetic systems. It is highly parallel in searching for the global optimal solution of complex optimization problems [24]. The range in the daily return of the stock market is generally approximately –10% to 10%; hence, the realized volatility in this paper is defined as $y(t) = \frac{1}{5}\sum_{i=0}^{4} r^2(t-i)$, where $r(i)$ is the daily return of the stock market price at time i. Therefore, the parameter range is clearly bounded by the fuzzy-SVR model, as shown in Table 1. We use a GA to solve the fuzzy-SVR problem in Equation (12). This method can explore several points in the search space simultaneously, thereby increasing the opportunity to locate new points in the search space with an expected improvement in performance.

The GA is composed of three operations: 1) selection, 2) crossover, and 3) mutation. These operations are implemented by performing the basic tasks of copying strings, exchanging portions of strings, and changing the state of bits from 1's to 0's or vice versa. These operations ensure that the "fittest" members of the population survive, and their information is preserved and combined to generate "fitter" offspring.

Table 1
Parameter range of fuzzy-SVR model

Parameter	Lower bound L_i	Upper bound U_i
C_l	e^{-10}	e^{10}
ε_l	e^{-10}	e^{5}
σ_l	e^{-10}	e^{5}
m_{lj}	0%	10%
δ_{lj}	0%	10%

The result is an improvement in the next generation's performance. The GA is described in the following subsections.

3.1.1. Coding

The GA featuring a population of strings and binary coding was used in this study. According to the binary coding method, each parameter of the membership functions (m_{lj}, δ_{lj}) and SVR-determined parameters $(C_l, \varepsilon_l, \sigma_l)$ have their own string length, which consists of 0's and 1's. The choice of the bit number B_i for each parameter depends on the desired resolution R_i, which is calculated as

$$R_i = \frac{U_i - L_i}{2^{B_i} - 1}, \tag{14}$$

where U_i and L_i are respectively the upper and lower bounds of the parameter. In this study, the fuzzy-SVR model parameter was coded as the following binary string:

$$\begin{array}{ccccccccc}
B_1 & \cdots & B_{L\times n} & B_{L\times n+1} & \cdots & B_{2L\times n} & B_{2L\times n+1} & \cdots & B_{(2n+1)L} \\
\leftrightarrow & & \leftrightarrow & \leftrightarrow & & \leftrightarrow & \leftrightarrow & & \leftrightarrow \\
\underbrace{1\cdots 11} & \cdots & \underbrace{1\cdots 10} & \underbrace{0\cdots 11} & \cdots & \underbrace{1\cdots 00} & \underbrace{1\cdots 11} & \cdots & \underbrace{0\cdots 10} \\
m_{11} & \cdots & m_{Ln} & \delta_{11} & \cdots & \delta_{Ln} & C_1 & \cdots & C_L \\
B_{(2n+1)L+1} & \cdots & B_{(2n+2)L} & B_{(2n+2)L+1} & \cdots & B_{(2n+3)\times L} & & & \\
\leftrightarrow & & \leftrightarrow & \leftrightarrow & & \leftrightarrow & & & \\
\underbrace{1\cdots 10} & \cdots & \underbrace{1\cdots 10} & \underbrace{1\cdots 10} & \cdots & \underbrace{1\cdots 00} & & & \\
\varepsilon_1 & \cdots & \varepsilon_L & \sigma_1 & \cdots & \sigma_L & & &
\end{array} \tag{15}$$

3.1.2. Fitness function

The degree of fitness depends on the performance of the possible solution represented by that particular string. In this study's design problem, locating the minimum in Equation (13) is equivalent to obtaining a maximal fitness value by using the genetic search process. A chromosome that has a lower objection function should be assigned a higher fitness value. A simple linear relationship between a fitness function and an objective function is expressed as

$$Fit = mE + h,\ m = \frac{Fit_{best} - Fit_{worst}}{E_{smallest} - E_{l\arg est}},$$
$$h = Fit_{best} - mE_{smallest}, \tag{16}$$

where Fit is the fitness function, and $E_{smallest}$ and $E_{l\arg est}$ are the lowest and highest objective function values, respectively. Fit_{best} and Fit_{worst} are the corresponding fitness values.

3.1.3. Selection

Selection, based on the principle of survival of the fittest, is a process by which individual strings are copied and placed in a mating pool for further genetic operations consistent with their fitness value. The probability $PR(j)$ of the jth string with a fitness value $Fit(j)$ selected for mating and reproduction in the next generation is

$$PR(j) = \frac{Fit(j)}{\sum\limits_{i=1}^{Q} Fit(i)}, \tag{17}$$

where Q is the population size specified by [7]. Once the strings are reproduced or copied for possible use in the next generation in a mating pool, they wait for the action of the other two operations: crossover and mutation.

3.1.4. Crossover

Crossover is the primary exchange of information for a chromosome. This study used the one-point crossover method [7], which is conducted in three steps. First, the two newly reproduced strings are selected from the mating pool created through selection in the previous generation. Second, a position including the two strings is selected at random. The third step involves exchanging all the characters by following the crossing sit. The crossover operation occurs only with a probability p_c (crossover probability). The choice of p_c is known to critically affect the performance of a GA. In general, the value of p_c ranges 0.5 to 1.0. When combined with selection, crossover is an effective means of exchanging information and combining various elements of high-quality solutions.

3.1.5. Mutation

Selection and crossover yield most of the processing power of GAs. However, mutation, the third operation, enhances the ability of GAs to search for the optimal solution. Mutation is the occasional flip of each bit at a particular string position with a low probability of a chromosome from 1 to 0, or vice versa. The mutation operation is used to change some elements in selected individuals with a probability p_m (the mutation probability). In general, the p_m should be used sparingly because it is a random search with a high mutation probability.

3.2. Volatility forecasting

3.2.1. Adaptive volatility forecasting

The power of a model in forecasting volatility is crucial because volatility is a measure of risk in financial markets. A common approach involves using in-sample data to construct a model, and then making one-step-ahead predictions to obtain the future solution $\hat{y}(N+1)$ [35, 44, 45]. Realized volatility [31] was used to estimate the performance of forecasting volatility. For improved forecasting results, in this study, the recursive least squares (RLS) [35] error formula was used for forecasting volatility. First, the forecasting objective function is defined in this paper as

$$E_f = \sum_{t=N+1}^{N+M} \lambda^{t-N} \left[y(t) - \hat{y}(t)\right]^2, \tag{18}$$

where $y(t)$ is the realized volatility, defined as $y(t) = \frac{1}{5}\sum_{i=0}^{4} r^2(t-i)$; $r(i) = \{\log P(i) - \log P(i-1)\} \times 100\%$; $P(i)$ is the daily closing price at time i; $r(i)$ is the daily stock return on investment at time; $\lambda \in (0, 1]$ is a forgetting factor; and M is the amount of out-of-sample data. According to Equations (12, 18) can be expressed as

$$E_f = \sum_{t=N+1}^{N+M} \lambda^{t-N} \left\{ y(t) - \sum_{l=1}^{L} g_l(\mathbf{z}(t)) \left\{ \sum_{i=1}^{N} \left(\alpha_{i,l} - \alpha_{i,l}^*\right) K_l(\mathbf{x}(i), \mathbf{x}) + b_l \right\} \right\}^2$$
$$= \sum_{t=N+1}^{N+M} \lambda^{t-N} \left\{ y(t) - \Lambda^t(t)\phi(N) \right\}^2 \tag{19}$$

where

$$\Lambda(t) = \left[g_1(\mathbf{z}(t)) \cdots g_L(\mathbf{z}(t)) \right]^t$$

$$\phi(N) = \left[\sum_{i=1}^{N} \left(\alpha_{i,1} - \alpha_{i,1}^*\right) K_1(\mathbf{x}(i), \mathbf{x}) + b_1 \cdots \right. \tag{20}$$
$$\left. \sum_{i=1}^{N} \left(\alpha_{i,L} - \alpha_{i,L}^*\right) K_L(\mathbf{x}(i), \mathbf{x}) + b_L \right]^t .$$

The RLS of the following formulas are expressed as

$$\mathbf{P}(t) = \frac{1}{\lambda} \{\mathbf{P}(t-1) - \mathbf{P}(t-1)\phi(t)\left(\lambda + \phi^t(t)\right.$$
$$\left.\mathbf{P}(t-1)\phi(t)\right)^{-1} \phi^t(t)\mathbf{P}(t-1)\Big\} \tag{21}$$

$$\Lambda(t) = \Lambda(t-1) + \mathbf{P}(t)\phi(t)(r^2(t) - \Lambda^t(t-1)\phi(t)) \tag{22}$$

where $t = N+1, \ldots, N+M$ represents the out-of-sample data and the initialized matrix $\mathbf{P}(t)$ is $\vartheta\mathbf{I}$, where ϑ is a low positive constant, and $\mathbf{I}$ is an identity matrix.

3.2.2. Forecasting performance evaluation

The mean square error measure of Equation (13) is used to derive the forecasting model in the in-sample data process; however, it cannot be used alone as a conclusive measure for comparing different forecasting models [38]. Therefore, various statistics have been used to compare forecast errors: these include the mean square forecast error (MSFE), mean absolute forecast error (MAFE), and mean percentage forecast error (MPFE).

$$MSFE = \frac{1}{M} \sum_{t=N+1}^{N+M} (y(t) - \hat{y}(t))^2$$

$$MAFE = \frac{1}{M} \sum_{t=N+1}^{N+M} |y(t) - \hat{y}(t)|, \tag{23}$$

$$MPFE = \frac{1}{M} \sum_{t=N+11}^{N+M} \frac{|y(t) - \hat{y}(t)|}{y(t)}$$

where $y(t)$ is the realized volatility at the forecasting period t and $\hat{y}(t)$ is the forecast volatility at the forecasting period t.

3.3. Design procedure

According to the aforementioned analysis, the design procedure for the fuzzy-SVR model applied to forecast stock market volatility by using GA is divided into the following steps:

Step 0: Input N in-sample data and M out-of-sample.

Step 1: Generate a random population of Q chromosomes.

Step 2: Decode each string into the corresponding parameter vector.

Step 3: Build the fuzzy-SVR model in Equation (12).

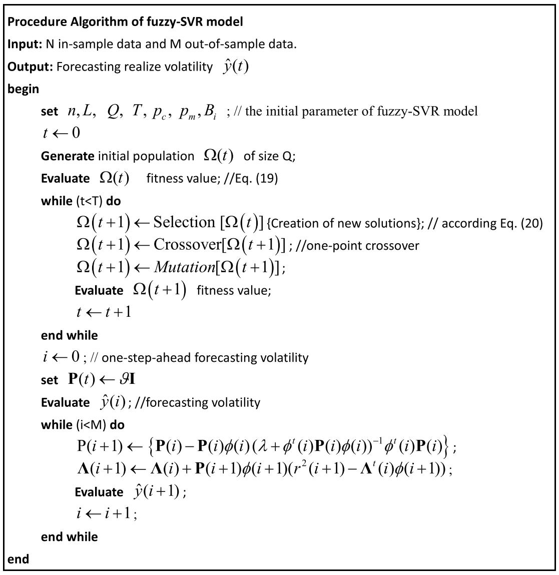
Step 4: Calculate the fitness values according to Equation (16).

Step 5: Keep the optimal chromosomes intact for the next generation.

Step 6: Use selection, crossover, and mutation to generate new chromosomes in the next generation.

Step 7: Repeat the procedure from Steps 2 to 6 until a suitable parameter set is obtained.

Step 8: Given the initial data $\phi(N)$ in Equation (20), use the parameter set from Step 7, and $\mathbf{P}(N) = \vartheta I$, where ϑ is a small positive constant and I is the identity matrix.



```
Procedure Algorithm of fuzzy-SVR model
Input: N in-sample data and M out-of-sample data.
Output: Forecasting realize volatility $\hat{y}(t)$
begin
    set $n, L, Q, T, p_c, p_m, B_i$ ; // the initial parameter of fuzzy-SVR model
    $t \leftarrow 0$
    Generate initial population $\Omega(t)$ of size Q;
    Evaluate $\Omega(t)$ fitness value; //Eq. (19)
    while (t<T) do
        $\Omega(t+1) \leftarrow$ Selection $[\Omega(t)]$ {Creation of new solutions}; // according Eq. (20)
        $\Omega(t+1) \leftarrow$ Crossover$[\Omega(t+1)]$ ; //one-point crossover
        $\Omega(t+1) \leftarrow Mutation[\Omega(t+1)]$;
        Evaluate $\Omega(t+1)$ fitness value;
        $t \leftarrow t+1$
    end while
    $i \leftarrow 0$ ; // one-step-ahead forecasting volatility
    set $\mathbf{P}(t) \leftarrow \vartheta\mathbf{I}$
    Evaluate $\hat{y}(i)$; //forecasting volatility
    while (i<M) do
        $\mathrm{P}(i+1) \leftarrow \{\mathbf{P}(i) - \mathbf{P}(i)\phi(i)(\lambda + \phi^t(i)\mathbf{P}(i)\phi(i))^{-1}\phi^t(i)\mathbf{P}(i)\}$ ;
        $\mathbf{\Lambda}(i+1) \leftarrow \mathbf{\Lambda}(i) + \mathbf{P}(i+1)\phi(i+1)(r^2(i+1) - \mathbf{\Lambda}^t(i)\phi(i+1))$ ;
        Evaluate $\hat{y}(i+1)$ ;
        $i \leftarrow i+1$ ;
    end while
end
```

Fig. 2. Pseudocode of fuzzy-SVR model.

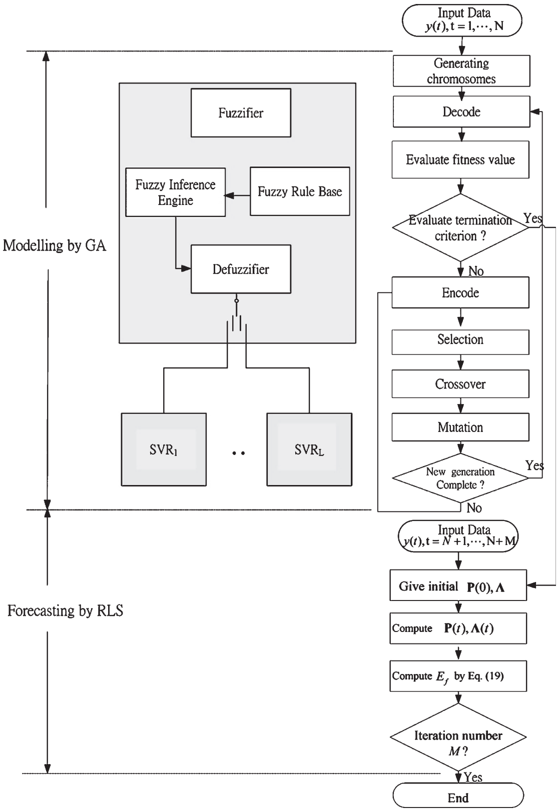


Fig. 3. Structure of the proposed method.

Step 9: Repeat the RLS algorithm to forecast the volatility in Equations (21) and (22) until the iteration returns M out-of-sample data.

Figure 2 shows the pseudocode of the fuzzy-SVR model, and Fig. 3 shows a flowchart of the design procedure for the fuzzy-SVR model.

4. Simulation

The data consisted of daily closing values for four stock indices from January 1, 2010, to December 31, 2013. This study focused on the Taiwan Stock Exchange weighted stock index (Taiwan), the NASDAQ Composite index, the Hang Seng index (Hong Kong), and the Shanghai Composite index (Shanghai) to illustrate the performance of the proposed method. The results were compared with those of other models, including the GARCH [4], GJR-CARCH [13], EGARCH [34], dynamic evolving neural-fuzzy inference system [21], SVR based on the GA [5], and various forecasting methods with multilayer perceptron (Combing MP) [41]. All the methods

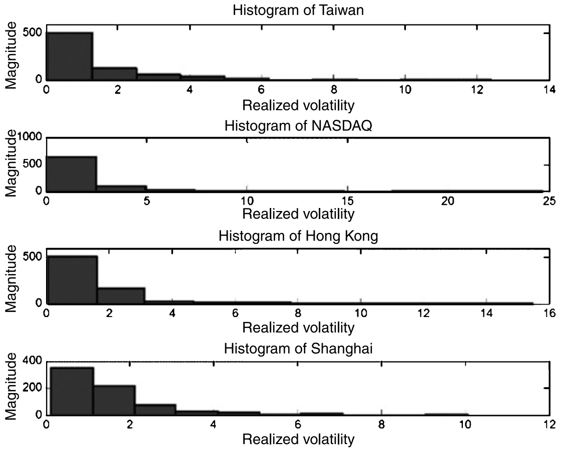


Fig. 4. Histograms of in-sample periods for realized volatility of Taiwan, NASDAQ, Hong Kong, and Shanghai.

Table 2
Results from out-of-sample volatility forecasting for markets in Taiwan, NASDAQ, Hong Kong, and Shanghai based on the proposed method with various number of fuzzy rules

		MAFE	MPFE	MSFE
Taiwan	L = 2	0.1278	0.2506	0.0476
	L = 3	0.1399	0.2824	0.0522
NASDAQ	L = 2	0.1409	0.2973	0.0551
	L = 3	0.1440	0.3132	0.0528
Hong Kong	L = 2	0.2597	0.3459	0.2114
	L = 3	0.2792	0.3635	0.2323
Shanghai	L = 2	0.5695	0.3781	1.5807
	L = 3	0.5617	0.3434	1.5351

were implemented in Matlab and evaluated on an Asus desktop PC with a 3.4 GHz i7-4770 CPU, RAM 16G, and Windows 8. The data were divided into two parts: the first half (January 1, 2010–December 31, 2012) of the sample comprised the in-sample data of the training set ($N = 748, 762, 750, 716$ of Taiwan, NASDAQ, Hong Kong, and Shanghai, respectively) and the second part (January 1, 2013–December 31, 2013) out-of-sample data of the test set ($M = 240, 259, 243, 254$ of Taiwan, NASDAQ, Hong Kong, and Shanghai, respectively). The GA optimizes parameters of the SVR model and proposed model. Practically, the generalization capability and accuracy of the SVR model and proposed model are determined according to searched problem parameters. In this study, the simulated results were obtained by averaging the results of 20 independent Monte Carlo (MC) runs to train the SVR model and proposed model for deriving a more reliable result [15]. For time-series forecasting, a cross-validation statistic was obtained using in-sample data to construct a model, and one-step-ahead predictions of out-of-sample data were then made to obtain the future solution [18].

Bollerslev [4] indicated that the GARCH (1,1) model accurately describes the volatility of financial data. Therefore, in this study, $n = 1$. Figure 4 presents histograms of the in-sample period volatility for the four financial markets, showing the concentrated clustering properties with few distribution regimes. The results indicated that volatility was concentrated in regimes with small fluctuations and few value distributions of the larger regime. The forecasting results according to various fuzzy rules are presented in Table 2. The results showed that $L = 2$ was more satisfactory than $L = 3$, but the Shanghai index has more distribution regimes than those of the other three financial markets. Therefore, the number of regimes is more than that of the other financial markets; the results are the same as those shown in Fig. 4. Moreover, the $L = 3$ and $L = 2$ performance is almost the same as that of the Shanghai index; thus, $L = 2$ was used. In general, the value of the daily stock return $r(t)$ is between –10% and 10%. Therefore, the upper and lower bounds of the parameters related to the fuzzy-SVR model are defined as

$$0 \leq m_{l1} \leq 10\%,\ 0 \leq \delta_{l1} \leq 10\%,\ \mathrm{e}^{-10} \leq C_l \leq \mathrm{e}^{10}, \\ \mathrm{e}^{-10} \leq \varepsilon_l \leq \mathrm{e}^5, \mathrm{e}^{-10} \leq \sigma_l \leq e^5 \ for\ l = 1, \ldots, L \tag{24}$$

The GA parameters [7, 24] are defined as

$$Q = 100, T = 20,\ p_c = 0.9, p_m = 0.01, \\ B_i = 12\ for\ i = 1, \ldots, (2n + 3) \times L, \tag{25}$$

where Q is the population size, T is the number of generations, p_c is the crossover rate, and p_m is the mutation rate. From the Table 1 show that $L = 2$ of the fuzzy rules, the fuzzy-SVR model with 10 parameters contains four parameters of the membership function (mean and variance) and six parameters of the SVR model; therefore, the chromosome total length is 120 bits. The simulated results were obtained by averaging 20 independent MC runs.

Table 3 lists the parameter estimates associated with the Taiwan, NASQ, Hong Kong, and Shanghai indices for the proposed method. As shown in Table 3, the two SVR models are distinct. The empirical results show that the volatility of the four markets exhibits characteristics such as clustering and exhibits the leverage effect. Table 4 depicts the various forecast statistics of the models. The results

Table 3
Values for the optimal parameters of the proposed method

		Mean	Standard deviation
Taiwan	$(m_{11},\ \delta_{11},\ \mathrm{C}_1,\ \varepsilon_1,\ \sigma_1)$	(0.0449, 0.1750, 740.13, 1.925, 2.45)	(0.0322, 0.0169, 333.07, 0.91, 1.463)
	$(m_{21},\ \delta_{21},\ \mathrm{C}_2,\ \varepsilon_2,\ \sigma_2)$	(1.6323, 0.7286, 1329.81, 2.22, 1.67)	(0.1232, 0.2643, 286.23, 1.67, 0.81)
NASDAQ	$(m_{11},\ \delta_{11},\ \mathrm{C}_1,\ \varepsilon_1,\ \sigma_1)$	(0.0526, 0.1832, 413.57, 2.08, 2.44)	(0.0408, 0.0339, 284.57, 0.98, 2.23)
	$(m_{21},\ \delta_{21},\ \mathrm{C}_2,\ \varepsilon_2,\ \sigma_2)$	(1.6409, 0.8052, 1809.33, 1.87, 2.19)	(0.1301, 0.2142, 873.41, 0.94, 1.64)
Hong Kong	$(m_{11},\ \delta_{11},\ \mathrm{C}_1,\ \varepsilon_1,\ \sigma_1)$	(0.0626, 0.1745, 520.23, 1.81, 2.32)	(0.0310, 0.0218, 385.63, 0.94, 0.91)
	$(m_{21},\ \delta_{21},\ \mathrm{C}_2,\ \varepsilon_2,\ \sigma_2)$	(1.5964, 0.6980, 1625.45, 1.49, 2.93)	(0.0666. 0.3307, 650.65, 0.79, 2.68)
Shanghai	$(m_{11},\ \delta_{11},\ \mathrm{C}_1,\ \varepsilon_1,\ \sigma_1)$	(0.0366, 0.1680, 585.60, 2.07, 2.34)	(0.0235, 0.01269, 312.53, 0.92, 1.38)
	$(m_{21},\ \delta_{21},\ \mathrm{C}_2,\ \varepsilon_2,\ \sigma_2)$	(1.5761. 0.7764, 1577.13. 2.78,3.04)	(0.0508, 0.2330, 551.05, 1.41, 1.08)

Table 4
Results from out-of-sample volatility forecasting for markets in Taiwan, NASDAQ, Hong Kong, and Shanghai based on the proposed method, GARCH, GJR-GARCH, EGARCH, and SVR based on the GA

			MATE	MPTE	MSTE	Training time (second)
Taiwan	GARCH		0.1609	0.4265	0.0696	0.0860
	GJR-GARCH		0.1661	0.4154	0.0798	0.1300
	EGARCH		0.1859	0.4406	0.0884	0.1300
	SVR based on GA		0.1555	0.3580	0.0611	141.68
	DENF1S with 40 rules		0.3243	0.4739	0.1648	0.8320
	Combing MP		0.3243	0.4739	0.1648	1.0584
	Proposed Method	Mean	**0.1278**	**0.2506**	**0.0476**	245.92
		(stand deviation)	(0.0048)	(0.0204)	(0.0026)	
NASDAQ	GARCH		0.2015	0.6416	0.1230	0.0940
	GJR-GARCH		0.1978	0.5454	0.1255	0.0980
	EGARCH		0.2158	0.6153	0.1286	0.1230
	SVR based on GA		0.1709	0.4540	0.0790	153.76
	DENF1S with 38 rules		0.4643	0.5584	0.3170	0.8620
	Combing MP		0.3243	0.4739	0.1648	1.0173
	Proposed Method	Mean	**0.1409**	**0.2973**	**0.0551**	
		(stand deviation)	(0.0018)	(0.0049)	(0.0010)	263.52
Hong Kong	GARCH		0.2962	0.6745	0.2460	0.0920
	GJR-GARCH		0.2859	0.6267	0.2372	0.0970
	EGARCH		0.2867	0.6391	0.2240	0.1120
	SVR based on GA		0.3275	0.6338	0.2975	140.70
	DENFTS with 38 rules		0.5457	0.7082	0.5002	0.8040
	Combing MP		0.3243	0.4739	**0.1648**	1.0994
	Proposed Method	Mean	**0.2597**	**0.3459**	0.2114	262.68
		(stand deviation)	(0.0043)	(0.0196)	(0.0034)	
Shanghai	GARCH		0.4174	0.6156	0.7255	0.1557
	GJR-GARCH		0.4096	0.5938	0.7108	0.5320
	EGARCH		**0.4049**	0.5684	**0.6911**	0.4700
	SVR based on GA		0.6206	0.5678	1.6693	131.2
	DENFTS with 44 rules		0.9707	0.7765	2.1336	0.8170
	Combing MP		0.5105	0.7386	0.7635	0.9706
	Proposed Method	Mean	0.5695	**0.3781**	1.5807	226.72
		(stand deviation)	(0.0048)	(0.0204)	(0.0096)	

Bold text is the best value of forecasting performance.

indicate that the volatility forecast from using the asymmetric GARCH models (e.g., the GJR-GARCH and EGARCH models) is superior to that of the GARCH model in the Hong Kong and Shanghai indices. This is because the realized volatility copes with the residual powers of past influence and negative shocks to returns give rise to greater volatility than equivalent positive shocks to returns [32]. Furthermore, the empirical results indicate that the Hong Kong and Shanghai stock market data are asymmetric. The information in Table 4 shows that the performance of the Taiwan, NASDAQ, and Hong Kong markets by SVR based on the GA is superior to that of the others models except for the proposed method because of the constructing data-driven capturing time-varying characteristics and the training model expending considerable time. The characteristic of the SVR is to minimize the generalization

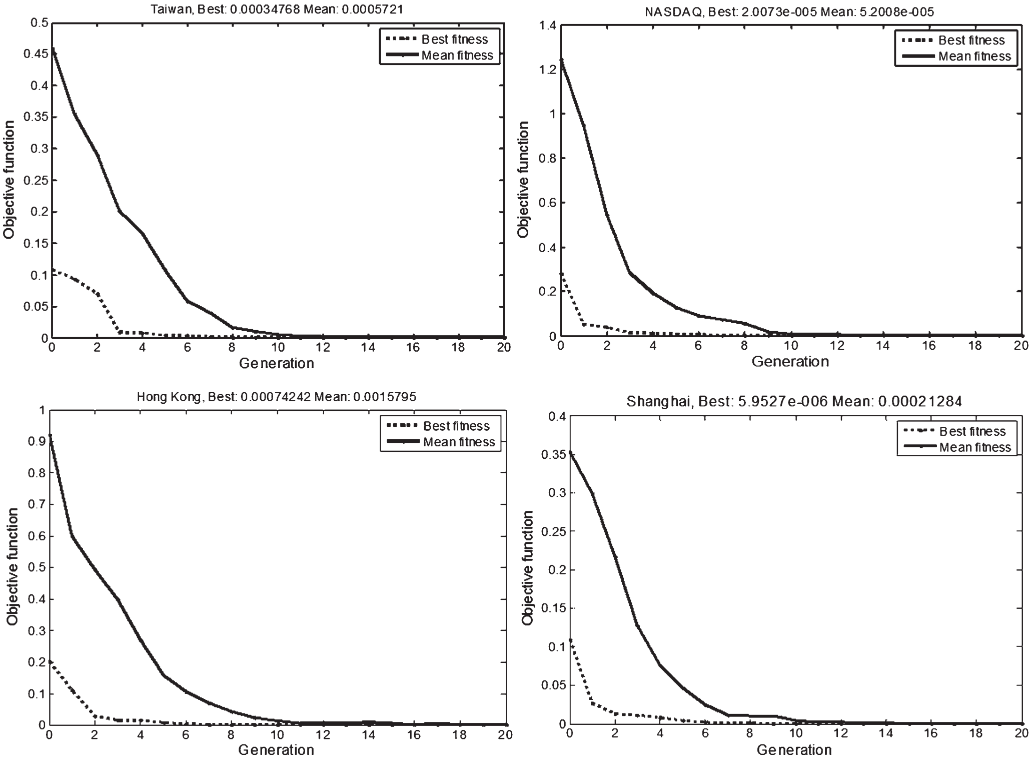


Fig. 5. Objective function E as evaluated using the GA for Taiwan, NASDAQ, Hong Kong, and Shanghai.

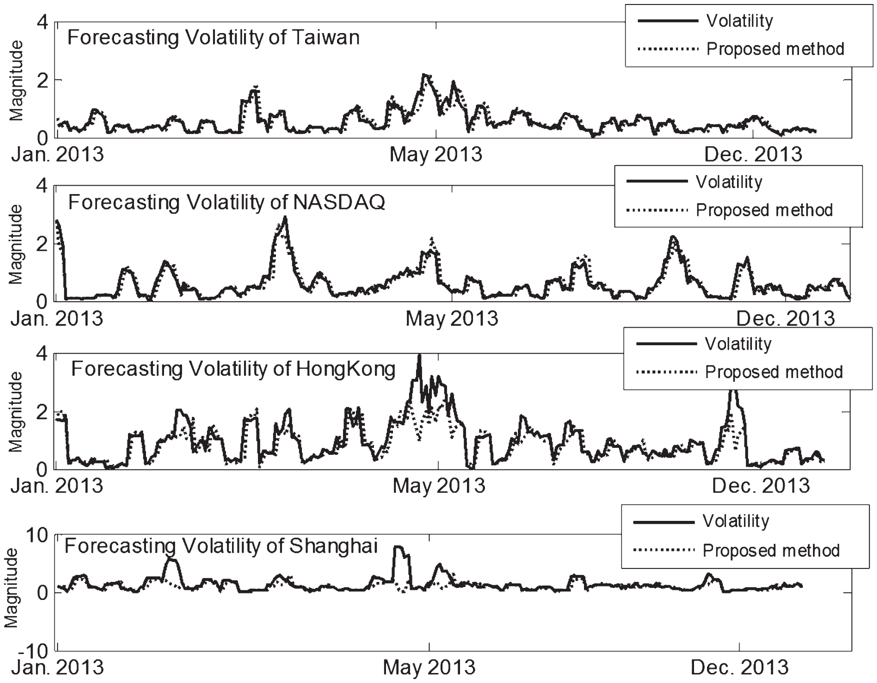


Fig. 6. Forecasting volatility by the proposed method for stock markets in Taiwan, NASDAQ, Hong, and Shanghai.

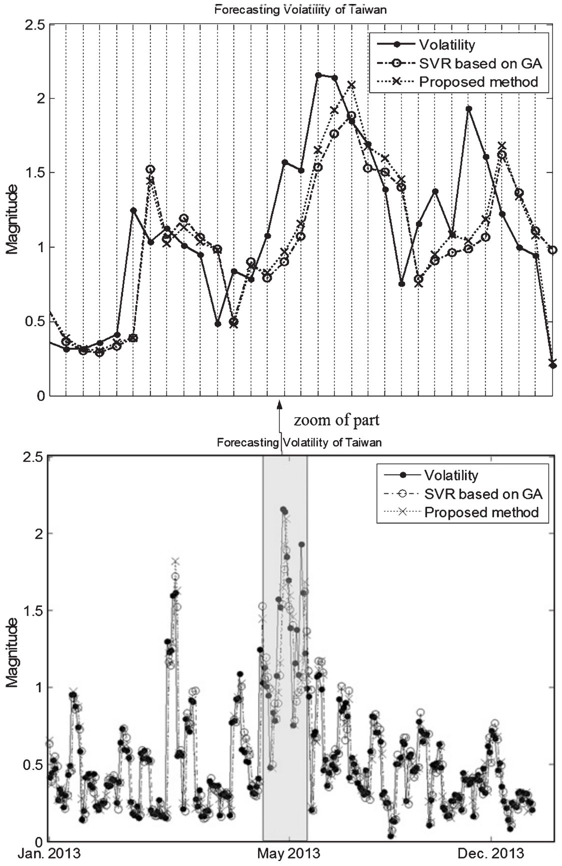


Fig. 7. Forecasting volatility and the zoomed part obtained using the proposed method and SVR based on the GA for stock markets in Taiwan.

error bound for achieving generalized performance, rather than minimizing only the mean square error over the data set [30]. The Shanghai realized volatility distribution regimes is more than those of the other three financial markets. Therefore, the based on the GA performance in MAFE and MSFE of Shanghai is worse than the persistence of volatility models (GARCH, GJR-GARCH, EGARCH) but the MPFE performance is almost better than the persistence of volatility model. The characteristic of SVR is to minimize the generalization error bound for achieving generalized performance, rather than minimizing only the mean square error over the data set [37]. In addition, the MPFE performance of the proposed method is superior to that of other models and the performance of the MAFE and the MSFE was more satisfactory than that of the other models, despite the training model expending considerable time. The proposed model led to an improvement exceeding 30% in the forecasting performance of the MPFE compared with that of the other models under is not taken training time. Therefore, both clustering and adaptive methods were used to capture time-varying data.

The convergence of the objective function E_f by using a GA-based estimator corresponding to a single run is shown in Fig. 5. Note that the cost functions of the GA-based estimator have exponential and rapid convergence at the beginning of generation and converge within eight generations. Figure 6 illustrates the volatility forecasting for the four markets according to the proposed method. The model involved using fuzzy rules to generate two SVR models and an adaptive method to forecast volatility. Therefore, the model could capture market volatility clustering and time-varying characteristics but not large change fluctuations, particularly for the part of Shanghai. Figure 7 shows the forecasting volatility and zoomed part for the Taiwan market obtained using the proposed method and the GA-based SVR model. According to the zoomed part, the proposed method could capture irregular behavior and fluctuations more effectively than the GA-based SVR model could, especially the fluctuations that rapidly change. The empirical results of this study indicated that the stock market data consist of clustering and time-varying fluctuations. The proposed fuzzy-SVR model outperformed the other models, as shown by the various rules concerning the forecast errors.

5. Conclusion

Volatility exhibits clustering and time-varying characteristics; moreover, many complex factors influence volatility. Therefore, this paper proposes a fuzzy-SVR artificial intelligence method. The predictive abilities of the volatility models were examined by comparing the forecast volatility measures defined in Equation (23). The proposed method appeared to be advantageous when we attempted to model both clustering and time-varying volatility. A GA-based design method was used to estimate parameters for the fuzzy-SVR model, and an adaptive algorithm was employed to forecast the volatility of various financial markets. The simulation results indicate that the proposed method afforded substantial improvements in forecasting performance. Moreover, the GA simultaneously evaluates many points in the search space. Therefore, specifying initial conditions is unnecessary for achieving improved results. The main disadvantage of the proposed method is that it cannot effectively capture large fluctuations of market volatility. The persistence models (GARCH, GJR-GARCH, EGARCH) involving conditional mean and conditional variance are frequently used to investigate the volatility in financial time series. They demonstrate the ability to capture the persistence of volatility but do not effectively capture clustering and time-varying characteristics. Future research problems entail designing a multifuzzy-SVR model or combining fuzzy rules to switch the persistence models or the SVR model to manage the varying fluctuation effects of forecasting volatility. Moreover, the optimal of the design model will is complexity problem and many local optimal may exist. In a future study, we will vary the meta-heuristics algorithm to evaluate the performance.

References

[1] S. Abbasbandy and B. Asady, Ranking of fuzzy numbers by sign distance, *Information Sciences* **176** (2006), 2405–2416.

[2] R.J. Almeida, N. Baştürk, U. Kaymak and M.C. João, Sousac, Estimation of flexible fuzzy GARCH models for conditional density estimation, *Information Sciences* **267** (2014), 252–266.

[3] G. Bloch, F. Lauer, G. Colin and Y. Chamaillard, Support vector regression from simulation data and few experimental samples, *Information Sciences* **178** (2008), 3813–3827.

[4] T. Bollerslev, Generalized autoregressive conditional heteroskedasticity, *Journal of Econometrics* **31** (1986), 307–327.

[5] E. Ceperic, V. Ceperic and A. Baric, A strategy for short-term load forecasting by support vector regression machines, *IEEE Transcations on Power Systems* **28** (2013), 4356–4364.

[6] S.M. Chen and Y.C. Chang, Multi-variable fuzzy forecasting based on fuzzy clustering and fuzzy rule interpolation techniques, *Information Sciences* **180** (2010), 4773–4783.

[7] J.J. Crefenstee, Optimization of control parameters for genetic algorithms, *IEEE Transactions on Systems, Man and Cybernetics* **16** (1986), 122–128.

[8] R. Engle, Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation, *Econometrica* **50** (1982), 987–1007.

[9] E.F. Fama, The behavior of stock market price, *Journal of Business* **38** (1965), 34–105.

[10] Y. Gao and Y.J. Liu, Adaptive fuzzy optimal control using direct heuristic dynamic programming for chaotic discrete-time system, *Journal of Vibration and Control* (2014), 1–9.

[11] V.V. Gavrishchaka and S.B. Ganguli, Volatility forecasting from multiscale and high-dimensional market data, *Neurocomputing* **55** (2003), 285–305.

[12] T. Van Gestel, J.A.K. Suykens, D.E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor and J. Vandewalle, Financial time series prediction using least squares support vector machines within the evidence framework, *IEEE Transactions on Neural Networks* **12** (2001), 809–821.

[13] L.R. Glosten, R. Jagannathan and D.E. Runkle, On the relation between the expected value and the volatility of the nominal excess return on stocks, *Journal of Finance* **48** (1993), 1779–1801.

[14] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, MA: Addition Wesley 1989.

[15] S.R. Gunn, Support Vector Machines for Classification and Regression, University of Southampton, 1998.

[16] P.Y. Hao and J.H. Chiang, A fuzzy model of support vector regression machine, *International Journal of Fuzzy Systems* **9** (2007), 45–50.

[17] C.F. Huang, A hybrid stock selection model using genetic algorithms and support vector regression, *Applied Soft Computing* **12** (2012), 807–818.

[18] J.C. Hung, Adaptive Fuzzy-GARCH model applied to forecasting volatility of stock markets using particle swarm optimum algorithm, *Information Sciences* **181** (2011), 4673–4683.

[19] K.C. Hung and K.P. Lin, Long-term business cycle forecasting through a potential intuitionistic fuzzy least-squares support vector regression approach, *Information Sciences* **224** (2013), 37–48.

[20] L.J. Kan, C.C. Chiu, C.J. Lu and J.L. Yang, Integration of nonlinear independent component analysis and support vector regression for stock price forecasting, *Neurocomputing* **99** (2013), 534–542.

[21] N.K. Kasabov and Q. Song, DENFIS: Dynamic evolving neural-fuzzy inference system and its application for time-series prediction, *IEEE Transactions on Fuzzy Systems* **10** (2002), 144–153.

[22] F. Klassen, Improving GARCH Volatility forecasts with regime-switching GARCH, *Empirical Economics* **27** (2002), 363–394.

[23] V. Kodogiannis and A. Lolis, Forecasting financial time series using neural network and fuzzy system-based techniques, *Neural Computing & Applications* **11** (2002), 90–102.

[24] T. Kozek, T. Roska and L.O. Chua, Genetic algorithm for CNN template learning, *IEEE Transcations Circuits Syst I* **40** (1993), 392–402.

[25] T.C. Lin, H.C. Huang, B.Y. Liao and J.S. Pan, An optimized approach on applying genetic algorithm to adaptive cluster validity index, *International Journal of Computer Sciences and Engineering Systems* **1** (2007), 253–257.

[26] Y.J. Liu, Y. Gao, S. Tong and Y. Li, Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete-time systems with dead-zone, *IEEE Transactions on Fuzzy Systems* **24** (2016), 16–28.

[27] Y.J. Liu and S. Tong, Adaptive fuzzy control for a class of nonlinear discrete-time systems with backlash, *IEEE Transactions on Fuzzy Systems* **22** (2014), 1359–1365.

[28] Y.J. Liu and S. Tong, Adaptive fuzzy identification and control for a class of nonlinear pure-feedback MIMO systems with unknown dead-zones, *IEEE Transactions on Fuzzy Systems* **23** (2015), 1387–1398.

[29] Y.J. Liu and S. Tong, Adaptive fuzzy control for a class of unknown nonlinear dynamical systems, *Fuzzy Sets and Systems* **263** (2015), 49–70.

[30] L. Maciel, F. Gomide and R. Ballini, Enhanced evolving participatory learning fuzzy modeling: An application for asset returns volatility forecasting, *Evolving Systems* **5** (2014), 75–88.

[31] M. Martens and D.V. Dijk, Measuring volatility with the realized range, *Journal of Econometrics* **138** (2007), 181–207.

[32] J. Mercer, Functions of positive and negative type and their connection with the theory of integral equations, *Philosophical Trans of the Royal Society A* **209** (1909), 415–446.

[33] R.C. Merton, On estimating the expected return on the market: An exploratory investigation, *Journal of Financial Economics* **8** (1980), 323–361.

[34] B. Nelson, Conditional heteroskedasticity in asset returns: A new approach, *Econometrica* **59** (1991), 347–370.

[35] L.M. San Jose-Revuelta, A new adaptive genetic algorithm for fixed channel assignment, *Information Sciences* **177** (2007), 2655–2278.

[36] L.J. Tashman, Out-of-sample tests of forecasting accuracy: An analysis and review, *International Journal of Forecasting* **16** (2000), 437–450.

[37] F.E.H. Tay and L. Cao, Application of support vector machines in financial time series forecasting, *Omega: The International Journal of Management Science* **29** (2001), 309–317.

[38] R.S. Tsay, Analysis of financial time series. John Wiley & Sons, USA, 2002.

[39] V. Vapnik, S. Golowich and A. Smola, Support vector method for function approximation, regression estimation, and signal processing, *Neural Information Processing Systems* **9** (1997), 281–287.

[40] B. Wang, H. Huang and X. Wang, A support vector machine based MSM model for financial short-term volatility forecasting, *Neural Computing and Applications* **22** (2013), 21–28.

[41] W. Xiao, Q. Fei and W. Chen, Forecasting Chinese Stock Markets Volatility Based on Neural Network Combining, *Fourth International Conference on Natural Computation IEEE-ICNC*, 2008, pp. 23–27.

[42] H. Yang, L. Chan and I. King, Support Vector Machine Regression for Volatile Stock Market Prediction, *Intelligent Data Engineering and Automated Learning: IDEAL*, 2002, pp. 391–396.

[43] L.A. Zadeh, Fuzzy set, *Information and Control* **8** (1965), 338–353.

[44] X. Zheng and B.M. Chen, Identification of stock market forces in the system adaptation framework, *Information Sciences* **265** (2014), 105–122.

[45] G. Zhiqiang, W. Huaiqing and L. Quan, Financial time series forecasting using LPP and SVM optimized by PSO, *Soft Computing* **17** (2013), 805–818.