Artificial Intelligence Review (2019) 52:743–773
https://doi.org/10.1007/s10462-018-9663-x

Forecasting financial series using clustering methods and support vector regression

Lucas F. S. Vilela[1] · Rafael C. Leme[1] · Carlos A. M. Pinheiro[1] · Otávio A. S. Carpinteiro[1]

Published online: 23 October 2018

Abstract
This paper proposes a two-stage model for forecasting financial time series. The first stage uses clustering methods in order to segment the time series into its various contexts. The second stage makes use of support vector regressions (SVRs), one for each context, to forecast future values of the series. The series used in the experiments is composed of values of an equity fund of a Brazilian bank. The proposed model is compared to a hierarchical model (HM) presented in the literature. In this series, the HM presented prediction results superior to both a support vector machine (SVM) and a multilayer perceptron (MLP) models. The experiments show that the proposed model is superior to HM, reducing the forecasting error of the HM by 32%. This means that the proposed model is also superior to the SVM and MLP models. An analysis of the construction and use of clusters associated with a series volatility study shows that data obtained from only one type of volatility (low or high) are enough to provide sufficient knowledge to the model so that it is able to forecast future values with good accuracy. Another analysis on the quality of the clusters formed by the model shows that each cluster carries different information about the series. Furthermore, there is always a group of SVRs capable of making adequate forecasts and, for the most part, the SVR used in forecasting is a SVR belonging to this group.

Keywords Financial time-series forecasting · Clustering · Support vector machine · Artificial intelligence

✉ Lucas F. S. Vilela
lucas@inatel.br

Rafael C. Leme
leme@unifei.edu.br

Carlos A. M. Pinheiro
pinheiro@unifei.edu.br

Otávio A. S. Carpinteiro
otavio@unifei.edu.br

[1] Research Group on Systems and Computer Engineering, Federal University of Itajubá, Itajubá, MG 37500-903, Brazil

1 Introduction

A time series S is a sequence of values (or observations) $\{s_1, s_2, \ldots, s_n\}$ of an event, where the values have a defined periodicity, such as hourly, daily, weekly, monthly, annual, among others (Box et al. 2008; Anderson et al. 2016). The forecast of future values of time series is commonly used, in a wide range of areas, to provide support for planning and decision making. Some factors may change the level of forecasting difficulty, making it either simpler or more complex. Some of these factors are knowledge about elements that may influence the behavior of the generating event of the time series, the amount of information available in the series, and the ability of previous forecasts to influence forecasts of future values of the series (Makridakis et al. 1998).

Financial time series are characterized by having noise and not being stationary. This makes both modeling and forecasting their future values more difficult. The noise is presented by the lack of information in the series, information that allows one to establish a correct relationship between past and future observations. Non-stationarity is related to the change in the statistical distribution of the series over time. Thus, to introduce more information in the series, which could lead to a decrease in the effect of the noise, would bring the problem of higher non-stationarity (Abu-Mostafa and Atiya 1996; Cao and Tay 2001a).

The creation of forecasting models for time series began with linear statistical models. These models had restrictions, such as not being able to capture all the information in the series, including volatility and periodicity. The evolution of the research led to the use of non-linear models, aimed at giving better treatment to the non-linearities of the series (Tong 2002). Some examples of non-linearities that may be mentioned are the different volatilities (low and high), and the change of the information relevant to the forecast, for different segments of the series (Clements et al. 2004).

The use of artificial neural models and learning machines to forecast future values came next. In addition to being non-linear models, they are constructed exclusively from the values of the series itself. Additionally, as these are machine-learning models, they are able to learn even with incomplete information, hence they are less susceptible to noise.

Machine learning models have been widely used for financial time series analysis and forecasting. For instance, Zhong and Enke (2017) tested dimensionality reduction techniques and artificial neural network (ANN) models to forecast values in a financial time series, Kara et al. (2011) compared classification performance of ANN and support vector machines (SVMs) models in predicting financial time series, and Kim and Enke (2016) developed a trading system based on rough set analysis. At last, Atsalakis and Valavanis (2009) surveys more than a hundred papers that describe the use of machine learning models in financial time series analysis and forecasting.

However, the application of machine learning models in time series forecasting is not restricted to financial series (Perwej and Perwej 2012; Shen and Xing 2009; Wang et al. 2011; Yizhen et al. 2011). It is also used in Internet traffic forecasting (Chabaa et al. 2010; Haviluddin and Alfred 2015) and in weather forecasting (Singh et al. 2011), among others.

Among machine learning models, SVMs are widely used in forecasting time-series values (Sapankevych and Sankar 2009; Van Gestel et al. 2001; Rosowsky and Smith 2013). Their accuracy in forecasting values is usually superior to that of neural models in a number of studies (Cao and Tay 2001a, 2003; Carpinteiro et al. 2012).

Several studies have extended the forecasting models, and these have also proposed to treat the different contexts in the time series. Contexts are the different periods of the series, which have unique characteristics due to their non-stationarity.

The treatment of the context of a piece of information consists of segmenting this information, according to its characteristics, and treating these segments in a different way so that, in the end, they can be used according to their characteristics. Some methods (or models) used to treat contexts are self-organizing maps (SOMs) (Carpinteiro et al. 2012; Cao and Tay 2001b) and genetic algorithms (Armano et al. 2005), being used successfully not only in time series forecasting (Carpinteiro et al. 2012; Cao and Tay 2001b; Armano et al. 2005), but also in pattern recognition (Kutics et al. 2013; Popovici and Andonie 2015).

This paper proposes a two-stage model for forecasting financial time series. The first stage uses clustering methods—K-Means or Fuzzy C-Means—to segment the time series in its various contexts. The second stage makes use of support vector regressions (SVRs), one for each context, to forecast future values of the series. Thus, for each pattern presented to the model, the relevance of this pattern to the clusters is verified, and then the forecast is made. Two processes are used to carry out the forecast. In the first one, only the SVR related to the cluster most pertinent to the pattern presented is used to forecast the future value. In the second one, the forecast value is a weighted composition of the forecasts of the SVRs of all clusters, and the weights of each forecast are defined according to the membership of the pattern presented in relation to each cluster.

The series used in the experiments is composed of values of an equity fund of a Brazilian bank. The proposed model is compared, on this same series, to a hierarchical model (HM) presented in Carpinteiro et al. (2012). The HM is composed of an SOM, responsible for the segmentation of the series in contexts, and an SVM, responsible for the forecast. In Carpinteiro et al. (2012), the HM presented higher forecast results than a model composed of an SVM and to those of another model composed of a multilayer perceptron (MLP). Thus, the comparison of the proposed model with the HM also allows an indirect comparison with the SVM and MLP models.

The remainder of this paper is organized as follows. The second section reviews some existing works on forecasting in financial series, addressing the models used as well as the results achieved. The third section contains a brief review on the K-Means and Fuzzy C-Means clustering methods, and on SVM and SVR. The following section describes the architecture of the proposed model. The fifth section presents the financial series used in the experiments as well as the treatments applied therein. The sixth section describes the experiments performed. The seventh section presents and discusses the results obtained, and also provides analyses on the clustering of the values of the series and on the treatment and usage of volatility by the proposed model. Lastly, the eighth section concludes the paper and presents directions for future work.

2 Related work

SVMs are widely used in models for forecasting values in time series. The following paragraphs describe some of the SVM-based models proposed in the literature.

Cao and Tay (2001a) proposed the use of an SVM to forecast financial time series. The series used in the experiments were extracted from five futures contracts of the Chicago Mercantile Market, using the closing price of the contract in horizons of approximately three and a half years as samples. The forecast is made over a five-day horizon. The authors analyze the impact of the free parameters of the SVM on the quality of the forecast. The SVM forecasts were compared to those of an MLP, using four different metrics. The forecasts provided by the SVM were more accurate than those provided by the MLP.

Cao and Tay (2003) analyze in detail the effects of the different values of the free parameters of the SVM in the outcome of financial series forecasting. Thus, they propose an SVM with adaptive free parameters. The authors show that the parameters C and δ^2 directly influence the result of the forecast, but that it is insensitive to the parameter ε, which influences the number of support vectors. The SVM uses C and ε as adaptive parameters, so as to give more weight to samples closer to the period to be predicted. The series used in the experiments were the same as those used in Cao and Tay (2001a). The adaptive SVM forecasts were compared to those of a standard SVM, of an MLP, and of a radial basis function (RBF) network. The adaptive SVM forecasts were more accurate than those of the standard SVM, of the MLP and of the RBF.

Yang et al. (2002) analyzed the effect of the SVM's margin on the forecast result, since small margins lead to overfitting and high-value margins can produce over generalization. They then proposed an adaptive margin, calculated according to the standard deviation of the series, which indicates the volatility of the asset. In this way, small and large margins can be used in periods of low and high volatility, respectively. Using, as a financial series, the closing price of the HSI index of the Hong Kong Stock Exchange, the authors showed that the SVM with adaptive margin produces more accurate forecasts than those produced by a standard fixed-margin SVM.

Yang et al. (2004) extended the use of adaptive margin for the treatment of outliers. Using a loss function insensitive to ε, outliers are detected when the distance between them and the margin exceeds a given threshold. Once detected, the margin is adapted in order to diminish their effects on the SVM. The series used in the experiments were extracted from the NASDAQ, HSI, and FTSE 100 indexes. The results indicated that reducing the effects of the outliers increases the accuracy of the SVM.

The approach used by Huang et al. (2005) differs from the others, because it seeks to predict the direction of the movement of financial time series, rather than to forecast future values per se. The series used in the experiments consisted of the weekly value of the Japanese NIKKEI 255 index and by two exogenous variables—US-based S&P500 index, and exchange rate between the US dollar and the Japanese yen—which have a strong correlation with the NIKKEI 255 index. The authors compared the results obtained from the SVM with those from the linear and quadratic discriminant analysis and the Elman neural model. The results indicated that the SVM motion direction hit rate is 73%, which is higher than the other two models.

Kara et al. (2011) and Patel et al. (2015) compared classification performance of ANNs and SVMs in predicting financial time series. Instead of forecasting stock prices, the authors focused in predicting the direction of movement. In modeling ANNs, they considered multilayer perceptrons with feedforward training approach. As for SVMs, they considered radial and polynomial basis SVMs. While Kara et al. (2011) have shown that ANNs performed better than SVMs in predicting the direction of movement, Patel et al. (2015) found that SVMs performed better in their experiments.

Yu et al. (2015) proposed the usage of Least Squares Support Vector Regression (LSSVR) for crude oil price forecasting. The proposed work uses a hybrid grid method and genetic algorithm approach to select the LSSVR parameters. Yu et al. (2016) proposed another method for selecting LSSVR parameters, again applied to crude oil price forecasting. The proposed work uses an ensemble learning paradigm, in which the user-defined parameters are considered as random factors to construct the model. Both the proposed models, compared to other LSSVR variants and to other forecasting techniques, were considered effective in terms of prediction accuracy and processing time.

Yu et al. (2017) assessed the usage of SVR for crude oil price forecasting in terms of one-step ahead prediction. The SVR model was compared to other five different models—a feed-forward neural network model, an auto-regressive integrated moving average (ARIMA) model, a fractional integrated ARIMA model (ARFIMA), a Markov-switching ARFIMA model, and a random walk model. The results showed that the SVR model outperformed the other five models in both prediction accuracy and directional prediction.

Limei and Xuan (2017) proposed a novel multi-kernel algorithm applied to a weighted SVR, aiming to forecast power system short-term load. Experimental results showed good generalization performance with shorter prediction time.

Azad et al. (2018) and Lin et al. (2018) used, respectively, SVR and LSSVR to forecast electric load demand. Both used past records of load consumption as input and weather factors as exogenous information. Liu et al. (2015) used SVR to forecast building energy consumption. Experimental results showed that SVR-based models are effective for load forecasting.

Zhao and Yu (2017) proposed a combined forecast model, composed of a discrete gray (DG) model, responsible for predicting the original time series, and a LSSVR model, used to correct residual errors. The DGM-LSSVR model proved to be advantageous in terms of effectiveness and feasibility.

The works listed below propose models that aim to separate and/or give different treatment to the different contexts of the financial series. What they all have in common is the fact that the models are composed of two stages. The first stage is always to extract, from the series, their respective contexts. The second stage uses the contexts formed by the previous stage to produce the forecast values. As will be seen, the use of two stages greatly increases the forecasting accuracy of the models.

Cao and Tay (2001b) proposed a two-stage model for forecasting values in financial time series. The first stage consists of several SOMs that cluster the series values into regions of their maps. The second one is composed of expert SVMs, relating to each region. In their experiments, the authors used two financial series, obtained from two different sources—Santa Fe Time Series Prediction Analysis Competition and Chicago Mercantile Market. The results indicated that the proposed model produced forecasts with higher quality than those produced by a single SVM.

In Cao (2003), the use of expert SVMs was extended to non-financial series. The model used is identical to the one proposed in Cao and Tay (2001b). The results of the experiments again demonstrated the superiority of the two-stage model over a model composed of a single SVM, in terms of accuracy of the forecasts.

Armano et al. (2005) also proposed a model composed of two stages. The first stage makes use of genetic algorithms to separate the values of the series into contexts. The second one, comprising expert MLPs relating to clusters, performs the forecasts. The series used in the experiments was extracted from the values of the COMIT and S&P500 indexes. The authors compared the results of the model with those of simulated operations using the Buy & Hold technique (buy a security and keep it for a long time). The proposed model was shown to be more efficient, offering more profitable options for the investor.

Carpinteiro et al. (2012) also proposed a two-stage HM. The first stage, consisting of an SOM, clusters the values of the series into regions of its map. The second one, consisting of an SVM, performs the forecasts. The series used in their experiments is the same one used in the experiments described herein. The authors compared the results obtained by the HM with those obtained by an MLP neural network and an SVM. The results showed that the HM has higher accuracy than the MLP and SVM.

Lastly, Sapankevych and Sankar (2009) survey the application of SVMs in forecasting time series values. The authors survey sixty-six papers, one-third of which address forecasting financial series values. They affirm that financial series are intrinsically non-linear and that SVMs are more suitable models to forecast values of these series, since they model non-linear processes in a more satisfactory way.

3 Brief review of the theory involved

3.1 K-Means

The K-Means clustering method aims to separate n objects, belonging to a data set, into k disjoint clusters (MacQueen 1967). Its concept can be extended to any clustering method that determines the membership of an object in a cluster, based on the distance between the object and the centroid of the cluster (Everitt et al. 2011).

The centroid c_i of a cluster is the average of all the objects belonging to the cluster, calculated through Eq. 1, where $u_{ij} \in \{0, 1\}$ is the membership of the object x_j in cluster i.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij} x_j}{\sum_{j=1}^{n} u_{ij}} \tag{1}$$

The basic rules of the K-Means method are:

1. empty clusters are not allowed;
2. an object must belong to a single cluster;
3. the union of all clusters forms the original set of objects.

Clustering methods that satisfy the second rule are called hard clustering methods. K-Means is therefore a hard-clustering method.

The basic steps of the algorithm of the K-Means clustering method are as follows:

1. k initial clusters with random centroids are created;
2. each object of the set is assigned to the cluster whose centroid is closest to it;
3. after the inclusion of an object in a cluster, its centroid is recalculated;
4. steps 2 and 3 are repeated until there is no change of objects between clusters.

A clustering is considered optimal when the sum of the distances between the objects belonging to a cluster and its centroid, for all clusters, is minimal (Oliveira and Pedrycz 2007). Different centroids created in the first step of the algorithm can lead to different clusterings, which means that the algorithm can fall in local optima. Hence, the algorithm is sensitive to initial centroids (Everitt et al. 2011), and should be performed several times to avoid local optima (Oliveira and Pedrycz 2007).

3.2 Fuzzy C-Means

Unlike hard clustering methods, the Fuzzy C-Means method allows partial membership of objects to clusters. If u_{ij} is the membership of object x_j in cluster i, its value can vary from 0, which means no membership, up to 1, indicating total membership. Therefore, $u_{ij} \in [0, 1]$. Methods that allow partial membership are called soft clustering methods. The concept of degree of membership originates from the fuzzy sets, described in Zadeh (1965).

The use of partial memberships relaxes the second basic rule of the K-Means method, described above, and makes objects belonging to a given cluster similar, to some degree, to objects in other clusters (Zadeh 1965).

Membership u_{ij} of object x_j in cluster i can be calculated through Eq. 2,

$$u_{ij} = \frac{1}{\sum_{l=1}^{c} \left(\frac{d_{ij}^2}{d_{il}^2} \right)^{\frac{1}{m-1}}} \tag{2}$$

where d_{ij} is the distance between the object x_j and the centroid c_i of cluster i, and m, called the fuzzifying parameter, indicates how fuzzy the clusters are. When $m = 1$, membership is full, just as in hard clustering methods. When $m > 1$, membership is partial. The value $m = 2$ is a common value for this parameter (Oliveira and Pedrycz 2007).

The calculation of the centroid c_i of a cluster i not only considers its objects, but also their memberships in all clusters, as described by Eq. 3.

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}^m x_j}{\sum_{j=1}^{n} u_{ij}^m} \tag{3}$$

The basic rules of the Fuzzy C-Means method are:

1. empty clusters are not allowed;
2. the sum of the object's memberships in the clusters must be equal to 1.

The basic steps of the Fuzzy C-Means method algorithm are (Bezdek et al. 1984):

1. parameters c and m are chosen. The membership matrix $U = (u_{ij} \in [0, 1])$ is created with random values;
2. the centroids $c_i^{(k)}$ are calculated through Eq. 3;
3. the matrix $U^{(k+1)}$ is updated through Eq. 2, making use of the centroids calculated in step 2;
4. the matrices $U^{(k+1)}$ and $U^{(k)}$ are compared. If the difference between them is less than a given threshold, the algorithm is terminated. Otherwise, the algorithm goes back to the second step.

Other possible alternatives for the stopping criterion of the algorithm can be defined, such as when the algorithm reaches a maximum number of iterations or when the variation of centroids between iterations reaches a minimum value (Oliveira and Pedrycz 2007).

An optimum Fuzzy C-Means clustering can be identified through a commonly used function (Bezdek et al. 1984), described by Eq. 4.

$$\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^m d_{ij}^2 \tag{4}$$

3.3 SVM and SVR

Support vector machine (SVM), initially described in Boser et al. (1992), and detailed in Vapnik (1998), is a learning machine capable of providing optimal solutions for pattern classification by constructing a hyperplane that maximizes the separation between positive and negative patterns (Haykin 2009).

An optimum hyperplane, defined by Eq. 5, can be obtained by finding the optimal values for w and b.

$$w^{\mathrm{T}}x + b = 0 \tag{5}$$

Such values can be found through the solution of the Lagrange function (Vapnik 1998), described in Eq. 6.

$$L = \frac{1}{2}w^{\mathrm{T}}w - \sum_{i=1}^{N}\alpha_i\left[d_i(w^{\mathrm{T}}x_i + b) - 1\right] \tag{6}$$

For non-linear classification problems, the input patterns must be transported to a high-dimensional space, where they can be linearly separable. This transformation of the data is represented by the function $\phi(x)$, taking the Lagrange function (Eq. 6) to be solved through Eq. 7.

$$L = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j d_i d_j \phi^{\mathrm{T}}(x_i)\phi(x_j) \tag{7}$$

Provided that there is a function, as shown in Eq. 8, that returns the scalar product of two vectors, it is not necessary to have knowledge about the transformation of the data per se.

$$K(x_i, x_j) = \phi^{\mathrm{T}}(x_i)\phi(x_j) \tag{8}$$

This function is called the kernel function. Any function that satisfies Mercer's theorem (Mercer 1909) can be used as a kernel function (Vapnik 1999). Among the existing kernel functions, the most widely used ones are linear, polynomial, and radial basis functions (Haykin 2009).

SVMs can be used for regression. In order to do so, one must minimize the risk function, described in Eq. 9, which in turn uses the loss function insensitive to ε, described in Eq. 10. This last function is commonly used in regression SVMs. A regression SVM is known as a support vector regression (SVR) (Vapnik 1998).

$$R(w, x, b) = \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}L_\varepsilon(d, y) \tag{9}$$

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon, & \text{for } |d - y| \geq \varepsilon \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

These functions have the following restrictions:

$$\begin{aligned} d_i - y_i &\leq \varepsilon + \xi_i \\ y_i - d_i &\leq \varepsilon + \xi_i^* \\ \xi_i &\geq 0 \\ \xi_i^* &\geq 0 \end{aligned} \tag{11}$$

where ξ_i and ξ_i^* are independent, non-negative slack variables describing the loss function insensitive to ε (Eq. 10) (Haykin 2009).

Thus, as in pattern classification problems, this optimization problem is also solved through Lagrange multipliers.

The support vector machines have the characteristic of good generalization capacity, reaching high performance in both pattern classification and regression problems (Haykin 2009).

3.4 Error measurement used

Error measurements play an important role in forecasting time series. They act as quality indicators, allowing the comparison between different forecasting models.

Among the existing error measures, the mean absolute percentage error (MAPE) stands out because it is an absolute measure, i.e. its value is not affected by the magnitude of the data (Armstrong and Collopy 1992). The MAPE value is calculated through Eq. 12, where e_t is the difference between the actual and predicted values and Y_t is the actual value.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{e_t}{Y_t}100\right| \tag{12}$$

The standard deviation measures the dispersion of the values in relation to their mean. The standard deviation was used to measure the dispersion of the MAPEs. Low standard deviation values are desirable because they indicate that errors are concentrated in a range of values close to the mean.

Predictive models that produce similar errors are more reliable than those that produce errors that are at times very small, and other times very large. It is important, therefore, for a predictive model to produce small average errors, which indicates accuracy, with low standard deviation, which indicates reliability.

3.5 Statistical measurement used

In addition to error measurements, statistical measurements of forecast values can also be used as quality indicators of forecasting models. Theil's U statistic is an accuracy measurement that highlights the largest errors produced by a model (Small and Wong 2002).

The Theil's U statistic value can be calculated through Eq. 13, where Y is the real value of the series and F, the forecast value.

$$U = \sqrt{\frac{\sum_{i=1}^{n-1}\left(\frac{F_{t+1}-Y_{t+1}}{Y_t}\right)^2}{\sum_{i=1}^{n-1}\left(\frac{Y_{t+1}-Y_t}{Y_t}\right)^2}} \tag{13}$$

The Theil's U statistic can be interpreted as the relation between the Root Mean Square Error (RMSE) of the forecasting model and that of the no-change naïve model (Makridakis et al. 1998). The no-change naïve model is a model in which the forecast value produced is always equal to the previous value. The RMSE is calculated through Eq. 14.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_t - F_t)^2} \tag{14}$$

The value produced by Theil's U statistic is interpreted as:

- Value 1 means that both the forecasting model and the no-change naïve model yield forecasts with the same accuracy;
- Values smaller than 1 mean that the forecasting model yields more accurate forecasts than the no-change naïve model;
- Values greater than 1 mean that the forecasting model yields less accurate forecasts than the no-change naïve model. In this case, the forecasting model is of no value.

3.6 Statistical predictive accuracy

The statistical Harvey, Leybourne and Newbold (HLN) test is carried out to check significant differences between forecast models. HLN test is an adjusted Diebold–Mariano (DM) test for small samples, which compares the forecasts of competing time series models i and j. Considering a loss function $g(.)$ and a prediction error $e_{.t}$, the difference between the forecasts of models i and j may be written as $d_t = g(e_{it}) - g(e_{jt})$. Thus, HLN tests the null hypothesis $H_0 : d_t = 0$ against the alternative $H_1 : d_t \neq 0$ (see Diebold and Mariano 2002; Harvey et al. 1998, for discussion). The HLN statistical test for h time horizon forecast is given by

$$HLN = \frac{1}{h}\sqrt{\frac{(1-h+h(h-1))}{\left(\gamma_0 + 2\sum_{l=1}^{h}\gamma_k\right)}}\sum_{t=1}^{h} d_t \tag{15}$$

where γ_l refers to the autocovariance of lag-l, discussed in the next section. Harvey et al. (1998) shows that the statistic HLN follows a t distribution with $h-1$ degrees of freedom.

4 Financial series

As mentioned above, a time series is defined as a sequence of observations obtained in a chronological order. Time series usually present a random behavior. Although one cannot forecast exactly (deterministically) their future values, they present a defined type of stochastic behavior.

In general, when analyzing financial time series, the continuous compound return is considered. The continuous compound return series is a function of asset prices P_t at time t, and is given by Eq. 16.

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) = \ln(P_t) - \ln(P_{t-1}) \tag{16}$$

Continuous compound return series (from now on, referred to as return time series) present desirable features for analyzing asset price behavior. For instance, return time series usually do not show trend or scale components, and are weakly stationary.

When a time series is stationary, the random variables that describe the stochastic process over time present strictly the same probability distribution. On weakly stationary series, however, such assumption is relaxed to a constant mean and autocovariance between r_t and r_{t-l} over time, where l is an arbitrary integer.

The weakly stationarity assumption allows one to estimate the autocovariance function by computing the lag-l autocovariance $\gamma_l = Cov(r_t, r_{t-l})$. Similarly, the autocorrelation function may also be estimated by computing the lag-l autocorrelation $\rho_l = Cor(r_t, r_{t-l})$. Equation 17 tests its statistical significance on a hypothesis test of null hypothesis $H_0 : \rho_l = 0$ against the alternative $H_1 : \rho_l \neq 0$

$$t_{ratio} = \frac{\hat{\rho}_l}{\sqrt{1 + 2\sum_{i=1}^{l-1}\hat{\rho}_i / T}} \tag{17}$$

where $\hat{\rho}_l$ is the estimated value for lag-l autocorrelation ρ_l. A deep discussion on financial time series features and analysis may be found in Tsay (2010).

The financial series used in the experiments consists of the series of daily price values of the "IBrX Indexado" equity fund, managed by Banco do Brasil S.A. (Bank of Brazil 2010).

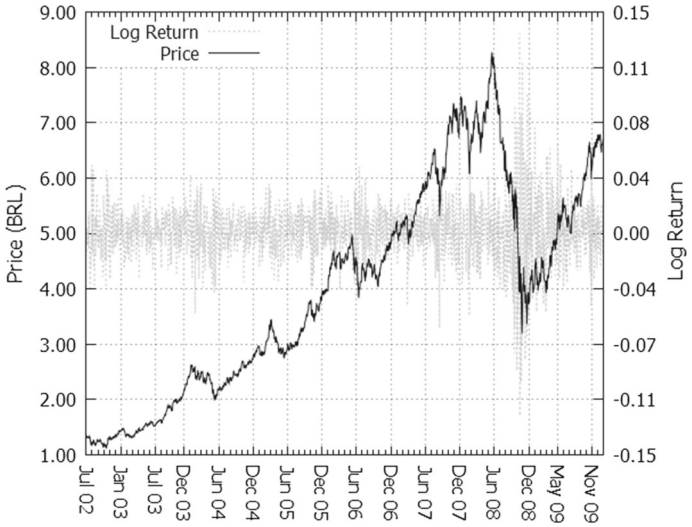

Fig. 1 Price series and return series

This fund follows the behavior of the IBrX index (BOVESPA 2017), which measures the performance of the 100 most highly traded shares on the Brazilian stock exchange (BOVESPA). The range of the analyzed series comprises the period between July 1, 2002 and December 31, 2009, totaling 1889 values.

The price series and the return series, obtained by Eq. 16, are shown in Fig. 1. In the figure, one can see that trend and scale components have been removed from the original series, converting the return series into a horizontal form, with values around zero. One can also see that the periods of greater magnitude of the return series are those in which the price values varied more sharply.

The statistical method of autocorrelation was applied to the return series. The autocorrelation function is depicted in the Fig. 2, where each stem refers to the estimated lag-l autocorrelation $\hat{\rho}_l$, and the lines refer to the critical t_{ratio} to reject $H_0 : \rho_l = 0$. That is, the estimated $\hat{\rho}_l$, outside the bounds of the lines, shows an evidence against the null hypothesis, so that $H_1 : \rho_l \neq 0$ is accepted. Here, a significance of 5% was considered. Figure 2 is usually referred to as autocorrelogram or, simply, correlogram.

Note, from Fig. 2, that the autocorrelation function indicates which previous values of the series are correlated with the future value that one wishes to forecast, thus reducing the complexity of the forecast. Through the correlogram, one can see that there are three past values that have significant correlation with the next value of the series. They are the autocorrelation values $\hat{\rho}_1$, $\hat{\rho}_2$ and $\hat{\rho}_3$, also known as r_{t+f-1}, r_{t+f-2}, r_{t+f-3}. Although $\hat{\rho_{17}}$ and $\hat{\rho_{18}}$ also seem significant, they are usually spurious, and may be neglected in the time series modeling.

Another feature of this time series may be observed in the Fig. 1. As usually found in financial time series, uptrend periods usually present lower volatility than downtrend periods. Indeed, it is well known in stock market literature that a rising price (bullish) market usually moves slower than a declining price (bearish) market, so that the return of security may also be impacted by its volatility (Tsay 2010).

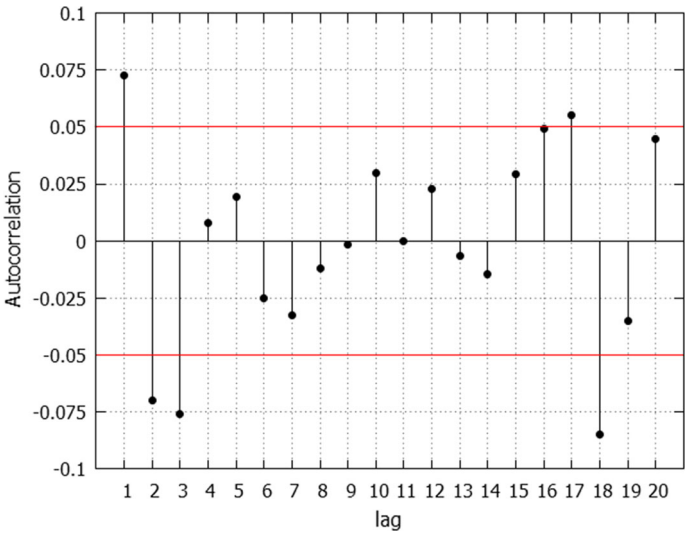

Fig. 2 20-day correlogram of the return series

Thus, based on the available data, the return series may be divided into periods of high and low volatility. The volatility of a value s_i of the series is given by comparing the standard deviation of the return series with its local standard deviation, defined as the standard deviation of the range of the twenty values prior to s_i. If the local standard deviation is greater than the standard deviation of the series, the value s_i is considered as having high volatility. Otherwise, it is considered to have low volatility. The usage of the standard deviation as a measurement to obtain volatility is appropriate, since it provides information about the historical dispersion of the values of the series in relation to the mean value of the values thereof (Yang et al. 2002).

Figure 3 compares the standard deviation of the return series with the local standard deviation. The values in which the solid line is above the dashed threshold are considered as having high volatility and are always associated with high variations of the price series. All other values have low volatility. In all, 22% of the return series values have high volatility and 78% have low volatility.

Return series values are arranged as patterns in a decision table, as shown in Table 1. The decision table consists of patterns that contain condition attributes. A decision attribute is associated to each pattern. The condition attributes are formed by the three past values—r_{t+f-1}, r_{t+f-2}, and r_{t+f-3}—correlated with the value to be forecast, and by the volatility of the value to be forecast. Volatility is a binary value, where the values 0 and 1 mean low and high volatility, respectively. The decision attribute is formed by the value to be forecast, i.e., future values r_{t+f} of the return time series.

The decision table contains 1868 patterns, which represent the 1889 series values in the period analyzed. The time series, as well as the period analyzed, are exactly the same ones used in Carpinteiro et al. (2012), so that the results obtained by the proposed model can be compared with those obtained by the HM proposed in the aforementioned paper.

5 Proposed model

The proposed model, illustrated in Fig. 4, consists of two stages. The first one consists of clusters containing the patterns, arranged by similarity, of the decision table described in

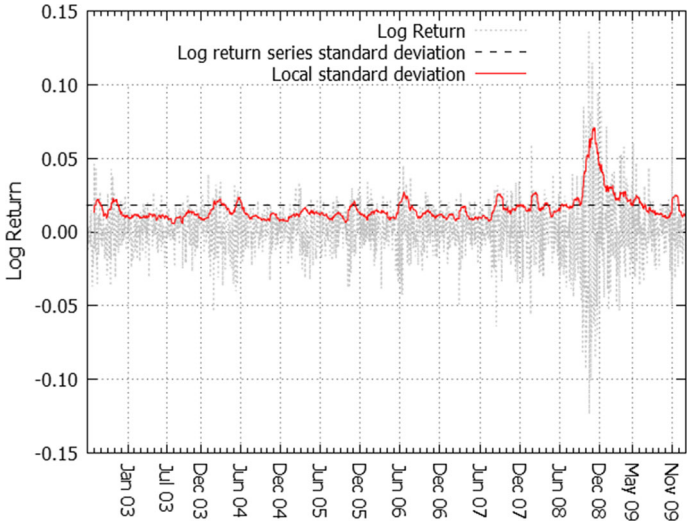

Fig. 3 Return series volatility

Table 1 Format of decision table patterns

Pattern (condition attributes)				Decision attribute
r_{t+f-3}	r_{t+f-2}	r_{t+f-1}	Vol	r_{t+f}

the previous section (Sect. 4). The second stage consists of SVRs that model the time series behavior in each cluster.

5.1 Training

The training process of the model consists of two phases—segmentation of the patterns into clusters and modeling the time series behavior in each cluster.

The first training phase—segmentation of the patterns into clusters—comprises, obviously, the creation of clusters. Clusters are formed through the execution of either the K-Means method or the Fuzzy C-Means method on the 1868 patterns $p_i \in \Re^4$ of the decision table. Thus, the two methods are responsible for segmenting the patterns—which are, in turn, formed by the values of the return series—into clusters. The assignment of patterns to clusters is given by the similarity between the patterns. The similarity measure used by both methods is the Euclidean distance. Thus, the clustering methods group in a same cluster patterns which are closer (i.e., more similar) to each other.

The next training phase—modeling the time series behavior in each cluster—involves, for each cluster created in the previous phase, the training of its corresponding SVR on the patterns inside the cluster. After trained, the SVRs are responsible for forecasting the future values of the return series. Patterns from any cluster j, which are dissimilar to patterns from cluster i, do not interfere in the training and, consequently, in the knowledge of the SVR of cluster i. As each SVR is trained only on the patterns of a single cluster, it becomes a specialist in forecasting values referring to patterns belonging to this cluster. For training the SVRs, obviously, the input patterns are associated with their respective decision attributes.

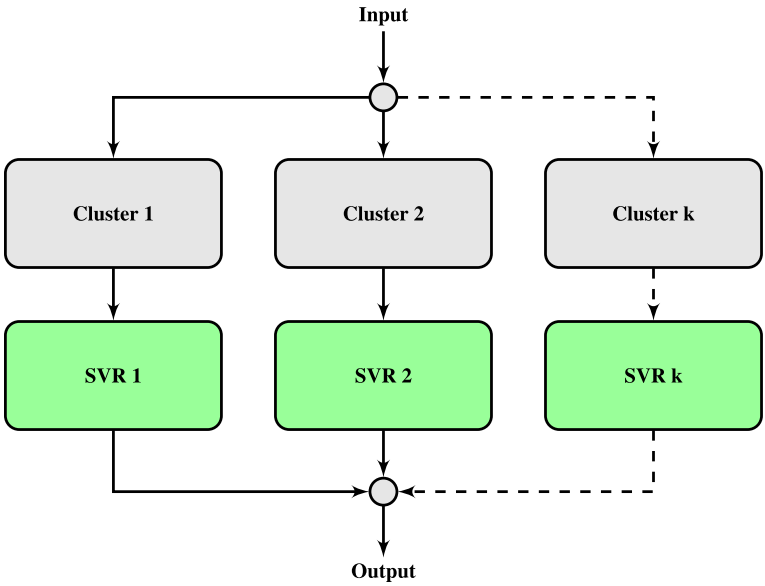

Fig. 4 Predictive model

5.2 Predicting

Each input pattern of the proposed model has the format of the decision table patterns. The output of the model is the forecast of the next value of the return series, corresponding to the input pattern.

When an input pattern is presented to the model, its first stage determines the membership of the pattern in relation to the k clusters of the model. Next, the SVRs in the second stage perform the regression, using the knowledge previously acquired in their respective training, to predict the next value of the return series. Two approaches can be followed in this second stage:

1. A single SVR, relating to the cluster to which the input pattern has the highest membership, is responsible for the forecast. This approach can be used either by hard clustering methods, given that the pattern's degree of membership, in this case, is always total to a single cluster and zero to all others, as well as by soft clustering methods, given that only the cluster to which the input pattern has highest degree of membership is chosen. The SVR selected in this approach is thus the most specialized one to predict the future value, for it has been trained exclusively on the patterns more similar to the input pattern;
2. All SVRs are responsible for the forecast, each one yielding its own forecast value. The degrees of membership of the input pattern relating to the clusters are used as weights to generate, as the final forecast, the weighted composition of the individual forecasts of each SVR. Only soft clustering methods can be used in this approach. Due to the usage of degrees of membership, the SVRs trained on patterns more similar to the input pattern have greater influence on the final predicted value than those trained on patterns dissimilar to the input pattern.

The second approach follows a principle similar to that of the HM proposed by Carpinteiro et al. (2012), where all contexts are used to produce the forecast value.

Table 2 Time periods used in the experiments and their characteristics

Experiment	Time period		Volatility	Trend
	Start	End		
1	2004-11-22	2004-12-17	Low	Uptrend
2	2007-05-28	2007-06-25	Low	Uptrend
3	2009-09-11	2009-10-08	Low	Uptrend
4	2006-05-19	2006-06-16	High	Correction
5	2008-01-18	2008-02-18	High	Horizontal
6	2009-10-14	2009-11-11	High	Horizontal

6 Experiments

Six experiments were performed. Each experiment is aimed at forecasting a period of time, comprising twenty consecutive days, of the return series. Each period of twenty consecutive days corresponds approximately to one-month trading, considering the fact that the stock market is closed during weekends and holidays.

The time periods are described in Table 2. The first three experiments include periods of low volatility, whereas the last three include periods of high volatility.

The six time periods belong to different contexts of the time series. The first three, are embedded in a clear uptrend movement. The fourth, belongs to a correction (downward) movement after a previous uptrend movement. The fifth period belongs to a horizontal movement occurred inside an uptrend. Finally, the sixth period belongs to a long-term horizontal movement.

These time periods are identical to the six periods used in Carpinteiro et al. (2012), so that the results obtained by the proposed model can be compared with those obtained by the HM and by other models described in that work.

In each experiment, obviously, the values of the time period to be forecast were removed from the return series both for forming the clusters and for training the SVRs. The forecast is made one step ahead. The forecast value is placed into a new input pattern, which is presented to the model, thereby generating a new forecast value. This process is performed successively until the values of all twenty days in the period are forecast.

Both clustering methods—K-Means and Fuzzy C-Means—were used in each experiment. With the Fuzzy C-Means method, the second stage of the model used both of the approaches described in the previous section (Sect. 5). With the K-Means method, only the single SVR approach was used, for it is a hard clustering method. The SVRs used the radial basis function as the kernel function, since this produced the best results reported by Carpinteiro et al. (Carpinteiro et al. 2012).

For implementation of the SVRs, the LIBSVM library (Chang and Lin 2011) was used. The accuracy of the forecasts of the proposed model was compared to that of the HM, SVM, MLP, and carbon copy (CRB) models, through the MAPE and its standard deviation as well as through the Theil's U statistical measurement and the HLN test. The HM, SVM and MLP models are described in Carpinteiro et al. (2012). CRB model is a no-change naïve model, that predicts the next time series value to be the same as the preceding value.

Table 3 Error values (MAPE) in each experiment

Experiment	K-Means	Hard fuzzy C-Means	Soft fuzzy C-Means	HM	SVM	MLP	CRB
1	0.67	1.14	2.26	0.99	2.30	2.93	4.30
2	0.87	0.94	0.97	1.92	1.29	1.76	3.21
3	0.89	0.89	2.71	2.57	2.51	4.12	4.04
Mean (low volatility)	0.81	0.99	1.98	1.83	2.03	2.94	3.85
4	2.72	2.87	6.92	5.02	8.13	9.11	5.76
5	1.96	2.04	2.33	2.15	2.63	3.56	4.11
6	2.19	2.15	2.13	1.93	2.62	2.37	2.20
Mean (high volatility)	2.29	2.35	3.79	3.03	4.46	5.01	4.02
Total mean	1.55	1.67	2.88	2.43	3.24	3.97	3.93
Standard deviation	0.85	0.80	2.06	1.37	2.44	2.65	1.19

7 Results

7.1 Error analysis

The MAPE of each experiment, obtained by the proposed model and by the HM, SVM, MLP [described in Carpinteiro et al. (2012)], and CRB, is shown in Table 3.

The first three columns comprises the experiments with the proposed model. The experiments described in the first and second columns made use, respectively, of the K-Means and Hard Fuzzy C-Means methods in the first stage, and of the single SVR approach in the second stage. The experiments described in the third column made use of the Soft Fuzzy C-Means method in the first stage, and of the all-SVR approach in the second stage. In all three cases, the SVR parameters $C = 2550$ and $\varepsilon = 0.0401$ were used, for they produced the best results. The proposed model with K-Means clustering used 12 clusters, and with Fuzzy C-Means, 14 clusters. Both values are the ones that produced the best results.

The HM, SVM, and MLP are described in Carpinteiro et al. (2012). The HM is composed of a SOM with map size of 50×50 in its first stage and a SVR with radial basis function and with parameters $C = 1000$ and $\varepsilon = 0.005$ in its second stage. The SVM model uses the radial basis function with parameters $C = 550$ and $\varepsilon = 0.0061$. The MLP model has a single hidden layer with 30 neurons. At last, the CRB model is a no-change naïve model.

One can see that the total means (i.e., the averages of the MAPEs) of 1.55 and 1.67, obtained by the proposed model with the single SVR approach (first two columns), are lower than those obtained by the HM. Analyzing the experiments individually, the proposed model with K-Means method in its first stage showed better accuracy than the HM in five of the six experiments. With Hard Fuzzy C-Means method, in four of the six.

Translated into numbers, the proposed model with K-means and Hard Fuzzy C-Means methods presented errors 35% and 31% lower than those of the HM, respectively. In turn, when comparing the clustering methods of the proposed model, one can see that the K-Means method showed an error 7% lower than that of the Hard Fuzzy C-Means method.

The proposed model with Soft Fuzzy C-Means method was not capable of achieving better results than the HM. The HM presented an average error 15% better than that presented by the proposed model with Soft Fuzzy C-Means. Both models are similar, from a functional

viewpoint, for they make use of the knowledge of all contexts of the series to produce the forecast value.

The three total means (i.e., the three averages of the MAPEs) obtained by the proposed model are even better than those obtained by the single stage—SVM, MLP and CRB—models. In the extreme case, the proposed model with K-Means method is 61% better than the MLP model.

The proposed model with the single SVR approach (first two columns) achieved standard deviation values of the MAPE lower than those achieved both by itself with the all-SVR approach (third column) and by all other models. This means that the MAPEs of the proposed model are more stable and, consequently, that its forecast values are more reliable. For example, the difference between the highest and the lowest MAPEs presented by the HM is 4.03. In turn, the difference between the highest and lowest MAPEs presented by the model with the K-means method is only 2.05.

Finally, the table indicates that all models produced higher MAPE values during periods of high volatility (experiments 4, 5 and 6) than those produced during periods of low volatility (experiments 1, 2 and 3). This result was expected, since periods of high volatility include abrupt changes in values of the series, making forecasting more difficult. However, once again, the proposed model with the single SVR approach showed mean MAPE values lower than those showed by the HM, not only during periods of low volatility but also during periods of high volatility.

7.2 Graphical analysis

The graphical analysis aims to compare the models that produced the best results. Hence, SVM, MLP and CRB models are not considered, for they produced poorer results when compared to those of the other models.

Figure 5 graphically shows the results of the experiments 1 to 6, as presented in Table 3, as well as the values of the actual price series in the time periods covered by the experiments.

Experiments 1 to 3 are aimed at forecasting values ranging over a period of time where the series has low volatility behavior. In the first three graphics of the figure, corresponding to the first three experiments, one can see the values forecast by the proposed model, with the hard-clustering methods (K-Means and Hard Fuzzy C-Means), followed the trend of the values of the actual price series during the 20-day periods with considerable accuracy. In the first experiment, even the price movement was caught by the proposed model with K-Means method. In all three experiments, the accuracy of the proposed model was better than that of the HM. It is worth noticing that the proposed model with K-Means and with Hard Fuzzy C-Means methods produced an equal result in the experiment 3. The reason for this is that both methods produced exactly the same clusters. With the Soft Fuzzy C-Means method, the model also followed the price trend, but with less intensity and without following the price movements.

In turn, experiments 4 to 6 are aimed at forecasting values ranging over a period of time where the series shows high volatility behavior. In the last three graphics of the figure, corresponding to the last three experiments, one can see that the proposed model, with the hard-clustering methods (K-Means and Hard Fuzzy C-Means), followed the trend of the values of the actual price series in two of the experiments. It presented accuracy far lower than that presented in the low volatility experiments. All other models obtained even worse performance. This may be justified, at least in part, by the intrinsic difficulty of forecasting financial time series values in periods of high volatility.

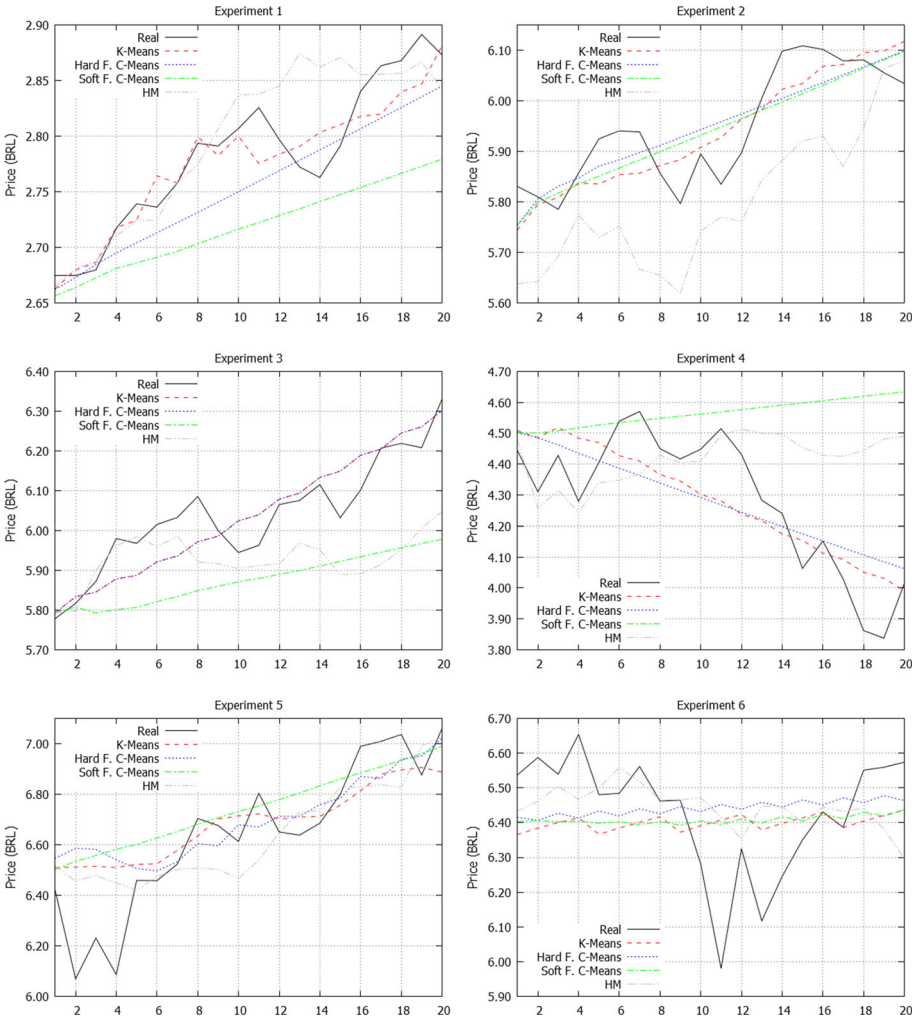

Fig. 5 Experiments 1 to 6—graphical analysis

7.3 Statistical analysis

The statistical analysis aims to provide new insights from the results. It makes use of the Theil's U statistic to compare the CRB—a no-change naïve model—with the proposed model and with the HM. Only these two models are considered because they produced the best results.

Table 4 shows the RMSE obtained in each experiment by the mentioned models. Table 5 shows the Theil's U statistic value calculated from the RMSE values.

The two models produced Theil's U values smaller than 1, indicating that they are able to produce more accurate forecast values than that produced by the CRB model. It is worth noticing that the values produced by the proposed model with K-Means and with Hard Fuzzy

Table 4 Error values (RMSE) in each experiment

Experiment	K-Means	Hard Fuzzy C-Means	Soft Fuzzy C-Means	HM	CRB
1	0.024	0.037	0.072	0.041	0.137
2	0.061	0.066	0.066	0.165	0.223
3	0.066	0.066	0.186	0.149	0.279
4	0.132	0.145	0.375	0.282	0.314
5	0.174	0.190	0.196	0.179	0.328
6	0.168	0.174	0.164	0.169	0.169
Mean	0.104	0.113	0.176	0.164	0.242

Table 5 Theil's U for each model

K-Means	Hard Fuzzy C-Means	Soft Fuzzy C-Means	HM
0.431	0.467	0.730	0.679

C-Means methods were considerably smaller than 1, corroborating the smaller error values obtained by the model in the experiments.

Similarly, the predictive accuracy is also analyzed by the HLN test. HLN test analyzes the significant differences between two models in predictions over 20-step-ahead horizons. Table 6 shows the analysis.

Except for the comparison between the models with K-Means and Hard Fuzzy C-Means methods, one can observe from Table 6 that the HLN tests show evidence that all models are statistically different, since all p-values tend to zero. Therefore, one can reject the null hypothesis that the statistical difference between two models is not significant under HLN tests.

Conversely, the HLN tests show evidence that the models with K-Means and Hard Fuzzy C-Means methods are not statistically different, since their p-values do not tend to zero. This is due to the fact, as mentioned before, that the two models yield the same predictions over 20-step-ahead horizons in the experiment 3 as a consequence of the fact that both methods produced exactly the same clusters.

7.4 Usage of clusters

The number of clusters formed by both the K-Means and Fuzzy C-Means methods is defined by the user. The numbers used in the experiments were the ones that produced the best results. To find them, several tests using different numbers of clusters were conducted. It is worth mentioning that even for a given number of clusters, different sets of cluster, comprising different patterns, may be generated when executing the methods several times. Thus, the sets chosen were those that produced the best results in the experiments. On the other hand, the division of the patterns of the time series into low and high volatility clusters and the number of clusters of each volatility is defined by the methods themselves.

The input data for the clustering methods consists of the 1868 patterns $p_i \in \Re^4$ of the decision table (Table 1). As shown in the table, each pattern p_i includes the volatility information in its fourth coordinate of $\Re^4$. The volatility information is given by a binary value. The low and high volatilities are represented by the values 0 and 1, respectively.

Table 6 HLN tests over 20-step-ahead horizons—numbers in parenthesis refer to the *p*-value of testing the hypothesis that the models yield equal predictions (i.e., the null hypothesis) against the alternative that they yield different predictions over over 20-step-ahead horizons

	Hard Fuzzy C-Means	Soft Fuzzy C-Means	HM		Hard Fuzzy C-Means	Soft Fuzzy C-Means	HM
Experiment 1				Experiment 2			
K-Means	− 3520.34	− 19, 565.62	− 4557.17	K-Means	− 2706.79	− 2796.22	− 1.94e + 09
	(0.000)	(0.000)	(0.000)		(0.000)	(0.000)	(0.000)
Hard Fuzzy C-Means	–	− 8.72e + 08	− 1036.83	Hard Fuzzy C-Means	–	− 89.43	− 1.88e + 09
	–	(0.000)	(0.000)		–	(0.000)	(0.000)
Soft Fuzzy C-Means	–	–	4.71e + 08	Soft Fuzzy C-Means	–	–	− 1.09e + 09
	–	–	(0.000)		–	–	(0.000)
Experiment 3				Experiment 4			
K-Means	0.00	− 1.23e + 09	− 3.13e + 08	K-Means	− 15, 073.22	− 3.97e + 08	− 4.47e + 08
	(1.000)	(0.000)	(0.000)		(0.000)	(0.000)	(0.000)
Hard Fuzzy C-Means	–	− 1.23e + 09	− 3.13e + 08	Hard Fuzzy C-Means	–	− 4.97e + 08	− 2.48e + 05
	–	(0.000)	(0.000)		–	(0.000)	(0.000)
Soft Fuzzy C-Means	–	–	51, 558.28	Soft Fuzzy C-Means	–	–	2.59e + 05
	–	–	(0.000)		–	–	(0.000)
Experiment 5				Experiment 6			
K-Means	− 1.59e + 08	− 3.42e + 08	− 9.73e + 07	K-Means	− 1.08e + 08	2.86e + 08	− 1118.79
	(0.000)	(0.000)	(0.000)		(0.000)	(0.000)	(0.000)
Hard Fuzzy C-Means	–	− 2.07e + 08	8.48e + 07	Hard Fuzzy C-Means	–	13, 244.77	6855.30
	–	(0.000)	(0.000)		–	(0.000)	(0.000)
Soft Fuzzy C-Means	–	–	2.73 + 08	Soft Fuzzy C-Means	–	–	− 8.68e + 07
	–	–	(0.000)		–	–	(0.000)

Table 7 Clusters created in each experiment

Experiment	Number of clusters			
	K-Means		Fuzzy C-Means	
	Low	High	Low	High
1	6	6	8	6
2	5	7	7	7
3	5	7	5	9
4	5	7	4	10
5	4	8	8	6
6	6	6	11	3

Table 8 Number of low and high volatility SVRs used in each experiment

Experiment	Number of clusters used			
	K-Means		Hard Fuzzy C-Means	
	Low	High	Low	High
1	5	0	2	0
2	5	0	6	0
3	2	0	2	0
4	3	0	1	0
5	4	0	6	0
6	4	0	4	0

To divide the input patterns into clusters, the methods make use of the Euclidean distances between the patterns in $\Re^4$. Hence, the clustering methods take into consideration the volatility to form the clusters. The clustering methods always group input patterns of periods of low and high volatility, respectively, into clusters of low and high volatility. This behavior of the methods is due to the fact that input patterns of periods of low volatility are more distant from those of periods of high volatility.

Table 7 shows the number of low and high volatility clusters formed by both clustering methods in each experiment.

The table shows that, unlike the Fuzzy C-means method, the K-means method always formed nearly the same quantities of low and high volatility clusters in all the experiments. When the model uses the K-means or Hard Fuzzy C-Means methods in its first stage, only one SVR, relating to the cluster whose centroid is closest to the input pattern, produces the forecast value. Thus, for each forecast made by the model, it can be determined whether it was produced by a low or high volatility SVR, i.e., by an SVR relating to a cluster of low or high volatility.

Table 8 shows the total number of low and high volatility SVRs used in each experiment to produce the forecast values of the 20-day period. In other words, if an SVR was responsible for the forecast on at least one day of the period, it is considered as being used in the experiment.

The decision whether to use low or high volatility SRVs during prediction depends on two factors. First, the pertinence of the input pattern to the clusters, given by the Euclidean distances between the pattern and each cluster centroid. As mentioned above, since the distance is calculated over $\Re^4$, the volatility of the input pattern ensures that the most pertinent cluster chosen is the one which shares the same volatility with the pattern. Second, the forecast values produced in sequence. Each forecast value is produced by using past values, which

includes previously forecast values. Therefore, the accumulation of errors in a sequence of predictions may cause large prediction errors, and consequently, may move the forecast values in direction of clusters of opposite volatility.

As can be seen in the Table 8, in all experiments and with both clustering methods, only low volatility SVRs were used. This behavior of the model is expected in the experiments that comprise periods of low volatility, but not for those that comprise periods of high volatility. Such behavior justifies, in part, the lower accuracy achieved by the model in the experiments 4, 5 and 6, comprising periods of high volatility. This behavior also shows that SVRs are unable to produce forecast values with magnitude required to make the local standard deviation higher than the standard deviation of the series. In other words, the accumulation of errors in the sequence of predictions caused large prediction errors, and consequently, moved the forecast values in direction of the low-volatility clusters.

7.5 Cluster choice analysis

In this section, it is analyzed each cluster choice—and, consequently, each SVR choice—made by the two hard methods (K-Means and Hard Fuzzy C-Means) of the model when it receives a pattern at its input. The analysis is made by comparing the accuracy of the forecast value of the SVR of the chosen cluster to the accuracy of the forecast values of the SVRs of the other clusters. In this section, the composition of the forecast values of the proposed model, with Soft Fuzzy C-Means method, is also analyzed, by comparing the accuracy of this composition with the accuracy of the forecast value of each SVR separately.

For this analysis, the six experiments described in Table 2 were conducted again. However, they were conducted without feedback of the predicted value, which means that the forecast always has a horizon of one step ahead. Thus, in each experiment, i.e., in each of the six periods of time, the forecasts of the SVRs relating to all clusters existing in the first stage of the model were obtained and analyzed on a horizon of one step ahead.

Figures 6 and 7 show the results of the analysis of forecast values provided by the SVR of each cluster of the model, with the K-Means method, in experiments 1 and 6, respectively. Each gray line represents the forecast values of the SVR of a cluster existing in the first stage of the model. Each red circle represents the forecast value of the SVR relating to the cluster selected by the model to provide the forecast.

Analyzing the values forecasted by SVRs on each day of the 20-day period, one can see that the SVRs are capable of producing values both close and distant from the value produced by SVR relating to the cluster selected by the model. This indicates that the K-Means method apportioned the knowledge about the series, by similarity, among the various clusters.

The model most often chose one of the SVRs most highly capable of performing the task of forecasting, i.e. the SVRs producing forecast values close to the values of the actual price series. However, on most of the days, this choice does not fall upon the SVR that produces the best value. This means that, each day, other SVRs may possibly produce forecast values better than the one of the SVR chosen by model. This finding paves the way for the development of improvements in the model.

Figures 8 and 9 present the results of the analysis of forecast values provided by the SVR of each cluster of the model proposed, with the Hard Fuzzy C-Means and Soft Fuzzy C-Means methods in experiments 1 and 6, respectively. Each red circle represents the forecast value of the SVR relating to the cluster selected by the model, with the Hard Fuzzy C-Means method, to provide the forecast. In turn, each blue square represents the forecast value composed of the values produced by all of the SVRs of the model, with the Soft Fuzzy C-Means method.

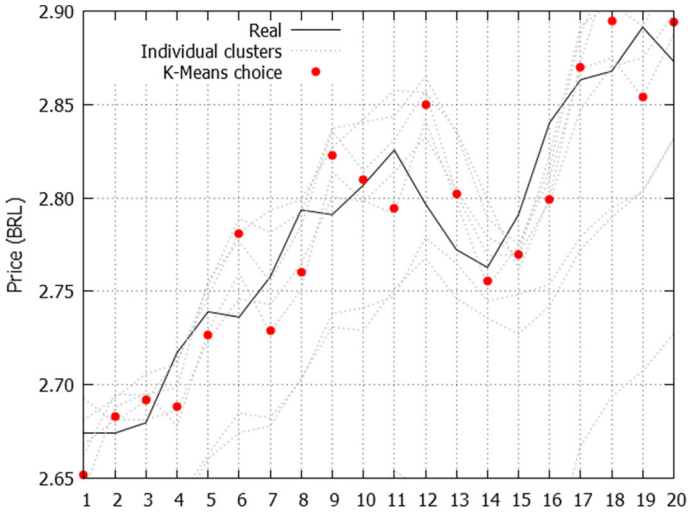

Fig. 6 Experiment 1—analysis of K-Means clusters

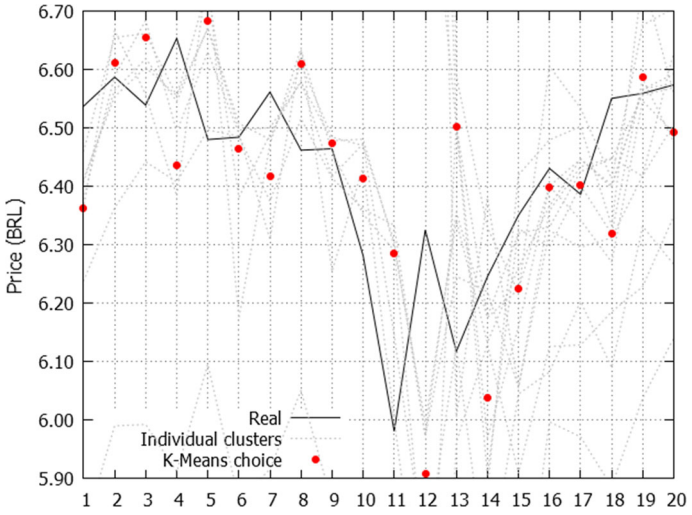

Fig. 7 Experiment 6—analysis of K-Means clusters

The results obtained from the analysis of the SVRs of the model with K-Means clusters are also valid for the SVRs of the model with Fuzzy C-Means clusters. The Fuzzy C-Means method also apportioned the knowledge about the series, by similarity, among the various clusters. Additionally, the model, with the Hard Fuzzy C-Means method, in most cases also selected one of SVRs more highly capable of performing the task of forecasting, although on most days, this choice does not fall upon the SVR that produces the best forecast value.

The analysis of the composition of forecast values made by the model, with the Soft Fuzzy C-Means method, indicates that there are several days on which the value from this composition is practically equal to the value produced by the model with the Hard Fuzzy C-Means method (days where only the blue squares are visible on the graph). This happens

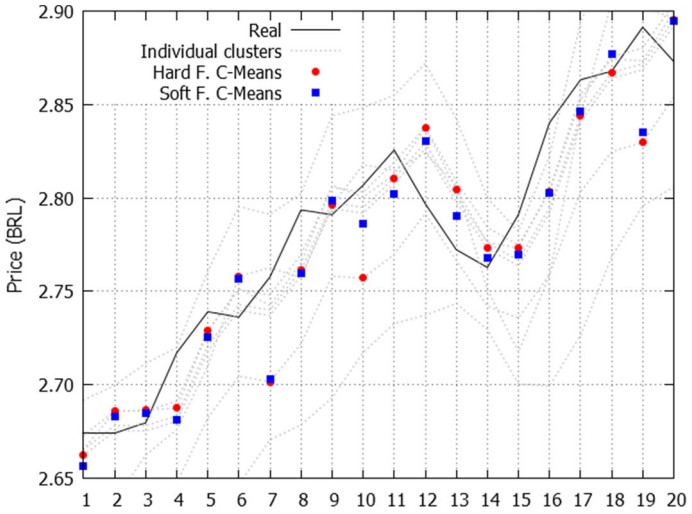

Fig. 8 Experiment 1—Fuzzy C-Means cluster analysis (hard and soft)

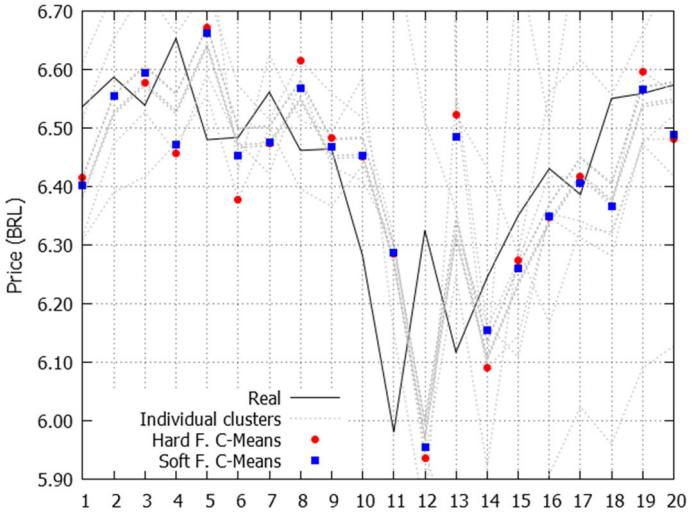

Fig. 9 Experiment 6—Fuzzy C-Means cluster analysis (hard and soft)

when the membership of the input pattern in the Hard Fuzzy C-Means cluster selected by the model is very high in relation to the degree of membership in the other clusters, making contribution of the SVRs of the other clusters practically zero in the composition of the forecast value.

7.6 Volatility analysis

As noted in Table 8 (Sect. 7.4), in all the experiments and with both clustering methods, only low volatility SVRs were used. Due to this behavior of the model, the six experiments were conducted again. This time, however, with two differences. First, only the patterns

Table 9 Error values (MAPE) in each experiment with high volatility patterns

Experiment	K-Means	Hard Fuzzy C-Means
1	0.75	0.69
2	0.94	0.93
3	0.80	0.82
Mean (low volatility)	0.83	0.81
4	2.96	2.44
5	1.99	1.99
6	1.93	2.08
Mean (high volatility)	2.29	2.17
Total mean	1.56	1.49
Standard deviation	0.88	0.76

relating the periods of high volatility of the series were used in training the model. Thus, only 22% of the lines of the decision table (Table 1), referring to periods of high volatility, were used by the clustering methods, for the formation of the clusters, and by the SVRs, in their training, and the remainder were disregarded. Second, information about volatility was removed from the patterns (Table 1), since the model no longer needs to distinguish among different volatilities. For this second set of experiments, the number of clusters, for both K-Means and Fuzzy C-Means methods, was set to 24, because, with this number of clusters, the model achieved the best results. The parameter values of the SVRs were the same used in the first set of experiments.

The aim of conducting this second set of experiments was to compare the knowledge on the general behavior of the series, existing in high volatility patterns, with that existing in the low volatility patterns, by comparing the forecast values produced by both sets of experiments. The results of this second set of experiments, called experiments with clusters of high volatility patterns, are shown in Table 9.

The comparison of the model's results in both sets of experiments (Tables 3 and 9) indicates that, with the K-means method, the average error stays practically identical. In turn, with the Hard Fuzzy C-means method, the model presented an average error 10% lower in the second set than the one shown in the first set.

Comparing the results in periods of low volatility, it can be seen that the model, with the K-means method, showed an average error 2% lower in the first set of experiments than the one shown in the second set. This behavior was expected, since only patterns of high volatility periods were used in the formation of clusters and the training of the model's SVRs. On the other hand, the model, with the Hard Fuzzy C-Means method, showed an average error 18% lower in the second set of experiments than the one shown in the first set. This result is surprisingly good, for it was the opposite of what was expected.

In high volatility periods, the model with the Hard Fuzzy C-Means method showed an average error 7% lower in the second set of experiments than that shown in the first set. This behavior was expected due to the nature of the training data. With the K-means method, the model presented an almost identical average error in both sets.

The standard deviations of the error presented by the model, with the K-means and Fuzzy C-Means methods, were similar in both sets of experiments.

These results indicate that the model, whether using only clusters of low volatility periods or using only clusters of high volatility periods, is able to produce good forecast values. In other words, patterns corresponding to periods of only one of the volatility types carry

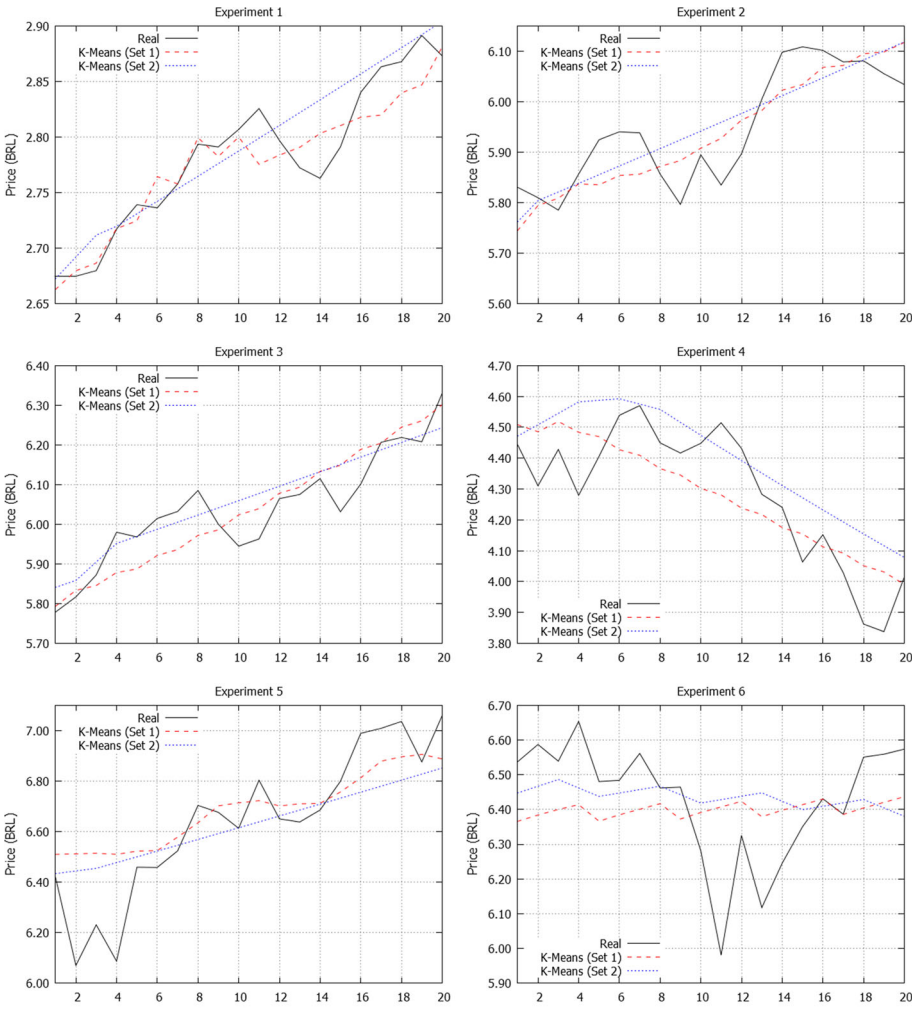

Fig. 10 Experiments 1 to 6 (sets 1 and 2)—K-Means method

sufficient information about the general behavior of the series. It is noteworthy that the model, when using only clusters of high volatility periods in the second set of experiments, used less than one fourth of the patterns of the series in forming its clusters and training its SVRs.

Figures 10 and 11 graphically show the model's forecast values, with the K-Means and Hard Fuzzy C-Means methods, in both sets of experiments (Tables 3 and 9), as well as the values of actual price series in the time period covered by the experiment.

The graphics of the figures show that the forecast values produced by the model follow with good accuracy the trend of the actual price series. The results in both sets indicate that the model is capable of accurately producing forecast values either by using only clusters of low volatility patterns (first set of experiments) or by exclusively using clusters of high volatility patterns (second set of experiments).

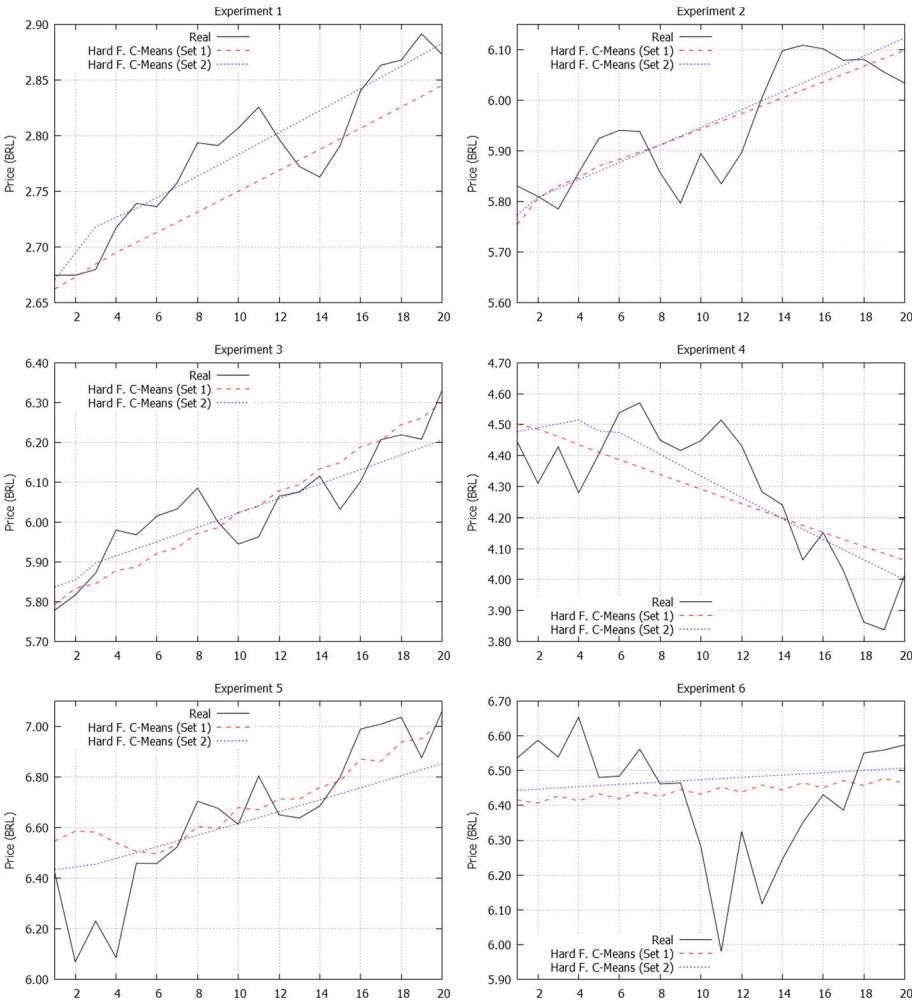

Fig. 11 Experiments 1 to 6 (sets 1 and 2)—Hard Fuzzy C-Means method

8 Conclusion

Correct prediction of values in financial series is difficult. Not only the law of supply and demand, but also facts, rumors, local and global economic conditions, and external factors, among others, directly or indirectly influence the price change of an asset. Thus, financial series present erratic moves when analyzed without adequate models.

Machine learning-based models are therefore used to capture the dynamics of the series, allowing its future values to be forecast. This paper proposes a two-stage model for forecasting financial time series. The first stage uses a clustering method—K-Means or Fuzzy C-Means—to segment the time series into its various contexts. The second stage makes use of support vector regressions (SVRs), one for each context, to forecast future values of the series.

The model is evaluated experimentally on a series composed of values of an equity fund of a Brazilian bank. The proposed model is also compared to an HM (Carpinteiro et al. 2012) presented in the literature.

The HM uses all the contexts of the series to produce the forecast values. The proposed model, with the Soft Fuzzy C-Means method, uses this same paradigm. Upon comparing the two, the HM has an error 15% lower than that of the proposed model in forecasting.

The proposed model, with either the K-Means or the Hard Fuzzy C-Means methods, uses another paradigm. In this one, the model uses only one of contexts of the series—the most relevant—to produce a forecast value. When compared to the HM, the model with the K-Means and Hard Fuzzy C-Means methods, respectively, had errors 35% and 31% lower than that of the HM in the forecasts. It also produced forecast values more accurate than those of the HM in most time periods analyzed, whether for periods of low or high volatility. Furthermore, the standard deviation of MAPE produced by the proposed model is less than the one produced by the HM.

The results achieved by the proposed model, with either K-Means or the Hard Fuzzy C-Means, are more accurate than those of the proposed model with Soft Fuzzy C-Means and of the HM. The results thus indicate that using only the patterns of the time series similar to the input pattern is better than using all patterns of the series to produce the forecast value.

Compared to the single stage models—SVM, MLP, and CRB—, the proposed model, with both clustering methods and with both hard and soft approaches, produced much more accurate results.

An analysis of the model's forecasts was conducted. In this analysis, the forecast value produced by each SVR existing in the model was compared to the forecast generated by the model, i.e. to the forecast produced by SVR relating to the cluster most pertinent to the input pattern. The analysis showed that there is always a group of SVRs capable of forecasting future values close to the actual value and that the SVR selected by the model to produce the forecast value almost always falls upon one SVR of this group. On the other hand, the forecast generated by the model is the best among the forecasts produced by SVRs of the group only some of the times, so there is room for improvement.

Another analysis of the construction and usage of clusters based on periods of volatility in the series was conducted. This analysis showed that patterns from only a single type of volatility (low or high) are able to provide sufficient knowledge to the model so that it is able to forecast future values with good accuracy. Considering accuracy only, there is no considerable distinction in using one volatility or the other. However, the use of high volatility training data only may be more interesting, for there are less high volatilities entries in the time series, rendering the training process faster.

Besides the good results achieved, another advantage of the proposed model over the HM is that clustering data is less complex than building SOMs. The process of creating and training SOMs is slow and cumbersome, because it includes several different free parameters. On the other hand, clustering data is fast and has only one free parameter, which is the number of clusters.

However, the proposed model lacks a technique to find, for a given number n of clusters, the optimal set of n clusters. The clustering process can form several different sets of n clusters that satisfy the method criteria, and finding the optimal set is still an open point.

From the analysis of the results, it is possible to suggest four directions for future work. First, evaluate other metrics of distance between patterns. Other metrics might better correlate the patterns by changing the formation of clusters to lead the model to produce more accurate forecasts. Second, evaluate techniques that, based on the series, are able to identify the best values for not only the free parameters of the SVRs but also the number of clusters. Third,

evaluate heuristics and techniques to find, for a given number n of clusters, the optimal set of n clusters during the training of the model. Finally, execute trading simulations with the proposed model, to verify if the prediction values produced are translated into profitable market operations.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments.

References

Abu-Mostafa YS, Atiya AF (1996) Introduction to financial forecasting. Appl Intell 6(3):205–213
Anderson DR, Sweeney DJ, Williams TA, Camm JD, Cochran JJ, Fry MJ, Ohlmann JW (2016) An introduction to management science: quantitative approaches to decision making. Int J Forecast 8(1):69–80
Armano G, Marchesi M, Murru A (2005) A hybrid genetic-neural architecture for stock indexes forecasting. Inf Sci 170(1):3–33 (computational Intelligence in Economics and Finance)
Armstrong J, Collopy F (1992) Error measures for generalizing about forecasting methods: empirical comparisons. Int J Forecast 8(1):69–80
Atsalakis GS, Valavanis KP (2009) Surveying stock market forecasting techniques—part II: soft computing methods. Expert Syst Appl 36(3, Part 2):5932–5941. https://doi.org/10.1016/j.eswa.2008.07.006
Azad MK, Uddin S, Takruri M (2018) Support vector regression based electricity peak load forecasting. In: 2018 11th international symposium on mechatronics and its applications (ISMA), pp 1–5. https://doi.org/10.1109/ISMA.2018.8330143
Bezdek JC, Ehrlich R, Full W (1984) FCM: the Fuzzy C-Means clustering algorithm. Comput Geosci 10(2–3):191–203
Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the ACM annual workshop on computational learning theory (COLT), pp 144–152
BOVESPA (2017) http://www.bmfbovespa.com.br/pt_br/produtos/indices/indices-amplos/indice-brasil-100-ibrx-100-1.htm
Box GEP, Jenkins GM, Reinsel GC (2008) Time series analysis, forecasting and control. Wiley, Hoboken
Bank of Brazil (2010) http://www37.bb.com.br/portalbb/fundosInvestimento/fundosinvestimento/gf07,802,10340,10340,1,0.bbx?fundo=6
Cao L (2003) Support vector machines experts for time series forecasting. Neurocomputing 51:321–339
Cao L, Tay FEH (2001a) Application of support vector machines in financial time series forecasting. Omega 29(4):309–317
Cao L, Tay FEH (2001b) Improved financial time series forecasting by combining support vector machines with self-organizing feature map. Intell Data Anal 5(4):339–354
Cao L, Tay FEH (2003) Support vector machine with adaptive parameters in financial time series forecasting. IEEE Trans Neural Netw 14(6):1506–1518
Carpinteiro OAS, Leite JPRR, Pinheiro CAM, Lima I (2012) Forecasting models for prediction in time series. Artif Intell Rev 38(2):163–171
Chabaa S, Zeroual A, Antari J (2010) Identification and prediction of internet traffic using artificial neural networks. J Intell Learn Syst Appl 2(3):147–155
Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):27
Clements MP, Franses PH, Swanson NR (2004) Forecasting economic and financial time-series with non-linear models. Int J Forecast 20(2):169–183
Diebold FX, Mariano RS (2002) Comparing predictive accuracy. J Bus Econ Stat 20(1):134–144. https://doi.org/10.1198/073500102753410444
Everitt BS, Landau S, Leese M, Stahl D (2011) Cluster analysis. Wiley, Hoboken
Harvey DI, Leybourne SJ, Newbold P (1998) Tests for forecast encompassing. J Bus Econ Stat 16(2):254–259
Haviluddin, Alfred R (2015) A genetic-based backpropagation neural network for forecasting in time-series data. In: Proceedings of the international conference on science in information technology (ICSITech), pp 158–163
Haykin S (2009) Neural networks and learning machines. Prentice Hall, Upper Saddle River
Huang W, Nakamori Y, Wang SY (2005) Forecasting stock market movement direction with support vector machine. Comput Oper Res 32(10):2513–2522

Kara Y, Boyacioglu MA, Baykan ÖK (2011) Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the istanbul stock exchange. Expert Syst Appl 38(5):5311–5319. https://doi.org/10.1016/j.eswa.2010.10.027
Kim Y, Enke D (2016) Developing a rule change trading system for the futures market using rough set analysis. Expert Syst Appl 59:165–173. https://doi.org/10.1016/j.eswa.2016.04.031
Kutics A, O'Connell C, Nakagawa A (2013) Segment-based image classification using layered-SOM. In: Proceedings of the IEEE international conference on image processing, pp 2430–2434
Limei L, Xuan H (2017) Study of electricity load forecasting based on multiple kernels learning and weighted support vector regression machine. In: 2017 29th Chinese control and decision conference (CCDC), pp 1421–1424. https://doi.org/10.1109/CCDC.2017.7978740
Lin Q, Wang Q, Zhang G, Shi Y, Liu H, Deng L (2018) Maximum daily load forecasting based on support vector regression considering accumulated temperature effect. In: 2018 Chinese control and decision conference (CCDC), pp 5199–5203. https://doi.org/10.1109/CCDC.2018.8408035
Liu D, Chen Q, Mori K (2015) Time series forecasting method of building energy consumption using support vector regression. In: 2015 IEEE international conference on information and automation, pp 1628–1632. https://doi.org/10.1109/ICInfA.2015.7279546
MacQueen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the Berkeley symposium on mathematical statistics and probability
Makridakis S, Whellwright SC, Hyndman RJ (1998) Forecasting: methods and appplications. Wiley, Hoboken
Mercer J (1909) Functions of positive and negative type, and their connection with the theory of integral equations. Philos Trans R Soc Lond A: Math Phys Eng Sci 209(441–458):415–446
Oliveira JV, Pedrycz W (2007) Advances in fuzzy clustering and its applications. Wiley, Hoboken
Patel J, Shah S, Thakkar P, Kotecha K (2015) Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Syst Appl 42(1):259–268. https://doi.org/10.1016/j.eswa.2014.07.040
Perwej Y, Perwej A (2012) Prediction of the Bombay Stock Exchange (BSE) market returns using artificial neural network and genetic algorithm. J Intell Learn Syst Appl 4(2):108–119
Popovici R, Andonie R (2015) Music genre classification with self-organizing maps and edit distance. In: Proceedings of the international joint conference on neural networks (IJCNN), pp 1–7
Rosowsky YI, Smith RE (2013) Rejection based support vector machines for financial time series forecasting. In: Proceedings of the international joint conference on neural networks (IJCNN), pp 1–7
Sapankevych NI, Sankar R (2009) Time series prediction using support vector machines: a survey. IEEE Comput Intell Mag 4(2):24–38
Shen W, Xing M (2009) Stock index forecast with back propagation neural network optimized by genetic algorithm. In: Proceedings of the international conference on information and computing science, vol 2, pp 376–379
Singh S, Bhambri P, Gill J (2011) Time series based temperature prediction using back propagation with genetic algorithm technique. Int J Comput Sci Issues 8(5):28
Small GR, Wong R (2002) The validity of forecasting. In: Pacific rim real estate society international conference Christchurch
Tong H (2002) Nonlinear time series analysis since 1990: some personal reflections. Acta Math Appl Sin 18(2):177–184
Tsay R (2010) Analysis of financial time series, Wiley series in probability and statistics, 3rd edn. Wiley-Interscience, Hoboken
Van Gestel T, Suykens JAK, Baestaens DE, Lambrechts A, Lanckriet G, Vandaele B, De Moor B, Vandewalle J (2001) Financial time series prediction using least squares support vector machines within the evidence framework. IEEE Trans Neural Netw 12(4):809–821
Vapnik VN (1998) Statistical learning theory. Wiley, Hoboken
Vapnik VN (1999) An overview of statistical learning theory. IEEE Trans Neural Netw 10(5):988–999
Wang JZ, Wang JJ, Zhang ZG, Guo SP (2011) Forecasting stock indices with back propagation neural network. Expert Syst Appl 38(11):14346–14355
Yang H, Chan L, King I (2002) Support vector machine regression for volatile stock market prediction. In: Proceedings of the international conference on intelligent data engineering and automated learning (IDEAL), pp 391–396
Yang H, Huang K, Chan L, King I, Lyu MR (2004) Outliers treatment in support vector regression for financial time series prediction. In: Proceedings of the international conference on neural information processing (ICONIP), pp 1260–1265
Yizhen L, Wenhua Z, Ling L, Jun W, Gang L (2011) The forecasting of Shanghai index trend based on genetic algorithm and back propagation artificial neural network algorithm. In: Proceedings of the international conference on computer science education (ICCSE), pp 420–424

Yu L, Dai W, Tang L, Wu J (2015) A hybrid grid-GA-based LSSVR learning paradigm for crude oil price forecasting

Yu L, Xu H, Tang L (2016) LSSVR ensemble learning with uncertain parameters for crude oil price forecasting

Yu L, Zhang X, Wang S (2017) Assessing potentiality of support vector machine method in crude oil price forecasting. Eurasia J Math Sci Technol Educ 13(12):7893–7904. https://doi.org/10.12973/ejmste/77926

Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353

Zhao C, Yu Z (2017) The research on forecasting model based on support vector machine and discrete grey system. In: 2017 international conference on computing intelligence and information system (CIIS), pp 104–107. https://doi.org/10.1109/CIIS.2017.24

Zhong X, Enke D (2017) Forecasting daily stock market return using dimensionality reduction. Expert Syst Appl 67:126–139. https://doi.org/10.1016/j.eswa.2016.09.027