ORIGINAL RESEARCH

Prediction of exchange rates using averaging intrinsic mode function and multiclass support vector regression

Bhusana Premanode[1], **Jumlong Vonprasert**[2], **Christofer Toumazou**[1]

1. Centre Bio-Inspired Technology, Imperial College London, South Kensington, London, UK. 2. Ubon Rachathani Rajabhat University, Ubon Ratchathani Province, Thailand

Correspondence: Bhusana Premanode. Address: Centre Bio-Inspired Technology, Imperial College London, South Kensington, London, SW7 2AT, UK. Email: Bhusana@gmail.com.

Received: October 30, 2012 **Accepted:** November 18, 2012 **Online Published:** February 18, 2013
DOI: 10.5430/air.v2n2p47 **URL:** http://dx.doi.org/10.5430/air.v2n2p47

Abstract

Prediction of nonlinear and nonstationary time series datasets can be achieved by using support vector regression. To improve the accuracy, we propose a new model ‘averaging intrinsic mode function’ which is a derivative of empirical mode decomposition to filter datasets of an exchange rate, followed by using a new algorithm of multiclass Support Vector Regression (SVR) for prediction. Simulation results show that the proposed model significantly improves prediction yields of the exchange rates, compared to simulation of SVR model without filtering and multiclass.

Key words

Prediction, Exchange rates, Empirical mode decomposition, Mean reversion, Multiclass, Support vector regression

1 Introduction

In terms of the classification and prediction of nonlinear and nonstationary time series data such as exchange rates, few models are migrated from different origins, including i) the Empirical Mode Decomposition (EMD) de-noising model, which emerges from the signal processing area and ii) the Support Vector Machine (SVM) model, which originates from the statistical learning theory. The following sections provide a background on EMD and Support Vector Regression (SVR).

1.1 Background on the EMD algorithm

The analysis of nonlinear and nonstationary data becomes relatively important in many applications such as bioinformatics [1, 2], signal processing [3, 4] geophysics [5, 6]; and finance [7, 8]. In 1996, Norden Huang et al, cited by Huang and Okine [4], invented a posteriori algorithm with adaptive control over a separate data structure. The invention was later called the Hilbert-Huang transform (HHT) [3]. This new transform enhances the limitation of Hilbert transform, which is only suitable for a narrow band-passed signal. The key part of the HHT algorithm is the EMD, in which any complicated dataset can be decomposed into a finite and often small number of Intrinsic Mode Functions (IMF) that admit a well-behaved Hilbert transform. Since the decomposition is based on the local characteristic time scale of the data, it is applicable to nonlinear and nonstationary processes [4]. In signal processing, high frequency noise from input data may be considered as different simple intrinsic mode oscillations [3].

The EMD uses a sifting process and curve spline technique to decompose a signal into a new oscillatory signal, intrinsic mode function (IMF). After a number of iterations of decomposition, the characteristics of the IMF meet the two conditions: i) in the entire dataset, the number of extrema (maxima plus minima) and the number of zero crossings must either be equal to or differ by at most one, and ii) at any point, the mean value of the envelope defined by the local maxima and the envelope defined by the local minima are zero [3]. The EMD algorithm, which is fundamental to the HHT, can reduce a high frequency noise from input data such as noise from retail trades in the stock exchange.

1.2 Background on the multiclass SVR model

In 1936, Fisher [9] invented the first algorithm for pattern recognition. A few decades later, Vapnik and Lerner [10] introduced the Generalized Portrait algorithm which has been the template for Support Vector Machine (SVM). Later in 1973, Duda, Richard, and Hart [11] discussed large margin hyperplanes in the input space. At the Conference on Learning Theory (COLT) in 1992, Boser, Guyon and Vapnik [12] introduced SVMs by combining the Generalized Portrait algorithm comprising a large margin hyperplane and kernel functions. Nowadays, SVM has empirically shown a better performance than other machine learning methods, including artificial neural networks (ANN). This is because of the employment of structural risk minimisation, in which they can avoid multiple local minima. Moreover, the computational complexity of SVM does not depend on the dimensionality of the input space. This is what makes an SVM so flexible when selecting large controlling parameters. Therefore, the SVM model is suitable for handling datasets that are complex in dimensionality such as exchange rates [13].

In the regression analysis, the SVM model was first compared against the benchmark of time series prediction tests with the Boston housing problem, ANN, and (using artificial data) the PET operator inversion problem [14]. Technically, SVMs consist of a set of related supervised learning methods. The algorithm sets a hyperplane characterising a functional margin that holds all possible data points situated in a finite dimensional nonlinear space. A kernel function k(x, x′), defines the cross-products that have been separated by the hyperplane. Each data point demonstrates its vector potential depending on its distance from the hyperplane. In the peak computational mode, the SVM model may have difficulty in finding the best-fit hyperplane. As a result it, can present overflow memory errors [14-16].

In the current methodologies in adopting SVMs to classification problems, Abe [17] reported that there are three main approaches, namely; i) multiclass ranking SVMs – one SVM decision function attempts to classify all classes, ii) one-against-all classification, in which is introduced by Vapnik [18] – there is one binary SVM for each class, and iii) pairwise classification – there is one binary SVM for each pair of class.

1.3 Background on the mean reversion

There are many definitions of mean reversion; in general, it is an asset model that shows that the asset prices tend to fall (rise) after hitting a maximum (minimum). The process of mean reversion is a log-normal diffusion, but with the variance not growing proportionally to the time interval. The variance grows in the beginning, and after sometime, stabilizes at a certain value. The most basic mean-reversion model is the arithmetic) [19], in which it is a stochastic process describing the velocity of a massive Brownian particle under the influence of friction. The process is, however, stationary, Gaussian, and Markovian. In another theme, it is an autoregression process in which the value drifts to its mean in the long run.

Currently, there are two outstanding methodologies used for measuring mean reversion, namely; i) variance ratio and ii) regression. Cochrane [20] used the variance ratio to measure the relative importance of the random walk component. Poterba and Summers [21] and Lo and Mackinlay [22] compared the relative variability of returns over different time horizons with the variance ratio in the discrete time series. Bali and Demirtas [23] confirmed that those reports showed that, in the ‘high variance’ scenario, mean reversion causes a negative drift, and vice versa. With regard to using regression tests, Fama and French [24] determined a correlation in currency asset returns, and Chen and Jeon [25] measured its mean reversion behaviour and found that the returns were positively autocorrelated over shorter periods but negatively autocorrelated over longer periods.

1.4 Objectives

We propose a new class of prediction model combining with an algorithm averaging intrinsic mode function, termed '*aIMF*', and a new multiclass SVR model using 'mean reversion and CV' technique which is explained in Section 3.3. The *aIMF* serves to filter datasets of an exchange rate which characterised as nonlinear nonstationary time series. Design of the *aIMF* filter is a new algorithm by Premanode and Toumazou [26] using derivation of EMD in conjunction with a new algorithm using coefficient of variance (CV) to achieve multi kernel parameters under SVR model. It is expected that the proposed model will improve prediction yields of the exchange rates significantly.

Section 2 is a collection of exchange rates and other relevant datasets, in addition to, a test to verify nonlinear and nonstationary characteristics of the dataset. Following to this, we introduce mathematical considerations of the *aIMF* algorithm and the multiclass SVR model. Section 3 consists of simulations and results of the proposed model, including their measurements and robust test of the proposed model. Finally, Section 4 concludes this paper and summarises the discussion.

2 Methods

In Section 2.1, we first collect daily trading data of EUR-USD, later called EUR-USD, and other relevant data from Bloomberg. In Section 2.1, since each time span of data may not be fully matched to the same day, we then adjust all the datasets by using linear interpolation. The next step is to test their relations with Granger causality test. For a test of nonlinear nonstationary time series, we introduce normality and unit root tests. The datasets after testing will be injected as input to EMD process.

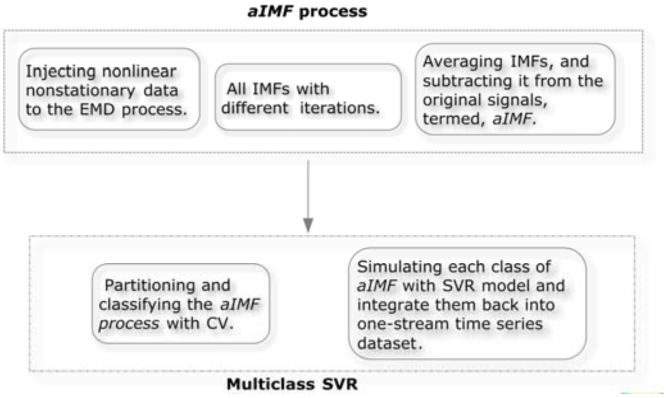

aIMF process
Injecting nonlinear nonstationary data to the EMD process.
All IMFs with different iterations.
Averaging IMFs, and subtracting it from the original signals, termed, aIMF.
Partitioning and classifying the aIMF process with CV.
Simulating each class of aIMF with SVR model and integrate them back into one-stream time series dataset.
Multiclass SVR

Figure 1. Block diagram of the proposed *aIMF* algorithm and multiclass SVR model

The EMD algorithm commences by using a sifting technique to decompose the original signal/datasets of the EUR-USD, as demonstrated in Figure 1 and detailed in Section 2.2. The output from iteration of EMD process is called intermediate mode frequency (IMF). The new algorithm starts by averaging every IMF and then subtracts the averaged IMF from the original datasets. The averaged IMF is later found to be normally distributed, similar to the characteristics of white Gaussian noise. Inevitably, it is safe to assume that removing the averaged IMF can reduce any unwanted signals i.e. white Gaussian noise. Finally, a new digital filter is created. Following, we simulate SVR using multiclass technique, which employs reversion of coefficient of variance (CV) partitioning and classifying the datasets that have been filtered from the *aIMF* algorithm.

2.1 Data preparation

Foreign exchange (FX) data are numbers with two to six decimal points coded in ASCII, and are nonlinear and nonstationary with exogenous influence. Normally, official FX data cannot be obtained publicly, but they are available at financial institutions and media companies, such as Bloomberg and Olsen Associates. The datasets used in this paper are EUR-USD, trade exchange rates, commodity indices, major foreign exchange rates, and other macroeconomic data such as interest rates and government bonds, and so on. In regression analysis, the SVR model requires both dependent and independent variables [26].

We collected day-to-day trading data from Bloomberg, excluding weekends and holidays from January 2, 2001 to June 1, 2012, totalling 2322 sets. The data first had different time-spanned bases since some datasets changed every minute, while others altered on an hourly or daily basis. We then adjusted those deviations by linear interpolation, converting them to daily deviation. This was done to ensure that the time span of all 65 datasets which are fully matched to the same day, without any irregularities. The datasets collected are divided to four groups, namely: i) Foreign exchanges (FOREX); EUR-USD, EUR-GBP, EUR-CNY, EUR-JPY, EUR-RUB, EUR-AUD and EUR-CHF, ii) Money market interest rate (MMIR); 1-month, 3-month, 6-month and 12-month EURIBOR, 1-month, 3-month, 6-month and 12-month LIBOR, EONIA and Federal fund rate; iii) Government bonds; 2-year, 5-year, 10-year and 30-year US Treasuries, 1-year, 3-year, 5-year and 10-year ECB, 5-year, 10-year, 15-year and 20-year UK gilts, 1-year, 5-year, 10-year, 15-year and 20-year Japanese, 1-year, 5-year and 10-year German, 1-year, 5-year and 10-year French, 1-year 5-year and 10-year Italian, 2-year, 5-year and 10-year Swiss, 1-year, 5-year and 10-year Australian; iii) Indices; Dow Jones, Euro50, NASDAQ, US S&P500, Nikkei225, FTSE100, Dax and CAC; and iv) commodities; platinum, nickel, zinc, silver, gold, tin, cooper, and cacao. Table 1 Normality test for EUR-USD datasets.

Table 1. Normality test for EUR-USD datasets

Methods	Statistics	*p*-value
Anderson-Darling	A = 36.095	< 2.2e-16
Cramer-von Mises	W = 5.6641	7.37e-10
Kolmogorov-Smirnov	D = 0.1029	< 2.2e-16
Pearson chi-square normality	P = 1128.76	< 2.2e-16

Table 2. Independent variables that are correlated with EUR-USD

Independent variables	F-Statistic	*p*-value
FOREX		
EUR-CNY	2.45847	0.0001
EUR-GBP	2.30192	0.0003
EUR-JPY	2.19065	0.0007
EUR-RUB	1.90520	0.0051
EUR-CHF (Swiss)	1.83181	0.0082
Government bonds		
15-year Japanese	1.59032	0.0344
20-year Japanese	1.79611	0.0102
1-year Australian	1.96054	0.0035
10-year Australian	1.75448	0.0132
20-year UK gilts	1.55266	0.0424
Commodities		
silver	2.39814	0.0002
gold	1.71102	0.0171

For a definition in regression analysis, we assign the EUR-USD as ‘dependent variables’, and the other data as ‘independent variables’. The next step aims to test nonlinear and nonstationary attributes of dependent variables by using

Normality test. The testing results shown in Table 1 indicate that dependent variables are nonlinear and nonstationary. Finally in Table 2, we test intra and inter correlations of dependent variables and independent variables by using Granger causality test, and found that in the level of confidence at 95%, only 12 independent variables from the three groups are correlated with dependent variables. Those group details are: i) FOREX; EUR-CNY, EUR-GBP, EUR-JPY, EUR-RUB and EUR-CHF (Swiss), ii) Government bonds; 10-year and 20-year Japanese, 1-year and 10-year Australian, and 20-year UK gilts, and iii) commodities; silver and gold.

2.2 EMD algorithm

In terms of signal processing, the EMD algorithm is seen as an a posteriori method based on adaptive characteristic scale separation. This process is useful when the input signal oscillation is nonlinear and/or nonstationary. The EMD is not a sampling with a fixed time slot in the time series; its local mean of a signal is defined through enveloping without resorting to any time scale. In general, EMD employs a sifting process and cubic spline technique for smoothing and filtering a signal [4]. The cubic splining technique performs as a two-sided filter, improving confident interval of the dataset distribution. The generic EMD algorithm is followed by the following steps [3] and shown from (1) to (8). The new add-on procedure after (8) until (9) is new algorithm set that generates a filtered dataset.

A nonlinear and nonstationary time series dataset is denoted as $X_i(t)$.The IMF, $c_i(t)$ is defined under the conditions [5] that (i) the numbers of extrema (maxima plus minima) and zero crossings in the entire data series must either be equal or differ by at most one, and (ii) at any point, the mean value of the envelope defined by the local maxima and that defined by the local minima, must be zero; and these conditions are provided by

$$\begin{bmatrix} nU_i(t)+nL_i(t)=nZ_i(t), \\ nU_i(t)+nL_i(t)=nZ_i(t)\pm 1 \end{bmatrix} = \begin{bmatrix} \frac{U_i(t)+L_i(t)}{2}=0 \end{bmatrix} \tag{1}$$

where $nUi(t)$, $nL_i(t)$, and $nZ_i(t)$, are the numbers of maxima (upper peak), minima (lower peak) and zero crossing, respectively. The *aIMF* algorithm can be presented as follows:

Steps 1: Spline $X_i(t)$ by interrelating with $U_i(t)$ and $L_i(t)$ which is given by

$$f_{int}\{U_i(t),X_i(t)\}=U_{1int}(t),U_{2int}(t),U_{3int}(t),....,U_{nint}(t) \tag{2}$$

$$\{L_i(t),X_i(t)\}=L_{1int}(t),L_{2int}(t),L_{3int}(t),....,L_{1int}(t) \tag{3}$$

The algorithm used for a parabolic interpolation can be described as follows:

i) When constructing the upper and lower envelopes, we calculate the parabola coefficients of aX^2+bX+c using $X(k-1),X(k),X(k+1)$.

ii) If the second-degree coefficient, a, equals zero, $X(k)$ is certainly not an extremum; thus the sliding window moves further on one discrete of $X(k)$.

iii) If the first-degree coefficient, b, equals zero, $X(k)$ is an extremum, either a maximum or minimum depending on the sign. We then calculate the top of this parabola by introducing $t_{top}=\frac{-b}{2a}$.

iv) Repeat ii) and iii) and stop after executing $X(k+n)$.

Finally, we explored the upper maxima and lower minima under (2) and (3).

Step 2: Average the maxima and minima in order of the time series, which is represented as

$$m_i(t)=\frac{U_{1int}+L_{1int}}{2},\frac{U_{2int}+L_{2int}}{2},\frac{U_{3int}+L_{3int}}{2},...,\frac{U_{nint}+L_{nint}}{2} \quad (4)$$

Step 3: Subtracting the original datasets $X_i(t)$ from the average of the local extrema (maxima and minima) $m_i(t)$ In order of the time series, the new decomposed signal is

$$h_i(t)=X_i(t)-m_i(t)\,. \quad (5)$$

Step 4: Repeat steps (i) through (iii) k times until $h_{1k}(t)$ equals $c_1(t)$. Referring to (5) where $h_{1(k-1)}(t)\text{-}m_{1k}=h_{1k}(t)$, we designate $c_1(t)$ as the first IMF.

Step 5: Find other IMFs by calculating the first residual, given by

$$R_1(t)=x_1(t)\text{-}c_1(t). \quad (6)$$

We derive $c_2(t),c_3(t),c_4(t),...,c_n(t)$ so that $R_1(t)$ is treated as a new dataset in the next loop; and complete obtaining $C_n(t)$.This procedure is represented by

$$R_2(t)=R_1(t)\text{-}c_2(t) \quad (7)$$

………….

$$R_n(t)=R_{n\text{-}1}(t)\text{-}c_n(t)\,. \quad (8)$$

2.3 Proposed *aIMF* algorithm

The EMD process in (8) stops when the residual $R_n(t)$ becomes either over-distorted or mono-component from which no further IMF can be decomposed. Characteristics of all IMFs decomposed $c_n(t)$appear to be different, detail in Figure 2. We average out all the IMFs decomposed $c_n(t)$, and termed to $c_a(t)$, in which represents by

$$c_a(t)=\frac{\sum_{i=1}^{n}c_i(t)}{n}\,. \quad (9)$$

We measured distribution of the averaged IMF $c_a(t)$ using Normality test and found to be normally distributed, in which details are demonstrated in Figure 2 and Table 3. Having analysed, we discovered in Section 3.2 that characteristics of the averaged IMF, $c_a(t)$ similar to the properties of white Gaussian noise which is normally distributed,. Therefore, it is safe to assume that removing $c_a(t)$ inevitably reduces white Gaussian noise. Followed this, we subtract the original datasets $X_n(t)$ with $c_a(t)$, given by

$$Y_n(t)=X_n(t)\text{-}c_a(t) \quad (10)$$

where $Y_n(t)$ is a function of the *aIMF* algorithm.

On conclusion, we create a now algorithm that filter and smooth nonlinear and nonstationary time series datasets whereas details of simulations of the EMD process and the *aIMF* algorithm are demonstrated in Section 3.1 and 3.2.

2.4 Proposed data classification using mean reversion and CV

We introduce a new technique for the multiclass SVR model using a mean reversion with coefficient of variance (CV) to partition time horizon (span) of the datasets that have been decomposed by the *aIMF* algorithm. As a typical curve of exchange rates tends to shift towards the mean, the point being reversed can be used to determine changes in its directions i.e. from up to down, and vice versa. At the reversed point, the datasets are partitioned. Since the standard deviations of a nonstationary dataset are not the same, we measure datasets between the reversed points and input them to the SVR model as per detail in Section 3.3. The procedure using mean reversion with CV is presented as follows:

i) Compute the mean $\mu_n(t)$ of random variables $X_n(t)$.

ii) Compute the variance $V_n(t)$ of $X_n(t)$.

iii) By normalising each $V_n(t)$ through $\mu_n(t)$ we obtain $\frac{V_n(t)}{\mu_n(t)}$

iv) In an upward scenario where $V_1(t) < V_2(t),\ldots,n,$ or a downward where

a. if $\frac{V_2(t)}{\mu_2(t)} < \frac{V_1(t)}{\mu_1(t)}$ or $\frac{V_2(t)}{\mu_2(t)} > \frac{V_1(t)}{\mu_1(t)}$, mark the intercept point on the x-axis and denote it as M_1, the value is $X_{rn}(t)$, *where* $r = 1,2,\ldots,c$. Here c is last class generated by CV; or

b. if $\frac{V_{n-1}(t)}{m_{n-1}(t)} = \frac{V_n(t)}{m_n(t)}$, do not the intercept point on the x-axis.

v) Repeat iv) and stop when $\frac{V_n(t)}{\mu_n(t)}$ becomes the last data point (n); next, plot $M_2,\ldots,M_n$ and draw the line or curve between them.

vi) Compute CV of the data $X_{rn}(t)$ in blocks of $M_1, M_2, \ldots, M_n,$ where n is the number of partitions/blocks.

The original datasets $X_n(t)$ have been classified into different classes of CV. The next step indicated in Section 3.3 aims to individually simulate $X_n(t)$ that are in every CV's group with the SVR model. As a result, we create multiple kernel parameters that are suitable for every class of CV.

2.5 Proposed multiclass SVR model

This section first introduces a generic SVM model and later optimises multiple kernel parameters following the logic in Section 2.4. In terms of prediction, variables Premanode and Toumazou (2012) used the SVR model which is a set of training data can be defined as $\{(x_1, y_1), \ldots., (x_\ell, y_\ell)\}$, where $x_i \subset R^n$ denotes the input space of the sample, and $y_i \subset R$ for i=$1,\ldots,l$ where l corresponds to the size of the training dataset. The idea of the SVM problem is to determine a function that can approximate future values y_i; and the generic SVM estimating function is given by

$$y_i = f(x) = (w \cdot \Phi(x)) + b \tag{11}$$

where $b \subset R, w \subset R^n$, and Φ denotes a non-linear transformation from R^n to a high dimensional space. In (11), the dot product can be replaced with a kernel function. This function enables the dot product to be performed in the high-dimensional feature space using low-dimensional space data input independently from the transformation Φ. However, the kernel functions must satisfy Mercer's condition that corresponds to the inner product of feature space. We introduce the Laplacian kernel function is the best kernel function which is given by

$$k(x, x') = \exp\left(-\sigma \|x - x'\|\right). \tag{12}$$

By substituting (11) with the kernel function (12), and the multiclass SVM with, for an example, the Laplacian function is obtained as

$$\mathrm{f}_j(x, w) = w^T k_j(x, x') + b_j \tag{13}$$

where j is the classes derived by block of the mean reversion in Section 2.4.

To solve the (11), we must find the value of w and b such that values of x can be determined by minimising the regression risk. Scholkopf and Smola [27] introduced the SVR equation, and is given as

$$R_{reg}(f) = C\sum_{i=0}^{\ell}\Gamma(f(x_i) - y_i) + \frac{1}{2}\|w\|^2 \tag{14}$$

where $\Gamma(\cdot)$ is a cost function, and C is a constant. In (14), the constant C determines the penalties of estimation errors. A large value of indicates a more error free equation, whereas a small value indicates a high degree of error. In the simulation, the value of C value is optimised by a set of programming instructions.

The vector w can be written in terms of data points using the Lagrange multipliers as follows

$$w = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\Phi(x_i) \tag{15}$$

then, substitute (14) into (11), the generic equation can be rewritten as

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)(\Phi(x_i)\cdot\Phi(x)) + b \tag{16}$$

In (14) the cost function can be substituted by ε-insensitive loss functions which is

$$\Gamma(f(x) - y) = \begin{cases} |f(x) - y| - \varepsilon, & for \quad |f(x) - y| \ge \varepsilon \\ 0 & otherwise \end{cases} \tag{17}$$

then, the regression risk in (13) and the ε-insensitive loss function (17) shall be minimised to

$$\frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(x_i, x_j) - \sum_{i=1}^{\ell}\alpha_i^*(y_i - \varepsilon) - \alpha_i(y_i + \varepsilon) \tag{18}$$

subject to

$$\sum_{i=1}^{\ell}\alpha_i - \alpha_i^* = 0, \quad \alpha_i, \alpha_i^* \in [0, C]$$

Referring to (18), the Lagrange multiplier α_i and α_i^* represent a solution to the quadratic problem. Simulating the SVR model, we use the Lagrange multipliers in (17) that is in the $|f(x) - y| \ge \varepsilon$ state; because being in the other stage, simulation results will be biased. To continue solving variable b, we thus define (15) and (16) to

$$\begin{cases} \alpha_i(\varepsilon + \zeta_i - y_i + (w, x_i) + b) = 0 \\ \alpha_i^*(\varepsilon + \zeta_i^* + y_i - (w, x_i) - b) = 0 \end{cases} \tag{19}$$

and

$$\begin{cases} (C - \alpha_i)\zeta_i = 0 \\ (C - \alpha_i^*)\zeta_i^* = 0 \end{cases} \tag{20}$$

where ζ_i and ζ_i^* are slack variables used to measure errors outside the ε-tube. Since the Lagrange multipliers $\alpha_i, \alpha_i^* = 0$, and $\zeta_i^* = 0$ for $\alpha_i^* \in (0, C)$, the variable b can be computed as follows

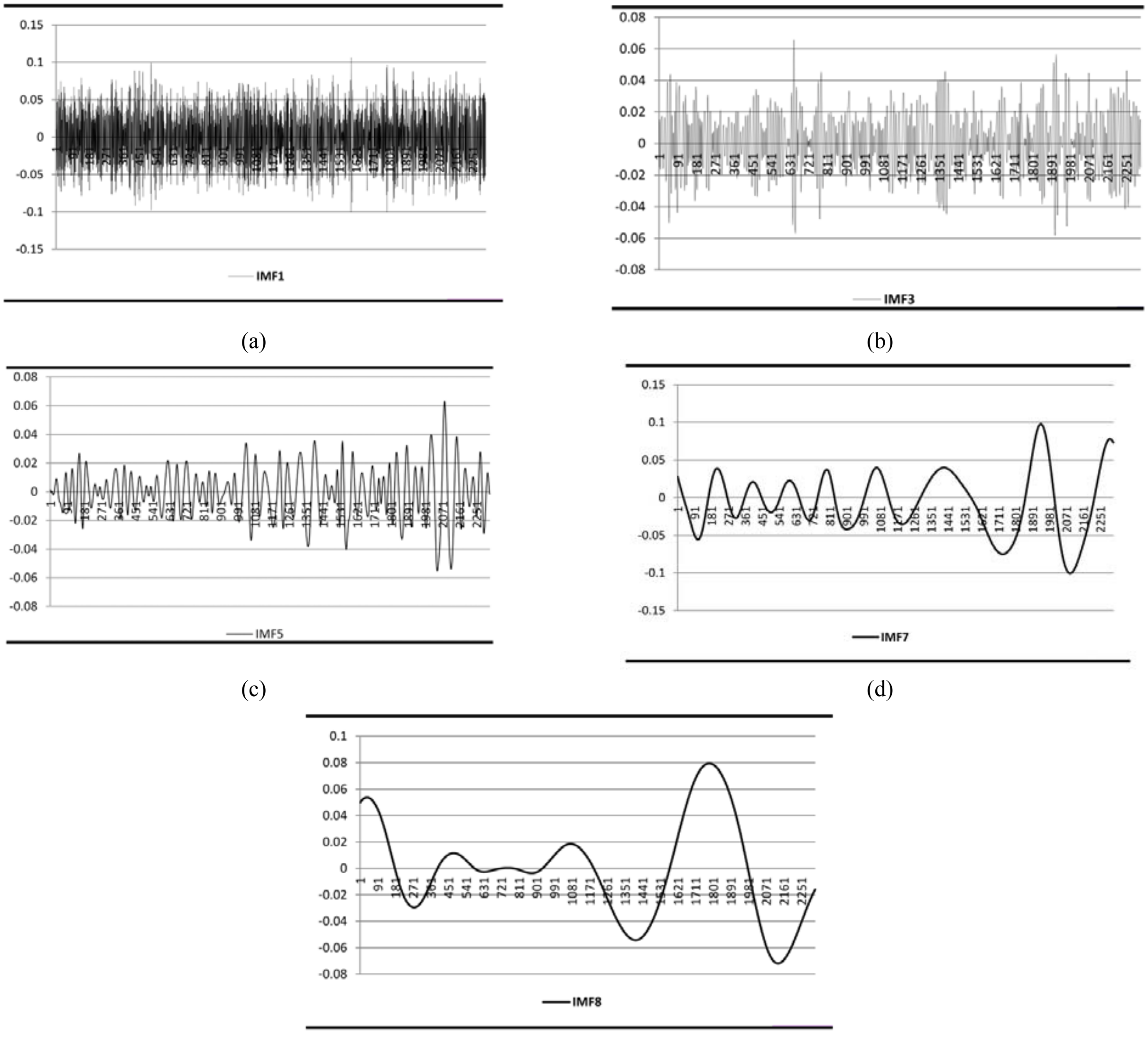
$$\begin{cases} b = y_i - (w, x_i) - \varepsilon & for\ \alpha_i \in (0, C) \\ b = y_i - (w, x_i) + \varepsilon & for\ \alpha_i^* \in (0, C) \end{cases}. \tag{21}$$

3 Simulations and results

This section discusses the simulation works and results of a combined mode consisting of the *aIMF* algorithm and the multiclass SVR model. The simulations in Section 3.1 and 3.2 are executed by EMD decomposition introduced by Kim and Oh [28] respectively; and the results are demonstrated in Figure 2 and Figure 3. We introduce out-of-sample prediction with ratios between training and test data at 30:70, 50:50, and 70:30; and later found that the ratio of 70:30 performed the best. This serves to imply that 30% of original datasets will be left for comparing with the results from proposed model, *aIMF* and multiclass SVR.

3.1 Simulations of EMD

To decompose the EMD process, the original datasets in Section 2.1 are simulated with the R-Programming introduced by Kim and Oh, 2009 [28], followed by the steps in Section 2.2. The process terminates once the number of extrema and zero crossings are equal to or differ by at most one inasmuch as the mean of each local IMF is zero. In this particular case, the last IMF is found to be the 8th IMF since the 9th IMF becomes mono-component; therefore, it is not useful to decompose the EMD further [3]. For ease in presentation, Figure 2 (a) to (e) shows only five samples out of ten from the EMD process.



(a)

(b)

(c)

(d)

(e)

Figure 2. For ease in presentation, graphs (a) to (e) show our plots of IMF1, IMF3, IMF5, IMF7 and IMF8, respectively

3.2 Simulation of the *aIMF* algorithm

Following to Section 3.1, we test normally distributed characteristics of all of the IMFs including the averaged IMF, $c_a(t)$ with the Anderson-Darling, Kolmogorov-Smirnov, and Pearson chi-squared methods. From the results indicated in Table 3, we noticed that the averaged IMF8 represented the highest degree of normal distribution, given a *p*-value of more than 0.05 whereas the second highest one was IMF7 and so on. Therefore, we reject null hypothesis that all IMFs are not normally distributed.

Table 3. Normality test of all IMFs decomposed and averaged IMF

IMF no.	Anderson-Darling	Kolmogorov-Smirnov	Pearson chi-square
1	< 2.2e-16	1.543e-10	< 2.2e-16
2	< 2.2e-16	1.925e-15	1.997e-09
3	< 2.2e-16	1.28e-08	1.824e-08
4	< 2.2e-16	< 2.2e-16	< 2.2e-16
5	< 2.2e-16	3.616e-13	< 2.2e-16
6	4.737e-05	0.006598	< 2.2e-16
7	3.288e-14	8.593e-09	< 2.2e-16
8	0.0593	0.0656	0.05808
$c_a(t)$[1]	0.6875	0.6020	0.68391

Note. 1=averaged IMF

The next step is to obtain the *aIMF* values from (10), and later compare it with test it the original datasets. The results demonstrated in Figure 3 indicated the upward and downward directions of those two graphs which are fully synchronised. For ease in presentation, we show a graph, in Figure 3, plotting only 50 samples of the results from the *aIMF* algorithm, compared to the original datasets of EUR-USD.

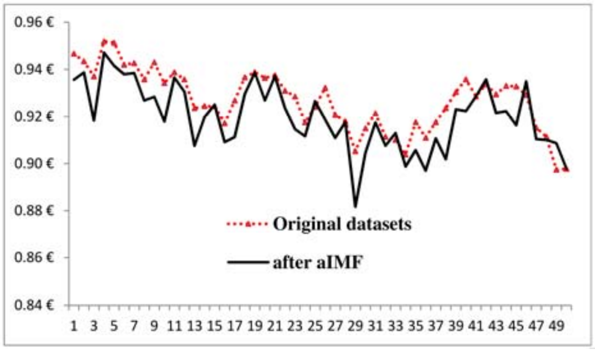


Figure 3. Graph of simulation results of the *aIMF* algorithm compared with the graph of original datasets. The x-axis represents 50 samples of the exchange rates the y-axis represents EUR-USD.

We verify the results generated from the *aIMF* algorithm using Normality test, namely: Anderson-Darling, Cramer-von Mises, Kolmogorov-Smirnov and Pearson chi-square normality methods, detail in Table 4.

Table 4. Normality test of the *aIMF* algorithm

Methods	Test statistic	*p*-value
Anderson-Darling	A = 36.0954	< 2.2e-16
Cramer-von Mises	W = 5.66410	7.4e-10
Kolmogorov-Smirnov	D = 0.10290	< 2.2e-16
Pearson chi-square normality	P= 1128.761	< 2.2e-16

3.3 Simulation of SVR with multiclass with mean reversion and CV technique

In this section, we compute CV of the Original datasets and the filtered dataset using the *aIMF* algorithm with the data points are between two reversed points (up and down), and then classify them manually into groups. For this eample, there are six groups out of 48 blocks of the datasets reversed in time series, as detail shown in Table 5. The datasets in every block are then simulated by the SVR model which automatically optimises parameters and distribution of kernel functions. As the results in each block, we select the best kernel functions and parameters from which different parameters and distributions of the kernel functions being used in the simulations. Apparently, the input datasets to the SVR simulation are: i) original datasets, ii) the filtered datasets by the *aIMF* algorithm in Section 3.2, and iii) the 12 independent variables from Section 2.1. In order to fit the kernel functions and parameters, we simulate the SVR model using R-programming [29]. Next, we re-introduce the SVR model to perform a simulation of test data, in which input datasets are i) 30% of the impendent variables, and ii) kernel parameters generated from the SVR model in the training data stage. Finally, we created two new datasets that is a prediction result of the EUR-USD –one is the Original datasets and the other is the filtered datasets using the *aIMF* algorithm.

3.4 Performance measurements of the *aIMF* and multiclass SVR model

Considering a prediction of the EUR-USD, we introduce cross validation technique to optimise the ratios of training and test data which are 30:70, 50:50, and 70:30. Table 5 indicates that the ratio of 70:30 presented the best result. As a result from Section 3.3, the training data (70%) of new *aIMF* blocks of datasets, which have been computed by mean reversion and CV technique, are injected into the SVR model one by one. For each block, we simulate it with different sets of the kernel functions, namely; Radial, Polynomial, Linear, Laplacian, Hyperbolic, Bessel, and ANOVA. We have found that in each CV class, the kernel parameters and functions used in the SVR model are different. Next, we select the best performance of the parameters and distributions from each block and rearrange them into a new dataset in time series. The different parameters and numbers of CV groups are demonstrated in Table 5. Having simulated the multiclass SVR model, we found that the Laplacian kernel function performed outstandingly, compared to the other kernel functions.

Table 5. Details of parameters of CV, numbers of blocks per CV class and its corresponding to the Laplacian kernel function

Class	Blocks	CV	Parameter of Laplacian kernel function
1	5	1.022778	$f(x,w) = w^T \exp\left(-0.09112303256\lVert x - x' \rVert\right) + b$
2	7	0.025920	$f(x,w) = w^T \exp\left(-0.10402077809\lVert x - x' \rVert\right) + b$
3	5	0.020776	$f(x,w) = w^T \exp\left(-0.09951566635\lVert x - x' \rVert\right) + b$
4	17	0.053612	$f(x,w) = w^T \exp\left(-0.16360524402\lVert x - x' \rVert\right) + b$
5	5	0.018410	$f(x,w) = w^T \exp\left(-0.83084376701\lVert x - x' \rVert\right) + b$
6	6	1.012593	$f(x,w) = w^T \exp\left(-0.28321301519\lVert x - x' \rVert\right) + b$

For a performance verification of the proposed model, *aIMF* and multiclass SVR by comparing to the test data (30% of the original datasets of EUR-USD exchange rates) with datasets generated in the following scenarios:

i) The generic SVR model simulating the EUR–USD exchange rates (dependent variables) and 12 independent variables from Table 2 without multiclass algorithm and the *aIMF* algorithm; called ‘SVR only’

ii) The generic SVR model simulating the EUR–USD exchange rates (dependent variables) and 12 independent variables from Table 2 with the aIMF algorithm, but without a multiclass algorithm; called ‘*aIMF* and SVR’

iii) The multiclass SVR model simulating the EUR–USD exchange rates, (dependent variables) and 12 independent variables from Table 2 without the *aIMF* algorithm, but the multiclass SVR still remained; called 'multiclass SVR'.

The prediction results of the three scenarios compared with the plots of the test data are displayed in Figure 4 using classification or 'accuracy test', which is used to compare up and down movements of the original datasets and datasets generated from the model. It is confirmed that our proposed *aIMF* and multiclass SVR model be outstandingly performed among others. For an ease of presentation we plotted 30 data points out of the test data.

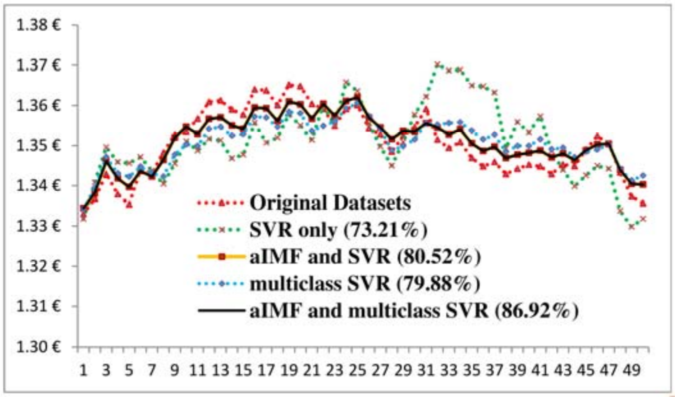


Figure 4. For ease of presentation we plotted 50 data points out of the test data. Graph of simulation results of the *aIMF* and multiclass SVR model compared with the graphs of original datasets, SVR only, *aIMF* and SVR, and multiclass SVR.

With respect to measure prediction performance of the proposed *aIMF* and multiclass SVR, we introduce loss estimators, namely; MSE, MAE, MAPE, R^2-squared, AIC $(= -2\ln \ell_{\max} + 2k,$ where $\ell_{\max}$ is the maximum likelihood achievable by the model and k the number of parameters of the model), and BIC (where N is the number of data points used in the fit. It comes from approximating the evidence ratios of models, known as the Bayes factor) including accuracy test. In Table 6, Under Laplacian kernel function with ratio of training and test data at 70:30, the performance accuracy of 'aIMF and multiclass SVR' is 86.92%, compared to those of 'SVR only', '*aIMF* and SVR', and 'multiclass SVR' which are 73.21%, 80.52% and 79.88%, respectively. Additionally, the estimation losses, namely; MSE, MAE, MAPE, R^2-squared, AIC and BIC derived from the simulations of '*aIMF* and multiclass SVR' are better than those from simulating 'SVR only', '*aIMF* and SVR', and 'multiclass SVR' as detailed in Table 6 and in Figure 4.

Table 6. Simulation results of the proposed '*aIMF* and multiclass SVR', compared to simulations of 'SVR only', '*aIMF* and SVR' and 'multiclass SVR' using the original datasets as a based reference

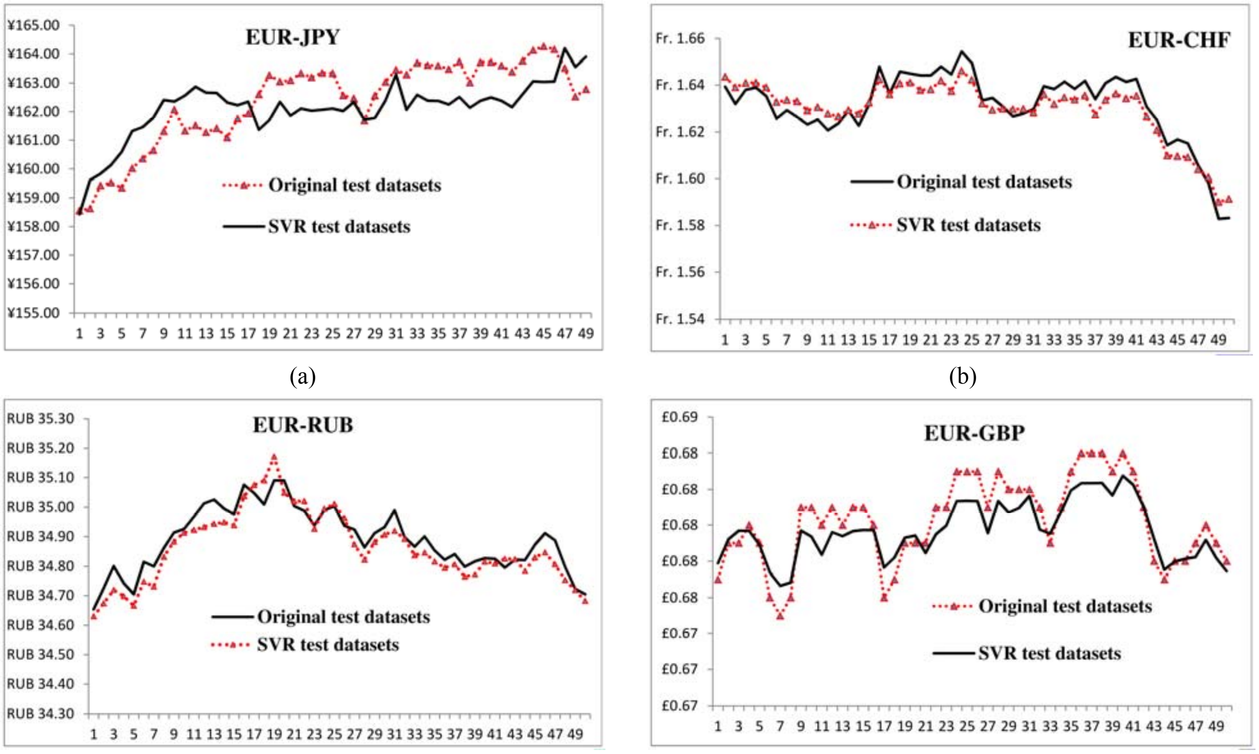
Loss estimators	**SVR only**	***aIMF* and SVR**	**Multiclass SVR**	***aIMF* and multiclass SVR**
Accuracy test (%)				
30:70	70.26	77.66	78.21	85.11
50:50	70.60	79.81	78.92	85.33
70:30	73.21	80.52	79.88	86.92
MSE	0.0001429	0.0001448	3.242E-05	1.089E-05
MAE	0.0098611	0.0097964	0.0046254	0.0025102
MAPE	0.6946638	0.6735869	0.3602532	0.3602532
R^2-squared	0.9827	0.9981	0.9992	0.9999
AIC	-4213.8360	-5738.339	-5209.846	-3173.644

3.5 Robustness test of the propose model (*aIMF* and multiclass SVR)

In this section we aim to test robustness of the proposed model (*aIMF* and multiclass SVR) by using different datasets, namely; EUR-JPY, EUR-CHF, EUR-RUB and EUR-GBP exchange rates as new dependent variables including the independent variables in Section 2.1. These independent variables will be simulated at the same way as happened in the early stage. Next, we test Normality for EUR-JPY, EUR-CHF, EUR-RUB and EUR-GBP exchange rates and using Grange causality test to select the datasets that correlated with each of them (new dependent variables). Following, from the new dependent variables we simulate EMD, the *aIMF* algorithm and SVR with multiclass using 'mean reversion and CV' technique under the criteria and procedures listed in Section 3.1, 3.2, and 3.3, respectively.

We measure the new dependent variables, using the same cross validation technique to optimise the ratios of training and test data which are 30:70, 50:50, and 70:30, and found that the ratio of 70:30 presented the best result. We then employed the same procedure in Section 3.3 and simulated the multiclass SVR model, we have found that the Laplacian kernel function performed outstandingly, compared to the other kernel functions.

Next, we perform verification of the proposed model, *aIMF* and multiclass SVR comparing to the test datasets (30% of the original datasets of each new dependent variables) simulating each new dependent variables with their corresponding 12 independent variables that has added back the EUR-USD exchange rates. By using classification or 'accuracy test', which is used to compare up and down movements of the original datasets and datasets generated from the model, the prediction results of the *aIMF* and multiclass SVR model with the plots of the test data are displayed in Figure 5 (a), (b), (c), and (d). It is confirmed that our proposed *aIMF* and multiclass SVR model still be outstandingly performed similar to the result in Section 3.3. For an ease of presentation, we plotted 50 data points out of test data.



(a) (b)

(c) (e)

Figure 5. For ease of presentation we plotted 50 data points out of the test data. Four Graphs of simulation results of the *aIMF* and multiclass SVR model simulating with the new dependent variables compared with the graphs of their corresponding original datasets (a) EUR-JPY, (b) EUR-CHF, (c) EUR-Ruble, and EUR-GBP

With respect to measure prediction performance resulted from of the proposed *aIMF* and multiclass SVR, we introduce loss estimators, namely; MSE, MAE, MAPE, R^2-squared, AIC, and BIC including accuracy test. Under Laplacian kernel function with ratio of training and test data at 70:30, the performance accuracy of: (a) EUR-JPY is 81.18%, (b) EUR-CHF is 88.50%, (c) EUR-RUB is 83.33%, and EUR-GBP is 84.19%, detailed in Figure 5. Additionally, these estimation losses derived from the simulations of '*aIMF* and multiclass SVR' are better than those from simulating 'SVR only', '*aIMF* and SVR', and 'multiclass SVR' as detailed in Table 7. We confirmed that the proposed model, *aIMF* and multiclass are highly robust in prediction the nonlinear nonstationary time series variables.

Table 7. Simulation results of the proposed '*aIMF* and multiclass SVR' of the new dependent variables compared to their corresponding original datasets

Description	EUR-JPY	EUR-CHF	EUR-RUB	EUR-GBP
Accuracy test (%)	81.18	88.50	83.33	84.19
MSE	1.594129064	0.000115822	0.213858861	8.34811E-06
MAE	0.815677713	0.005985487	0.140293456	0.001606438
MAPE	0.565511583	0.338472120	0.337532199	0.196598220
R^2-Squared	0.9945	0.9937	0.9873	0.9989
AIC	2294.993	-4374.498	896.4742	-6176.207
BIC	2308.629	-4360.862	910.1102	-6162.571

4 Conclusions and discussion

Predicting nonlinear and nonstationary EUR-USD exchange rates is very challenging, since many correlated independent random variables are involved. This paper introduces a new model comprised the *aIMF* algorithm and the multiclass SVR. While the *aIMF* algorithm functions as a filter demonising any nonlinear and nonstationary time series datasets, the multiclass SVR model predicts the future value of the exchange rates; whereas the independent and dependent variables were retrieved from the Bloomberg terminal from 2001 to 2012.

In Table 6, we employed a number of estimation losses, namely; MSE, MAE, MAPE, R^2-squared, AIC, BIC, and accuracy tests for measuring the performance of the our proposed model, aIMF and multiclass SVR. The results showed that the *aIMF* and multiclass SVR model outperformed the following simulations: SVR only, *aIMF* and SVR, and multiclass SVR. In terms of out-of-sample prediction, the *aIMF* yielded 86.92%, compared to the other models. It is now prudent to confirm that the *aIMF* and the multiclass SVR model can be a candidate estimator for nonlinear and nonstationary time series class of random variables. As regard to testing robustness of the model, we used EUR-JPY, EUR-CHF, EUR-RUB, and EUR-GBP exchange rates as dependent variables in the *aIMF* and multiclass SVR model and found the results similar to the simulations for the EUR-USD. We therefore confirmed that the proposed model, *aIMF* and multiclass SVR is highly robust in the prediction of the nonlinear nonstationary time series variables.

In the process of building up the *aIMF* and multiclass SVR, we encountered a complexity of creating a new novel, *aIMF* and multiclass SVR, in which are interdisciplinary exercise referring to digital signal processing and machine learning. Additionally, in the simulating process of the *aIMF* algorithm we experienced a computational limitation during the simulation process, which may be caused by the server – Intel(R) Xeon(R), with 2x2.4GHz E5620CPUs, 3.99GB RAM, and 64-bit Microsoft Windows Operating System. Having anticipated the issue and finally overcoming them, the proposed model is quite easy to use. In practice, we don't need to run the *aIMF* algorithm all time because the current training datasets can be used until the ratio of training and test data exceeds a certain ratio which produce inaccuracy of the outcome. Alternatively, the filtered data can be updated by the *aIMF* algorithm during non-peak hours.

Having created a model that can predict nonlinear nonstationary time series datasets with high accuracy of over 80% in less than one minute, we deem that our effort to build the *aIMF* and multiclass SVR model is worthwhile and justified.

Acknowledgements

This work is fully inspired by the collaborations of The Department of Electrical and Electronic Engineering and Centre of Bio-Inspired Technology, Imperial College London. The authors are grateful to Prof. Nicos Christofides, Wong Fan Voon and Janpen Jantorntrakul who read, edit and provide helpful suggestions.

References

[1] Shi, F., Chen, Q.J. Li, N.N. Hilbert Huang transform for prediction proteins subcellular location. Journal of Biomedical Science and Engineering. 2008: 59-63. http://dx.doi.org/10.4236/jbise.2008.11009

[2] Huang, W., Shen, Z., Huang, N.E. Fung, Y.C. Nonlinear Indicial Response of Complex Nonstationary Oscillations as Pulmonary Hypertension Responding to Step Hypoxia. Proceedings of the National Academy of Sciences. 1996; 96: 1834-1839. http://dx.doi.org/10.1073/pnas.96.5.1834

[3] Huang, N.E., Shen, Z., Long, S.R. The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Nonstationary Time Series Analysis. Journal of Proceedings of the Royal Society of London Series A (454). 1998; 903-995.

[4] Huang, N.E., Attoh-Okine, N.O. The Hilbert Transform in Engineering, CRC Press, Taylor & Francis Group. 2005.

[5] Wang, L., Koblinsky, C., Howden, S. Huang, N.E. Interannual Variability in the South China Sea from Expendable Bathythermograph Data, Journal of Geophysical Research. 1990; 104(23): 509-523.

[6] Datig, M., Schlurmann, T. Performance and limitations of the Hilbert-Huang Transformation (HHT) with an application to irregular water waves. Ocean Engineering. 2004; 1783-1834. http://dx.doi.org/10.1016/j.oceaneng.2004.03.007

[7] Huang, N.E., Shen, S.S.P. Hilbert-Huang Transform and Its Applications Interdisciplinary Mathematical Sciences 5 World Science. 2005; 2-26.

[8] Guhathakurta, K., Mukherjee, I. Roy, A. Empirical mode decomposition analysis of two different financial time series and their comparison, Chaos., Solutions & Fractals. 2008; 37 (4): 1214-1227. http://dx.doi.org/10.1016/j.chaos.2006.10.065

[9] Fisher, R.A. The use of multiple measurements in taxonomic problems. Annals of Eugenics. 1936; 111-132.

[10] Vapnik, V., Lerner, A. Pattern recognition using generalized portrait method. Automation and Remote Control.1963; 24 (1): 774-780.

[11] Duda, Richard, O., Hart, P.E. Pattern Classification and Scene Analysis, John Wiley & Sons Inc, NewYork. 2003.

[12] Boser, Bernhard, E., Guyon, I.M. Vapnik, V. A training algorithm for optimal Margin classifiers, COLT. Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM Press New York. 1992; 144-152..

[13] Basak, D., Pal, S. Patranabis, D.C. Support Vector Regression, Neural Information Processing – Letters and Reviews, 2007; 11 (10).

[14] Burges, C.J.C. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery. 1998: 121-167. http://dx.doi.org/10.1023/A:1009715923555

[15] Rychetsky, M. Algorithms and Architectures for Machine Learning based on Regularized Neural Networks and Support Vector Approaches, Shaker Verlag GmbH, ISBN-13: 9783826596407, 3-36.2001.

[16] Kim, K.J. Financial time series forecasting using support vector machine. Neurocomputing. 2003; 55: 307-309. http://dx.doi.org/10.1016/S0925-2312(03)00372-2

[17] Abe, S. Analysis of Multiclass Support Vector Machines. International Conference on Computational Intelligence for Modelling Control and Automation. 2003; 385-396. PMid:14670839

[18] Vapnik, V. The Nature of Statistical Learning Theory. Springer-Verlag, London. 1995. PMid:8555380

[19] Ornstein, L.S., Uhlenbeck, G.E. On the Theory of Brownian Motion. Physical Review, 36. 1930.

[20] Cochrane, J.H. How Big Is the Random Walk in GNP?. The Journal of Political Economy. 1998; 96 (5): 893-920. http://dx.doi.org/10.1086/261569

[21] Poterba, J.M., Summers L.H. Mean Reversion in Stock Prices Evidence and Implications. Journal of Financial Economics.1998; 22: 27-59. http://dx.doi.org/10.1016/0304-405X(88)90021-9

[22] Lo, A. W., MacKinlay, C. Stock Market Prices Do Not Follow Random Walks: Evidence from a Simple Specification Test. Review of Financial Studies1. 1998; 41-66. http://dx.doi.org/10.1093/rfs/1.1.41

[23] Bali, T.G., Demitras, K.O. Testing Mean Reversion in Stock Market Volatility. Journal of Futures Markets. 2008; 28 (1): 1-33.

[24] Fama, E.F., French, K.R. Permanent and temporary components of stock Prices. Journal of Political Economy. 1988; 96: 246-273. http://dx.doi.org/10.1086/261535

[25] Chen, S.N., Jeon, K. Mean Reversion Behavior of the Returns on Currency Assets. International Review of Economics and Finance. 2000; 7(2): 185-200. http://dx.doi.org/10.1016/S1059-0560(98)90039-9

[26] Premanode, B., Toumazou C. Improving Prediction of Exchange Rates using Differential EMD. Expert Systems with Applications. 2012; 40 (1): 377-384. http://dx.doi.org/10.1016/j.eswa.2012.07.048

[27] Smola A.J., Schölkopf, B. A Tutorial on Support Vector Regression. NeuroCOLT2 Technical Report Series. 1998.

[28] Kim, D., Oh, H.S. A Package for Empirical Mode Decomposition and Hilbert Spectrum. The R Journal, 1, ISSN 2073-4859. 2009; 40-46.

[29] Karatzoglou, A., Smola, A., Hornik, K. R-programming scripts of Support Vector Regression model [Internet]. Available from: http://cran.r-project.org/web/packages/kernlab/index.html.2012.