Computational Intelligence Prediction Model Integrating Empirical Mode Decomposition, Principal Component Analysis, and Weighted $k$-Nearest Neighbor

Li Tang* | He-Ping Pan | Yi-Yong Yao

Abstract—On the basis of machine leaning, suitable algorithms can make advanced time series analysis. This paper proposes a complex $k$-nearest neighbor (KNN) model for predicting financial time series. This model uses a complex feature extraction process integrating a forward rolling empirical mode decomposition (EMD) for financial time series signal analysis and principal component analysis (PCA) for the dimension reduction. The information-rich features are extracted then input to a weighted KNN classifier where the features are weighted with PCA loading. Finally, prediction is generated via regression on the selected nearest neighbors. The structure of the model as a whole is original. The test results on real historical data sets confirm the effectiveness of the models for predicting the Chinese stock index, an individual stock, and the EUR/USD exchange rate.

Index Terms—Empirical mode decomposition (EMD), $k$-nearest neighbor (KNN), principal component analysis (PCA), time series.

1. Introduction

Time series analysis, especially financial time series analysis, has attracted serious attention of financial and cross-disciplinary researchers. By analyzing and predicting financial time series, the essential information and features of market can be found for supporting the financial activities and decision-making, which has very important practical significance. There are a lot of literature research on the methods of economic prediction, including typical ones: Autoregressive integrated moving average (ARIMA) models, autoregressive conditional heteroscedasticity (ARCH) models, generalized autoregressive conditional heteroscedasticity (GARCH)
Although these computational intelligence models are equivalent, it has turned out that the feature extraction for generating essential information from original data as input is a foremost key challenge to prediction. Actually, the feature extraction problem is equivalent to signal processing. Therefore, the techniques in signal representation can be applied for the feature extraction of financial time series. The empirical mode decomposition (EMD)\cite{16} is widely used for signal decomposition, even in financial prediction\cite{19,21,22}. It has been demonstrated that EMD is an effective algorithm for dealing with nonlinear and non-stationary time series. But most of the researches have not considered the stability of decomposition near the end. The forward rolling EMD\cite{19} extracts the most recent data using a sliding window can tackle this problem. It should be noted that the high-dimension output of EMD as the input of prediction can make the model unstable usually. Thus, there should be a dimensionality reduction. The principal component analysis (PCA)\cite{20} maps the original high-dimensional data set to low-dimensional space and transforms it to a set of principal components, which can reduce data dimensionality effectively\cite{21}. For prediction modeling, KNN is a classical nonparametric algorithm. Generally, a simple mean of $k$-nearest neighbors are used\cite{20}. However, the simple mean can not reflect the amount of information from KNN synthetically. In order to tackle this problem, we propose a new weighted KNN classifier with more comprehensive information as input and more reasonable classify algorithms for modeling the prediction of financial time series.

Along the direction of signal decomposition, feature extraction, and prediction modeling, this paper proposes a computational intelligence prediction model for financial time series. This model applies a complex feature extraction process integrating the forward rolling EMD for financial time series signal decomposition and PCA for the dimension reduction. The information-rich features extracted by this process are input to a weighted KNN classifier with PCA loading as weights and transformed into prediction via regression on the selected $k$-nearest neighbors. Therefore, this model can be called as a financial time series EMD-PCA-weighted KNN (FEPK) model.

2. FEPK Prediction Model

2.1. Framework of FEPK Prediction Model

As a computational intelligence prediction model, FEPK needs to learn from a historical data set. Generally, we should define a time frame of the historical series first. This paper only focus on daily time frame, thus assume that the historical daily time series long enough exists. The financial price time series $X(t)$ of time period $t$ consists of the open price $O(t)$, high price $H(t)$, low price $L(t)$, close price $C(t)$, and volume $V(t)$. In this paper, $X(t)$ means $C(t)$ which is the only consideration, and others will be considered in future work.

In general, we apply a sliding window to intercept the historical financial time series $DX(t,N)$ long enough

$$DX(t,N) = (X(t - (N - w) + 1)), \cdots, X(t - 1), X(t))$$ (1)

where $t$ is the most recent time, $N$ is the total number of days, and $w << N$ is the width of a sliding window, thus $DX(t,N)$ also can be expressed as $DX(t,w)$. For any $X(t)$, define a relative return as

$$R(t, \lambda) = (X(t) - X(t - \lambda)) / X(t - \lambda)$$ (2)

where $\lambda$ is the prediction step length with a basic value of $\lambda = 1$, thus use $R(t)$ for $R(t, \lambda)$ without any other specification in this paper. Correspondingly, a relative return data set can be expressed as

$$DR(t, w) = (R(t - (N - w) + 1)), \cdots, R(t - 1), R(t))$$ (3)

Therefore, we can define a general FEPK prediction model as

$$R(t + \lambda) = \text{KNN}[^{PCA}[^{EMD}(DR(t,w))], k]$$ (4)

where $k$ is the number of KNN and PCA* denotes the feature extraction process with PCA.
2.2. Nonlinear Feature Extraction Procedure Using EMD and PCA

An FEPK prediction model contains a complex nonlinear feature extraction process, integrating the forward rolling EMD and PCA. The forward rolling EMD, which uses a sliding window with EMD, is applied specially for financial time series signal analysis because it can satisfy the real-time requirement and tackle the end effect of EMD\cite{19}, which is called FtsEMD.

For training an FEPK model, the historical relative return data set \( \text{DR}(i - 1, N) \), where \( i = t, t - 1, \ldots, t - (N - w) + 1 \) as the input should be decomposed by FtsEMD with a sliding window, and \( \text{R}(i) \), where \( i = t, t - 1, \ldots, t - (N - w) + 1 \) is the output. Thus an input-output data set as

\[
\begin{align*}
\text{DT}(t, N - w) = & \begin{pmatrix}
\text{FtsEMD}(\text{DR}(t - 1), w) \rightarrow R(t) \\
\text{FtsEMD}(\text{DR}(t - 2), w) \rightarrow R(t - 1) \\
\vdots & \vdots \\
\text{FtsEMD}(\text{DR}(t - (N - w)), w) \rightarrow R(t - (N - w) + 1)
\end{pmatrix},
\end{align*}
\]

(5)

It should note that multilevel intrinsic mode functions (IMFs) are obtained by EMD. According to the definition of IMF\cite{16}, IMF should satisfy two criteria: 1) In whole time series, the number of zero-crossing points and that of extrema are equal or differ at most one; 2) for any time period, the mean of the upper envelope formed by local maxima and the lower envelope formed by local minima is zero.

A process of EMD for details can be described as follows (e.g. EMD of \( \text{DR}(t - 1) \)):

1) Find all the local extrema of \( \text{DR}(t - 1) \), including the minima and maxima.

2) Use cubic spline to form a lower envelope \( \text{le}(t - 1) \) with the minima and an upper envelope \( \text{ue}(t - 1) \) with the maxima, while calculating their mean value

\[
m_i(t - 1) = (\text{ue}(t - 1) + \text{le}(t - 1))/2.
\]

(6)

3) Calculate the details of decomposition

\[
h_i(t - 1) = \text{dr}(t - 1) - m_i(t - 1)
\]

(7)

where \( i = 1, 2, \ldots \) denotes the \( i \) decomposition. When \( i = 1 \), \( \text{dr}(t - 1) = \text{DR}(t - 1) \).

The steps 1) to 3) constitute one sifting process.

4) Check whether \( h_i(t - 1) \) is IMF:

i) If it is true, \( h_i(t - 1) \) is IMF and \( r(t - 1) \) is the residual, then

\[
c(t - 1) = h_i(t - 1)
\]

\[
r(t - 1) = \text{dr}(t - 1) - c(t - 1).
\]

(8)

(9)

Then, let \( \text{dr}(t - 1) = r(t - 1) \) and go through the next sifting process.

ii) If it is not true, let \( \text{dr}(t - 1) = h_i(t - 1) \) and go through the next sifting process.

The main meaning of IMF is to define an instantaneous frequency. In order to ensure a sufficient physical significance of the instantaneous frequency, \cite{16} defines a metric called SD for stopping sifting:

\[
\text{SD} = \sum_{i=1}^{N} \left[ \frac{|h_{ni}(t - 1) - h_i(t - 1)|^2}{(h_i(t - 1))^2} \right].
\]

(10)

When the value of SD is between 0.2 and 0.3, stop the sifting process. Meanwhile, if the extrema of \( r(t - 1) \) is less than 2, stop sifting and return all the results of iterations as

\[
\text{DR}(t - 1) = \left( \sum_{i=1}^{n} a_i \right) + r.
\]

(11)
It also can be expressed as

\[
DR(t - 1) = \begin{pmatrix}
IMF_1(t - 1, w) \\
IMF_2(t - 1, w) \\
\vdots \\
IMF_n(t - 1, w)
\end{pmatrix} + r.
\]

(12)

The original data set has been decomposed into \( n \) multilevel IMFs (usually, let \( n \leq 5 \)) and one residual \( r \), thus rewrite (5) as

\[
DT(t, N - w) = \{D \rightarrow R\}
\]

(13)

\[
D = \begin{pmatrix}
IMF_1(t - 1, w) & \cdots & IMF_n(t - 1, w) \\
IMF_1(t - 2, w) & \cdots & IMF_n(t - 2, w) \\
\vdots & \ddots & \vdots \\
IMF_1(t - (N - w), w) & \cdots & IMF_n(t - (N - w), w)
\end{pmatrix}
\]

(14)

\[
R = \begin{pmatrix}
R(t) & R(t - 1) & \cdots & R(t - (N - w) + 1)
\end{pmatrix}^T.
\]

(15)

Note each row of matrix \( D \) is composed of IMF time series, and it is a high-dimensional vector with redundancy information. Thus, we can apply PCA for the dimension reduction and feature extraction with rich-information.

PCA transforms the high-dimension data to a set of principal components which are arranged according to their variance contribution rates from large to small. The first principal component is the most able to reflect the original information. Thus, we can apply PCA for the dimension reduction and feature extraction with rich-information.

First, matrix \( D \) should be normalized to matrix \( Z \) and then do the singular value decomposition of matrix \( Z \)

\[
Z = (\text{normalization}(D))^T
\]

(16)

\[
Z = U \Sigma W^T
\]

(17)

where \( U \) and \( W \) are eigenvector matrices of \( ZZ^T \) and \( Z^T Z \), respectively, and both of them are orthogonal matrices; \( \Sigma \) is a nonnegative rectangular diagonal matrix with a diagonal matrix comprising the eigenvalues \( \sigma_i \) \((i = 1, 2, \ldots, r)\) of \( ZZ^T \) in the left part. Therefore, we can obtain a transformed matrix by

\[
Y = Z^T U = W \Sigma^T U^T U = W \Sigma^T
\]

(18)

where each column of matrix \( Y \) consists of each principal component in turn.

Practically, the information of matrix \( Z \) is mainly concentrated on the first \( p << r \) principal components. However, how to find the value of \( p \)? An effective way is to apply the cumulative contribution rate (CCR). In general, CCR is required to be more than a threshold (such as 85%)

\[
\text{CCR}_p = \frac{\sum_{i=1}^{p} \sigma_i}{\sum_{i=1}^{r} \sigma_i} > 85\%
\]

(19)

where \( \sigma_i \) \((i = 1, 2, \ldots, r)\) is also the variance contribution rate (VCR) corresponding to each principal component

\[
\text{VCR}_i = \sigma_i.
\]

(20)

Thus we can construct a new \( p \times r \) matrix \( \Sigma_p \)

\[
\Sigma_p = I_{p \times r} \Sigma.
\]

(21)
Correspondingly, a low-dimensional matrix $\mathbf{U}_p$ of $\mathbf{U}$ and matrix $\mathbf{Y}_p$ are obtained
\[ \mathbf{Y}_p = \mathbf{Z}^\top \mathbf{U}_p = \mathbf{W} \mathbf{S}_p \]
where $\mathbf{Y}_p$ is a low-dimensional matrix.

The information-rich features comprise matrix $\mathbf{Y}_p$ and VCR, $(i = 1, 2, \cdots, p)$ should be then input to a weighted KNN classifier.

### 2.3. New KNN Classifier with PCA Loading as Weights

As a classical nonparametric algorithm, KNN can match the most similar $k$-nearest neighbors by the historical data set itself for prediction. In this paper, we propose a new KNN classifier with PCA loading as weights to generate the prediction via regression on the selected nearest neighbors.

We assume $x(t + \lambda) = \text{DR}(t + \lambda, N)$ is the point that should be predicted. Let $x(t) = \text{DR}(t, N)$, matrix $\mathbf{Y}_p$ and VCR, $(i = 1, 2, \cdots, p)$ are the input of KNN, thus we can generate a prediction model as
\[ x(t + \lambda) = \text{KNN} \left( x(t), \mathbf{Y}_p, \text{VCR}, k \right). \]

Calculate the similarity between $x(t)$ and any sample $x_i$ $(i = 1, 2, \cdots, p)$ in matrix $\mathbf{Y}_p$
\[ S(x(t), x_i) = -||x(t) - x_i||^2. \]

In this paper, the Euclidian measure is used as a similarity metric and anyone special for financial time series may be used in our future work. Sort all values of $S$ to find the first $k$-largest values and the most similar KNN $x_i$ $(i = 1, 2, \cdots, k(k < p))$.

Note that the new KNN classifier generates the prediction with PCA loading as weights, thus the VCR, $(j = 1, 2, \cdots, k(k < p))$ corresponding to $k$-nearest principal components should be selected to be the weight coefficients. Then we can generate the prediction via regression on the selected $k$-nearest neighbors as
\[ x(t + \lambda) = \sum_{j=1}^{k} \text{VCR}_j x_j \]
where $x(t + \lambda) = \text{DR}(t + \lambda, N)$. According to (3), the prediction $R(t + \lambda)$ can be obtained.

### 2.4. Key Parameters of FEPK Prediction Model

According to (4), a specific FEPK prediction model is determined by three key parameters: 1) $\lambda$ is the step length of prediction, 2) $w$ is the width of a sliding window in FtsEMD, and 3) $k$ is the $k$-nearest neighbors for prediction.

### 3. Empirical Test of FEPK Models

#### 3.1. Performance Metrics of Specific FEPK Prediction Models

The direction of market trend implies the key information, thus it is the most important for market risk management and investment trading strategies. We should first predict the direction of the market trend and then the range of rise and fall which is affected by the information that appears during the day. Therefore, we use a metric called hit rate (HR) which just measures the accuracy of the prediction direction in this paper. Naturally, in our future research with comprehensive information as input, we will focus on predicting the range of rise and fall which should be also evaluated by other indicators (such as the mean absolute percentage error (MAPE), root mean square error (RMSE), mean absolute difference (MAD), and the sharp ratio).

\[ HR = \frac{1}{n} \sum_{i=1}^{n} d(i), \quad d(i) = \begin{cases} \begin{array}{ll} 1, & R_i \times R^* > 0 \\ 0, & R_i \times R^* < 0 \end{array} \end{cases} \]


where $R_i$ is the real value of the relative return, $R^*$ is the predicted value, and $n$ is the total number of samples.

### 3.2. Three Specific FEPK Models and Test

In order to test the FEPK prediction models on real financial time series, three specific FEPK prediction models are constructed:

1. FEPK_HS300_D1 for the $t+1$ daily return of the HS300 index prediction
   \[ R(t + 1) = \text{KNN}([\text{PCA}^*[\text{EMD}(\text{HS300D1}(t, w))]], k). \] (27)

2. FEPK_yonyou_D1 for the $t+1$ daily return of the individual stock-yonyou (600588.SH) prediction
   \[ R(t + 1) = \text{KNN}([\text{PCA}^*[\text{EMD}(\text{yonyouD1}(t, w))]], k). \] (28)

3. FEPK_EURUSD_D1 for the $t+1$ daily return of the EUR/USD exchange rate prediction
   \[ R(t + 1) = \text{KNN}([\text{PCA}^*[\text{EMD}(\text{EURUSD}(t, w))]], k). \] (29)

The historical date sets on HS300 and yonyou (600588.SH), which span over the period of 4 January 2007 to 28 July 2017 consisting of 2571 trading days, are used for FEPK_HS300_D1 and FEPK_yonyou_D1 model training and testing. And the historical date sets on the EUR/USD exchange rate, which spans over the period of 2 January 2007 to 24 November 2017 consisting of 2830 trading days, are used for FEPK_EURUSD_D1 model training and testing. The data sets are all divided into two parts, the earlier 80% data for in-sample training and later 20% data for out-of-sample testing. In terms HR, Table 1 shows the testing result of FEPK_HS300_D1. Its best HR is 0.754 (75.4%) with $w = 100$ and $k = 7$. Table 2 shows the testing result of FEPK_yonyou_D1. Its best HR is 0.741 (74.1%) with $w = 150$ and $k = 2$. Table 3 shows the testing result of FEPK_EURUSD_D1. This model achieves the best HR of 0.742 (74.2%) with $w = 300$ and $k = 6$. Thus it can be said that the FEPK models are effective prediction models for predicting the $t+1$ daily return of HS300, an individual stock, and EUR/USD exchange rate.

### 3.3. Comparison with EMD-KNN and KNN Prediction Models

FEPK models are compared with other six EMD-KNN and KNN models, including:

1. EMD-KNN_HS300_D1

\[ R(t + 1) = \text{KNN}([\text{EMD}(\text{HS300D1}(t, w))], k). \] (30)
2) EMD-KNN_yonyou_D1
\[ R(t + 1) = \text{KNN}\{\text{yonyouD1}(t, w), k\}. \] (31)

3) EMD-KNN_EURUSD_D1
\[ R(t + 1) = \text{KNN}\{\text{EURUSDD1}(t, w), k\}. \] (32)

4) KNN_HS300_D1
\[ R(t + 1) = \text{KNN}\{\text{HS300D1}(t, w), k\}. \] (33)

5) KNN_yonyou_D1
\[ R(t + 1) = \text{KNN}\{\text{yonyou}(t, w), k\}. \] (34)

6) KNN_EURUSD_D1
\[ R(t + 1) = \text{KNN}\{\text{EURUSDD1}(t, w), k\}. \] (35)

These six models are trained and tested in the same data sets and ways as the FEPK models. The best HR of each model is chosen for comparison as shown in Table 4. In terms of HR, the FEPK models perform the best, and the EMD-KNN models perform better than the KNN models. The comparison result implies IMFs generated by FtsEMD can reflect the data features effectively, the principal components obtained by PCA are information-rich, and they can improve the performance of the KNN prediction.

4. Conclusions

In this paper, we have proposed a computational intelligence prediction model, integrating the forward rolling EMD with a sliding window for the financial time series decomposition, PCA for the dimension reduction, and a weighted KNN for the prediction. In general, the structure of this model is original, with more comprehensive information as input and more reasonable classify algorithms than the original KNN. The feature extraction process integrating FtsEMD and PCA is an advanced special extraction method for financial time series signal analysis, which has the adaptability, comprehensiveness, and orthogonality of feature extraction. Moreover, a weighted KNN with PCA loading as weights is more reasonable and has a better effect on classifying than the simple KNN, thus it has better prediction performance.

The specific FEPK models are implemented and tested for predicting the \( t + 1 \) daily return of the HS300 index, yonyou stock, and EUR/USD exchange rate, resulting the best HR of 0.754 (75.4%) on HS300. A comparison of FEPK models with the EMD-KNN and KNN models shows the FEPK models are the superior ones.

For further work, we will focus on more comprehensive information as input, and a more suitable similarity metric for financial time series. Moreover, other effective nonlinear dimensionality reduction methods or prediction algorithms will be applied, such as the auto encoder and random forest.

**Disclosures**

The authors declare no conflicts of interest.
References


Li Tang was born in Sichuan in 1984. She received the B.S. degree in finance from Sichuan Agricultural University, Chengdu in 2007 and the M.S. degree in business information system from City University of Hong Kong, Hong Kong in 2010. She is currently pursuing the Ph.D. degree with the School of Management and Economics, University of Electronic Science and Technology of China (UESTC), Chengdu. And she works as an associate professor with the Tianfu College, Southwestern University of Finance and Economics, Chengdu. Her research interests include financial engineering, intelligent finance, and machine learning.

He-Ping Pan was born in Shanxi in 1961. He received the B.S. degree in photogrammetry and remote sensing from Wuhan Technical University of Surveying and Mapping (now merged into Wuhan University), Wuhan in 1982, and the Ph.D. degree in information science from Twente University, Enschede in 1990. He was the Cheung Kong Scholars Distinguished Professor of Chinese Ministry of Education; the Director of Digital Intelligence Research Center, Wuhan University, Wuhan; the senior investigator and the research leader of the Information Fusion Group, Data and Information Fusion Program, Cooperative Research Center for Sensor Signal and Information Processing (CSSIP), Adelaide, incorporated by University of Adelaide, University of South Australia, Flinders University, University of Melbourne, and University of Queensland; the visiting professor at University of British Columbia, Vancouver and University of Ballarat, Ballarat; a professor with Southwestern University of Finance and Economics, UESTC, and Chongqing Technology and Business University (Chongqing); the Vice President of Chongqing Institute of Finance, Chongqing. He published over 160 refereed publications and supervised over 20 Ph.D. and 150 master candidates. He won the Best Paper Awards of 1995 CPGIS and 1996 CSSIP Techfest Conference. His current research interests include intelligent finance, deep intelligent investing, and global macro hedge funds.

Yi-Yong Yao was born in Sichuan in 1975. He received the B.S. and M.S. degrees, both in computer science from Sichuan University, Chengdu in 1998 and 2004, respectively. He is currently pursuing the Ph.D. degree with the School of Management and Economics, UESTC. He is an associate professor in computer science with the Tianfu College, Southwestern University of Finance and Economics. His research interests include intelligent finance, quantitative transactions, and data analysis.