Liu and Cheng
EURASIP Journal on Advances in Signal Processing (2024) 2024:19
https://doi.org/10.1186/s13634-024-01115-5

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

De-noising classification method for financial time series based on ICEEMDAN and wavelet threshold, and its application

Bing Liu[1,2] and Huanhuan Cheng[1*]

*Correspondence:
hh_cheng@126.com

[1] School of Economics and Management, Huainan Normal University, Huainan 232038, China
[2] School of Mathematics and Statistics, Central China Normal University, Wuhan 430079, China

Abstract

This paper proposes a classification method for financial time series that addresses the significant issue of noise. The proposed method combines improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) and wavelet threshold de-noising. The method begins by employing ICEEMDAN to decompose the time series into modal components and residuals. Using the noise component verification approach introduced in this paper, these components are categorized into noisy and de-noised elements. The noisy components are then de-noised using the Wavelet Threshold technique, which separates the non-noise and noise elements. The final de-noised output is produced by merging the non-noise elements with the de-noised components, and the 1-NN (nearest neighbor) algorithm is applied for time series classification. Highlighting its practical value in finance, this paper introduces a two-step stock classification prediction method that combines time series classification with a BP (Backpropagation) neural network. The method first classifies stocks into portfolios with high internal similarity using time series classification. It then employs a BP neural network to predict the classification of stock price movements within these portfolios. Backtesting confirms that this approach can enhance the accuracy of predicting stock price fluctuations.

Keywords: Time series classification, Decomposition-ensemble, Empirical mode decomposition, Quantitative portfolio investment

1 Introduction

The classification of time series is a crucial research area with applications in healthcare, econometrics, and voice recognition, among other fields. As a result, numerous time series classification methods have been developed. However, the accuracy of classification algorithms, particularly those based on Euclidean and DTW distances, consistently declines [1] as the noise standard deviation increases. Noise has become a critical challenge in time series classification.

The literature indicates that the wavelet method is utilized for signal decomposition and de-noising [2, 3]. Similar to the empirical mode decomposition method, the wavelet method offers a multi-frequency and multi-scale analysis [4, 5], and it has been

extensively researched and applied [6, 7]. Fractal images or fractal noise, which are present in chaotic systems across various fields such as physics, biology, psychology, economics, and finance [8–10], have led scholars to integrate fractal theory [11, 12] into the Wavelet method to develop fractal wavelet techniques [13–16].

Inspired by this, scholars have focused on various joint de-noising methods combining modal decomposition with wavelet threshold, including EMD and wavelet threshold de-noising[17–19], CEEMDAN and wavelet threshold de-noising [20], ICEEMDAN and wavelet threshold de-noising [21], and variational mode decomposition (VMD) with wavelet threshold de-noising [22]. A prevalent challenge in these methods is determining whether an IMF component is dominated by noise. Common practice involves calculating the Pearson correlation coefficient between the IMF component and the original signal to gauge the IMF's information content. A threshold is set, below which the IMF components are deemed noise-dominated. However, using Pearson's correlation coefficient to identify noise components presents two problems: first, a lower degree of linear correlation between the IMF component and the original signal does not necessarily mean that the IMF component is noise; second, setting the correlation threshold is somewhat subjective and lacks convincing justification. To address this, this paper introduces a joint verification method that employs the t test and unit root test to ascertain whether an IMF component is noise-dominated. This method is rooted in the nature of noise and offers a clear parameter testing approach. It can replace the correlation coefficient test in various modal decomposition methods combined with wavelet threshold de-noising.

Building on this, the paper proposes a time series classification method based on ICEEMDAN and Wavelet Threshold joint de-noising. The process begins with ICEEMDAN, which decomposes the time series into a series of IMF components and residuals. The noise component verification method proposed here is then applied to categorize the IMF components and residuals into noise and de-noised elements. The noise components are subsequently de-noised using the wavelet threshold method, resulting in non-noise sequences. The final de-noised output is formed by combining these non-noise sequences with the de-noised components, after which the nearest neighbor 1-NN algorithm is used for time series classification.

Wang et al. [23] proposed a two-stage investment strategy for bear markets, initially using tail correlation coefficients for hierarchical clustering of assets based on fuzzy matrices, then selecting one asset from each cluster to form an investment portfolio. Empirical evidence showed that this method could construct portfolios more resistant to risk during bear markets. Gupta et al. [24] introduced a two-step investment framework that first employs a Bayesian classifier to identify investment targets for a portfolio, and then applies multiple criteria decision-making (MCDM) techniques to devise investment strategies.

This paper contends that the essence of financial time series classification methods lies in identifying the similarity among various financial time series. Technical indicators derived from portfolios with higher internal similarity are posited to be more effective than those from less similar portfolios. Consequently, a two-step classification prediction method for stock portfolios is introduced. This method first uses a time series classification algorithm to select stocks with higher similarity within a certain industry, then

employs a BP neural network to predict the classification of stock price movements within the portfolio.

The marginal contributions of this paper are threefold: (1) It proposes a noise component verification method with an objective and clear judgment standard, applicable to noise verification of IMF components across all modal decomposition methods. (2) The de-noising method put forward ensures that essential information is preserved and can effectively precede all time series data mining tasks. (3) It introduces a two-step stock classification prediction method that combines time series classification with a BP neural network, aiming to improve the accuracy of predicting stock price movements in investment portfolios.

2 De-noising classification method based on ICEEMDAN and wavelet threshold

2.1 Improved complete ensemble empirical mode decomposition with adaptive noise

The CEEMDAN algorithm in the current technology can effectively reduce the error in signal reconstruction, restoring the completeness of EMD. However, IMF components are easily affected by noise, and problems with residual noise and pseudo-modal components persist. The improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) algorithm [25] introduces a local envelope average, allowing for the decomposition of IMF components with less noise and greater physical significance. Let $x(t)$ represent the original time series, $E_j(\cdot)$ represent the j-th IMF component obtained after EMD decomposition, ω^i represent the i-th added Gaussian white noise, β_k the amplitude coefficient of the added noise, i.e., the signal-to-noise ratio in the k-th stage, and $i = 1, \ldots, I$ represent the number of experiments. The specific steps of the ICEEMDAN algorithm are as follows:

Step 1 Add noise to the time series $x(t)$ to construct a new time series.

$$x_i(t) = x(t) + \beta_0 E_1(\omega^i) \tag{1}$$

Step 2 Through EMD, calculate the i-th local average in Eq. (1), obtaining the first stage residual:

$$r_1(t) = < M(x_i(t)) > \tag{2}$$

where $< M(\cdot) >$ is the operator to calculate the average.

The noisy signal $x_i(t)$ obtains the first modal component of ICEEMDAN through EMD, i.e.,

$$\mathrm{IMF}_1(t) = x(t) - r_1(t) \tag{3}$$

Step 3 Similarly, perform I experiments ($i = 1, \ldots, I$), calculate the local average of the signal $r_1(t) + \beta_1 E_2(\omega^i)$, and obtain the second stage residual:

$$r_2(t) = < M(r_1(t) + \beta_1 E_2(\omega^i)) > \tag{4}$$

Subtract Eq. (2) from Eq. (4) to obtain the second IMF component of the original sequence:

$$\mathrm{IMF}_2(t) = r_1(t) - r_2(t) \tag{5}$$

Step 4 Repeat Step 3 until the extreme points of the residual do not exceed 2. The recursive formula for the k-th residual is as follows:

$$r_k(t) = \left\langle M(r_{k-1}(t) + \beta_{k-1}E_k(\omega^i)) \right\rangle \tag{6}$$

and the k-th component of the original sequence is obtained:

$$\mathrm{IMF}_k(t) = r_{k-1}(t) - r_k(t) \tag{7}$$

The final residual is:

$$R(t) = x(t) - \sum_{k=1}^{K} \mathrm{IMF}_k \tag{8}$$

Thus, the original time series $x(t)$ is ultimately decomposed into:

$$x(t) = \sum_{k=1}^{K} \mathrm{IMF}_k + R(t) \tag{9}$$

2.2 Noise component test method

In reality, financial time series contain a significant amount of noise. Assuming noise $\varepsilon(t) = 0$ is unrealistic; the actual data are often $\varepsilon(t) \neq 0$. Without loss of generality, let us assume that random noise $\varepsilon(t)$ exists in the time series $x(t)$. This noise reflects the impact of random factors on the de-noised time series $\tilde{x}(t)$. Consequently, we construct

$$x(t) = \tilde{x}(t) + \varepsilon(t) \tag{10}$$

Under the Gaussian assumption, the noise is considered white noise, meaning it follows a normal distribution with a mean of 0 and a variance of σ^2. It is denoted as $\varepsilon(t) \sim N(0, \sigma^2)$. At this point, the noise $\varepsilon(t)$ should exhibit zero mean and homoscedasticity. However, heteroscedasticity tests often necessitate the use of explanatory variables from the original model to construct an auxiliary regression model. This model helps determine whether random errors display heteroscedasticity. Conducting this test is challenging without first building a regression model.

In cointegration tests, if the variables X_t and Y_t are both first-order integrated $I(1)$, we assume the original model is $Y_t = \beta_0 + \beta_1 X_t + \varepsilon_t$. In tests for cointegration relationships, if $\varepsilon(t)$ is stationary with a mean of 0, it suggests that X_t and Y_t have a cointegrating relationship, ensuring that random errors in the equation do not accumulate. Conversely, if $\varepsilon(t)$ follows a random walk (unit root process), it implies that random errors in the equation will accumulate, leading to persistent deviations from equilibrium that cannot self-correct. If the random time series X_t is stationary, then:

(1) The mean of X_t does not change over time, $E(X_t) = \mu$.
(2) The variance of X_t does not change over time, $\mathrm{VAR}(X_t) = E(X_t - \mu)^2 = \sigma^2$.

(3) The covariance between X_t and X_{t-k} at any two periods relies solely on the distance or lag length (k) between these periods and does not depend on other variables (for all k). This is expressed as the covariance between X_t and X_{t-k}:

$$\gamma_k = E[(X_t - \mu)(X_{t+k} - \mu)] \tag{11}$$

If any of the above properties are not met, X_t is said to be non-stationary.

Given that this paper focuses on time series data, we can substitute the Gaussian model's test for random disturbances with an examination of whether $\varepsilon(t)$ is a stationary process with a mean of 0. When $\varepsilon(t)$ is stationary with a mean of 0, deviations from $\tilde{x}(t)$ are corrected promptly. The elimination of random noise $\varepsilon(t)$ does not affect the long-term trend of $\tilde{x}(t)$.

In the financial markets, the Shenzhen Component Index is one of the indices that most accurately represents the Chinese stock market. This paper compiles a financial time series sample using the daily closing prices of the Shenzhen Component Index from 2000 to 2021. The sample comprises 5332 data points for the Shenzhen Component Index. As depicted in Fig. 1, several IMF components and residues were derived from the CEEMDAN decomposition of the Shenzhen Component Index. The red line showcases distinct heteroscedasticity in the high-frequency IMF components during certain periods. Further analysis reveals that although the composite component of the initial high-frequency components passes the zero mean and stationarity tests, it exhibits heteroscedasticity in specific periods. This is a characteristic outcome, akin to "leptokurtic" and "volatility clustering" observed in financial time series. These heteroscedasticities signify that high-frequency IMF components are not solely composed of noise. Consequently, this paper suggests that if the composite component of high-frequency components, decomposed from a time series via the empirical mode decomposition method, is stationary with a mean of 0, it lacks long-term trend elements and is presumed to be primarily noise. This component is referred to as the "noise-containing component" in this paper. Further decomposition of this noise-containing component is necessary to extract valuable information.

Following the decomposition of a time series into a series of modal components and residues via the empirical mode decomposition method, the modal components and residues can be aggregated into two categories. Without loss of generality, if the division is

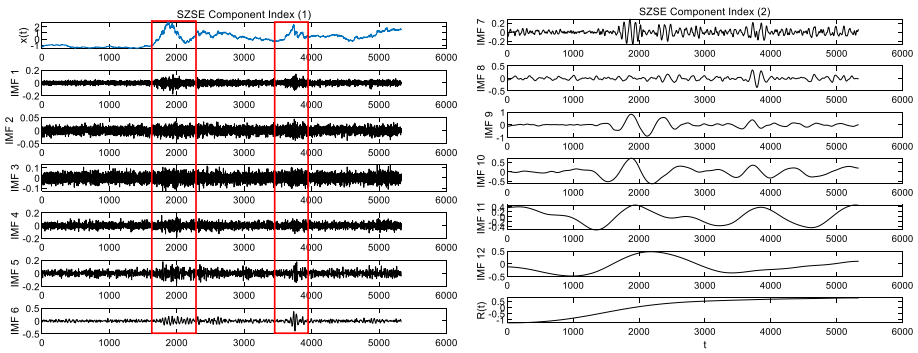

Fig. 1 CEEMDAN decomposition of the Shenzhen component index

between $1, \ldots, i$ and $i+1, \ldots$, the noise-containing component and the de-noised component can be obtained, denoted respectively as $x(t)_{\text{noise}}$ and $x(t)_{\text{non\_noise}}$:

$$x(t)_{\text{noise}} = \sum_{k=1}^{i} \text{IMF}_k \tag{12}$$

$$x(t)_{\text{non\_noise}} = \sum_{k=i+1}^{K} \text{IMF}_k + R(t) \tag{13}$$

The decomposition should satisfy the following conditions:

(1) When $k = 1, \ldots, i$ is present, the overall mean of IMF_k equals 0.
(2) The overall mean of $\sum_{k=1}^{i} \text{IMF}_k$ equals 0.
(3) When $k = 1, \cdots, i$ is present, IMF_k is stationary.
(4) $\sum_{k=1}^{i} \text{IMF}_k$ is stationary.

At this point, $\tilde{x}(t) = [\sum_{k=1}^{i} \text{IMF}_k - \varepsilon(t)] + \sum_{k=i+1}^{K} \text{IMF}_k + R(t)$.

For testing conditions (1) and (2), a population mean test can be conducted on IMF_k $(k = 1, \ldots, i)$ and $\sum_{k=1}^{i} \text{IMF}_k$ respectively, denoted as $H_0 : \mu = 0, H_1 : \mu \neq 0$. The t test statistic can be constructed as follows:

$$t = \frac{\overline{x}}{s/\sqrt{n}} \sim t(n-1) \tag{14}$$

Hence, the rejection region is $\{|t| > t_{\alpha/2}(n-1)\}$.

For testing conditions (3) and (4), the ADF test can be used.

2.3 Wavelet threshold de-noising

Donoho [26] proposed a de-noising method based on wavelet transformation, known as the wavelet threshold de-noising method. This method has been widely studied and applied [27–29]. It involves selecting suitable wavelet basis functions and decomposition levels, performing wavelet decomposition on the noise-containing signal, and obtaining a series of low-frequency and high-frequency wavelet coefficients. These coefficients are then processed with a threshold function. After processing, the high-frequency and low-frequency coefficients are reconstructed to produce a signal from which noise has been removed.

2.3.1 Threshold selection criteria

1 In wavelet threshold de-noising, the criteria for threshold selection typically include:
2 Fixed threshold (sqtwolog), $\lambda_1 = \sigma_n\sqrt{2\ln N}$, where σ_n is the noise standard deviation and N is the signal length.
3 Unbiased risk estimate threshold (rigrsure), based on Stein's unbiased risk estimate principle for adaptive threshold selection. The threshold is $\lambda_2 = \sigma_n\sqrt{\omega_b}$, where σ_n is the noise standard deviation and ω_b is the risk function.

2.3.2 Threshold functions

After the noise-containing component $x(t)_{\text{noise}}$ undergoes wavelet decomposition, the wavelet coefficients are de-noised using threshold functions. This process separates the noise component $\varepsilon(t)$ from the non-noise component $\vec{x}(t)$, where $\vec{x}(t) = \sum_{k=1}^{i} \text{IMF}_k - \varepsilon(t)$. The wavelet coefficient processing includes soft threshold functions, hard threshold functions, and some improved threshold functions. Assuming $\omega_{j,k}$ is the wavelet coefficient, $\hat{\omega}_{j,k}$ is the quantized wavelet coefficient, sgn is the sign function, and λ is the threshold, the functions are as follows:

(1) Soft threshold function [30]

$$\hat{\omega}_{j,k} = \begin{cases} \text{sgn}(\omega_{j,k})\left(\left|\omega_{j,k}\right| - \lambda\right), & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases} \tag{15}$$

(2) Hard threshold function [31]

$$\hat{\omega}_{j,k} = \begin{cases} \omega_{j,k}, & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases} \tag{16}$$

(3) Improved threshold function (a1) [32]

$$\hat{\omega}_{j,k} = \begin{cases} \text{sgn}(\omega_{j,k})\left(\left|\omega_{j,k}\right|^2 - \lambda^2\right)^{\frac{1}{2}}, & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases} \tag{17}$$

(4) Improved threshold function (a2) [33]

$$\hat{\omega}_{j,k} = \begin{cases} \text{sgn}(\omega_{j,k})\left(\left|\omega_{j,k}\right| - 2^{(\lambda - \left|\omega_{j,k}\right|}\right), & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases} \tag{18}$$

(5) Improved threshold function (a3) [34]

$$\hat{\omega}_{j,k} = \begin{cases} \text{sgn}(\omega_{j,k})\left(\left|\omega_{j,k}\right| - \frac{2\lambda}{\exp\left(\frac{\left|\omega_{j,k}\right| - \lambda}{\lambda}\right) + 1}\right), & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases} \tag{19}$$

2.4 Euclidean distance

The similarity measure $D(x_i, x_j)$ between time series $x_i(t)$ and $x_j(t)$ is a function that takes two time series $x_i(t)$ and $x_j(t)$ as inputs and returns the distance d between the two time series.

Euclidean distance (ED) [35] is one of the most commonly used methods for measuring similarity in time series classification. It can be understood as the length of the straight line segment connecting two points and measures the absolute distance between two points in multidimensional space. The formula for Euclidean distance is as follows:

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \tag{20}$$

2.5 Nearest neighbor algorithm

First, we find the k-nearest neighbor samples of the studied sample in the training data set. If most of the k-nearest neighbor samples belong to a certain category, then the sample also belongs to this category, which is the *k*-nearest neighbor algorithm (KNN) [36]. The specific formula is as follows:

Importing: training datasets

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$$

where $x_i \in \chi \subseteq R^n$ is the time series of the sample, $y_i \in \mathbf{y} = (c_1, c_2, \ldots, c_K\}$ and is the category of the sample, $i = 1, 2, \ldots, N$.

Output: Class Y to which test set sample x belongs.

(1) According to the Euclidean distance metric, find k points nearest to the test set sample X in the training set t, and record the field of X covering these k points as $N_k(x)$;
(2) Determine the category y of X in $N_k(x)$ according to the classification decision rules (majority voting):

$$y = \arg\max_{c_j} \sum_{x_i \in N_k(x)} I(y_i = c_j), i = 1, 2, \ldots, N; j = 1, 2, \ldots, K \tag{21}$$

where I is the indicating function, that is, at that time $y_i = c_j$ I is 1, otherwise I is 0.

The special case of *k*-nearest neighbor algorithm is the case of $k=1$, which is called nearest neighbor 1-NN algorithm. Because the nearest neighbor 1-NN algorithm has the advantage of no parameters, it is more convenient to compare among various methods, so this paper selects the nearest neighbor 1-NN algorithm to determine the classification label of samples.

2.6 Classification method steps

The ICEEMAN method is employed here for empirical mode decomposition. This paper proposes a financial time series de-noising classification method based on ICEEMDAN and wavelet threshold, with the steps detailed as follows:

Step 1 Utilize ICEEMDAN to decompose the time series, resulting in IMF components and a residual component.
Step 2 Carry out t tests and unit root tests for each IMF_k and $\sum_{k=1}^{i} \mathrm{IMF}_k$, then gather the IMF components and the residual components to form the noise-containing component $x(t)_{\mathrm{noise}}$ and the noise-removed component $x(t)_{\mathrm{non\_noise}}$.
Step 3 Apply wavelet threshold de-noising to the noise-containing component $x(t)_{\mathrm{noise}}$, breaking it down into the noise component $\varepsilon(t)$ and the retained noise-free component $\vec{x}(t)$. $\varepsilon(t)$ will be the final noise component. Integrate the retained

noise-free component in the noisy component $\vec{x}(t)$ to the noise-removed component $x(t)_{\text{non\_noise}}$, obtaining the final de-noised signal $\tilde{x}(t)_{\text{non\_noise}}$. The de-noising methods proposed in this paper are outlined above.

$$\begin{aligned} x(t) &= \sum_{k=1}^{K} \text{IMF}_k + R(t) \\ &= x(t)_{\text{noise}} + x(t)_{\text{non\_noise}} \\ &= \varepsilon(t) + \vec{x}(t) + x(t)_{\text{non\_noise}} \\ &= \varepsilon(t) + \tilde{x}(t) \end{aligned} \tag{22}$$

Step 4 Calculate the Euclidean distances $D(\tilde{x}_i, \tilde{x}_j) = \sqrt{\sum_{k=1}^{n} (\tilde{x}_{ik} - \tilde{x}_{ik})^2}$ of the de-noised signals from the training and testing sets, applying 1-NN to label the test set with its category index. This results in classifying each time series in the test set.

3 Two-step stock classification forecasting method based on classification method and BP neural network

3.1 BP neural network

The BP (Backpropagation) neural network [37] refers to one of the most classical neural networks. It is a neural network that uses the backpropagation algorithm. Backpropagation involves gathering errors produced in the simulation process, feeding the aforementioned errors back to the output values, then adjusting the weights of the neurons with these errors, thus generating an artificial neural network system that is capable of simulating the original problem.

A BP neural network primarily consists of an input layer, one or more hidden layers, and an output layer, each with a certain number of nodes (neurons). Typically, the input data of a neural network moves forward through the input layer, hidden layer, and output layer. Furthermore, the BP neural network covers backpropagation, i.e., the output errors start to move backward from the output layer. The specific steps [38] are elucidated as follows:

Step 1 Initialize weights;
Step 2 Forward move the signal, obtain model output $\mathbf{y}$, compute error vector $\mathbf{E}$, and calculate the delta $\boldsymbol{\delta}$ of output nodes;

$$\mathbf{e} = \mathbf{d} - \mathbf{y} \tag{23}$$

$$\boldsymbol{\delta} = \phi'(\mathbf{V})\mathbf{E} \tag{24}$$

Step 3 Calculate the delta AA of the backpropagated output nodes and the delta of the next layer of nodes;

$$\mathbf{E}^{(k)} = \mathbf{W}^T \boldsymbol{\delta} \tag{25}$$

$$\boldsymbol{\delta}^{(k)} = \phi'(\mathbf{V}^{(k)})\mathbf{E}^{(k)} \tag{26}$$

Step 4 Repeat Step 3 until it calculates the hidden layer on the right side of the input layer;
Step 5 Adjust the weight values according to the following formula, i.e.,

$$\Delta w_{ij} = \alpha \delta_i x_j \tag{27}$$

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \tag{28}$$

Step 6 Repeat Steps 2 to 5 for all training data nodes;
Step 7 Repeat Steps 2 to 6 until the neural network has received suitable training.

3.2 Method steps

The two-step stock classification forecasting method based on the classification method and the BP neural network encompasses the following steps:

Step 1 Tag multiple industry indices with category labels, and combine the closing prices of these indices at the first time stage as the training set for the first step of the time series classification stage; select all stocks in a certain industry into the portfolio as the control group for the second step of the forecast stage; and select the adjusted prices of the stocks at the first time stage of this control group as the test set for the first step.
Step 2 Use the time series decomposition-ensemble classification method to select the investment portfolio of the control group, eliminate stocks with significant differences in morphological features from the industry index, and select stocks with a high degree of industry morphological similarity to form an investment portfolio, which is referred to as the experimental group.
Step 3 Use the data of the second time stage to calculate the technical indicators of the experimental group and the control group, respectively, to characterize the statistical features of the stocks. The technical indicators of the experimental group and the control group are split in time order into the training set and the prediction set in terms of the second step of the forecast stage.
Step 4 Define the historical samples of the experimental group and the control group as good, bad, or average, and tag them with rise and fall category labels.
Step 5 Adopt the mean–variance normalization method to normalize the training set and prediction set of the experimental group and the control group at the prediction stage.
Step 6 To avoid ineffective technical indicators reducing prediction performance, the correlation coefficient method is employed to determine the correlation between the technical indicators of the prediction stage training set and the stock category labels, thereby eliminating irrelevant technical indicators.
Step 7 Use the prediction stage training set to train the BP neural network, then use the prediction set to forecast the rise and fall classification of stocks, and compare the prediction accuracy of the experimental group and the control group.

4 Numerical experiments of classification method

4.1 De-noising experiment

To validate the proposed de-noising method, as shown in Fig. 2, we selected the function HeaviSine ($f(t) = 4\sin 4\pi t - \mathrm{sgn}(t-0.3) - \mathrm{sgn}(0.72-t)$) proposed by Donoho [26] for testing, with the noise standard deviation set as $\sigma = 0.2$.

In the parameter setting, the wavelet function uses db5, the decomposition level is 5, and the threshold λ adopts the unbiased risk estimation threshold (rigrsure) criterion. The threshold functions used include soft threshold function, hard threshold function, and improved threshold functions a1, a2, and a3. Here, the de-noising experiment uses these five threshold functions for wavelet threshold de-noising as the control group, and under the method proposed in this paper, these five threshold functions are used as the experimental group, and comparative experiments are conducted.

This paper uses the signal-to-noise ratio (SNR), mean square error (MSE), and waveform correlation coefficient (NCC) as evaluation indicators of de-noising performance. The higher the SNR, the more significant the noise suppression effect. The MSE reflects the similarity between the de-noised signal and the noise-free signal, and the smaller the error value, the better the de-noising performance. The calculation methods of SNR, MSE, and NCC are shown below:

$$\mathrm{SNR} = 10 \times \log_{10}\left[\frac{\sum_{k=1}^{n} x^2(k)}{\sum_{k=1}^{n}[y(k)-x(k)]^2}\right] \tag{29}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{k=1}^{n}[y(k)-x(k)]^2 \tag{30}$$

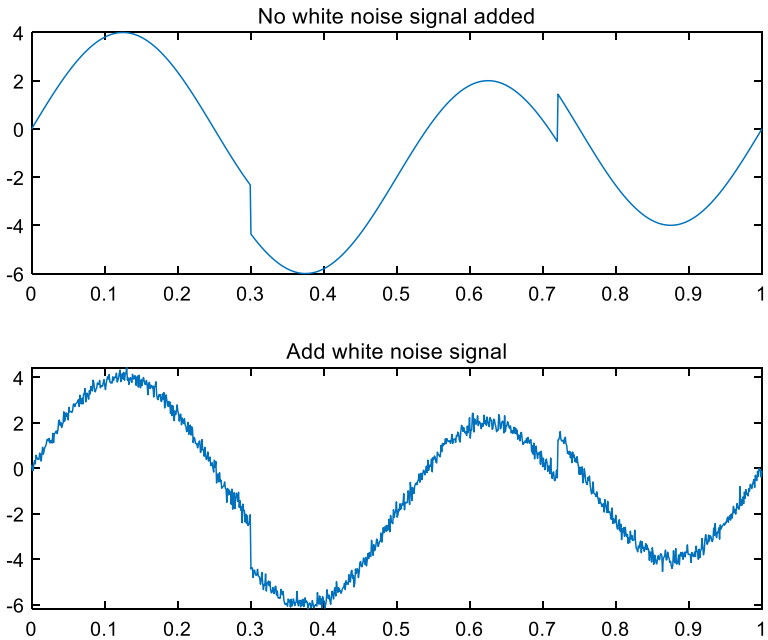

Fig. 2 Noise-free and noisy signals of HeaviSine

$$\mathrm{NCC} = \frac{\sum_{k=1}^{n} [x(k) \times y(k)]}{\sqrt{\sum_{k=1}^{n} x^2(k) \times \sum_{k=1}^{n} y^2(k)}} \tag{31}$$

where $x(k)$ is the noise-free signal, $y(k)$ is the de-noised signal, and n is the length of the signal.

From Table 1, it can be found that whether it is the soft threshold, hard threshold, or a1, a2, a3, the improved method proposed in this paper has shown excellent performance.

4.2 Classification experiment

4.2.1 Data source

This research validates the performance of the proposed algorithm using the UCR dataset [39]. Due to the ICEEMDAN method's requirement for data to reach a certain time series length, to verify the effectiveness of the proposed classification method, as shown in Table 2, we utilize the UCR dataset with time series length greater than 255, totaling 68 datasets, ordered by time series length.

4.2.2 Experimental results comparison

To better compare and validate the effectiveness of classification methods, this research selected the baseline algorithm as the nearest neighbor 1-NN algorithm based on Euclidean distance (ED), denoted as ED. As shown in Table 3, among the algorithms based on the nearest neighbor ED, the proposed financial time series classification method based on ICEEMDAN and wavelet threshold (deED) showed optimal performance 46 times, outperforming ED. Regarding the mean accuracy rate, deED is 0.6407, and ED is 0.6312, indicating that deED also surpasses ED.

5 Application of the classification method in quantitative portfolio investment: numerical experiment

5.1 Classification experiment data

To evaluate the effectiveness of the two-step stock classification prediction method, which integrates a classification technique and a BP neural network, a numerical experiment was conducted. The IndexShares487 dataset was compiled for the purpose of sample classification. As indicated in Tables 4 and 5, this dataset includes indices from four specific industries for training: Food and Beverage, Pharmaceuticals and Biotech, Defense, and Banking. The listed companies in the banking industry, comprising state-owned commercial banks, joint-stock commercial banks, city commercial banks, and rural commercial banks, were selected as the test set. However, the banking sector is considered a narrow-based industry due to the significant impact of industry-specific factors on listed companies.

The period selected for sample classification spanned from January 1, 2020, to December 31, 2021, encompassing two years of daily closing price data. The data for the listed companies in the test set have been adjusted for rights, and companies that have been designated were excluded. Missing values were imputed using the closing price of the preceding trading day. The training set is composed of 51 samples, and the test set includes 28 samples. The time series length for this dataset is 487, with all data obtained

Table 1 SNR, MSE, and NCC values of results obtained after simulation de-noising

De-noising method	Soft threshold wavelet de-noising	ICEEMDAN-soft threshold wavelet de-noising	Hard threshold wavelet de-noising	ICEEMDAN-hard threshold wavelet de-noising	a1 wavelet de-noising	ICEEMDAN-a1 wavelet de-noising	a2 wavelet de-noising	ICEEMDAN-a2 wavelet de-noising	a3 wavelet de-noising	ICEEMDAN-a3 wavelet de-noising
SNR	26.3226	28.3100	26.4494	28.3829	26.4656	28.4038	26.4627	28.4051	26.4523	28.4081
MSE	0.0222	0.0140	0.0216	0.0138	0.0215	0.0137	0.0215	0.0137	0.0215	0.0137
NCC	0.9988	0.9993	0.9989	0.9993	0.9989	0.9993	0.9989	0.9993	0.9989	0.9993

Table 2 Dataset information

No	Dataset	Number of classes	Training set	Test set	Time series length
1	InsectWingbeatSound	11	220	1980	256
2	FiftyWords	50	450	455	270
3	WordSynonyms	25	267	638	270
4	Trace	4	100	100	275
5	ToeSegmentation1	2	40	228	277
6	Coffee	2	28	28	286
7	CricketX	12	390	390	300
8	CricketY	12	390	390	300
9	CricketZ	12	390	390	300
10	FreezerRegularTrain	2	150	2850	301
11	FreezerSmallTrain	2	28	2850	301
12	UWaveGestureLibraryX	8	896	3582	315
13	UWaveGestureLibraryY	8	896	3582	315
14	Lightning7	7	70	73	319
15	ToeSegmentation2	2	36	130	343
16	DiatomSizeReduction	4	16	306	345
17	FaceFour	4	24	88	350
18	Symbols	6	25	995	398
19	Yoga	2	300	3000	426
20	OSULeaf	6	200	242	427
21	Ham	2	109	105	431
22	Meat	3	60	60	448
23	Fish	7	175	175	463
24	Beef	5	30	30	470
25	FordA	2	3601	1320	500
26	FordB	2	3636	810	500
27	ShapeletSim	2	20	180	500
28	BeetleFly	2	20	20	512
29	BirdChicken	2	20	20	512
30	Earthquakes	2	322	139	512
31	Herring	2	64	64	512
32	ShapesAll	60	600	600	512
33	OliveOil	4	30	30	570
34	Car	4	60	60	577
35	InsectEPGRegularTrain	3	62	249	601
36	InsectEPGSmallTrain	3	17	249	601
37	Lightning2	2	60	61	637
38	Computers	2	250	250	720
39	LargeKitchenAppliances	3	375	375	720
40	RefrigerationDevices	3	375	375	720
41	ScreenType	3	375	375	720
42	SmallKitchenAppliances	3	375	375	720
43	NonInvasiveFetalECGThorax1	42	1800	1965	750
44	NonInvasiveFetalECGThorax2	42	1800	1965	750
45	Worms	5	181	77	900
46	WormsTwoClass	2	181	77	900
47	UWaveGestureLibraryAll	8	896	3582	945
48	Mallat	8	55	2345	1024

Table 2 (continued)

No	Dataset	Number of classes	Training set	Test set	Time series length
49	MixedShapesRegularTrain	5	500	2425	1024
50	MixedShapesSmallTrain	5	100	2425	1024
51	Phoneme	39	214	1896	1024
52	StarLightCurves	3	1000	8236	1024
53	Haptics	5	155	308	1092
54	EOGHorizontalSignal	12	362	362	1250
55	EOGVerticalSignal	12	362	362	1250
56	ACSF1	10	100	100	1460
57	SemgHandGenderCh2	2	300	600	1500
58	SemgHandMovementCh2	6	450	450	1500
59	SemgHandSubjectCh2	5	450	450	1500
60	CinCECGTorso	4	40	1380	1639
61	EthanolLevel	2	322	139	512
62	InlineSkate	7	100	550	1882
63	HouseTwenty	2	40	119	2000
64	PigAirwayPressure	52	104	208	2000
65	PigArtPressure	52	104	208	2000
66	PigCVP	52	104	208	2000
67	HandOutlines	2	1000	370	2709
68	Rock	4	20	50	2844

from the RESSET database. The products and services offered by listed companies in the banking industry are relatively homogeneous, and the industry is significantly influenced by regulatory policies. The time series classification method introduced in this paper has been shown to produce a high degree of similarity within the screened investment portfolio for the banking industry, indicating strong interconnectivity.

5.2 Classification experiment results

The classification method described previously was first utilized to designate classification labels, which led to the acquisition of the classification results. As depicted in Table 6, samples labeled with 'Y' in the classification results were chosen to construct the investment portfolio for the experimental group. The control group consisted of all samples from the banking industry (bank), while the experimental group (banksel) included samples marked with 'Y' from the industry's integrated classification results. This same stock price increase/decrease classification prediction method was then applied to both the experimental and control groups, allowing for a comparative analysis of the two groups' performance in stock classification prediction.

5.3 Stock price rise/fall classification prediction experiment data and technical indicators

In this section, we adopt the methodology used by Zhuo Jinwu and Zhou Ying [40], employing 20 technical indicators as presented in Table 8. The sample range for calculating these indicators is the last 100 trading days leading up to December 31, 2022. The classification of each stock on a daily basis is determined by the stock's price increase

Table 3 Classification accuracy rates of algorithms based on nearest neighbor ED

No	Dataset	ED	deED
1	InsectWingbeatSound	0.5616	**0.5717**
2	FiftyWords	0.6308	**0.6462**
3	WordSynonyms	0.6176	**0.6191**
4	Trace	**0.7600**	**0.7600**
5	ToeSegmentation1	0.6798	**0.6842**
6	Coffee	**1.0000**	**1.0000**
7	CricketX	0.5769	**0.6154**
8	CricketY	**0.5667**	0.5641
9	CricketZ	0.5872	**0.6128**
10	FreezerRegularTrain	0.8049	**0.8600**
11	FreezerSmallTrain	0.6758	**0.6789**
12	UWaveGestureLibraryX	0.7393	**0.7409**
13	UWaveGestureLibraryY	0.6616	**0.6714**
14	Lightning7	0.5753	**0.6164**
15	ToeSegmentation2	0.8077	**0.8308**
16	DiatomSizeReduction	**0.9346**	**0.9346**
17	FaceFour	**0.7841**	0.7386
18	Symbols	0.8995	**0.9005**
19	Yoga	**0.8303**	0.8300
20	OSULeaf	0.5207	**0.5248**
21	Ham	**0.6000**	0.5143
22	Meat	**0.9333**	**0.9333**
23	Fish	**0.7829**	0.7486
24	Beef	**0.6667**	**0.6667**
25	FordA	**0.6652**	0.6583
26	FordB	**0.6062**	0.5852
27	ShapeletSim	0.5389	**0.5667**
28	BeetleFly	**0.7500**	**0.7500**
29	BirdChicken	**0.5500**	**0.5500**
30	Earthquakes	**0.7122**	0.6619
31	Herring	0.5156	**0.5469**
32	ShapesAll	**0.7517**	0.7500
33	OliveOil	**0.8667**	0.6667
34	Car	**0.7333**	0.7167
35	InsectEPGRegularTrain	0.6787	**0.6948**
36	InsectEPGSmallTrain	0.6627	**0.6747**
37	Lightning2	0.7541	**0.7705**
38	Computers	0.5760	**0.5840**
39	LargeKitchenAppliances	0.4933	**0.5013**
40	RefrigerationDevices	0.3947	**0.4053**
41	ScreenType	0.3600	**0.3680**
42	SmallKitchenAppliances	**0.3413**	0.3387
43	NonInvasiveFetalECGThorax1	**0.8290**	0.8254
44	NonInvasiveFetalECGThorax2	**0.8799**	0.8692
45	Worms	**0.4545**	0.4286
46	WormsTwoClass	**0.6104**	0.5714
47	UWaveGestureLibraryAll	0.9481	**0.9509**
48	Mallat	0.9143	**0.9147**
49	MixedShapesRegularTrain	0.8973	**0.8990**

Table 3 (continued)

No	Dataset	ED	deED
50	MixedShapesSmallTrain	0.8355	**0.8388**
51	Phoneme	**0.1092**	0.1055
52	StarLightCurves	**0.8488**	0.8481
53	Haptics	**0.3701**	**0.3701**
54	EOGHorizontalSignal	**0.4171**	**0.4171**
55	EOGVerticalSignal	**0.4420**	0.4392
56	ACSF1	**0.5400**	0.5200
57	SemgHandGenderCh2	0.7617	**0.8917**
58	SemgHandMovementCh2	0.3689	**0.6400**
59	SemgHandSubjectCh2	0.4044	**0.8311**
60	CinCECGTorso	0.8971	**0.8986**
61	EthanolLevel	0.2740	**0.2800**
62	InlineSkate	**0.3418**	**0.3418**
63	HouseTwenty	**0.6639**	0.6555
64	PigAirwayPressure	0.0577	**0.0625**
65	PigArtPressure	0.1250	**0.1394**
66	PigCVP	**0.0817**	0.0673
67	HandOutlines	0.8622	**0.8676**
68	Rock	**0.8400**	**0.8400**
Mean accuracy rate		0.6312	0.6407
Number of optimal performances		33	46

Bold values indicate better classification accuracy than another algorithm

Table 4 IndexShares487 training dataset

Category	Number of Samples	Training set security codes	Category codes
Food and beverage index	13	000807 000815 399,396 801,120 930,653 930,682 930,696 CN6033 H30020 H30176 H30177 H30192 H30205	1
Pharmaceuticals and biotech index	22	000037 000075 000109 000121 000808 000814 000857 000913 000933 000978 000991 399,275 399,280 399,386 399,394 399,441 399,618 399,647 399,674 399,676 930,726 930,791	2
Defense index	10	399,368 399,813 399,959 399,967 399,973 801,740 930,617 930,875 931,066 H50036	3
Banking index	6	000134 000951 399,431 399,986 H30022 L11641	4
Total	51		

Data sourced from the RESSET database (www.resset.cn)

Table 5 IndexShares487 test dataset

Category	Number of samples	Test set security codes	Category codes
Bank	28	000001 002142 002807 002839 002936 600,000 600,015 600,016 600,036 600,908 600,919 600,926 601,009 601,128 601,166 601,169 601,229 601,288 601,328 601,398 601,577 601,818 601,838 601,939 601,988 601,997 601,998 603,323	4

Table 6 Classification results for samples in the banking industry

Sample	Original Category	deED	Ensemble
000001	4	1	
002142	4	1	
002807	4	4	Y
002839	4	3	
002936	4	4	Y
600000	4	4	Y
600015	4	4	Y
600016	4	4	Y
600036	4	1	
600908	4	3	
600919	4	4	Y
600926	4	1	
601009	4	1	
601128	4	4	Y
601166	4	4	Y
601169	4	4	Y
601229	4	4	Y
601288	4	4	Y
601328	4	4	Y
601398	4	4	Y
601577	4	4	Y
601818	4	4	Y
601838	4	1	
601939	4	4	Y
601988	4	4	Y
601997	4	4	Y
601998	4	4	Y
603323	4	4	Y

Table 7 Number of training samples and prediction quantities for each portfolio in the banking industry

Portfolio	Bank	Banksel
Number of training samples	2576	1840
Number of prediction samples	140	100

over the next 1-day and 3 days periods. A stock is categorized as 'good' if its price rises by 2% the next day and by 3% over the next three days. Conversely, if the stock price declines on the next day and also over the next three days, it is categorized as 'bad'. All other stocks are designated as 'average'. Since December 31, 2022, falls on a weekend when the markets are closed, there are no stock category labels for the last three trading days of the year, from December 28 to December 30, 2022. Consequently, neither the training nor the prediction samples include data from these days. As indicated in Table 7, the final five trading days out of the 27 are set aside as the prediction samples, with the rest allocated as training samples.

Table 8 Technical indicators selected for each portfolio

No	Indicator	Bank	Banksel
1	Daily increase	False	False
2	2-day increase	False	False
3	5-day increase	False	False
4	10-day increase	False	False
5	30-day increase	False	False
6	10-day rise/Fall ratio	False	False
7	10-day relative strength indicator (ADR)	False	False
8	Daily K-line value	False	False
9	3-day K-line value	False	False
10	6-day K-line value	False	False
11	6-day bias rate (BIAS)	False	False
12	10-day bias rate (BIAS)	False	False
13	9-day RSV	True	True
14	30-day RSV	False	False
15	90-day RSV	False	False
16	Daily OBV quantification	False	False
17	5-day OBV quantification	False	False
18	10-day OBV quantification	False	False
19	30-day OBV quantification	False	False
20	60-day OBV quantification	False	False
Number of selected indicators		1	1

Table 8 illustrates the filtration of the 20 technical indicators for both the control group (bank) and the experimental group (banksel). Figures 3 and 4 show the degree of linear correlation between the technical indicators and stock categories for the bank and banksel groups, respectively. A low correlation between a technical indicator and stock category could negatively impact its effectiveness as a predictive parameter in the model. Hence, a threshold is usually established for selecting technical indicators. In this paper, 0.2 is the chosen threshold; technical indicators with an absolute value of the correlation coefficient greater than 0.2 with the stock category are selected as inputs for the model. Table 8 lists the technical indicators chosen for each portfolio, where TRUE signifies that an indicator has been selected, and FALSE indicates that the indicator's correlation is below 0.2 and is, therefore, not selected.

5.4 Stock rise/fall classification prediction experimental result comparison

For selecting the number of hidden layer nodes, we reference the empirical formula [41]: $2^X > N$, where X represents the count of nodes in the hidden layer, and N is the number of samples. Table 9 shows the correct prediction rates for the control group (bank) and the experimental group (banksel) with different numbers of hidden layer nodes. The experimental group (banksel) consistently outperforms the control group (bank) in terms of classification prediction accuracy across various node counts, achieving an average increase in accuracy of 7%. Although the two-step stock portfolio rise/fall classification prediction method does not directly result in an investment strategy, its superior performance in classifying stock price movements can inform an investment strategy that involves buying stocks predicted as 'good' and selling those predicted as 'bad.'

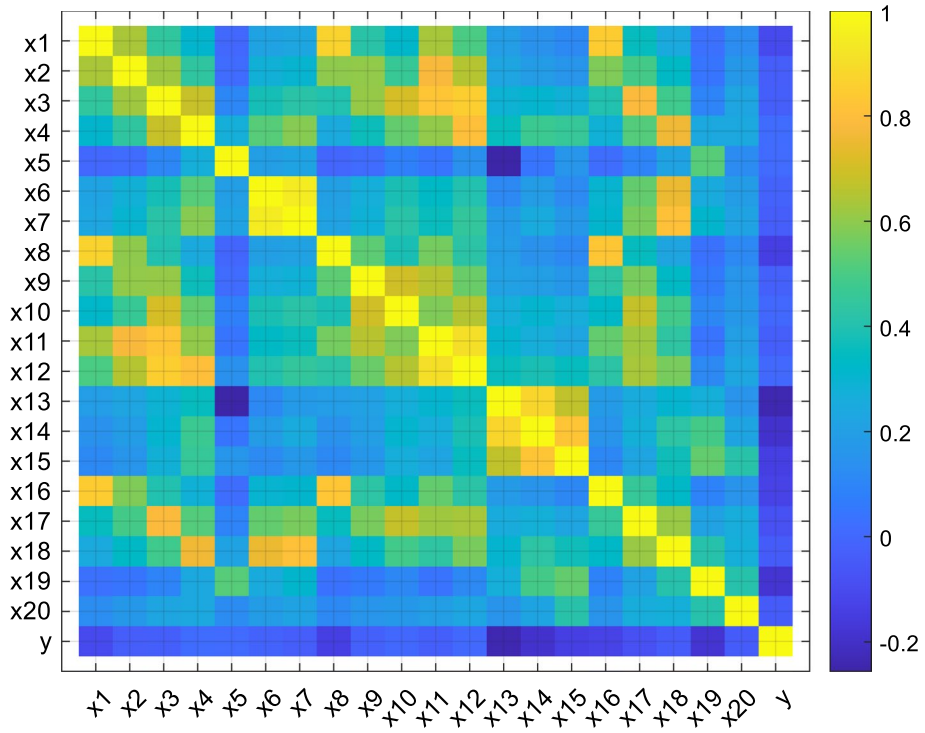

Fig. 3 Correlation between technical indicators and stock categories in the control group (bank)

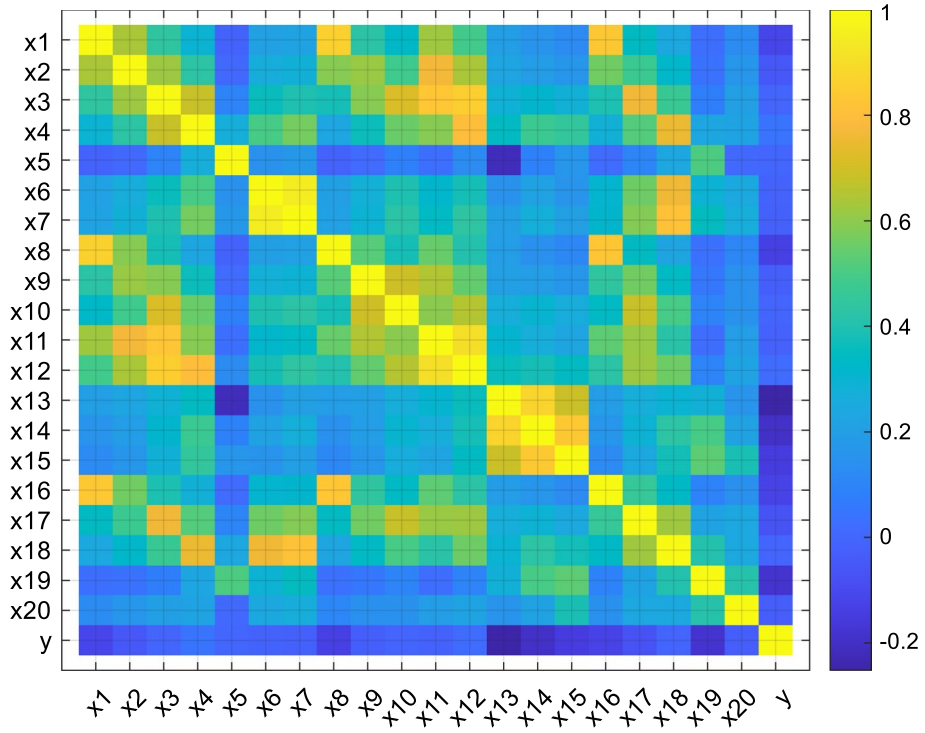

Fig. 4 Correlation between technical indicators and stock categories in the experimental group (banksel)

Table 9 Correct prediction rate of control group (bank) and experimental group (banksel) at different hidden layer node counts

Node count	12	14	16	18	20	Average
Bank	0.77	0.77	0.75	0.77	0.77	0.77
Banksel	0.84	0.84	0.84	0.83	0.84	0.84

6 Conclusion

This paper addresses the significant noise present in financial time series by introducing a time series classification method that combines improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) and wavelet threshold for noise reduction. Initially, the method employs ICEEMDAN to decompose the time series into a set of IMF components and a residue. A noise detection method proposed in this paper is then used to categorize the IMF components and residue into components with and without noise. Subsequently, noise-inclusive components are processed through wavelet threshold de-noising, resulting in a non-noise sequence. This sequence, merged with the noise-reduced components, forms the final de-noised output, which is then classified using the 1-NN nearest neighbor algorithm. This approach not only enhances the benchmark method's performance, but also has significant application potential. For the first time, a combination of the t test and unit root test is introduced to detect noise in time series components, offering new tools for the analysis of time series data. Through de-noising simulation experiments, the method proposed in this paper demonstrates superior performance over the benchmark method across five different thresholds. In time series classification experiments with 68 UCR datasets, the proposed method outperforms the benchmark algorithm.

Additionally, this paper presents a two-step stock portfolio classification prediction method that utilizes both a time series classification method and a BP neural network. Initially, the time series classification method screens stocks within a specific industry, resulting in a selected stock portfolio. Subsequently, the BP neural network algorithm predicts the directional movement of stock prices within this portfolio. Comparative experiments on quantitative portfolio investment show that the two-step classification prediction method offers stable and improved performance over the direct prediction method with various configurations of hidden layer nodes. The practical application confirms that the time series classification method proposed improves the predictive performance of existing methods and has promising prospects in quantitative portfolio investment.

The success of this approach lies in its ability to construct an investment portfolio closely aligned with the learning objective through empirical learning. This not only strategically leverages investors' experiential knowledge, but also promotes high similarity within the selected portfolio, enhancing the statistical effectiveness of its technical indicators.

In practical applications, to manage the impact of sudden market events on stock price behavior, a threshold should be set. Monitoring for unexpected events that could shift stock price patterns is crucial, with strategies adjusted when win rates or returns

dip below this threshold. Rolling training of stock classification attributes and strategy parameters should be conducted to improve the adaptability of investment strategies.

To mitigate systemic risk through diversification, practical implementation may first involve categorizing stocks in the market using clustering or industry benchmarks. By setting an upper limit on the investment ratio for each stock category and allocating funds accordingly, diversification goals can be achieved. Finally, the method proposed in this paper can be applied to select an investment portfolio within each category, with each category's allocated funds used for quantitative investment. This approach not only achieves diversification, but also aims to enhance the investment performance within each category.

Author contributions

BL involved in conceptualization (lead); data curation (lead); formal analysis (lead); investigation (equal); methodology (lead); project administration (equal); resources (lead); software (lead); visualization (equal); writing—original draft (equal); and writing—review and editing (equal). HC involved in funding acquisition (lead); project administration (equal); visualization (equal); writing—original draft (equal); and writing—review and editing (equal).

Funding

This work is supported by Youth Fund for Humanities and Social Sciences Research of the Ministry of Education (21YJC910005), Key Research Projects of Anhui Humanities and Social Sciences (SK2021A0544, 2023AH051519), and Key Scientific Research Projects of Huainan Normal University (2023XJZD002).

Availability of data and materials

Related data were available from the UCR archives provided by Dau et al. (2019) (https://www.cs.ucr.edu/~eamonn/time_series_data_2018/) and RESSET database (www.resset.cn).

Declarations

Competing interests

Authors have no conflict of interest relevant to this article.

Received: 21 November 2023 Accepted: 15 January 2024
Published online: 26 January 2024

References

1. P. Schäfer, The BOSS is concerned with time series classification in the presence of noise. Data Min. Knowl. Disc. **29**(6), 1505–1530 (2015)
2. X. Liu, H. Zhang, Y.M. Cheung, X. You, Y.Y. Tang, Efficient single image dehazing and denoising: an efficient multi-scale correlated wavelet approach. Comput. Vis. Image Underst. **162**, 23–33 (2017)
3. R.C. Guido, F. Pedroso, A. Furlan, R.C. Contreras, L.G. Caobianco, J.S. Neto, CWT × DWT × DTWT × SDTWT: clarifying terminologies and roles of different types of wavelet transforms. Int. J. Wavel. Multiresolut. Inf. Process. **18**(06), 2030001 (2020)
4. S.G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans. Pattern Anal. Mach. Intell. **11**(7), 674–693 (1989)
5. X. Zheng, Y.Y. Tang, J. Zhou, A framework of adaptive multiscale wavelet decomposition for signals on undirected graphs. IEEE Trans. Signal Process. **67**(7), 1696–1711 (2019)
6. E. Guariglia, R.C. Guido. Chebyshev wavelet analysis. J. Funct. Spaces **2022**, 5542054 (2022)
7. L. Yang, H. Su, C. Zhong et al., Hyperspectral image classification using wavelet transform-based smooth ordering. Int. J. Wavel. Multiresolut. Inf. Process. **17**(06), 1950050 (2019)
8. T. Stadnitski, Measuring fractality. Front. Physiol. **3**, 127 (2012)
9. B. Hoop, C.K. Peng, Fluctuations and fractal noise in biological membranes. J. Membr. Biol. **177**, 177–185 (2000)
10. F. Klingenhöfer, M. Zähle, Ordinary differential equations with fractal noise. Proc. Am. Math. Soc. **127**(4), 1021–1028 (1999)
11. E. Guariglia, Entropy and fractal antennas. Entropy **18**(3), 84 (2016)
12. E. Guariglia, Primality, fractality, and image analysis. Entropy **21**(3), 304 (2019)
13. E. Guariglia, S. Silvestrov, *Fractional-Wavelet Analysis of Positive definite Distributions and Wavelets on D'(C). Engineering mathematics II: Algebraic, Stochastic and Analysis Structures for Networks, Data Classification and Optimization* (Springer, New York, 2016), pp.337–353
14. M. Ghazel, G.H. Freeman, E.R. Vrscay, Fractal-wavelet image denoising revisited. IEEE Trans. Image Process. **15**(9), 2669–2675 (2006)
15. P. Afzal, K. Ahmadi, K. Rahbar, Application of fractal-wavelet analysis for separation of geochemical anomalies. J. Afr. Earth Sc. **128**, 27–36 (2017)

16. P. Podsiadlo, G.W. Stachowiak, Fractal-wavelet based classification of tribological surface. Wear **254**(11), 1189–1198 (2003)
17. R.P. Shao, J.M. Cao, Y.L. Li, Gear fault pattern identification and diagnosis using time-frequency analysis and wavelet threshold de-noising based on EMD. J. Vib. Shock **31**(08), 96–106 (2012)
18. Y. Gan, L. Sui, J. Wu et al., An EMD threshold de-noising method for inertial sensors. Measurement **49**, 34–41 (2014)
19. S. Shukla, S. Mishra, B. Singh, Power quality event classification under noisy conditions using EMD-based de-noising techniques. IEEE Trans. Ind. Inf. **10**(2), 1044–1054 (2013)
20. Y. Xu, M. Luo, T. Li et al., ECG signal de-noising and baseline wander correction based on CEEMDAN and wavelet threshold. Sensors **17**(12), 2754 (2017)
21. L. Feng, J. Li, C. Li et al., A blind source separation method using denoising strategy based on ICEEMDAN and improved wavelet threshold. Math. Probl. Eng. **2022**, 3035700 (2022)
22. M. Ding, Z. Shi, B. Du et al., A signal de-noising method for a MEMS gyroscope based on improved VMD-WTD. Meas. Sci. Technol. **32**(9), 095112 (2021)
23. H. Wang, R. Pappadà, F. Durante, E. Foscolo, *A Portfolio Diversification Strategy via Tail Dependence Clustering Soft Methods for Data Science* (Springer, New York, 2017), pp.511–518
24. S. Gupta, G. Bandyopadhyay, S. Biswas et al., An integrated framework for classification and selection of stocks for portfolio construction: evidence from NSE, India. Decis. Mak. Appl. Manag. Eng. **6**, 1–29 (2022)
25. M.A. Colominas, G. Schlotthauer, M.E. Torres, Improved complete ensemble EMD: a suitable tool for biomedical signal processing. Biomed. Signal Process. Control **14**, 19–29 (2014)
26. D.L. Donoho, J.M. Johnstone, Ideal spatial adaptation by wavelet shrinkage. Biometrika **81**(3), 425–455 (1994)
27. R.C. Guido, Wavelets behind the scenes: practical aspects, insights, and perspectives. Phys. Rep. **985**, 1–23 (2022)
28. R.C. Guido, Effectively interpreting discrete wavelet transformed signals. IEEE Signal Process. Mag. **34**(3), 89–100 (2017)
29. R.C. Guido, Practical and useful tips on discrete wavelet transforms. IEEE Signal Process. Mag. **32**(3), 162–166 (2015)
30. D.L. Donoho, De-noising by soft-thresholding. IEEE Trans. Inf. Theory **41**(3), 613–627 (1995)
31. T.F. Sanam, C. Shahnaz, Noisy speech enhancement based on an adaptive threshold and a modified hard thresholding function in wavelet packet domain. Digit. Signal Process. **23**(3), 941–951 (2013)
32. W.L. Sun, C. Wang, Power signal denoising based on improved soft threshold wavelet packet network. J. Nav. Univ. Eng. **31**(04), 79–82 (2019)
33. C. Liu, L.X. Ma, P. Jinfeng, Ma. Zhen, PD signal denoising based on VMD and improved wavelet threshold. Modern Electron. Technol. **44**(21), 45–50 (2021)
34. P.L. Zhang, X.Z. Li, S.H. Cui, An improved wavelet threshold-CEEMDAN algorithm for ECG signal denoising. Comput. Eng. Sci. **42**(11), 2067–2072 (2020)
35. J.C. Gower, Properties of euclidean and non-euclidean distance matrices. Linear Algebra Appl. **67**, 81–97 (1985)
36. H. Li, *Statistical Learning Methods* (Tsinghua University Press, Beijing, 2012), pp.37–38
37. D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors. Nature **323**(6088), 533–536 (1986)
38. P. Kim, *Deep Learning for Beginners: With MATLAB Examples* (Verlag Nicht Ermittelbar, Freiburg im Breisgau, 2016), pp.23–54
39. H.A. Dau, A. Bagnall, K. Kamgar et al., The UCR time series archive. IEEE/CAA J. Autom. Sin. **6**(6), 1293–1305 (2019)
40. J.W. Zhuo, Y. Zhou, *Quantitative Investment: Data Mining Technology and Practice* (Electronic Industry Press, Delaware, 2015), pp.366–380
41. S.E. Yang, L. Huang, Financial crisis warning model based on BP neural network. Syst. Eng. Theory Pract. **01**, 12–18+26 (2005)

Publisher's Note