Faculty of Natural,
Mathematical Engineering Sciences
Department of Informatics

7CCSMPRJ

Individual Project Submission 2021/22

Name:	Tiantian Zhang
Student Number:	21021501
Degree Programme:	MSc Computational Finance
Project Title:	Compare denoising methods based on EMD, CEEMDAN and VMD
Supervisor:	Dr Bart de Keijzer
Word Count:	9466

Plagiarism Statement

All work submitted as part of the requirements for any examination or assessment must be expressed in your own words and incorporates your own ideas and judgements. Plagiarism is the taking and using of another person's thoughts, words, judgements, ideas, etc., as your own without any indication that they are those of another person.

Plagiarism is a serious examination offence. An allegation of plagiarism can result in action being taken under the *B3 Misconduct Regulations*.

I acknowledge that I have read and understood the above information and that the work I am submitting is my own.

Signature: Tiantian Zhang **Date:** August 8, 2022

Department of Informatics
King's College London
WC2R 2LS London
United Kingdom

Compare denoising methods based on EMD, CEEMDAN and VMD

Tiantian Zhang
Student Number: 21021501
Course: MSc Computational Finance

Supervisor: Dr Bart de Keijzer

Thesis submitted as part of the requirements for the award of the MSc in Computational Finance.
7CCSMPRJ - MSc Individual Project - 2022

Abstract

Financial time series are nonlinear, nonstationary, and contaminated by noise. Since many research and models are sensitive to noise, it is possible to reach inaccurate findings if the original noisy time series are used directly. Consequently, this study undertakes extensive research on denoising. Empirical Mode Decomposition (EMD), Complete Ensemble Empirical Code Decomposition with Adaptive Noise (CEEMDAN), and Variational Mode Decomposition (VMD) can be used to decompose nonlinear and non-stationary time series into a set of modes. Thus, this research employs these decomposition techniques to exclude the modes which contain noise and add the remaining modes to reconstruct data. Detrended Fluctuation Analysis (DFA) is implemented to determine which modes constitute noise. Permutation and combination of the three fundamental decomposition methods yield four noise reduction techniques: EMD, CEEMDAN, VMD-DFA, and VMD-EMD.

In order to test the four noise reduction methods, we examine and compare them in the following three aspects: simulation experiments, prediction of low-frequency data and classification of high-frequency data. In general, EMD has the best performance, which not only improves the signal-to-noise ratio (SNR) in simulation trials but also increases the accuracy of prediction and classification. And it is robust and efficient in signal extraction. The overall performance of CEEMDAN is superior to that of VMD-DFA and VMD-EMD. In the prediction and classification models, the denoising effect of VMD-DFA and VMD-EMD-DFA is inconsistent, sometimes outperforming the benchmark model and other times underperforming it.

Contents

1 Introduction **1**

1.1 Background . . . 1
1.2 Motivation . . . 2
1.3 Aims and Objectives . . . 3
1.4 Report Structure . . . 4

2 Literature Review **5**

3 Background Theories **8**

3.1 Decomposition methods . . . 8
3.1.1 EMD . . . 8
3.1.2 CEEMDAN . . . 10
3.1.3 VMD . . . 11
3.2 DFA . . . 14
3.3 XGBoost . . . 16

4 Model Design **18**

4.1 EMD . . . 18
4.2 CEEMDAN . . . 19
4.3 VMD-DFA . . . 19
4.4 VMD-EMD-DFA . . . 21

5 Main Results **23**

5.1 Simulation . . . 23
5.1.1 Trend-stationary time series . . . 23
5.1.2 Non-stationary time series . . . 27
5.2 Prediction Tests . . . 29
5.2.1 Data . . . 29
5.2.2 Data preprocessing and prediction framework . . . 31
5.2.3 Results . . . 34
5.3 Classification tests . . . 38
5.3.1 Data . . . 38
5.3.2 Data preprocessing and classification framework . . . 39
5.3.3 Results . . . 41

6 Conclusion **44**
6.1 Future work 44

References **47**

A Appendix **54**
A.1 ACF and PACF diagrams of five financial assets 54
A.2 Comparison between the denoised data and the pure data 55

List of Figures

1 The structure of the EMD algorithm . . . 9
2 The structure of the CEEMDAN algorithm . . . 12
3 The structure of the VMD algorithm . . . 15
4 The structure of the VMD-DFA algorithm . . . 21
5 The pure signal . . . 23
6 Analog signal with noise . . . 25
7 denoised signal and pure signal . . . 26
8 Random walk with log-returns . . . 28
9 The denoised data of four methods and the pure data (SNR of 5 dB) . . . 28
10 The closing prices . . . 30
11 The the ACF and PACF of HSI . . . 31
12 Prediction with expanding windows . . . 34
13 The forecasting results . . . 35
14 The price series of Bitcoin one-minute . . . 38
15 The framework of classification . . . 41
16 F1 score with different thresholds . . . 42
17 The box plot of R_t and $\hat{R}_t$ ^{2} . . . 43
18 The the ACF and PACF of GDAXI . . . 54
19 The the ACF and PACF of Crude oil . . . 54
20 The the ACF and PACF of Apple . . . 55
21 The the ACF and PACF of Bitcoin . . . 55
22 The denoised data of four methods and the pure data (SNR of 1 dB) . . . 55
23 The denoised data of four methods and the pure data (SNR of 8 dB) . . . 56
24 The denoised data of four methods and the pure data (SNR of 10 dB) . . . 56

List of Tables

1 Comparison of SNR obtained by using different denoising methods . . . 24
2 Comparison of MSE obtained by using different denoising methods . . . 27
3 Comparison of SNR obtained by using different denoising methods . . . 29
4 Comparison of RMSE obtained by using different denoising methods . . . 29
5 The statstical tests for five underlying assets . . . 32
6 Predictive performance evaluation of different methods . . . 36
7 The cost time of different denoising methods . . . 37

8 The statistical tests for Bitcoin 1 min 39
9 The cost time of different denoising methods 43

Acknowledgements

I would like to thank my supervisor, Bart. In every group meeting, he listened to my project carefully and gave helpful suggestions. Also, I'd like to thank my tutors, Vardan, Vitali and Lingbo in Turin Tech. They provided me with a lot of new ideas on model design. Finally, I would like to thank those friends who helped me. Studying codes with them is the happiest time since I came to Britain. Although the process of failing to improve the model performance and starting over each time is painful, it becomes meaningful because of their company.

1 Introduction

1.1 Background

Financial time series exhibit significant levels of noise, nonlinearity, and complexity. A time series is a collection of data gathered at regular intervals (such as minutes, hours, daily, weekly, etc.). Forecasting time series is possible by evaluating historical data and creating forecasts based on this analysis. Financial time series have strong temporality and autocorrelation. Financial time series data describe, in the time dimension, the ongoing happenings in the financial market. Due to the irreversibility of time, a huge quantity of information is concealed in time series data, which is the objective people are continually pursuing, particularly in the financial industry. The nature of financial time series is complicated, highly noisy, chaotic, nonlinear, dynamic, and non-parametric [1]. Take the stock market as an example. The stock market is a nonlinear complex system controlled by a range of economic and social forces. Its price fluctuations exhibit non-stationary, nonlinear, and high-noise complex properties. In addition, there are different patterns in financial time series data, such as trend, seasonality, cycle and irregularity.

Economic noise is a concept first proposed by Fischer Black in 1986 [2]. In his theory, noise refers to information that is irrelevant to the actual value, false or distorted. Fama et al.[3] argue that in an efficient market, if all relevant information about the underlying asset can be fully and sufficiently reflected in the price, then the price of the financial asset is equal to or close to its true worth. In reality, though, things function differently. Information is intricate and costly to obtain. In addition, the capacity to collect and interpret data differs considerably amongst traders. Therefore, the knowledge held by various dealers is insufficient and unbalanced. Due to these factors, there is a deviation between the prices of financial assets established on financial markets and their values; this deviation is noise.

Noise is inevitable and will distort the original signal in the market. Black [2] pointed out that even if people receive noise, they will interpret it as information and act accordingly. As a result, even though market liquidity has improved, the stock market cannot be maintained at an efficient level. De Long et al. [4] proposed a model of investor trading behavior that divides investors into rational and noise traders. A noise trader is an investor with no private information and behaves irrationally. Noise traders get a lot of information from technical analysts, economic advisers, and stockbrokers. They erroneously assume they can foresee asset values and select portfolios based on this false belief. Rational traders leverage the irrational beliefs of noise traders to do the opposite action. They sell when noise traders bid up the price. When noise traders drive prices

down, investors purchase equities. Some financial market oddities, such as excessive stock price volatility, can be explained by this theory. In addition, they discovered that noise is unavoidable in financial markets and noted that the subjective consciousness of noise traders is an inherent risk affecting price stability in financial markets.

Compared with other markets, the signal-to-noise ratio (SNR) of financial data is lower. Economic asset prices are easily affected by many factors, such as news, wars, policies, upstream and downstream industries, and financial conditions. There are not only long-term effects but also short-term effects. Often modelling and analysis are based on signals, and the presence of noise can degrade performance. Therefore, lower SNR means less predictability in the financial system.

There is considerable noise in the financial markets. Noise is so pervasive in economic activity that it is difficult to distinguish between noise and information. So it is tough to denoise financial data. First, noise's definition is uncertain. Due to the lack of clean financial data, it is impossible to assess the impact of noise reduction directly. Additionally, it is challenging to tell noise from trends. It is difficult for investors to determine whether price increases are due to short-term fluctuations caused by noise or longer-term trends. Thirdly, the quantity of data is modest. Despite the fact that tick data can also reach many terabytes, it is more than two orders of magnitude less than industry data. In addition, the majority of sources of financial data only provide information from the last 100 years. With the increase in model complexity, the relative scarcity of data will affect the effect of the model.

1.2 Motivation

When there is a great deal of noise in the financial market, the forecast effect will not be good if the unprocessed data is directly used for modelling. Brown and Gregory [5] discovered that irrational investors acting in concert in response to a noisy signal might affect asset values and increase volatility. This causes a deviation between the market value and the actual worth. In financial analysis, it is essential to select the proper instruments for repairing distorted data signals in order to eliminate the influence of noise and prevent inaccurate conclusions.

As a result of the development of computational storage technologies, people are able to get information on more minor time scales, which is the foundation for the growth of quantitative finance. The time interval at which data are collected is closely correlated with the availability of information. The smaller the gaps, the more information is available; conversely, the longer the intervals, the more information is lost. Thus, an

increasing number of investors are attempting to profit from the information contained in high-frequency financial data.

With the rapid development of high-frequency finance, noise reduction has become a demand. In the financial market, information continuously affects the price movement of the stock market. In various financial markets, a particular unit of measurement is taken as the basic indicator, and the multiple values of the basic indicator are used as the measurement standard for various price changes, resulting in a discrete variable distribution of recorded transaction prices. The collecting of high-frequency data or even ultra-high-frequency data is based on each transaction that occurs within a single day. In addition, there are some artificial prescribed factors in the stock market and the discrete characteristics of the price itself, making it difficult for the continuity of the continuous trading variables to emerge. The discrete acquisition of data will inevitably lead to the loss of information to different degrees. Therefore, the higher the frequency of data collection, the lower the degree of information loss. But as the observation frequency increases, the SNR will further decrease. What's more, high-frequency data presents a thick-tailed distribution [6] and has different degrees of jumps [7]. A statistical analysis and data mining will draw incorrect conclusions due to these characteristics if the data is not denoised [8]. For example, ask-bid price bounce is an example of micro-noise in tick data. This bounce will cause significant distortions to the parameter estimates of high-frequency finance, especially volatility estimates. Therefore, it is necessary to remove the noise in data to help model and get a reliable and significant result, improving related work performance like risk control and returns of investment strategies.

1.3 Aims and Objectives

The main aims of the project are to use noise reduction techniques to remove noise from data and compare the effectiveness of various noise reduction methods in financial markets. To achieve these aims, we propose the following objectives :

- Use four denoising methods, EMD, CEEMDAN, VMD-EMD and VMD-DFA, to remove noise in low-frequency and high-frequency financial data to obtain reconstructed financial data.
- Evaluate whether the denoised financial data can improve the performance of subsequent classification and prediction models
- Compare the model performance with different denoising ways to compare the effect of noise reduction.

1.4 Report Structure

This paper focuses primarily on four denoising techniques for reducing noise in financial data with varying frequencies. Then the performance of those denoising techniques is evaluated and compared directly or indirectly.

The next chapter mainly discusses some research results, related background, research significance and research status of the denoising methods.

Chapter 3: Introduce the theoretical knowledge of the basic signal decomposition technique, EMD, CEEMDAN and VMD. DFA, which is used to detect whether components are noise, is also introduced. Lastly, we cover XGBoost, which is utilised as a prediction and classification model and provide the corresponding mathematical models.

Chapter 4: Provide the workflow for four denoising techniques. The process consists of using EMD, CEEMDAN and VMD to decompose the signal to obtain a series of modes and reconstructing the signal through DFA and the statistical properties of the modes.

Chapter 5: To determine if these four noise reduction techniques are useful in financial markets, we assess their performance in three areas: simulation experiment, prediction, and classification, and give relevant results.

Chapter 6: Conclude the whole project and discuss the future work.

2 Literature Review

This chapter mainly discusses the mainstream noise reduction methods, including the principle, development process, and advantages and disadvantages of a variety of noise reduction methods.

Although the traditional noise reduction methods Wiener filter and Kalman filter are excellent models [9], the Wiener filter is unable to deal with non-stationary data, and the Kalman filter needs to establish accurate functional relationships. Therefore, these two methods are not suitable for highly volatile and chaotic financial high-frequency data. Some researchers began to concentrate on the Wavelet Transformation, which can be applied widely. Wavelet Transformation is derived from Fourier transformation, and the theory points out that complex graphics can be decomposed into simple sine waves, square waves and sawtooth waves. Wavelet Transformation has been shown to be very effective in dealing with singularity detection and processing [10] [11] [12]. Some scholars use the Wavelet Transformation for denoising and reconstructing low-frequency financial data [13] [14] [15]. Due to the irregularities and roughness of economic data, the higher the sampling frequency of financial data, the greater the realised volatility, which is indicative of microstructure noise [16] [17]. Numerous academics have proposed innovative noise reduction techniques based on Wavelet Transformation to address this issue. Sun and Meinl [18] developed the algorithm for the local linear scaling approximation. Fan and Wang [19] used new wavelet methods to remove the jump in high-frequency data to estimate the integrated volatility accurately. However, Wavelet Transformation has the disadvantage that it needs to set the wavelet basis artificially, and the wrong choice of the function will lead to poor performance.

Huang et al. [20] invented and used an Empirical Mode Decomposition (EMD) approach to nonlinear and nonstationary signal sequences in 1998. They also introduced Hilbert Spectrum Analysis (HAS) called Hilbert-Huang Transform (HHT). EMD can decompose the data into several intrinsic mode functions (IMFs) in an adaptable manner without additional settings or user participation. It is appropriate for nonlinear or nonstationary signal sequence analysis. Due to these characteristics, EMD is employed in various fields. Veeraiyan et al. [21] combined the frequency domain and EMD, and the proposed frequency domain thresholding approach worked well for different wind noises. Li et al. [22] suggested an HHT-based denoising method that removes noise based on its energy contribution, enhancing the accuracy of the gold closing price forecast. Using the EMD, Nava et al. [23] studied the scaling features of 22 different stock market indices. They found that the scaling properties of developed markets are closer to the Brownian

motion. Besides, EMD can also be applied in seismic signals [24], medicine [25] [26], voiced speech signals [27], and fault detection [28].

EMD-based approaches have some limitations. EMD lacks solid mathematical theory. A firm theoretical foundation for EMD is primarily limited by its implementation-focused definition [29]. EMD is nonlinear due to the interpolation algorithm used to create the envelope and the stopping criteria applied to the iterations. The end effect is one of the most significant downsides, as it causes the signal's energy to be overestimated and the production of faulty components at the ends of the signal [30]. Another important drawback is the influence of mode mixing. It means that in the same IMFs, there are multiple signals with a wide-scale distribution, and in other IMFs, there are signals with a comparable size. The essence of the mode mixing is caused by the local extreme value jumping many times in a brief time interval in the process of decomposition.

To solve mode mixing, Wu and Huang [31] first proposed Ensemble Empirical Mode Decomposition (EEMD). White noise is added to the signals, and related time scale signals can be automatically separated to their respective reference scales. By using the white noise characteristic, the sub-signals uncorrelation degree is amplified so that two initially inseparable modes can be extracted. Gaci [32] proved the effectiveness of the EEMD-based denoising technique for seismic signals. However, it also created new problems. If the white noise sequences are all positive, then all the modes obtained by adding white noise do not add up to the original series. The amount of additional white noise is proportional to the number of decomposed modes, which increases the reconstruction error.

Yeh et al. [33] took EEMD a step further. To overcome the significant reconstruction error and poor decomposition completeness in EEMD, they introduced Complementary Ensemble Empirical Mode Decomposition (CEEMD), which enhances processing efficiency compared to EEMD. By adding positive and negative white noise sequences, the white noise can cancel each other in the decomposition process and reduce the reconstruction error.

Torres et al. [34] proposed Complete Ensemble Empirical Code Decomposition with Adaptive Noise (CEEMDAN) as an advanced version of EEMD. EEMD decompose several signals after adding white noise and then directly calculate the mean value among the corresponding IMFs. However, the CEEMDAN method adds white noise to the residual signals after calculating the specific order IMFs. Then calculates the mean value of the IMFs, and iterates successively. Compared to CEEMD, this method avoids the problem that the IMFs are different in number, which makes it challenging to align and set the average value.

CEEMDAN is widely used in a variety of fields, including hydraulic engineering [35], lane-level traffic flow [36], and daily peak load forecasting [37]. Moreover, it is combined with the long short-term memory (LSTM) to forecast financial time series [38] [39] [40]. Zhou et al. [41] forecast crude oil prices based on CEEMDAN and XGBoost.

Variational Mode Decomposition (VMD) is an adaptive signal processing method proposed by Dragomiretskiy and Zosso [42]. The VMD is a modification and improvement of the EMD. It searches iteratively for the optimal solution for variational modes and updates the function and centre frequency of each mode. Using the quadratic penalty function and the Lagrange multiplier, the constrained problem is converted into an unconstrained problem, which is then solved using the alternating direction multiplier approach. The modes of signal decomposition are obtained through iterative updating. The decomposed modes are composed of the modes containing the primary signal and the modes having the noise. It is possible to achieve denoising by rebuilding the mode containing the dominant signal.

VMD has some pros and cons. On the good side, VMD has a solid theoretical foundation based on the variational theory of functional analysis in mathematics. Moreover, VMD is resilient with respect to sample number and noise size. However, there are still some issues with VMD. The long-term mode spectrum may fluctuate substantially over time if the signal is long. And there may be global bandwidth overlap. Additionally, VMD is limited by boundary effects and burst signals and using VMD requires a predefined number of modes k.

When EMD cannot effectively separate data with similar frequencies, VMD can be used as an alternative to EMD to predict financial time series in combination with other models. The prediction accuracy of the hybrid model combined with VMD is generally higher than that of the single model [43] [44]. VMD can be used for signal decomposition. For instance, obtain some band-limited intrinsic mode functions (BLIMFs) through VMD and selectively sum BLIMFs to reconstruct data [45]. VMD can also be used to extract signal features [46], such as extracting components according to sequence entropy, sample entropy [47], energy entropy[48], correlation, etc.

In addition to the decomposition approaches outlined previously, many researchers also attempt to remove noise using deep learning models. For instance, Ghose et al. [49] used the convolutional neural network (CNN) to denoise photos. Lu et al. [50] successfully implemented the deep autoencoder (DAE) in the acoustic field to minimise voice noise. Yan et al. [51] proposed a new method based on deep learning that offers a novel solution for denoising in the optical area.

3 Background Theories

The background theories are organised into three sections that cover the project's essential technologies. The first section provides the mathematical formulas and flowcharts of three decomposition methods, EMD, CEEMDAN, and VMD. DFA is introduced because it is required to determine what constitutes noise and pick the number of modes k in VMD. The second section presents the theoretical foundations of DFA. The third section gives an overview of XGBoost, which is used as the prediction and classification model.

3.1 Decomposition methods

3.1.1 EMD

EMD is a time-frequency domain signal processing method that can process non-stationary and nonlinear data. EMD essentially functions as a dyadic binary bank in denoising [52]. Unlike Wavelet Transformation, this approach decomposes signals based on the time scale features of the data itself and does not require the mother wavelet to be set beforehand. In essence, EMD is a means of smoothing non-stationary signals. As a result, the fluctuation and trend of different scales in the signal are decomposed step by step to produce a series of data series with different characteristic scales. The decomposed sequences are called the Intrinsic Mode Functions (IMFs). The IMF is defined as follows:

- The difference between the number of extreme points and zero crossing points in the data should not exceed one at most.
- For any point, the average value of the envelope defined by the local extremum is 0.

The decomposition steps of EMD for $x(t)$ are as follows:

Step 1: Find all extrema and connect the local maximum points into the upper envelope and the local minimum points into the lower envelope through the cubic spline curve, denoted as $u(t)$ and $d(t)$ respectively.

Step 2: Calculate mean of extrema envelopes, $m_1(t) = (u(t) + d(t))/2$.

Step 3: Calculate the intermediate signal $h_1(t) = x(t) - m_1(t)$.

Step 4: Determine if $h_1(t)$ satisfies the requirements to be an IMF. If $h_1(t)$ meets the two conditions, then $h_1(t)$ is the first IMF, $I_1(t)$. If not, the iterative steps 1-4 are repeated starting with $h_k(t)$ until the decomposed signal satisfies the IMF conditions.

Step 5: Calculate the remaining part $r_1(t) = x(t) - I_1(t)$. Do Step 1-4 with $r_1(t)$ and get next IMF $I_2(t)$. Repeat till $r_j(t)$ cannot be decomposed.

The first IMF has the highest frequency. As the number of subscript increases, the frequency of IMFs reduce. $r_j(t)$ can be considered a trend. The signal $x(t)$ can be expressed as $x(t) = \sum_{i=1}^{j} I_i(t) + r_j(t)$.

Figure 1 displays the flow chart of EMD.

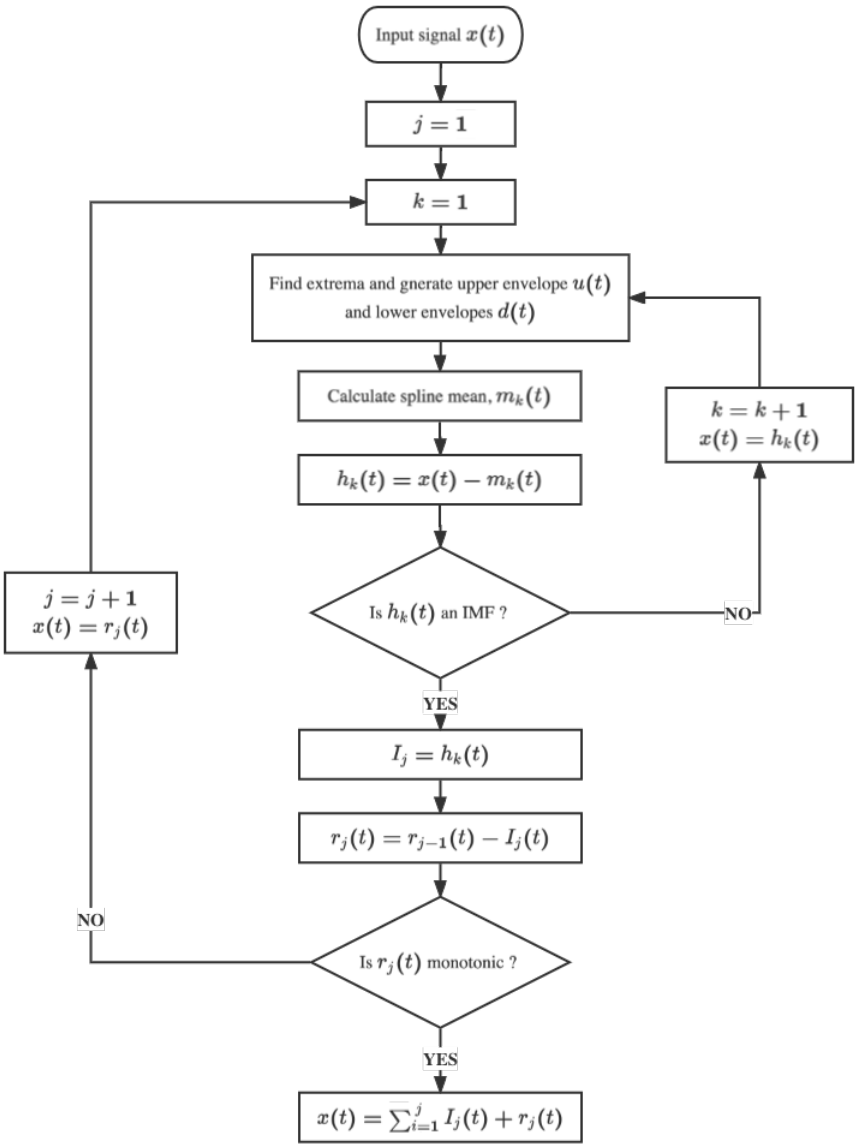

Figure 1: The structure of the EMD algorithm

3.1.2 CEEMDAN

Although EMD and EEMD can decompose the original signal very well, there are still some problems. The performance of EMD can be affected by the mode mixing problem. It indicates several signals with a broad distribution within the same IMF or comparable-sized signals in different IMFs. As for EEMD, it is time-consuming. CEEMDAN is an essential improvement over EMD and EEMD. CEEMDAN not only reduces the reconstruction error between the reconstructed data and the original data but also solves the problem of the varying number of decomposition modes caused by different noise addition techniques.

The CEEMDAN method consists of the following:

Step 1: Let $x(t)$ be the input time series. $x_i(t)$ is the time series added ith white noise. $\epsilon_i(t)$ is the ith white noise to be added. ω_0 is the noise scalar coefficient. I is the total number of added noise.

$$x_i(t) = x(t) + \omega_0 \epsilon_i(t), i \in \{1, ..., I\}. \tag{3.1}$$

Step 2: Feed $x_i(t)$ to EMD and get IMFs. Calculate the mean of the first IMF, IMF_1^i, denoted as $IMF_1(t)$.

$$IMF_1(t) = \frac{1}{I} \sum_{i=1}^{I} IMF_1^i. \tag{3.2}$$

Step 3: Calculate the first residual signals $r_1(t)$.

$$r_1(t) = x(t) - IMF_1(t). \tag{3.3}$$

Step 4: Obtain new time series $[r_1(t) + \omega_1 EMD[\epsilon_i(t)]]$ by adding adaptive noise and input them to EMD to get the second IMF, $IMF_2(t)$, where $EMD_i(\cdot)$ stands for the i-th component getting from EMD.

$$IMF_2(t) = \frac{1}{I} \sum_{i=1}^{I} EMD_1[r_1(t) + \omega_1 EMD_1(\epsilon_i(t))]. \tag{3.4}$$

Then, calculate the second residual $r_2(t) = r_1(t) - IMF_1(t)$.

Step 5: Similarly, continue calculating $IMF_i(t)$ like the following formula (3.5)(3.6)

shown, in which n means the number of whole components.

$$r_j(t) = r_{j-1}(t) - IMF_j(t), j = 2, 3, ..., n. \tag{3.5}$$

$$IMF_{j+1}(t) = \frac{1}{I}\sum_{i=1}^{I} EMD_1[r_j(t) + \omega_j EMD_j(\epsilon_i(t))]. \tag{3.6}$$

The end of Step 5 is the residual signal meets the stop condition of EMD.
Finally, the original time series can be expressed as follows:

$$x(t) = \sum_{j=1}^{n} IMF_j(t) + r_n(t).$$

To summarize, CEEMDAN has the following advantages:

- It introduces additional SNR for white noise to control the noise level in each decomposition process.
- Its IMFs can wholly reconstruct the original data almost without noise.
- Compared with EMD, EEMD and CEEMD, it requires fewer tests and has the highest decomposition efficiency.
- It avoids the problem that the decomposition effect of one stage is poor and affects the next stage.

But CEEMDAN also has some disadvantages :

- The residual noise is still included in IMFs, especially in the first couple of modes.
- The first two or three IMFs have similar signal scales.

Figure 2 shows the calculating procedure for this approach.

3.1.3 VMD

VMD assumes that any signal comprises a series of sub-signals with a specific central frequency and limited bandwidth. VMD algorithm mainly consists of the construction and solution of variational problems. Based on the classical Wiener filter, VMD obtains the central frequency and bandwidth limit by solving the variational problem, finds the components corresponding to each central frequency in the frequency domain, and finally obtains the IMFs.

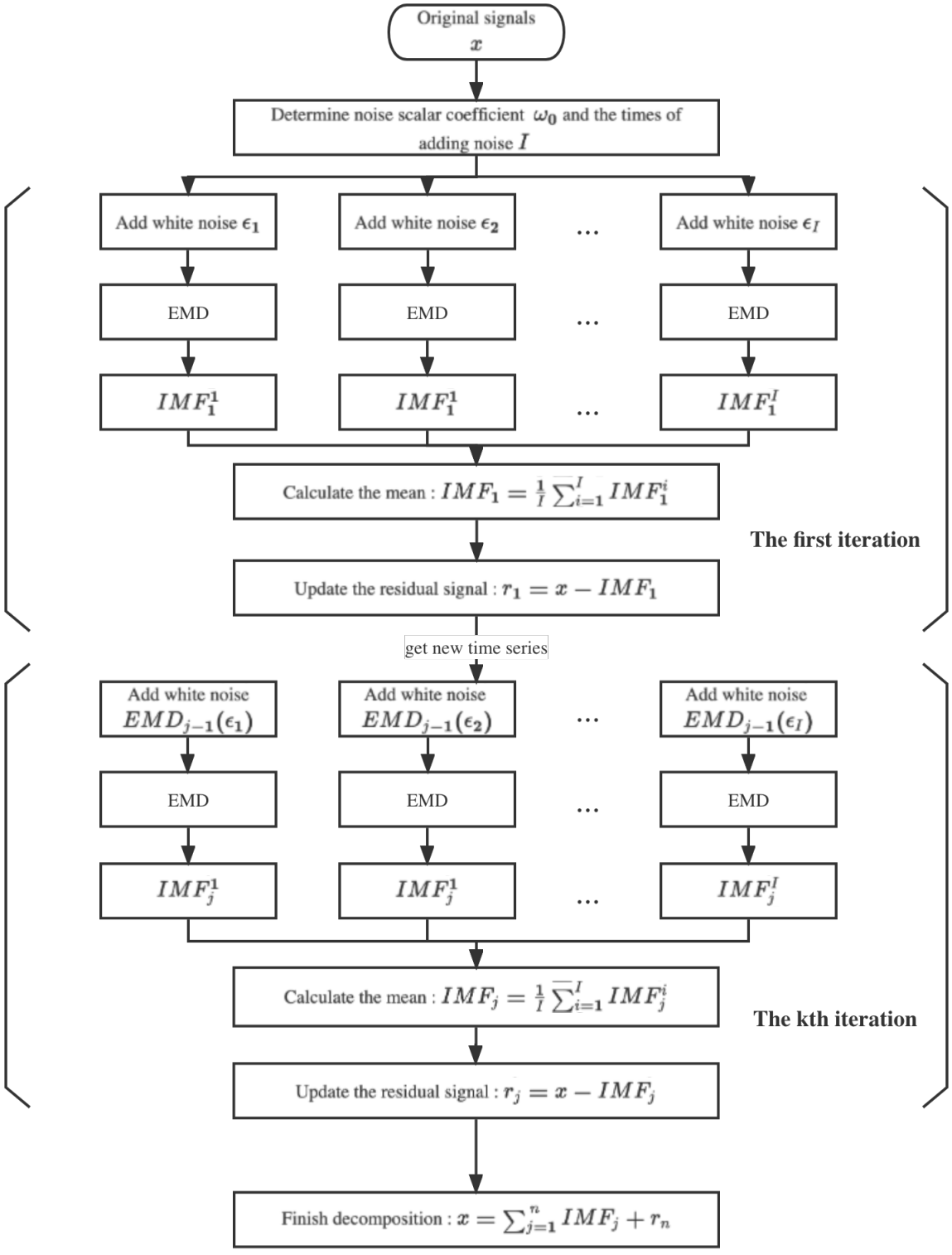

Figure 2: The structure of the CEEMDAN algorithm

Unlike EMD, VMD can specify the number K of modes. The mode mixing problem can be reduced if the K is appropriately chosen. In addition, the bandwidth of the decomposed IMFs is limited, which means IMFs have limited variance and some other statistical properties, such as stationarity, which is the premise of some financial models. But at the same time, the choice of penalty factor α and the number of decomposition layers K will affect the decomposition effect of VMD.

Different from the concept of IMFs proposed by Huang [20], Dragomiretskiy et al. [42] put forward band-limited intrinsic mode functions (BLIMFs), denoted as $u_k(t)$. The mathematical expression is as follows, in which $A_k(t)$ and $\phi_k(t)$ represent envelope amplitude of $u_k(t)$ and instantaneous phase, respectively.

$$u_k(t) = A_k(t)\cos(\phi_k(t)).$$

The specific principle of VMD is as follows:

- For each mode $u_k(t)$, relevant signals are calculated by Hilbert transformation to obtain the unilateral frequency spectrum.
- By adding an exponential function tuned to the respective estimated central frequency ω_k, the frequencies of each mode $u_k(t)$ are converted to the baseband.
- Gaussian smoothness is used to demodulate the signals and obtain the bandwidth for each $u_k(t)$. That is, it can decompose any signal $f(t)$ into several $u_k(t)$ which are around the central frequency ω_k.

The constructed constrained variational model in VMD is as follows:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_k \parallel \partial_t[(\delta(t) + \frac{j}{\pi t})u_k(t)] \exp^{-j\omega_k^t} \parallel_2^2 \right\} \tag{3.7}$$

$$f(t) = \sum_k u_k(t) \tag{3.8}$$

In formula (3.7) (3.8), $\{u_k\}$ stands for $\{u_1, u_2, ..., u_k\}$ and $\{\omega_k\}$ represent $\{\omega_1, \omega_2, ..., \omega_k\}$. δ_t is the impulse function. In order to obtain the optimal solution for formula (3.7) (3.8), the Lagrangian multipliers $\lambda(t)$ and the quadratic penalty factor α are introduced to transform the problem into an unconstrained one. Here is the formula of the augmented Lagrangian:

$$L(\{u_k\},\{\omega_k\},\lambda) := \alpha \sum_k \| \partial_t[(\delta(t) + \frac{j}{\pi t}) u_k(t)] \exp^{-j\omega_k^t} \|_2^2 + \| f(t) - \sum_k u_k(t) \|_2^2 + \left\langle \lambda(t), f(t) - \sum_k u_k(t) \right\rangle \tag{3.9}$$

The alternate direction method of multipliers (ADMM) is used to obtain the saddle points of the above Lagrange function. The specific steps are shown in Figure 3.

3.2 DFA

Hurst exponent is the most critical index in fractal market theory. It is defined by the asymptotic behaviour of the rescaled range as a function of the length of a time series and describes the degree of long memory of the time series. There are many methods for calculating the Hurst exponent, such as Rescaled Range (R/S) analysis, Detrended Fluctuation Analysis (DFA), and Periodogram Regression. Since time series are usually superimposed with noise and trend, using R/S analysis will lead to inaccurate scores [53]. DFA developed by Peng et al. [54] can efficiently filter out the trend components of each series order. Even with noisy and multinomial trend signals, DFA can discover long-range dependence. Consequently, it is appropriate for the long-range power-law dependence analysis of nonstationary time series.

The DFA involves the following steps:

Step 1: For a given time series $x(t), t \in \{1, 2, ..., N\}$, calculate the cumulative sum, where $\bar{x} = \frac{1}{N} \sum_{t=1}^{N} x(t)$.

$$Y(i) = \sum_{t=1}^{i} (x(t) - \bar{x}), i \in \{1, 2, ..., N\}.$$

Step 2: Divide $Y(i)$ into $n = [N/s]$ non-overlapping subseries with length s. To make use of all the data, $Y(i)$ is divided twice from the beginning to the end and from the end to the beginning, yielding $2n$ segments.

Step 3: The least squares method is used to fit the polynomial trend of each segment of data, usually using a polynomial of first degree $Y_s(t) = a_s x + b_s$. If higher order polynomials are used, the fitting degree will increase, but the time will increase accordingly.

Step 4 : Calculate the mean square error $F^2(v, s)$.

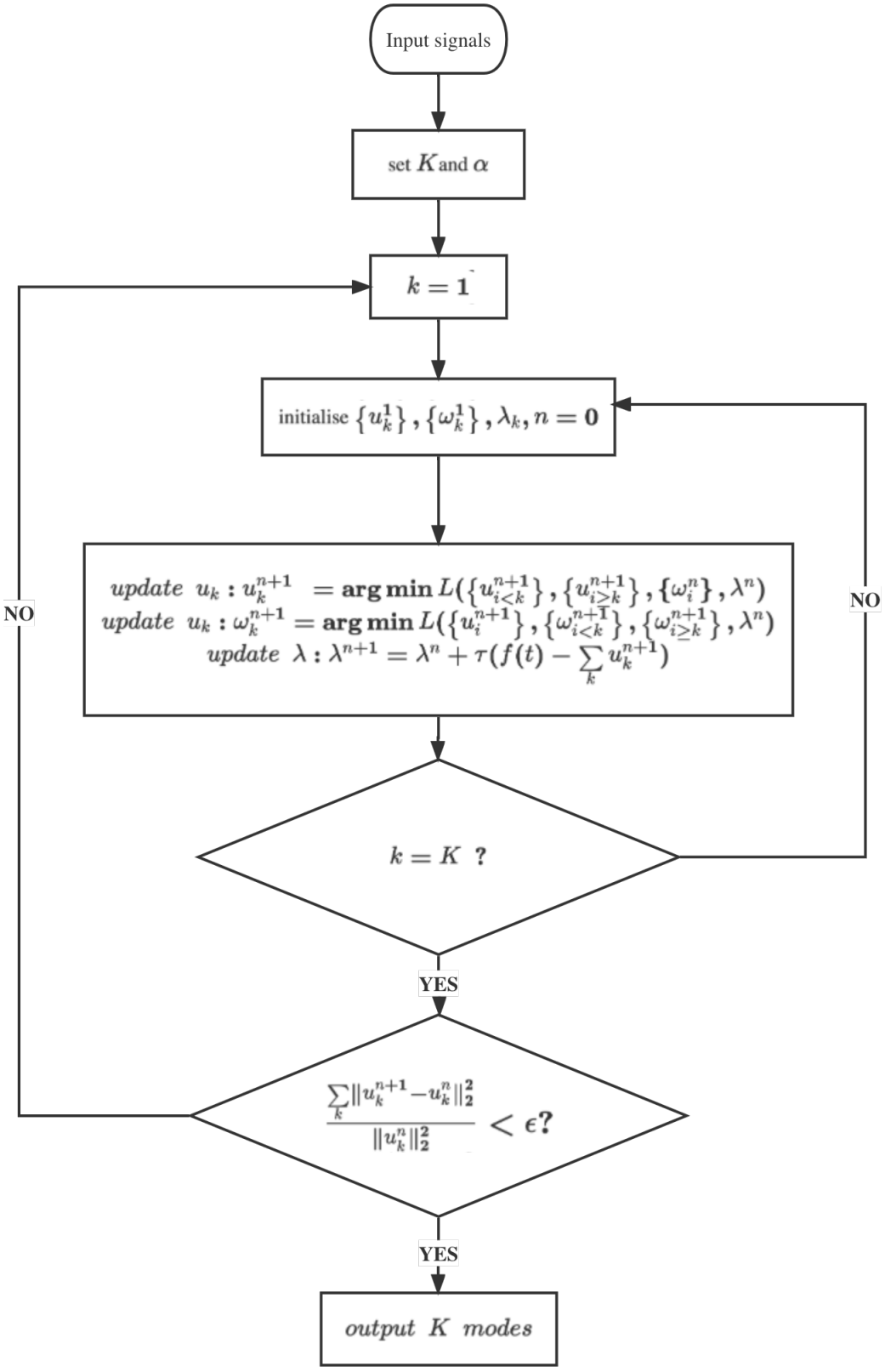

Figure 3: The structure of the VMD algorithm

$$F^2(v,s) = \frac{1}{s}\sum_{i=1}^{s}(Y((v-1)s+i) - y_v(i))^2, \qquad v = 1,2,...,n.$$

$$F^2(v,s) = \frac{1}{s}\sum_{i=1}^{s}(Y(n-(v-n)s+i) - y_v(i))^2, \quad v = n+1, n+2, ..., 2n.$$

Step 5: For $2n$ segments, calculate the fluctuation function $F(s)$.

$$F(s) = \sqrt{\frac{1}{2n}\sum_{v=1}^{2n} F^2(v,s)}.$$

Step 6: For each s, calculate the certain $F(s)$. Fit with linear least-squares regression, which is given as :

$$\ln(F(s)) = \alpha \ln(s) + constant.$$

This scaling exponent α delivers information on the autocorrelation of the series :

- If $0 < \alpha < 0.5$, it is a short-term dependence, indicating that the time series data have the opposite trend compared to the previous time series data.
- If $\alpha \approx 0.5$, the process is randomly distributed and is considered as white noise.
- If $0.5 < \alpha < 1$, then the series has long-range dependence. If α is closer to 1, then this long-range dependence is higher.
- If $\alpha \approx 1$, the process is a $1/f$ sequence, also known as pink noise. Unlike white noise, which has a uniform power across frequency bands, pink noise has an equal amount of power at each octave.
- If $\alpha \approx 1.5$, it is a sequence of Brownian noise.

3.3 XGBoost

In 2016, Chen [55] proposed the regression tree based lifting algorithm XGBoost, and proved that its model has the characteristics of low computational complexity, fast running speed and high accuracy. There are two types of trees in XGBoost, regression trees and classification trees.

Since XGboost is only used as a benchmark model to evaluate the denoising effect, the XGboost model with default parameters in the XGBoost module in Python is used in

the experiment. The mathematical definition of the objective function of the regression tree is given below.

XGBoost can be viewed as an additive regression tree model with K trees. $\hat{y}_i$ is the prediction value of sample x_i and f_k is the kth model.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i).$$

The objective function consists of the loss function L and the regularization term Ω that inhibits the complexity of the model :

$$Object = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k),$$

where l is a function of estimating the residual between the predicted values $\hat{y}_i$ and actual values y_i.

The idea of XGBoost is to add a tree per iteration to keep the predicted value close to the actual value, so the goal of iteration t is to minimize the following functions:

$$Object^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \sum_{j=1}^{t} \Omega(f_j), \tag{3.10}$$

in which $\hat{y}_i^t$ is the predicted value at the t iteration.

According to Taylor's formula, Taylor's second-order expansion of formula (3.10) is carried out.

$$Object^{(t)} = \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \sum_{j=1}^{t} \Omega(f_j), \tag{3.11}$$

in which $g_i = \frac{\partial (\hat{y}_i^{t-1} - y_i)^2}{\partial \hat{y}_i^{t-1}}$ and $h_i = \frac{\partial^2 (\hat{y}_i^{t-1} - y_i)^2}{\hat{y}_i^{t-1}}$.

Delete the constant term $\hat{y}_i^{t-1}$ in formula (3.11) and get :

$$Object^{(t)} = \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \sum_{j=1}^{t} \Omega(f_j).$$

4 Model Design

In this chapter, four noise reduction methods are described in detail. First, decompose the data, then remove the components judged as noise, and finally reconstruct the data.

4.1 EMD

Wu [56] found that the prod of the energy density of the IMF and its corresponding mean period must be a constant, which allows us to do a white noise statistical significance test of IMFs for any noisy data and it is used in this method.

The framework of this posterior statistical test is shown as follows:

Step 1: Implement EMD to time series $x(t)$ with length T and get $IMF_i, (i = 1, 2, 3, ..., N)$, in which the residal is denoted as IMF_N.

Step 2: For each IMF_i find the number p_i of peaks in IMF_i and calculate the mean-period I_i of the signal. Besides, calculate the energy density d_i.

$$I_i = \frac{p_i}{T},$$

$$d_i = \ln\big([\sum_{j=1}^{T} IMF_i(j)^2]/T\big).$$

Step 3: IMF1 has very low information content, so the energy of this IMF is assumed to come entirely from noise and its energy is allocated to the lowest value corresponding to the 5% line, denoted as $Factor$. $Factor$ is given as :

$$Factor = -I_1 + \big(k\sqrt{\frac{1}{T}e^{I_1}}\big)/2 - d_1,$$

where $k = |CDF^{-1}(\frac{1-0.05}{2})|$ and CDF^{-1} is the inverse of the cumulative distribution function of standard normal function.

Step 4: Get the scaled energy density sd_i and set the upper $limit$.

$$sd_i = Factor + d_i,$$

$$limit = -I_1 + (k\sqrt{\frac{1}{T}e^{I_1}})/2.$$

Step 4: Reconstruct data $\hat{x}(t)$.

$$\hat{x}(t) = \sum IMF_i, i = \{sd_i \geq limit\}.$$

4.2 CEEMDAN

Similarly, in this method just place the decomposition way EMD with CEEMDAN in Step 1 of section 4.1 and the remaining keep the same.

4.3 VMD-DFA

Before using VMD to decompose signals, the number K of modes should be set in advance. The effect of decomposition is mainly influenced by the selection of modes number. When K is small, the VMD algorithm is equivalent to an adaptive filter bank. Some vital information in the original signal will be filtered, affecting the accuracy of subsequent predictions. However, when K is large, the center frequencies of the adjacent mode components will be close to each other, resulting in mode duplication or additional noise. The main difference between the different modes is the difference in the center frequency. Therefore, it is necessary to select appropriate modes number by observing the distribution of center frequencies under different modal numbers. However, due to the different characteristics of different data, the method of trying the K from small to large one by one and determining the K by analysing the decomposition results is time-consuming and energy-consuming. Liu et al.[57] proposed a general method for selecting K based on DFA. We pick K according to the way in the paper and combine VMD and DFA to denoise the financial data.

Here are the steps of the VMD-DFA:

Step 1: Estimate α_{input} of the original data $x(t)$ by DFA. In DFA, the scaling range s of the segmentation interval is selected as $[4, 16]$, and the first-order polynomial is used to fit it. The details of DFA can be seen in Section 3.2.

Step 2: According to α_{input}, determine the median value L, which is the benchmark for determining the mode number K. The formula is as follows:

$$L = \begin{cases} 1, & \alpha_{input} \leq 0.75 \\ 2, & 0.75 < \alpha_{input} \leq 1 \\ 3, & 1 < \alpha_{input} \leq 1.5 \\ 4, & 1.5 < \alpha_{input} \end{cases}$$

Step 3: Set threshold θ. According to DFA, when scaling exponent α is about 0.5, the series behaves as a random walk; when α is less than 0.5, it has the characteristic of mean reversion; when the value is more than 0.5, it indicates that the time series has a long memory. If the first two types of sequences are not removed, then a lot of noise will be included in the reconstructed data. In addition, due to the uncertainty in the calculation, such as the selection of scaling range s, we increased the threshold θ for noise. The threshold θ is defined as $0.5 + 0.25 = 0.75$, like Liu et al. [57] did.

Step 4: Set K to L.

Step 5: Apply VMD to decompose the original data into K modes, denoted as $BLIMF_k, (k = 1, 2, ..., K)$. Here are the parameter settings in VMD. The moderate bandwidth constraint is set to 2000; the noise tolerance is set to 0; the tolerance of the convergence criterion is set to 1e-7; all estimated mode center frequencies start uniformly distributed.

Step 6: For each mode $BLIMF_k$, apply DFA and get the corresponding scaling exponent α_k. Count number M of the modes of which the α_k is greater than threshold θ. If M is not greater than L, then $K = K + 1$ and repeat step 5.

Step 7: Due to the limitation of the loop algorithm, the final $K = K - 1$. Then reconstruct the time series $\hat{x}(t)$ which is given as :

$$\hat{x}(t) = \sum_{l=1}^{L} BLIMF_l.$$

The above steps are simplified and shown in the Figure 4.

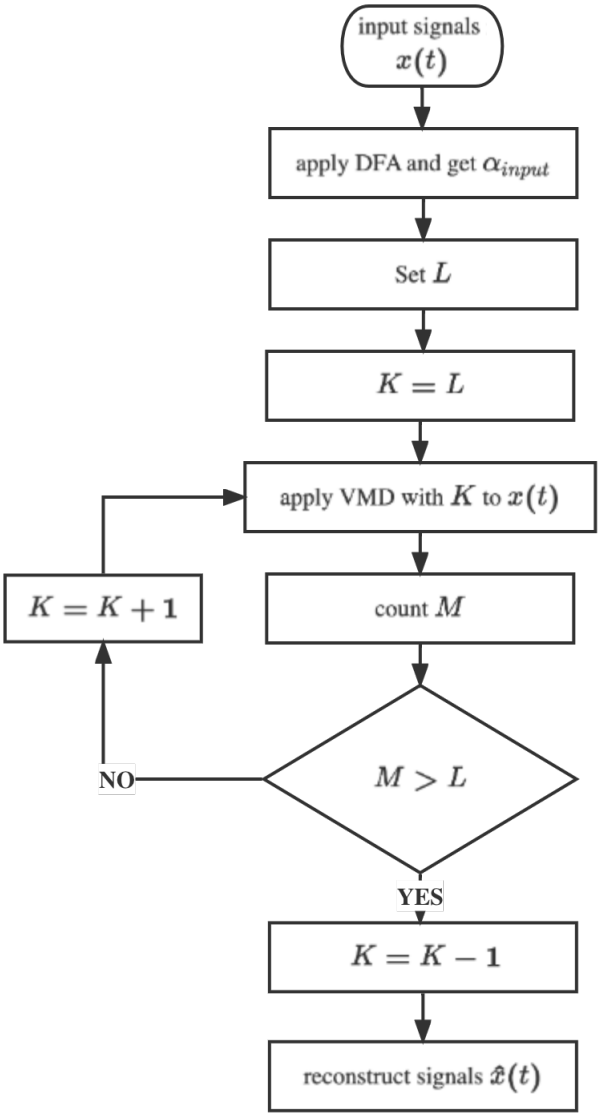

Figure 4: The structure of the VMD-DFA algorithm

4.4 VMD-EMD-DFA

The only difference between VMD-EMD-DFA and VMD-DFA is the way to choose K. In this method, the mode number K in VMD is determined by the total number N of IMFs and the residual extracted by EMD, as Lahmiri and Salim [58] did.

Below are the steps.

Step 1: Implement EMD to time series $x(t)$ and get the total number N of IMFs and the residual signal.

Step 2: Apply VMD to time series $x(t)$ and get $K = N$ modes, denoted as $BLIMF_k, (k = 1, 2, ..., K)$.

Step 3: Similarly, calculate the α_k of each mode $BLIMF_k$ by DFA.

Step 4: Sum the $BLIMF_k$ of which the α_k is larger than the threshold $\theta = 0.75$ and get the denoised data $\hat{x}(t)$

$$\hat{x}(t) = \sum BLIMF_k, k = \{\alpha_k \geq \theta\}$$

5 Main Results

This chapter reports simulation, prediction tests and classification tests. To examine the direct influence of noise reduction methods on various time series, they are evaluated on trend-stationary and non-stationary time series, respectively. In the prediction and classification tests, five low-frequency datasets and one high-frequency dataset are tested, respectively. The corresponding implementation, outcomes, and discussion are provided in each experiment.

5.1 Simulation

5.1.1 Trend-stationary time series

We generate a time series in order to conceptually assess the impact of noise reduction because pure data is not available in the financial sector. Four noise reduction techniques are used to minimise the noise after it has been artificially added to the time series in varying degrees. To assess the denoising effect, the denoised signal and the pure data were compared.

We used the following time series from a real model with references to Kim & Oh [59] in our simulation:

$$x(t) = 0.5t + \sin(\pi t) + \sin(2\pi t) + \sin(6\pi t) + r_n, t \in [0, 9],$$

where r_n is the Gaussian white noise and $n = \{1, 2, ..., 9000\}$. There were several parts to the first synthetic data x(t). The periodic variations are indicated as $sin(t)$, $sin(2t)$, and $sin(6t)$, whereas the long-term trend is expressed as $0.5t$. The sample sampling frequency is 1KHz. The pure signal is shown in Figure 5.

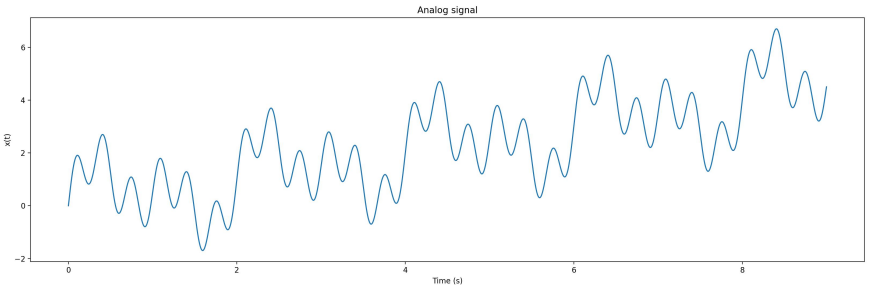

Figure 5: The pure signal

The amount of noise in financial data grows as the observational scale does. In order to simulate financial data of different scales, we add Gaussian white noise to the original time series $x(t)$ according to its power, so that the signal-to-noise ratio (SNR) of contaminated time series $\hat{x}(t)$ is reduced to 1, 5, 8, 10 dB. The corrupted time series $\hat{x}(t)$ are shown in Figure 6.

First, we compare all the denoising methods qualitatively by visual examination. The denoised signals are shown in Figure 7. As the SNR decreases, the noise almost covers the signal.

As shown in Figure 7, the data processed by EMD and CEEMDAN are smoother and more consistent with the pure signal. VMD-DFA has a little less effective noise reduction effect. VMD-EMD-DFA has the least effective noise reduction. After processing the noisy signal, there is still a great deal of noise in the reconstructed signal.

Then, the SNR and the Mean Square Error (MSE) are used for quantitative analysis of the denoised data. The comparison of the MSE and SNR is shown in Table 1 and 2. In general, the noise reduction effect of EMD and CEEMDAN is similar, and their performance is the best among the four noise reduction methods. Even in data with a SNR of 1dB, the noise can be significantly removed, increasing the SNR from 1dB to about 18dB. The noise reduction effect of VMD-DFA is slightly lower than that of EMD and CEEMDAN. VMD-EMD-DFA has the worst noise reduction effect. Except for EMD, other methods have higher SNR and lower MSE when the noise in the sequence is reduced. When the SNR is 10dB in noisy series, the noise reduction effect of EMD, CEEMDAN and VMD-DFA is nearly equivalent. It is worth noting that when the SNR increases from 8dB to 10dB, the noise reduction effect of EMD decreases instead. The reason may be that the SNR of the noisy signal is high, and EMD removes the signal while removing the noise.

Method	EMD	CEEMDAN	VMD-DFA	VMD-EMD-DFA
1 dB	18.124292	18.127836	15.892434	9.987242
5 dB	21.686777	21.273019	19.799117	13.668137
8 dB	24.788559	22.555694	22.754384	16.547693
10 dB	24.338655	24.332550	24.729285	18.429259

Table 1: Comparison of SNR obtained by using different denoising methods

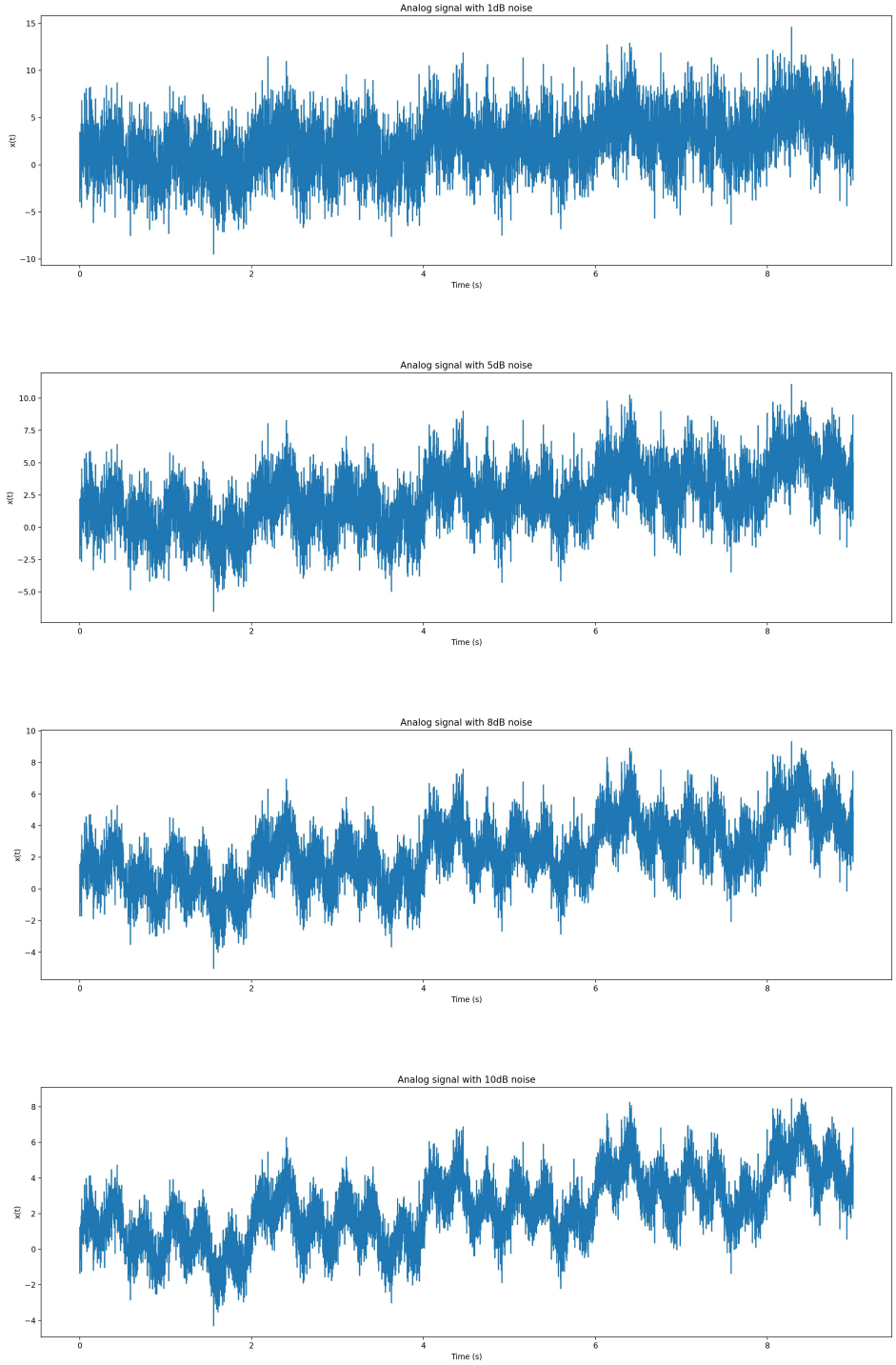

Figure 6: Analog signal with noise

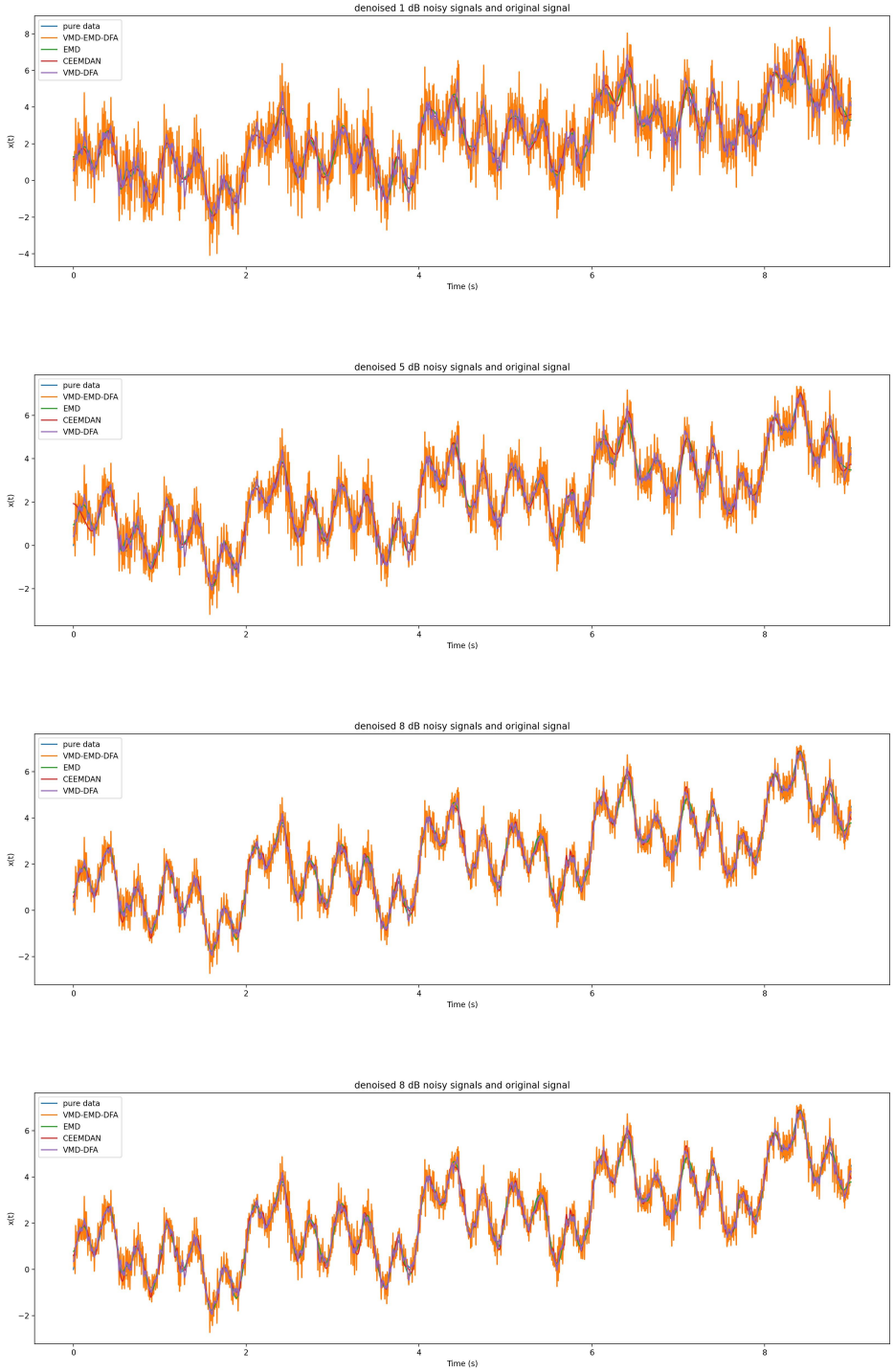

Figure 7: denoised signal and pure signal

Method	EMD	CEEMDAN	VMD-DFA	VMD-EMD-DFA
1 dB	0.134288	0.134490	0.227775	0.960693
5 dB	0.057853	0.063755	0.090125	0.382307
8 dB	0.028196	0.047449	0.045124	0.191626
10 dB	0.031289	0.031286	0.028493	0.122896

Table 2: Comparison of MSE obtained by using different denoising methods

5.1.2 Non-stationary time series

The Random Walk model is the most popular and effective statistical model for financial time series analysis and has been widely studied and used. This model assumes that the most recent observations are the best guide to the next period's predictions. The Random walk process is a special non-stationary stochastic process. In finance, the Random Walk assumption is often used to model stock prices and other factors. Mathematically, a simple Random Walk model is represented as follows:

$$y_t = y_{t-1} + \epsilon_t,$$

where y_t are the observations of time series, ϵ_t are white noise, and $\epsilon_t \sim N(0, \sigma^2)$.

To more realistically simulate how prices change over time, the next observed value is the currently observed value multiplied by a percentage:

$$y_t = y_{t-1} * \epsilon_t, \tag{5.1}$$

where $\epsilon_t = R_t + 1$ and $R_t \sim N(0, \sigma^2)$. When the simple return R_t is small, it can be approximated as the log-return $r_t = log(R_t + 1)$. So replacing R_t with r_t we can get the formula of Random Walk with log-returns:

$$y_t = y_0 * exp(\sum_{t=1}^{t} r_t), \tag{5.2}$$

in which y_0 is the start price of the asset and $r_t \sim N(0, \sigma^2)$.

We generate a time series using equation (5.2) with 1001 data points, which is shown in Figure 8. The beginning price is 100 and the σ^2 is set to 0.005.

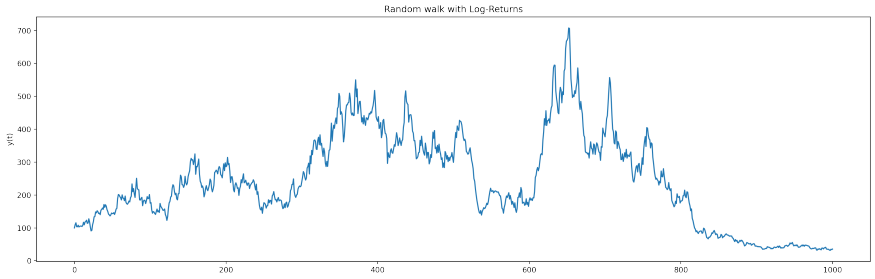

Figure 8: Random walk with log-returns

Similarly, like section 5.1.1, we add white noise into the pure data and get noisy data with SNR of 1, 5, 8 and 10 dB. After applying four denoising techniques to noisy data, the reconstructed data is obtained.

Take the results of the denoised data which comes from noisy price series with SNR of 5 dB as an example. As shown in Figure 9, the data denoised by VMD-DFA is more consistent with Pure data than other methods. The rest plots of the denoised data are in Appendix A.2.

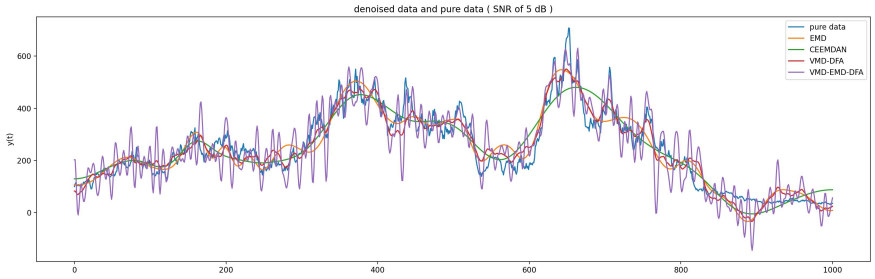

Figure 9: The denoised data of four methods and the pure data (SNR of 5 dB)

Since the starting price is high to better see the denoising effect, we select SNR and the root-mean-square error (RMSE) to evaluate. The Table 3 and 4 present the corresponding results. They show that the four noise reduction methods can effectively remove the noise in the simulated price series. VMD-DFA can successfully increase SNR and decrease RMSE, and its performance is superior to the other three techniques. VMD-EMD-DFA performs the worst, with the reconstructed data still containing a significant amount of noise.

Method	EMD	CEEMDAN	VMD-DFA	VMD-EMD-DFA
1 dB	12.585419	13.549275	14.106428	8.554190
5 dB	15.391764	14.751065	16.799497	12.366835
8 dB	16.252992	16.259653	18.246977	15.149699
10 dB	17.565126	17.728559	18.929572	17.222287

Table 3: Comparison of SNR obtained by using different denoising methods

Method	EMD	CEEMDAN	VMD-DFA	VMD-EMD-DFA
1 dB	65.696531	58.796255	55.143206	104.497022
5 dB	47.558180	51.198846	40.442552	67.370792
8 dB	43.068904	43.035886	34.234545	48.902206
10 dB	37.030291	36.340043	31.647164	38.521133

Table 4: Comparison of RMSE obtained by using different denoising methods

5.2 Prediction Tests

In this section, the main purpose is to study whether the four noise reduction methods can improve the accuracy of predicting the closing price of financial underlying.

5.2.1 Data

Hang Seng Index (HSI), DAX PERFORMANCE INDEX (GDAXI), Crude oil, Apple stock, and Bitcoin were chosen for study in this report. The data consists of major financial markets, such as stock indexes, commodities, cryptocurrencies, and IT giants' stocks. This study selects daily frequency data from the previous five years, including open price, high price, low price, closing price, and trade volume. Yahoo Finance provides the data beginning on July 26, 2017 and ending on July 26, 2022. Due to the varying trading time restrictions in each market, Bitcoin contains 1827 data points, but the rest of the financial data length is approximately 1250.

Figure 10 presents the closing prices of the five financial assets.

The stationarity, autocorrelation and normality of the close price of these five underlying assets are tested using statistical tests in the Statsmodels module in Python. The results and summary are presented in Table 5.

First, the Augmented Dickey-Fuller (ADF) test is used to examine the stationarity of the data. The results show that the p-values are all over 0.05 and cannot reject the null hypothesis. All those underlying assets are non-stationary.

Next, the autocorrelation is tested using the Ljung-Box method. The null hypoth-

Figure 10: The closing prices

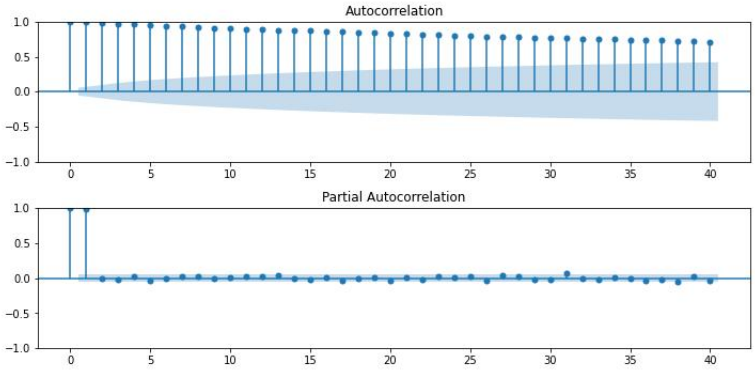

Figure 11: The the ACF and PACF of HSI

esis, which states that the data are independently distributed, is rejected. Since in the event of any lag, it reveals that all p-values are nearly 0.00 much smaller than 0.05.

Third, the Jarque-Bera test is applied to determine whether the data are normally distributed, and it demonstrates that the p-values are around 0.00 less than 0.05, rejecting the null hypothesis. All assets' close prices are non-normality.

Finally, the plot about Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) are drawn. Due to the length limitation of this paper, only ACF and PACF diagrams of HSI are presented in Figure 11. The rest of the figures are in the appendix A.1. For all five financial assets, the ACF decays exponentially, with the obvious trailing property. In addition, the PACF rapidly decreases to near 0 after the second order.

5.2.2 Data preprocessing and prediction framework

In order to increase the accuracy of the prediction, we construct some features based on the closing price to help forecast. Using the closing price as the target sequence, the TA-Lib module in Python is applied to compute the simple moving average(SMA_{lag}), exponential moving average(EMA_{lag}), standard deviation($STDDEV_{lag}$), and price before t time($Price_{lag}$) with lag 5, 10, 20, 60. Since the maximum lag is 60, the first 60 pieces of data contain missing values which are deleted after. In all, 16 new features have been added to the initial four, for a total of 20 features.

The first 80% of the data about 940 pieces of data (1414 for Bitcoin) is split into the training set and use the last 20% of the data as the test set with around 230 pieces of data (353 for Bitcoin). We adopt a one-step ahead prediction, and the data volume of 5 days is used as a group to predict the closing price of the next day. For example,

Assets	Statistical Tests	p-value	Implication
HSI	ADF	0.4683	Non-stationarity
	Ljung–Box test	$6.5035e^{-265}$	Autocorrelation
	Jarque–Bera test	$1.6883e^{-8}$	Non-normality
	ACF	/	Autocorrelation
	PACF	/	Autocorrelation
GDAXI	ADF	0.2358	Non-stationarity
	Ljung–Box test	$3.4763e^{-274}$	Autocorrelation
	Jarque–Bera test	0.0001	Non-normality
	ACF	/	Autocorrelation
	PACF	/	Autocorrelation
Crude Oil	ADF	0.7001	Non-stationarity
	Ljung–Box test	$1.2250e^{-270}$	Autocorrelation
	Jarque–Bera test	$7.1263e^{-48}$	Non-normality
	ACF	/	Autocorrelation
	PACF	/	Autocorrelation
Apple	ADF	0.9121	Non-stationarity
	Ljung–Box test	$3.3464e^{-275}$	Autocorrelation
	Jarque–Bera test	$1.9078e^{-31}$	Non-normality
	ACF	/	Autocorrelation
	PACF	/	Autocorrelation
Bitcoin	ADF	0.5510	Non-stationarity
	Ljung–Box test	0.000	Autocorrelation
	Jarque–Bera test	$5.9756e^{-80}$	Non-normality
	ACF	/	Autocorrelation
	PACF	/	Autocorrelation

Table 5: The statstical tests for five underlying assets

using the data from day 1 to day 5, predict the closing price on day 6.

The 20 features and the initial closing price are directly entered into the prediction model as a benchmark to produce the prediction, which is assessed using the following assessment criteria: coefficient of determination (R^2), the mean absolute error (MAE), the root mean of squared errors ($RMSE$) and the mean absolute percentage error ($MAPE$). While $MAPE$ assesses the relative magnitude of deviations, MAE and $RMSE$ measure the absolute size of the divergence between the real value and the anticipated value. R^2 is mostly used to evaluate how well the model fits the data.

For comparison, the original closing price series is applied with four denoising methods to remove the noise and reconstruct the data and the denoised data replaces the original closing price series in the model. In many academic works, researchers first break down the original time series to produce a number of $IMFs$. The prediction is then made by dividing the training set and the test set in $IMFs$. Because the core of EMD and derived denoising methods are actually denoising in accordance with the entire segment of data. For instance, the number and shape of the disaggregated $IMFs$ change as new data points are added. We can obtain some surprisingly positive results if we first decompose all the time series and then train the model using this data.

In order to get around this issue, we use a decomposition by expanding windows in the training set. In other words, the length of the decomposed sequence increases continuously with the length of the first split training set as the starting point. Decompose and reconstruct the data once more after each new data point becomes available. Even if the newly added data points are removed, these old and new data segments are not entirely consistent after re-decomposition and reconstruction of the data. The prediction would be greatly off if the newly added denoised data point is simply input into the original model, which is trained on the last reconstructed data. Therefore, it is necessary to use the latest reconstructed data to retrain the model and then predict the next value after re-decomposition. Figure 12 shows how the expanding window works.

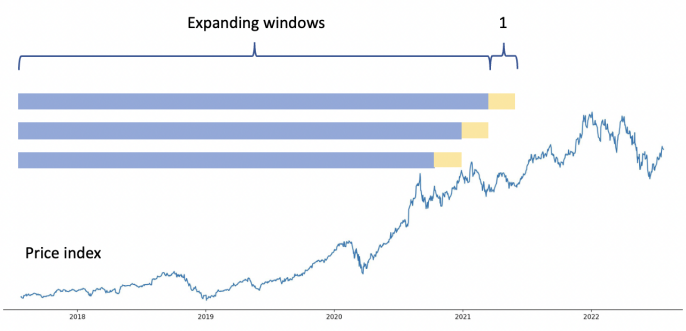

Figure 12: Prediction with expanding windows

XGBoost is the model to forecast the closing price. We use the XGBoost with the default parameters in the Python XGBoost module to ensure fairness and to make it clear that the improvement of prediction accuracy comes from the noise reduction method rather than from the parameter tuning. Compared with the Gradient Boosting Decision Tree, XGBoost introduces a regularization term in the cost function, which controls the complexity of the model and prevents the model from overfitting. Although it has been demonstrated that the deep learning model can significantly increase prediction accuracy when compared to the conventional approaches. No deep learning model is applied here. The reason is that the model needs to be repeatedly trained due to the characteristics of decomposition methods and the design of the prediction workflow. Since deep learning takes more time to complete, using deep learning models will significantly lengthen the training period. For example, reconstructing the data and retraining the model is required for each prediction. Each dataset in this section needs to be retrained at least 230 times.

5.2.3 Results

The Figure 13 and Table 6 show the prediction results of the four noise reduction methods applied and the benchmark on the test set. Benchmark means the model trained with the unprocessed closing price.

Apart from crude oil, EMD performs well in four out of the five datasets, and all four indicators are optimised. Except for Apple and crude oil, CEEMDAN performs well across all three datasets. Although EMD and CEEMDAN increase R^2 and decrease $RMSE$ in the crude oil dataset, they also increase MAE and $MAPE$. They behave generally the same as the baseline model in the oil dataset. Both EMD and CEEMDAN

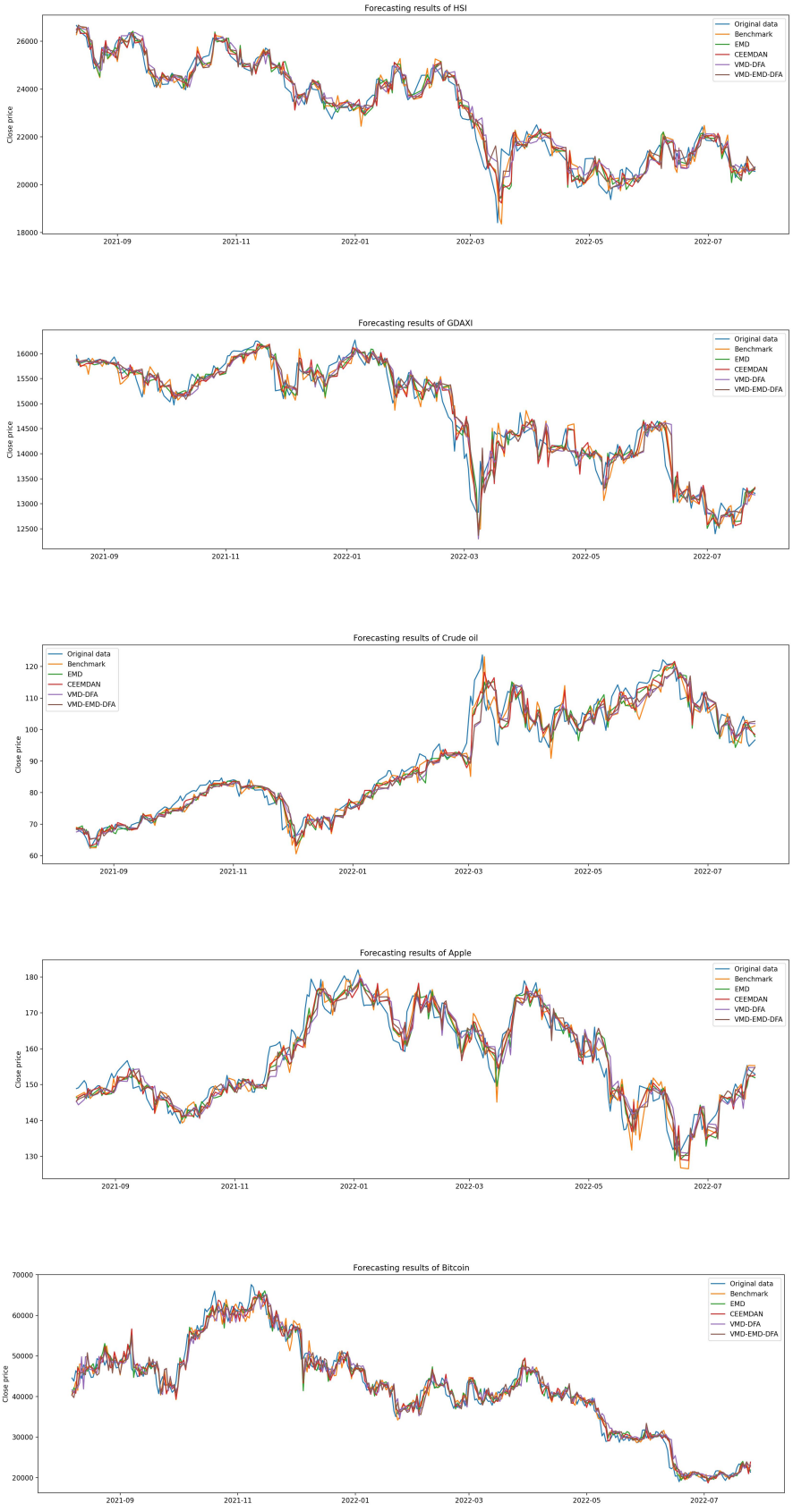

Figure 13: The forecasting results

Assets	Methods	R^2	$RMSE$	MAE	$MAPE$
HSI	Benchmark	0.936403	496.397745	374.084802	1.655815
	EMD	0.938712	487.300095	361.365376	1.602412
	CEEMDAN	0.940929	478.407881	354.831280	1.575213
	VMD-DFA	0.929115	524.069665	395.332315	1.764443
	VMD-EMD-DFA	0.941465	476.231052	354.105886	1.567463
GDAXI	Benchmark	0.929495	272.536385	198.237488	1.367527
	EMD	0.941685	247.858653	179.943962	1.245933
	CEEMDAN	0.938848	253.815194	186.132340	1.288407
	VMD-DFA	0.934076	263.533655	187.002490	1.295521
	VMD-EMD-DFA	0.928730	274.011134	194.324772	1.346217
Crude oil	Benchmark	0.932567	4.224682	2.883765	3.070597
	EMD	0.937531	4.066209	2.908605	3.114550
	CEEMDAN	0.935778	4.122863	2.911189	3.092991
	VMD-DFA	0.923402	4.502644	3.239623	3.432564
	VMD-EMD-DFA	0.922002	4.543598	3.248627	3.446808
Apple	Benchmark	0.902314	3.948423	3.099678	1.988443
	EMD	0.908624	3.818774	3.049980	1.952080
	CEEMDAN	0.902752	3.939574	3.137972	2.006215
	VMD-DFA	0.874175	4.481168	3.520600	2.247919
	VMD-EMD-DFA	0.895243	4.088842	3.275443	2.093338
Bitcoin	Benchmark	0.966595	2136.870277	1571.803708	3.756434
	EMD	0.968861	2063.122424	1508.186371	3.684011
	CEEMDAN	0.970302	2014.814592	1495.630450	3.664384
	VMD-DFA	0.959998	2338.384829	1754.297882	4.328088
	VMD-EMD-DFA	0.965346	2176.460689	1639.271838	4.035978

Table 6: Predictive performance evaluation of different methods

increase R^2 and decrease $RMSE$ across all datasets. For instance, EMD and CEEMDAN can, on average, reduce $RMSE$ by 4.27% and 3.76%, respectively. It is challenging to select the better approach for these two techniques, though. For example, CEEMDAN improves more than EMD in the Bitcoin dataset, whereas the opposite is true in the Apple Stock dataset.

These two methods do not improve much. The reason may be that the SNR of low-frequency data is relatively high, and the prediction obtained by the model trained on the original data can fit the target well. Therefore, none of the models developed using the reconstructed data is significantly improved. Another reason is that EMD has an end effect. In other words, the signal will be distorted at the edges of the decomposed sequence. The models are subject to end effects every time because we need to redecompose the model. Regarding CEEMDAN, some noise is still present in its modes.

Neither VMD-DFA nor VMD-EMD-DFA fared well in the Apple, Bitcoin or Crude Oil. The prediction accuracy of the models using these two denoising techniques is not better but worse when compared to the benchmark model. But these two denoising methods are not completely ineffective. VMD-EMD-DFA has optimization in HSI, while VMD-DFA improves the prediction accuracy in GDXAI. The signal is deleted simultaneously with the noise reduction, meaning that some information is lost, which could be the cause of the poor performance. Additionally, not much evidence exists to support the use of VMD for economic and financial data modelling and forecasting compared to EMD.

Table 7 shows the time taken to decompose and reconstruct the closing price 100 times in HSI. Compared to the other three methods, EMD requires a lot less time. EMD performs the best when taking into account computation time and denoising effect.

Method	EMD	CEEMDAN	VMD-DFA	VMD-EMD-DFA
Cost time	5.5874	676.2155	113.7324	48.0305

[1] Parameters: run = 100

Table 7: The cost time of different denoising methods

5.3 Classification tests

5.3.1 Data

We use the Bitcoin one-minute price in February 2021 as our research subject. Satoshi Nakamoto first proposed the idea of Bitcoin in 2008. Bitcoin possesses the traits of decentralization, low transaction costs, anonymity, tax exemption, free of regulation and more. As the price of bitcoin continues to rise, it has become the most watched investment product in the world. As of August 4, 2022, the total market capitalization of Bitcoin was about 437.3 billion dollars, ranking first in the total market capitalization of cryptocurrencies in the world.

The data are obtained from Kaggle. The exact time of data start is 00:00, February 1, 2021 to 23:59:00, Feb 28, 2021 and there are 40,242 data points. Figure 14 is the price trend chart for Bitcoin one-minute prices.

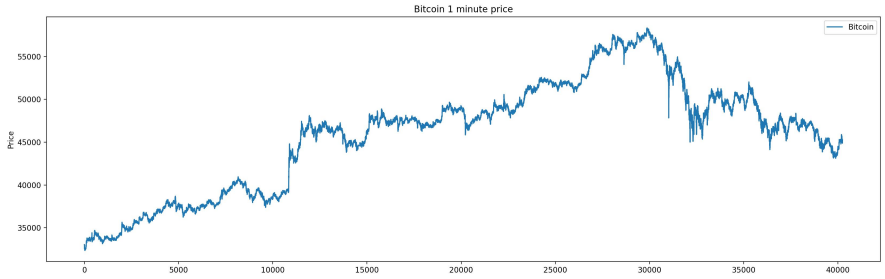

Figure 14: The price series of Bitcoin one-minute

Similar statistical tests are performed on the prices of the one-minute Bitcoin series for stationarity, autocorrelation and normality. The results and summary are presented in Table 8. To check for data stationarity, we run the ADF test. According to the findings, the p-value is greater than 0.05 and cannot rule out the null hypothesis. The price of bitcoin per minute is not stationary. Next, the Ljung-Box method and an ACF and PACF plot are used to test the autocorrelation. ACF decays exponentially and PACF rapidly decreases to near 0 after the second order. They all demonstrate how heavily autocorrelated the Bitcoin one-minute prices are. The Jarque-Bera test indicates that the Bitcoin one-minute prices are not normally distributed. In conclusion, the one-minute prices of bitcoin are non-stationary, not normally distributed, and strongly correlated.

Statistical Tests	p-value	Implication
ADF	0.2292	Non-stationarity
Ljung–Box test	0.0	Autocorrelation
Jarque–Bera test	0.0	Non-normality
ACF	/	Autocorrelation
PACF	/	Autocorrelation

Table 8: The statistical tests for Bitcoin 1 min

5.3.2 Data preprocessing and classification framework

The purpose of this experiment is to verify whether the denoised data can increase the accuracy of classification.

There are no labels in the original data, so we set up a simple method to add labels. First, compute the log returns R_t of the original price series. Because of the poor practicality of binary classification, we divided R_t into three categories. The value range of threshold $\sigma_i, (i = 1, 2, ..., 50)$ is from 0 to the 60% of maximum value of R_t on the test set, and 50 points are evenly divided the value range into fixed intervals. If R_t is greater than σ_i, the label is set to 1. If R_t is less than σ_i, the label is set to -1. If R_t does not meet the preceding two conditions, the label is 0.

XGBoost is chosen as the classification model. To be more specific, we use the XGBClassifier in the XGBoost package in python with the default parameters. XGBoost has the following advantages:

- Simple and easy to use. Users can utilise XGBoost with ease and achieve excellent results.
- High scalability. Large data sets can be processed quickly and effectively with it, and it doesn't require a lot of hardware resources like memory.
- Robust. It can produce results that are close to those of the deep learning model without fine parameter tuning.

We selected the F1-score as the classification evaluation criterion of the model. In statistics, the F1-score is used to assess the precision of a binary classification model. It takes both recollection and precision into account. The F1-score, which has a maximum value of 1 and a minimum value of 0, can be thought of as a weighted average of precision and recall. It is calculated using the True Positives (TP), False Positive (FP), False Negatives (FN) and True Positives (TP) and the F1-score equation for binary

classification is as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

F1 scores can also be used to rate how successful a classifier is in multi-categories, which can be calculated in the following three ways:

- Macro-F1. The TP, FP, FN and TP of each category are counted, and the precision and the recall of each category are calculated to obtain the F1 score of each category, and then the arithmetic mean is taken to obtain Macro-F1.
- Micro-F1. The TP, FP, and FN values for all categories are added and then substitute into the F1-score equation to obtain our micro-F1.
- Weighted-F1. It is similar to Macro-F1, except that the contribution of each category to F1-score is weighted by the size of the category.

Because the threshold is different, the number of labels in each category will be highly different, that is, this is an unbalanced dataset. So in order to assign a larger contribution to classes with more examples, we chose to use Weighted-F1.

Considering the characteristics of EMD and related derived decomposition methods, that is, those decomposition methods denoise based on the whole segment of data and both EMD and VMD are impacted by the end effect, so the classification model is designed as follows:

Model 1: The training set is the first 80% of log returns R_t of the original price series, with 32193 data points. We adopt one-step ahead prediction, which means predicting the label of the next day using the log return of the current day.

Model 2: Apply four denoising methods to the first 80% of the original price series to obtain the reconstructed price series, and calculate the new log return $\hat{R_t^i}(i = 1, 2, 3, 4)$ based on the reconstructed price series, where i is denoted according to the order of EMD, CEEMDAN, VMD-DFA, VMD-EMD-DFA.

Then, similarly use the $\hat{R_t^i}$ to do one-step-ahead predictions.

Finally, the last 20% of data, 8049 data points, is used as the test set. That is, input the log returns R_t in the test set into the two models above and calculate the F1-scores separately. The F1-score line of Model 1 is denoted as $F1 - score_0$ and the F1-score lines of Model 2 are denoted as $F1 - score_i(i = 1, 2, 3, 4)$

Figure 15 shows the framework of classification.

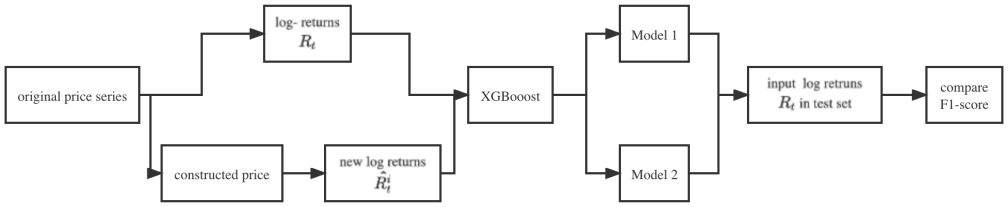

Figure 15: The framework of classification

5.3.3 Results

As can be seen from the Figure 16, all noise reduction methods can significantly improve the F1-score when the threshold σ_i is small. Figure 17 is the box plot of log returns R_t and new log returns $\hat{R}_t^i$ produced by different denoising methods in the training set. As we can see from the figure, most of R_t is distributed in the interval $[-0.001, 0.001]$. The classification model trained by R_t has a poor classification effect when σ_i is very small. However, in the model trained by $\hat{R}_t^i$, F1-score is better. It shows that the model trained by $\hat{R}_t^i$ can deal with complex classification situations.

However, when σ_i is large, the F1 score line of Model 2 with four denoising methods and the F1 score line of Model 1 appear to cross. In other words, in this crossover region, the F1 score of Benchmark is higher than that of the comparison model. The crossover between $F1 - score_0$ and $F1 - score_1$ appears late and the gap with $F1 - score_0$ is very small. CEEMDAN performs second only to EMD. The intersection point of $F1 - score_3$ and $F1 - score_0$ appears the earliest, and there is a big gap with $F1 - score_0$. The reason why the classification accuracy of the model trained with denoised data is not as good as that of Benchmark may be that, during denoising, sudden fluctuations are smoothed and the number of high $\hat{R}_t^i$ decreases rapidly, expecially for the VMD-DFA and VMD-EMD-DFA. Because the decomposition effect of VMD is limited by the burst signal.

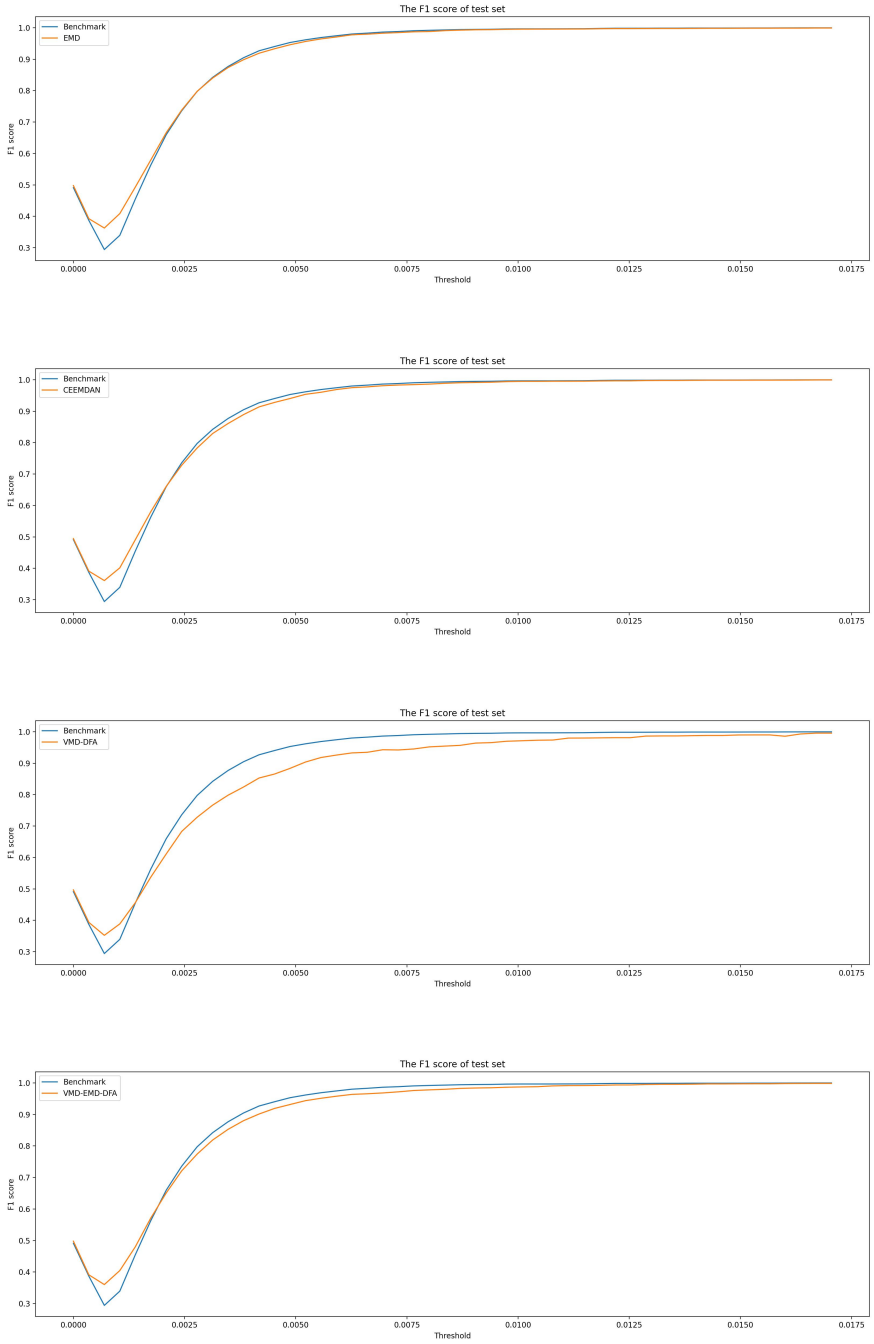

Figure 16: F1 score with different thresholds

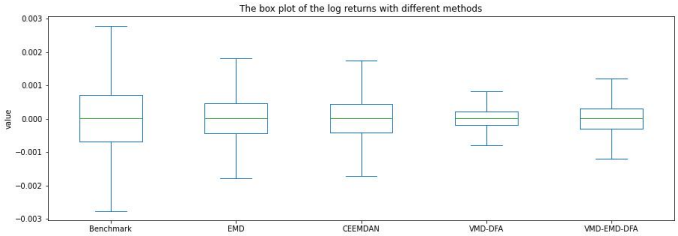

Figure 17: The box plot of R_t and $\hat{R}_t$ [2]

Table 9 shows the time spent by the four noise reduction methods to denoise and reconstruct the data in the training set. EMD takes much less time than the other methods, with CEEMDAN taking the longest.

Method	EMD	CEEMDAN	VMD-DFA	VMD-EMD-DFA
Cost time	6.0244	451.0556	18.7608	37.6316

Table 9: The cost time of different denoising methods

In summary, the four noise reduction methods rank from best to worst in the classification model: EMD, CEEMDAN, VMD-EMD-DFA and VMD-DFA.

[2]Benchmark: The F1-score line of the Model 1

6 Conclusion

Since the financial market is full of noise, which is the core problem of financial analysis and research, this paper aims to compare the noise reduction effect of four noise reduction methods based on EMD, CEEMDAN and VMD. By applying EMD, CEEMDAN and VMD, the time series are decomposed into different components, and then the noise is removed and the signal is reconstructed according to certain criteria. In order to evaluate the effectiveness of noise reduction, we conduct tests in three aspects: simulation experiment, prediction and classification. The performance of the four noise reduction methods is verified and compared several times. The main conclusions are as follows:

- In the simulation experiments, the comprehensive performance of EMD is the best for the trend-stationary analog signal. EMD can significantly improve the SNR and reduce the mean squared error. For non-stationary signals, VMD-DFA has the best performance.
- As for predictions, we test four methods in HSI, GDAXI, Crude Oil, Apple and Bitcoin datasets. The denoised data with EMD and CEEMDAN can improve the accuracy of the prediction model. However, VMD-DFA and VMD-EMD-DFA can improve the accuracy of forecasting models in stock index datasets, but such improvement is not stable. On most data sets, these two methods are less effective than raw data.
- As for classification, in the high-frequency Bitcoin dataset, four denoising methods can significantly improve the classification accuracy of the model when the classification labels are most complex. Among them, EMD has the best comprehensive performance in the classification model.

In conclusion, the algorithm of EMD is the simplest and takes the shortest time, and EMD performs well in all three experiments. The overall performance of CEEMDAN is second only to EMD, but it takes much longer than the other three methods. The results of VMD-DFA and VMD-EMD-DFA on classification and prediction models are not stable, that is, the results are mixed compared to the baseline model.

6.1 Future work

Specifically, the paper can be improved from the following two aspects:

- Try prediction models with different architectures. Since repeated decomposition in the current architecture is highly affected by end effects, different model architectures can be tried to avoid this. For example, EMD can be used to decompose the original signal into different sequences and input these sequences into the model to predict separately. Finally, deep learning models can be used to learn the optimal proportion of each sequence and add them together to get the reconstructed data, so as to reduce the impact of end effects.

- In the classification model, although EMD has improved the accuracy of classification, it has not reached the degree of application. Small errors can cause considerable losses in practical applications. In order to further improve the F1-score, we can also redesign the model structure or look at the confusion matrix to further clarify the defects.

References

[1] Y.-W. Si and J. Yin, "Obst-based segmentation approach to financial time series," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2581–2596, 2013.

[2] F. Black, "Noise," *The journal of finance*, vol. 41, no. 3, pp. 528–543, 1986.

[3] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," *The journal of Finance*, vol. 25, no. 2, pp. 383–417, 1970.

[4] J. B. De Long, A. Shleifer, L. H. Summers, and R. J. Waldmann, "Noise trader risk in financial markets," *Journal of political Economy*, vol. 98, no. 4, pp. 703–738, 1990.

[5] G. W. Brown, "Volatility, sentiment, and noise traders," *Financial Analysts Journal*, vol. 55, no. 2, pp. 82–90, 1999.

[6] W. Sun, S. Rachev, and F. J. Fabozzi, "Fractals or iid: evidence of long-range dependence and heavy tailedness from modeling german equity market returns," *Journal of Economics and Business*, vol. 59, no. 6, pp. 575–595, 2007.

[7] Y. Aït-Sahalia and J. Jacod, "Analyzing the spectrum of asset returns: Jump and volatility components in high frequency data," *Journal of Economic Literature*, vol. 50, no. 4, pp. 1007–50, 2012.

[8] S.-T. Au, R. Duan, S. G. Hesar, and W. Jiang, "A framework of irregularity enlightenment for data pre-processing in data mining," *Annals of Operations Research*, vol. 174, no. 1, pp. 47–66, 2010.

[9] M. S. Grewal and A. P. Andrews, "Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]," *IEEE Control Systems Magazine*, vol. 30, no. 3, pp. 69–78, 2010.

[10] S. E. Noel, Y. J. Gohel, and H. H. Szu, "Wavelet detection of singularities in the presence of fractal noise," in *Wavelet Applications IV*, vol. 3078, pp. 374–383, SPIE, 1997.

[11] T.-C. Hsung, D.-K. Lun, and W.-C. Siu, "Denoising by singularity detection," *IEEE Transactions on Signal Processing*, vol. 47, no. 11, pp. 3139–3144, 1999.

[12] V. Pakrashi, B. Basu, and A. O'Connor, "A statistical measure for wavelet based singularity detection," *Journal of vibration and acoustics*, vol. 131, no. 4, 2009.

[13] S. Li, "Volatility spillovers in the csi300 futures and spot markets in china: Empirical study based on discrete wavelet transform and var-bekk-bivariate garch model," *Procedia Computer Science*, vol. 55, pp. 380–387, 2015.

[14] D. B. Percival and A. T. Walden, *Wavelet methods for time series analysis*, vol. 4. Cambridge university press, 2000.

[15] R. Gençay, F. Selçuk, and B. J. Whitcher, *An introduction to wavelets and other filtering methods in finance and economics*. Elsevier, 2001.

[16] Y. Ait-Sahalia, P. A. Mykland, and L. Zhang, "How often to sample a continuous-time process in the presence of market microstructure noise," *The review of financial studies*, vol. 18, no. 2, pp. 351–416, 2005.

[17] F. M. Bandi and J. R. Russell, "Microstructure noise, realized variance, and optimal sampling," *The Review of Economic Studies*, vol. 75, no. 2, pp. 339–369, 2008.

[18] E. W. Sun and T. Meinl, "A new wavelet-based denoising algorithm for high-frequency financial data mining," *European Journal of Operational Research*, vol. 217, no. 3, pp. 589–599, 2012.

[19] J. Fan and Y. Wang, "Multi-scale jump and volatility analysis for high-frequency financial data," *Journal of the American Statistical Association*, vol. 102, no. 480, pp. 1349–1362, 2007.

[20] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C. C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, vol. 454, no. 1971, pp. 903–995, 1998.

[21] V. Veeraiyan, R. Velayutham, and M. M. Philip, "Frequency domain based approach for denoising of underwater acoustic signal using emd," *Journal of Intelligent Systems*, vol. 22, no. 1, pp. 67–80, 2013.

[22] Y. Li, H. Han, and Y. Li, "A new hht-based denoising algorithm for financial time series data mining," in *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pp. 397–401, IEEE, 2019.

[23] N. Nava, T. Di Matteo, and T. Aste, "Anomalous volatility scaling in high frequency financial data," *Physica A: Statistical Mechanics and its Applications*, vol. 447, pp. 434–445, 2016.

[24] T. Wang, M. Zhang, Q. Yu, and H. Zhang, "Comparing the applications of emd and eemd on time–frequency analysis of seismic signal," *Journal of Applied Geophysics*, vol. 83, pp. 29–34, 2012.

[25] M. A. Kabir and C. Shahnaz, "Denoising of ecg signals based on noise reduction algorithms in emd and wavelet domains," *Biomedical Signal Processing and Control*, vol. 7, no. 5, pp. 481–489, 2012.

[26] S. Agrawal and A. Gupta, "Fractal and emd based removal of baseline wander and powerline interference from ecg signals," *Computers in biology and medicine*, vol. 43, no. 11, pp. 1889–1899, 2013.

[27] K. Khaldi, M. T.-H. Alouane, and A.-O. Boudraa, "A new emd denoising approach dedicated to voiced speech signals," in *2008 2nd International Conference on Signals, Circuits and Systems*, pp. 1–5, IEEE, 2008.

[28] R. Shao, W. Hu, and J. Li, "Multi-fault feature extraction and diagnosis of gear transmission system using time-frequency analysis and wavelet threshold de-noising based on emd," *Shock and Vibration*, vol. 20, no. 4, pp. 763–780, 2013.

[29] N. Tsakalozos, K. Drakakis, and S. Rickard, "A formal study of the nonlinearity and consistency of the empirical mode decomposition," *Signal Processing*, vol. 92, no. 9, pp. 1961–1969, 2012.

[30] D. Ren, S. Yang, Z. Wu, and G. Yan, "Evaluation of the emd end effect and a window based method to improve emd," in *2006 International Technology and Innovation Conference (ITIC 2006)*, pp. 1568–1572, IET, 2006.

[31] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," *Advances in adaptive data analysis*, vol. 1, no. 01, pp. 1–41, 2009.

[32] S. Gaci, "A new ensemble empirical mode decomposition (eemd) denoising method for seismic signals," *Energy Procedia*, vol. 97, pp. 84–91, 2016.

[33] J.-R. Yeh, J.-S. Shieh, and N. E. Huang, "Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method," *Advances in adaptive data analysis*, vol. 2, no. 02, pp. 135–156, 2010.

[34] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 4144–4147, IEEE, 2011.

[35] M. Rezaie-Balf, S. R. Naganna, O. Kisi, and A. El-Shafie, "Enhancing streamflow forecasting using the augmenting ensemble procedure coupled machine learning models: case study of aswan high dam," *Hydrological Sciences Journal*, vol. 64, no. 13, pp. 1629–1646, 2019.

[36] W. Lu, Y. Rui, Z. Yi, B. Ran, and Y. Gu, "A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting," *IEEE Access*, vol. 8, pp. 42042–42054, 2020.

[37] S. Dai, D. Niu, and Y. Li, "Daily peak load forecasting based on complete ensemble empirical mode decomposition with adaptive noise and support vector machine optimized by modified grey wolf optimization algorithm," *Energies*, vol. 11, no. 1, p. 163, 2018.

[38] J. Cao, Z. Li, and J. Li, "Financial time series forecasting model based on ceemdan and lstm," *Physica A: Statistical mechanics and its applications*, vol. 519, pp. 127–139, 2019.

[39] R. d. Luca Avila and G. De Bona, "Financial time series forecasting via ceemdan-lstm with exogenous features," in *Brazilian Conference on Intelligent Systems*, pp. 558–572, Springer, 2020.

[40] Y. Lin, Y. Yan, J. Xu, Y. Liao, and F. Ma, "Forecasting stock index price using the ceemdan-lstm model," *The North American Journal of Economics and Finance*, vol. 57, p. 101421, 2021.

[41] Y. Zhou, T. Li, J. Shi, and Z. Qian, "A ceemdan and xgboost-based approach to forecast crude oil prices," *Complexity*, vol. 2019, 2019.

[42] K. Dragomiretskiy and D. Zosso, "Variational mode decomposition," *IEEE transactions on signal processing*, vol. 62, no. 3, pp. 531–544, 2013.

[43] H. Niu, K. Xu, and W. Wang, "A hybrid stock price index forecasting model based on variational mode decomposition and lstm network," *Applied Intelligence*, vol. 50, no. 12, pp. 4296–4309, 2020.

[44] S. Lahmiri, "Intraday stock price forecasting based on variational mode decomposition," *Journal of Computational Science*, vol. 12, pp. 23–27, 2016.

[45] Y. Zhang, B. Chen, G. Pan, and Y. Zhao, "A novel hybrid model based on vmd-wt and pca-bp-rbf neural network for short-term wind speed forecasting," *Energy Conversion and Management*, vol. 195, pp. 180–197, 2019.

[46] X. Zheng, S. Wang, and Y. Qian, "Fault feature extraction of wind turbine gearbox under variable speed based on improved adaptive variational mode decomposition," *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, vol. 234, no. 6, pp. 848–861, 2020.

[47] Q. Wu and H. Lin, "Daily urban air quality index forecasting based on variational mode decomposition, sample entropy and lstm neural network," *Sustainable Cities and Society*, vol. 50, p. 101657, 2019.

[48] F. Li, G. Ma, S. Chen, and W. Huang, "An ensemble modeling approach to forecast daily reservoir inflow using bidirectional long-and short-term memory (bi-lstm), variational mode decomposition (vmd), and energy entropy method," *Water Resources Management*, vol. 35, no. 9, pp. 2941–2963, 2021.

[49] S. Ghose, N. Singh, and P. Singh, "Image denoising using deep learning: Convolutional neural network," in *2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 511–517, IEEE, 2020.

[50] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder.," in *Interspeech*, vol. 2013, pp. 436–440, 2013.

[51] K. Yan, Y. Yu, C. Huang, L. Sui, K. Qian, and A. Asundi, "Fringe pattern denoising based on deep learning," *Optics Communications*, vol. 437, pp. 148–152, 2019.

[52] P. Flandrin, G. Rilling, and P. Goncalves, "Empirical mode decomposition as a filter bank," *IEEE signal processing letters*, vol. 11, no. 2, pp. 112–114, 2004.

[53] A. Mert and A. Akan, "Detrended fluctuation thresholding for empirical mode decomposition based denoising," *Digital signal processing*, vol. 32, pp. 48–56, 2014.

[54] C.-K. Peng, S. V. Buldyrev, S. Havlin, M. Simons, H. E. Stanley, and A. L. Goldberger, "Mosaic organization of dna nucleotides," *Physical review e*, vol. 49, no. 2, p. 1685, 1994.

[55] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[56] Z. Wu and N. E. Huang, "A study of the characteristics of white noise using the empirical mode decomposition method," *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 460, no. 2046, pp. 1597–1611, 2004.

[57] Y. Liu, G. Yang, M. Li, and H. Yin, "Variational mode decomposition denoising combined the detrended fluctuation analysis," *Signal Processing*, vol. 125, pp. 349–364, 2016.

[58] S. Lahmiri, "Comparing variational and empirical mode decomposition in forecasting day-ahead energy prices," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1907–1910, 2015.

[59] D. Kim and H.-S. Oh, "Emd: A package for empirical mode decomposition and hilbert spectrum.," *R J.*, vol. 1, no. 1, p. 40, 2009.

Declaration

I declare that this thesis is the solely effort of the author. I did not use any other sources and references than the listed ones. I have marked all contained direct or indirect statements from other sources as such.

Neither this work nor significant parts of it were part of another review process. I did not publish this work partially or completely yet. The electronic copy is consistent with all submitted copies.

Signature and date: Tiantian Zhang 2022.08.07

A Appendix

A.1 ACF and PACF diagrams of five financial assets

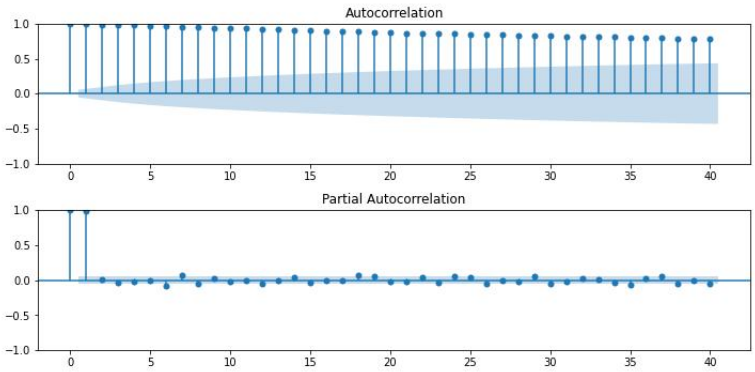

Figure 18: The the ACF and PACF of GDAXI

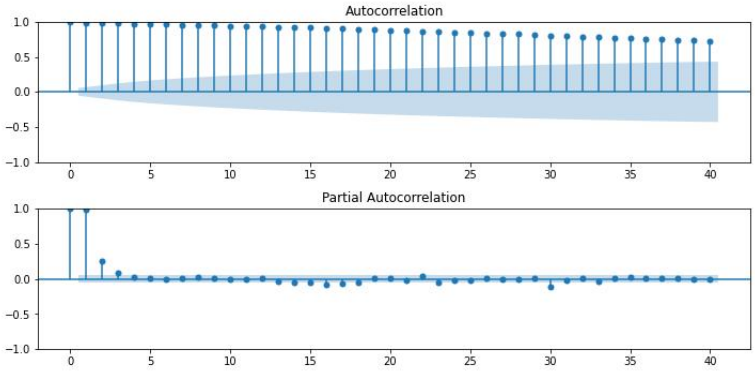

Figure 19: The the ACF and PACF of Crude oil

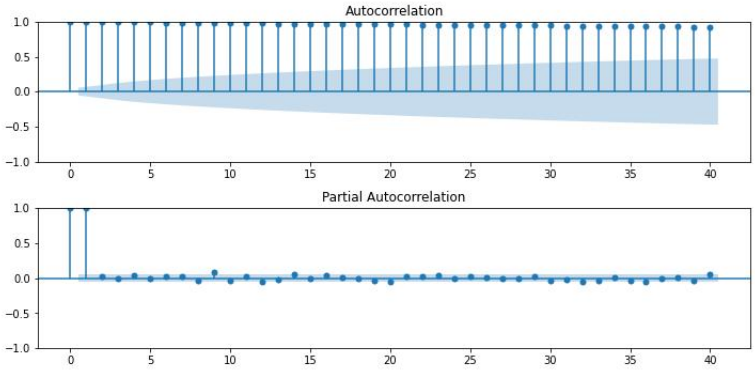

Figure 20: The the ACF and PACF of Apple

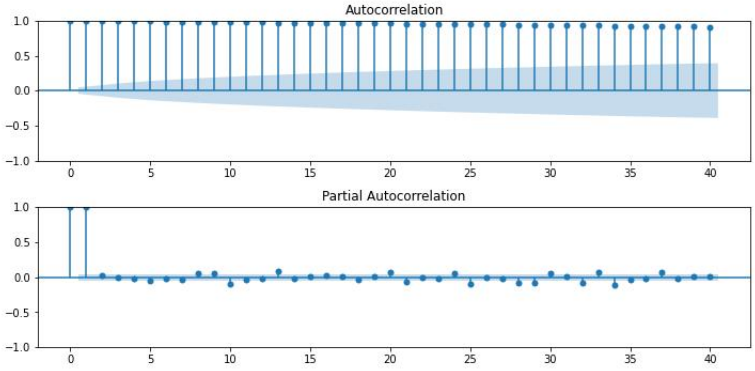

Figure 21: The the ACF and PACF of Bitcoin

A.2 Comparison between the denoised data and the pure data

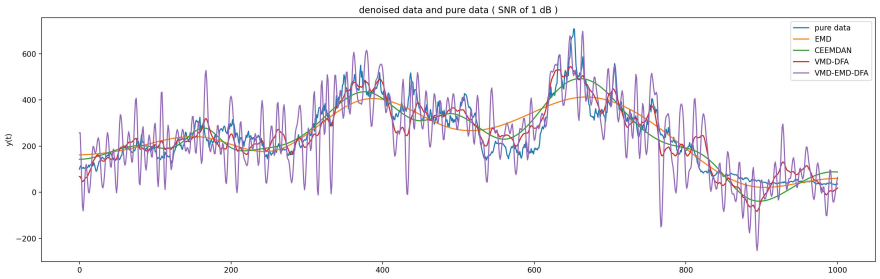

Figure 22: The denoised data of four methods and the pure data (SNR of 1 dB)

Figure 23: The denoised data of four methods and the pure data (SNR of 8 dB)

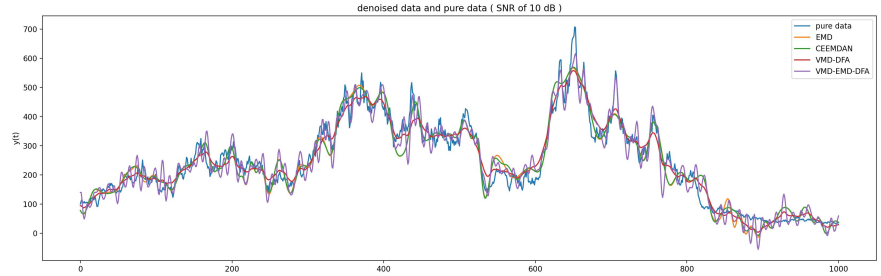

Figure 24: The denoised data of four methods and the pure data (SNR of 10 dB)