Multiscale Financial Signal Processing and Machine Learning

Zhengde Zhao

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2022

Reading Committee:
Siu-Tang Leung, Chair
Matthew Lorig
Ryan Donnelly

Program Authorized to Offer Degree:
Applied Mathematics

University of Washington

Abstract

Multiscale Financial Signal Processing and Machine Learning

Zhengde Zhao

Chair of the Supervisory Committee:
Siu-Tang Leung
Department of Applied Mathematics

Financial time series such as market indices and asset prices are shown to be driven by multiscale factors, ranging from long-term market regimes to rapid fluctuations. Multiscale analysis and signal processing not only reveal latent behaviors embedded in financial time series, but also help machine learning prediction tasks. In this thesis we focus on two different approaches tailored for daily and intraday financial time series respectively. In the first study, Hilbert-Huang transform is applied to daily prices and index values to reveal the underlying multiscale dynamics. In addition, a novel machine learning framework is proposed for identifying useful predictive features. An adaptive algorithm for highly nonstationary time series was introduced and applied to cryptocurrencies to show embedded structure and spectral properties. In the second study, we inspect the relations between statistical properties at different timescales, with the application to intraday high-frequency price data with noise. Functions describing the multiscale behaviors of volatility and correlation are defined and computed using empirical data. Models for high-frequency price processes are proposed and compared against empirical observations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

First and foremost, I would like to thank my PhD advisor Professor Tim Leung. Tim was the one who led me formally into the field of financial mathematics, back to the time when I was master student. He was an amazing advisor who provided me with the opportunities to get in touch with many interesting research topics, and also encouraged me to explore different ideas that attract me. From the high level, he shaped me with his insightful understanding about financial mathematics and related fields, as well as the general finance industry. His values making impact on real-world problems, which is something I could learn a lot from besides just knowledge. During my PhD career, I was not only able to do research in the academic setting, but also got valuable chance to apply my knowledge to real-world application. All of these not only were helpful for my PhD study, but left me a good asset for lifetime.

I also want to thank Matthew Lorig and Ryan Donnelly as my committee members for their advice to my PhD work and insightful discussion in my general and final exams. Matthew was also the professor for my stochastic course sequence, and helped me lay a good foundation for research in financial mathematics. I also learned a lot from Ryan as his teaching assistant for many quarters. Lastly, I want to thank the GSR Antonino Ferrante for being at my general and final exams and making insightful comments and discussions.

In addition to my committee members, I learned a lot from many other professors in The Department of Applied Mathematics as well as other departments across University of Washington, either being a student in their classes, or serving as a teaching assistant or research assistant. Especially I want to thank: Bernard Deconinck, Nathan Kutz, Thomas Rothvoss,

Aleksandr Aravkin, Steven Murray, Hong Qian, Zaid Harchaoui, Archis Ghate, Maryam Fazel, Ruanne Barnabas, Bahman Angoshtari. I also learned a lot from my fellow students, either in class, journal club, or in life. Special thank to Yang Zhou, Kelly Liu, Yuying Liu, Mo Zhao, Ying-Jen Yang, Natalie Wellen, Weston Barger, Katherine Lacy Owens, Roman Levin, Tyler Chen, Ryan Creedon. The Department of Applied Mathematics also provided me with an excellent program to pursue my study. I especially want to thank Lauren Lederer who gave me a lot of help during the six years I spent in the department.

My dear family members gave me a lot of help and support for my student life oversea and made me feel like home. I want to express my sincere appreciation to Jia and Scarlet for your care and love, and Richard and Melodie for your help and support. Another special thank to Preston for his support and always insightful discussions on all kinds of topics. Although not specifically mentioned, many other families and friends were also important for this period of my life.

Special thanks to my partner Meredith for her love during my PhD life at Seattle and beyond. Your understanding and support have always been important to me, even from across the ocean during this unusual time. Best wishes to you for your success in the upcoming PhD life!

Lastly, and most importantly, I would like to express my sincere appreciation to my parents, Jiahe and Weiying. You have always been the most important people in my life. Without your unconditional love and support for the 28 years in my life, I wouldn't make any of the achievements and be here today. During the past ten years when I left home to pursue higher education, you have always been there for me as the home. Even separated with geographical distance and haven't been able to see each other during this special time, I can always feel your greatest love from home.

DEDICATION

This PhD thesis is to my father, who has been a great professor and scholar.

You have the highest degree in my mind.

Chapter 1

INTRODUCTION

1.1 Multiscale Financial Signal Processing

Market observations and empirical studies have shown that financial time series such as market indices and asset prices are often driven by multiscale factors, ranging from long-term market regimes to rapid fluctuations (see [39, 58, 122, 77], among others). Stemming from the fundamental mechanics, different market players such as hedge funds and high-frequency trading firms target different horizons and operate on different timescales. All of these evidences call for methods to study and utilize the multiscale property of financial time series. One core objective is to understand their behaviors on different timescales, and how different scales are interconnected. For decades, researchers in the field developed various approaches to tackle the problem [23], ranging from model-based to data-driven, from time domain analysis to frequency domain analysis [59], and from linear, stationary models to nonlinear, nonstationary methods [75].

While the topic can be broad, in this thesis we narrow down to two seemingly distinct approaches, targeting two major types of financial time series: daily and intraday. In the first study, decomposition methods are applied to daily financial time series of various types. The challenges are low data frequency, weak dependency, and nonstationarity. We will show how empirical decomposition methods and spectral analysis help to reveal the underlying dynamics. In the second study, we turn to intraday high-frequency price data, facing the challenges of market microstructure noise and more delicate dependency in the time and scale domains. We will make necessary assumption and study how various statistical properties change with the timescale of measurement.

1.1.1 Nonstationary Decomposition and Spectral Analysis

Embedded with different timescales, many financial time series often exhibit nonstationary behaviors with trends and time-varying volatilities. These characteristics can hardly be captured by linear models and call for an adaptive and nonlinear approach for analysis. Traditionally, Fourier analysis has been a powerful tool in finding a spectral representation of time series. However, the Fourier bases, by assumption, are stationary and linear. On the other hand, nonstationary behaviors and nonlinear dynamics are often observed in financial time series. These characteristics can hardly be captured by linear models and call for an adaptive and nonlinear approach for analysis. For decades, methods based on short time Fourier transform and wavelets have been developed and applied to nonstationary time series, but there are still challenges in capturing nonlinear dynamics [54], and the often prescribed assumptions make the methods not fully adaptive. This gives rise to the need for an adaptive and nonlinear approach for analysis.

One alternative approach in adaptive time series analysis is the Hilbert-Huang transform (HHT) [54]. The HHT method can decompose any time series into oscillating components with nonstationary amplitudes and frequencies using the empirical mode decomposition (EMD). This fully adaptive method provides a multiscale decomposition for the original time series, giving richer information about its behavior. The instantaneous frequency and instantaneous amplitude of each component are later extracted using the Hilbert transform. The decomposition onto different timescales also allows for reconstruction up to different resolutions, providing a smoothing and filtering tool that is ideal for noisy financial time series. The method of HHT and its variations have been applied in numerous fields, from engineering to geophysics [56]. Applications of HHT to finance date back to the work by Huang and co-authors on modeling mortgage rate data [55]. The empirical mode decomposition (EMD) has been used for financial time series forecasting [95, 118] and for examining the correlation between financial time series [94].

In terms of methodology, there have been several studies on the variations and alternatives of EMD, including optimization-based methods [47, 48, 49, 51], ensemble empirical-mode decomposition (EEMD) for tackling mode mixing [121], and other noise-assisted approaches [124]. For financial time series with high level of intrinsic noise, we chose to use and improve the complementary ensemble empirical mode decomposition (CEEMD). Like EMD, CEEMD decomposes any time series – stationary or not – into a number of intrinsic mode functions representing the local characteristics of the time series at different timescales, but the timescale separation is improved by resolving mode mixing in EMD [53]. The noise-assisted approach is also more robust to intrinsic noise in the data. CEEMD have been found to be useful for forecasting [97, 112] and signal processing [76]. More recent application of EMD and the noise-assisted variations includes [111, 92, 91, 93].

1.1.2 Multiscale Volatility and Correlation

While many multiscale studies, including the ones introduced above, have been focusing on daily data, the interest in intraday price has rose recently. Explosively increasing amount of high frequency data emerging in the field allows and demands more complicated models [87]. However, models for low-frequency daily data can hardly fit the complex structure of intraday high frequency data [38], especially for volatility modeling [10]. How volatility and correlation behave at middle or high frequency is important for understanding the multiscale behavior of intraday price movement. The scaling property is also connected with the dependency structure, such as trending or mean-reverting, and therefore leads to implications in trading strategies [73] and a wide range of financial applications.

In order to better understand the multiscale properties, the scaling concept originated from complex system was introduced into financial mathematics [90]. As the base example, the original Brownian motion stock model [100] intrinsically determines return distribution at any timescale, as a result of the independent increment and memory-less properties, which

were also backed by the efficient market hypothesis (EMH) [37]. This standard model, however, has been questioned and modified afterwards, starting with the work of [81]. The generalization to fractional Brownian motion [84] based on self-similar introduced a larger class of stochastic processes, allowing for new scaling properties and long memory process. However, it also brought challenge to EMH and set room for arbitrage opportunities [110]. With the debates going on, fractional Brownian motion has become an important tool in financial modeling [99] including trading strategies [44].

On the other hand, as price dynamics and volatility modeling become more complex going into the intraday and high frequency region, researchers turned to empirical and non-parametric approaches to study the stochastic properties of return, including the measuring of realized volatility [14, 10]. However, people found that it is not optimal to directly use all the data at the highest available frequency due to the presence of market microstructure noise [4, 11]. Different techniques were designed for an accurate measurement of realized volatility for the "true price" [106]. One of the key ideas was to measure realized volatility at different timescales and combine the results to discover the latent value [126]. This multiscale technique was generalized to a wider class of models and other return statistics [38, 3, 18].

At first glance, the fractional Brownian motion model and the multiscale realized volatility estimator were designed for different purposes and are under different assumptions. Interestingly, they share some very similar form in terms of showing a multiscale behavior of randomness. This similarity encourages us to formally propose definitions to describe the multiscale properties of price process, and explore a wider class of models for high frequency financial time series. On the other hand, this indicates that the market microstructure noise can potentially bias estimation of the scaling parameters. In a related recent study, [28] showed one form of microstructure noise due to the discrepancy between the realized and the instantaneous volatility can lead to bias in Hurst exponent [57] estimation. The necessity

of new multiscale models is also justified by real-world observations in high-frequency data.

1.2 Multiscale Analysis and Machine Learning

With increasing volume of financial data and success of modern machine learning algorithms, the application of machine learning to finance is of high interest. However, even though machine learning has been proved to be successful in many fields such as computer vision and natural language processing, financial time series forecasting has been ever challenging. Multiscale analysis has been shown to enhance machine learning prediction on other time series, for example HHT based prediction was used for SVM and neural network in the electrical engineering field [69, 68, 67], and EMD was adopted for financial machine learning [97, 112].

While multiscale signal processing methods such as EMD and HHT have been shown to help machine learning, significant drawbacks remain. Firstly, the mathematical properties of the multiscale features from EMD and HHT were not fully investigated, and it is not clear which features are useful for prediction. Secondly, in most of the previous studies using EMD or HHT as forecasting features, the decomposition was pre-implemented on the whole time span, including training and testing data, which is not practical when doing real-time forecasting. Moreover, both EMD and HHT lend information from the past and the future to compute the current values, which can cause information leakage in the forecasting process. On the other hand, current versions of extrapolating prediction [95] are subject to the end effect of EMD, which makes the decomposition error quite high at the end of the time span. It is unknown from previous studies of how to overcome the challenge of end effect.

On the other hand, the key to improve financial machine learning is to understand the main obstacles. By its nature, financial time series are noisy, nonstationary, with subtle dependency structure. One question posed by bias-variance trade-off in machine learning is: should we use longer history for better fitting accuracy, or should we use short-term data to capture the timely information? Studying of multiscale behavior provides one aspect to

answer the question, and the scaling properties are also related to the dependency structure.

1.3 Thesis Outline

In Chapter 2 we present the method of complementary ensemble empirical mode decomposition (CEEMD) and Hilbert-Huang transform (HHT) for analyzing nonstationary financial time series. This noise-assisted approach decomposes any time series into a number of intrinsic mode functions, along with the corresponding instantaneous amplitudes and instantaneous frequencies. Using HHT, we generated a collection of new features and integrated them into machine learning models, such as regression tree ensemble, support vector machine (SVM), and long short-term memory (LSTM) neural network. Using empirical financial data, we compared several HHT-enhanced machine learning models in terms of forecasting performance.

We experiment the methods using daily data for the S&P 500 index, CBOE volatility (VIX) index, SPDR gold exchange-traded fund (GLD), and 10-year treasury yield (TNX). We show that while these time series have contrasting path behaviors (upward trending vs mean-reverting), their instantaneous energy-frequency spectra reveal that they share very similar average frequencies on both long and short timescales. In feature selection part, our comparison analysis identifies the most useful HHT features for prediction. We highlight the improved prediction performance of our method when benchmarked to the prediction using only the original time series. We then discussed the end effect of EMD and proposed a novel framework of extrapolating forecasting with end effect correction. The prediction was implemented on the above-mentioned dataset. Our results are potentially useful not only for prediction but also portfolio construction [125].

In Chapter 3, we present a novel method of applying adaptive complementary ensemble empirical mode decomposition (ACE-EMD) and Hilbert spectral analysis to study cryptocurrency price dynamics. As an emerging asset class, cryptocurrencies have a number of

salient features compared to traditional equities, including significantly higher volatility and rapidly changing directional trends [65, 13, 62, 104, 20, 50, 61, 114, 40]. Our noise-assisted approach is adaptive to the time-varying volatility of each cryptocurrency price evolution. Different combinations of modes allow us to reconstruct the time series using components of different timescales. We then apply Hilbert spectral analysis to define and compute the associated instantaneous energy-frequency spectrum to illustrate the properties of various timescales embedded in the original time series. In particular, we derive and compute the central frequency associated with a collection of cryptocurrencies as well as equity indices, which allows us to observe the distinct behaviors of cryptocurrency prices.

In Chapter 4 we define functions to describe multiscale volatility in asset price and investigated their behaviors in common classes of financial models. We then proposed models for intraday high-frequency price with their multiscale properties. Numerical estimation of model parameters was then proposed, including modified algorithm to evaluate the Hurst exponent [85], which describes the dependency structure of the time series and how the variability changes with the timescale. The value of the Hurst exponent is associated with the smoothness of the random process, long-memory, and fractal dimension [82], therefore widely used in financial modeling [42, 86, 115]. Experimental results on real-world high-frequency price data were shown, including multiscale volatility behavior and modified Hurst exponent estimation under microstructure noise.

In Chapter 5 we continued to study multiscale behaviors of multiple assets, focusing on how correlation change with the timescale. We first defined the multiscale correlation function and showed its behavior in common stochastic models. The multscale models for high-frequency price were generalized to multivariate forms, with their correlation studied. In the end, we provided experimental results of correlation in intraday high-frequency price data, and discussed the potential and limitation of our models.

Chapter 2

MULTISCALE FEATURES FOR MACHINE LEARNING

2.1 Overview

This chapter reviews the concepts of Hilbert-Huang transform (HHT) and multiscale features for machine learning. The empirical mode decomposition (EMD) and its variations are first introduced, with algorithms provided for intrinsic mode function extraction. The Hilbert spectral analysis is then reviewed to show the time-frequency information we can obtain from a time series. Next, we introduce the multiscale HHT features and how to integrate them into financial machine learning. The training and evaluation frameworks for in sample and extrapolating prediction are proposed, combined with a variety of machine learning models, including regression tree ensemble (RTE), support vector machine (SVM), and specially designed long short-term memory (LSTM) neural network. Lastly, we provide forecasting performance of the HHT-enhanced machine learning models on different types of financial time series, and discuss the feature selection problem.

2.2 Multiscale Decomposition Methodology

In this section, we present the methodology for processing a financial time series to generate its multiscale time-frequency profile. The major components are the ensemble empirical mode decomposition (EMD) and Hilbert spectral analysis. We also discuss some significant drawbacks of the algorithm and methods to overcome them. A brief comparison with related methods is provided at the end of this section.

2.2.1 Empirical Mode Decomposition

Empirical mode decomposition (EMD) is the first step of the multistage HHT procedure. For any given time series $x(t)$ observed over a period of time $[0, T]$, we decompose it in an iterative way into a finite sequence of oscillating components $c_1(t), \cdots, c_n(t)$, plus a non-oscillatory trend called the residue term $r_n(t)$. Precisely, we have

$$x(t) = \sum_{j=1}^{n} c_j(t) + r_n(t). \tag{2.1}$$

To ensure that $c_j(t)$ have the proper oscillatory properties, the concept of intrinsic mode function (IMF) was proposed [54]. The intrinsic mode functions are real functions in time that admit well-behaved and physically meaningful Hilbert transform. Mathematically, each IMF is defined by the following two criteria:

- No local oscillation: the number of extrema and the number of zero crossings must be equal or at most differ by one.

- Symmetric: the maxima of the function defined by the upper envelope and the minima defined by the lower envelope must sum up to zero at any time $t \in [0, T]$.

The definition of an IMF guarantees pure oscillation while allowing time-varying frequency and amplitude. Mathematically, an IMF $c(t)$ admits the expression

$$c(t) = a(t)\cos(\theta(t)), \tag{2.2}$$

where $a(t) \geq 0$ is the instantaneous amplitude, and $\theta(t)$ is the phase function with $\theta'(t) \geq 0$. Such property makes it ready for time-frequency analysis under the Hilbert spectral analysis to be discussed later.

As is standard [54, 109], we consider a sifting process that decomposes any time series into a finite set of IMFs that oscillate on different timescales, plus a non-oscillatory residue term. The key idea of the method is as follows: look for the finest oscillation by finding all local maxima and minima, and then subtract the remaining trend, until the rest of the function

satisfies the IMF conditions. Each IMF discovered is removed sequentially from the time series until only a non-oscillatory residue remains. The residue is a constant or monotonic function, or has at most one maximum or minimum. The algorithm is summarized as follows:

- Initialize the residue term as $r_0(t) = x(t)$ and set $j = 1$.
- While $r_{j-1}(t)$ does not satisfies the non-oscillatory condition, i.e. $r_{j-1}(t)$ has more than one maximum or minimum, do the following sifting process to extract the oscillation as an IMF denoted by $c_j(t)$.
- Initialize the component as $c_j(t) = r_{j-1}(t)$.
  - Interpolate the maxima of $c_j(t)$ using a cubic spline as the upper envelope $u(t)$, and interpolate the minima of $c_j(t)$ using a cubic spline as the lower envelope $l(t)$. Compute the mean of the upper and lower envelopes $m(t) = \frac{1}{2}(u(t) + l(t))$.
  - Iterate $c_j(t) \leftarrow c_j(t) - m(t)$.
  - Stop when $c_j(t)$ satisfies the criteria of an IMF. Let $c_j(t)$ be the j-th component, and iterate $r_j(t) = r_{j-1}(t) - c_j(t)$ and $j \leftarrow j + 1$.
- Return the IMFs $c_1(t), \cdots, c_n(t)$, and residue $r_n(t)$.

The sifting process described above is purely empirical and adaptive with very little assumption on the temporal change, making it ideal for nonstationary time series.

2.2.2 Ensemble Variations of EMD

Even though the EMD algorithm is designed for decomposing nonstationary time series into components from high frequency to low frequency, it has some significant drawbacks in practice. One of them is the phenomenon of mode mixing [53], which is defined as either one IMF consisting of widely disparate scales, or signals of similar scales residing in several IMF components [121]. Such a phenomenon poses potential challenges on the interpretation

of the IMFs and makes the construction of machine learning features rather difficult. This problem is particularly relevant for financial time series when high degree of nonstationarity and noise are observed.

To resolve the mode mixing issue, we consider the ensemble empirical mode decomposition (EEMD) proposed by [121] as the variation for our analysis. EEMD is a noise-assisted signal processing technique that extracts each mode from an ensemble mean computed based on N trials. In each trial i, an i.i.d. white noise w_i with a zero mean and finite variance σ is added to the original time series $x(t)$, and $x(t) + w_i(t)$ is referred to as the *signal* in this trial. The original EMD algorithm is then applied to the signal, outputting the IMFs $c_{ij}(t),\ j = 1, \cdots, n$, and the residual term $r_{in}(t)$. Finally, the ensemble means of the IMFs and residual terms across all the N trails is regarded as the true mode extraction. The resulting ensemble mean of the IMF components are given by

$$c_j(t) = \frac{1}{N}\sum_{i=1}^{N} c_{ij}(t), \quad r_n = \frac{1}{N}\sum_{i=1}^{N} r_{in}(t). \tag{2.3}$$

Using Eq. (2.1), we can write

$$\sum_{j=1}^{n} c_{ij}(t) + r_{in}(t) = x(t) + w_i(t). \tag{2.4}$$

Then, the ensemble mean of the IMF components sums up to

$$\sum_{j=1}^{n} c_j(t) + r_n(t) = \frac{1}{N}\sum_{i=1}^{N}(x(t) + w_i(t)) = x(t) + \frac{1}{N}\sum_{i=1}^{N} w_i(t)\,.$$

Therefore, the components from the ensemble mean provide a decomposition which converges to the original time series $x(t)$ almost surely according to the strong law of large numbers. The convergence rate is $\mathcal{O}(\frac{1}{\sqrt{N}})$, which means a large ensemble size is typically desired if possible. Implementing EMD with a large number of trials can be potentially prohibitive in terms of computational cost and speed. To address this issue, an alternative called the complementary ensemble empirical mode decomposition (CEEMD) method is introduced [124]. CEEMD expands the ensemble by adding a complementary negative noise $-w_i(t)$ for

each trial, expanding the total ensemble size to $2N$. Consequently, the components from the ensemble mean sum up to equal the original time series:

$$\sum_{j=1}^{n} c_j(t) + r_n(t) = x(t) + \frac{1}{2N}\sum_{i=1}^{N}(w_i(t) - w_i(t)) = x(t). \tag{2.5}$$

This holds regardless of the choice of N, thus reducing the need for a very large ensemble size. These favorable properties make CEEMD the default when using EEMD in many applications. Besides reducing mode mixing, both EEMD and CEEMD are also more robust to intrinsic noise in the original data, as these methods automatically average out extra independent noises in the process. Due to the high non-stationarity and noise level in financial time series, we use CEEMD as the noise-assisted decomposition to resolve the mode mixing problem. Results in the rest of this chapter are derived from the CEEMD algorithm.

2.2.3 Hilbert Spectral Analysis

By definition, an IMF lends itself to Hilbert spectral analysis, the second stage of HHT. An oscillating real-valued function can be viewed as the projection of an orbit on the complex plane onto the real axis. For any function in time $X(t)$, its Hilbert transform is given by

$$Y(t) = \mathcal{H}[X](t) := \frac{1}{\pi}\int_{-\infty}^{+\infty} \frac{X(s)}{t-s}\, ds, \tag{2.6}$$

where the improper integral is defined as the Cauchy principle value. The transform exists for any function in L^p [113]. As a result, $Y(t)$ provides the complementary imaginary part of $X(t)$ to form an analytic function in the upper half-plane defined by

$$Z(t) = X(t) + iY(t) = a(t)e^{i\theta(t)}, \tag{2.7}$$

where

$$a(t) = \|Z(t)\| = \sqrt{X^2(t) + Y^2(t)}, \tag{2.8}$$

$$\theta(t) = \arg Z(t) = \arctan(\frac{Y(t)}{X(t)}). \tag{2.9}$$

For a function of the form in Eq. (2.2), if the amplitude $a(t)$ and the frequency $\theta'(t)$ are slow modulations, the Hilbert transform will give a $\pi/2$ shift to the phase $\theta(t)$ [15, 98]. Therefore, the $a(t)$ given by Eq. (2.8) is exactly the instantaneous amplitude, and the $\theta(t)$ given by Eq. (2.9) is exactly the instantaneous phase function. The instantaneous frequency is then defined as the 2π-standardized rate of change of the phase function, that is,

$$f(t) = \frac{1}{2\pi}\dot{\theta}(t) = \frac{1}{2\pi}\frac{d}{dt}\left(\arctan(\frac{Y(t)}{X(t)})\right). \tag{2.10}$$

The definition of IMF ensures a well-defined Hilbert transform, meaning that $a(t)$ and $f(t)$ reflect the amplitude and frequency of the oscillation. Applying Hilbert transform to each of the IMF components individually yields a sequence of analytic signals [54]:

$$c_j(t) + i\mathcal{H}[c_j](t) = c_j(t) + i\hat{c}_j(t) = a_j(t)e^{i\theta_j(t)}, \tag{2.11}$$

for $j = 1, \cdots, n$. We refer to the $c_j(t) + i\hat{c}_j(t),\ j = 1, \cdots, n$ as the complex IMFs, where the IMF components are the real parts, and the corresponding Hilbert transforms serve as the imaginary parts. In turn, the original time series can be represented as

$$x(t) = \mathfrak{Re}\ \sum_{j=1}^{n} a_j(t)e^{i\int^t 2\pi f_j(s)ds} + r_n(t). \tag{2.12}$$

This decomposition can be viewed as a sparse spectral representation of the time series with time-varying amplitude and frequency. In other words, each IMF represents a generalized Fourier expansion that are suitable for nonlinear and nonstationary financial time series. The sequence of instantaneous amplitudes and instantaneous frequencies can be represented as the sparse Hilbert spectrum:

$$H(f,t) = \sum_{j=1}^{n} H_j(f,t), \quad \text{where } H_j(f,t) = \begin{cases} a_j(t), & f = f_j(t), \\ 0, & \text{otherwise.} \end{cases} \tag{2.13}$$

Define the instantaneous energy of the j-th component as

$$E_j(t) = |a_j(t)|^2. \tag{2.14}$$

We examine the behavior of the Hilbert spectrum through the pair $(f_j(t), E_j(t))$, which fits better with machine learning models for time series forecasting.

2.2.4 Implementation with Financial Data

In this section, we visualize implementation on four financial time series of distinct type, namely the S&P 500 index, the CBOE volatility index VIX, the gold price exchange-traded fund (ETF) GLD, and the 10-year treasury yield (TNX), over a 10-year period from 4/1/2010 to 3/31/2020. For S&P 500 and GLD, let $s(t)$ be the value of the time series and $x(t) := \log(s(t))$ is the log price. For VIX and 10-year treasury rate, $x(t)$ is the time series itself. In each case, CEEMD is applied to $x(t)$ to extract the IMFs and residual from the time series, the Hilbert spectrum of which is later computed with HHT. To reduce the error in instantaneous frequency computation, we have used the robust locally weighted scatter plot smoothing (robust LOWESS) [25].

Fig. 2.1 shows the IMFs extracted using CEEMD. The top row in each plot shows the original time series $x(t)$, followed by the IMFs with decreasing frequencies, ended with the residual term showing the overall trend. The modes on different scales indicate different behavior of the time series. Taking early 2020 for example, when S&P 500 began the sharp decline on 2/21/2020, modes 3 and 4 also show notable peak around the time period, but on earlier dates (2/25/2020 and 1/21/2020). This indicates that a mode on a longer timescale signals trends earlier than higher frequency modes. The opposite behavior is observed for VIX – the lower-frequency modes exhibit a steep increase towards the day when VIX spikes.

Using formula (2.6), we implement Hilbert transform to the IMFs extracted using CEEMD. Following Eq. (2.11), the IMF components are taken as the real parts and the corresponding Hilbert transform are the imaginary parts, plotted together in Fig. 2.2. We see that each real part (IMF component) and the corresponding imaginary part (Hilbert transform) are oscillations coupled together, where negative (positive) Hilbert transform corresponds to a increasing (decreasing) IMF component. The coupling behavior indicates the complex IMFs can be useful for time series forecasting.

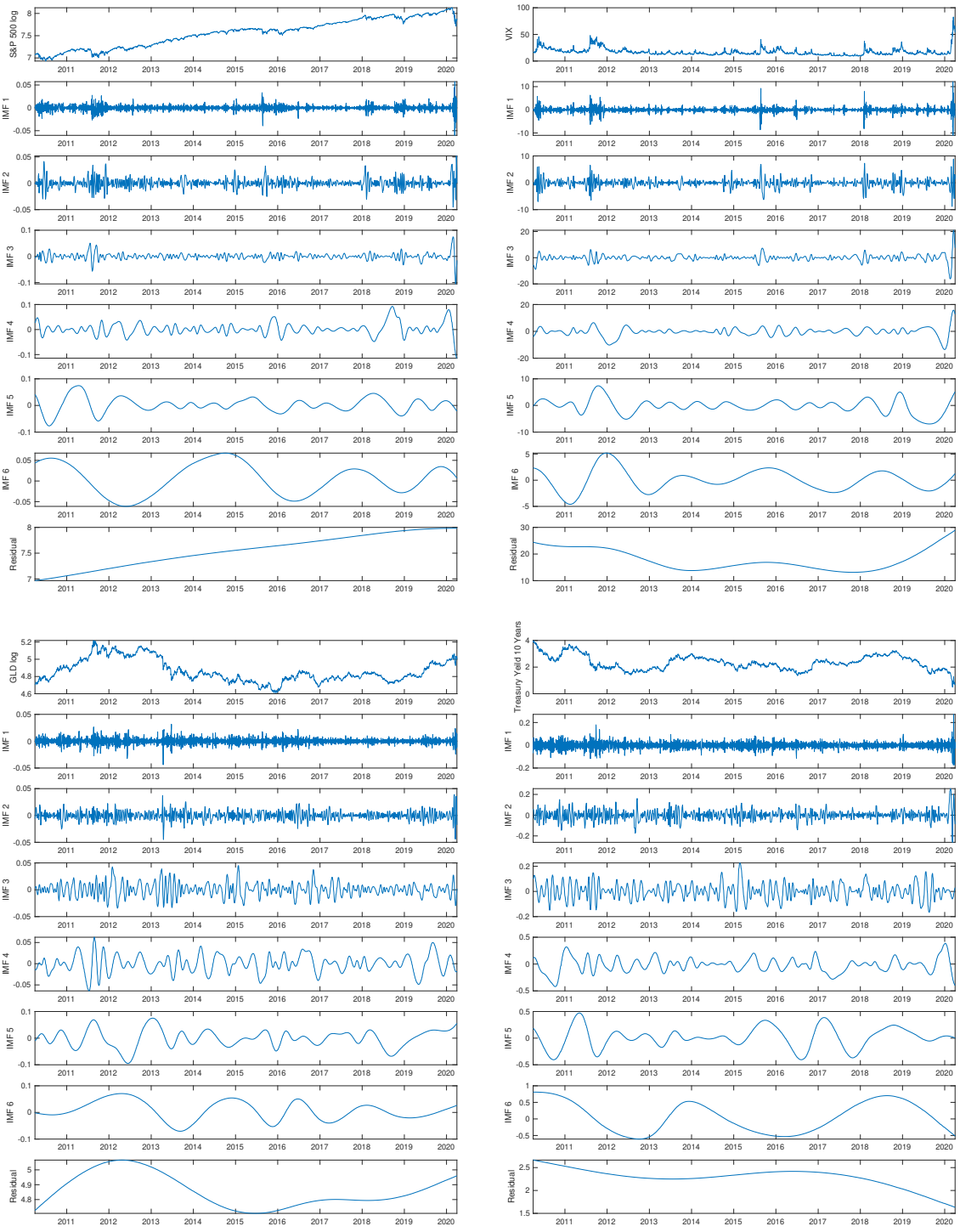

Figure 2.1: IMFs and residual terms extracted from CEEMD of S&P 500 (top left), VIX (top right), GLD (bottom left), and 10-year treasury yield (bottom right), from 4/1/2010 to 3/31/2020. The top row in each plot shows the original time series. The second to last but one rows show the IMF modes of the corresponding time series. The bottom row of each plot shows the residual term of the time series.

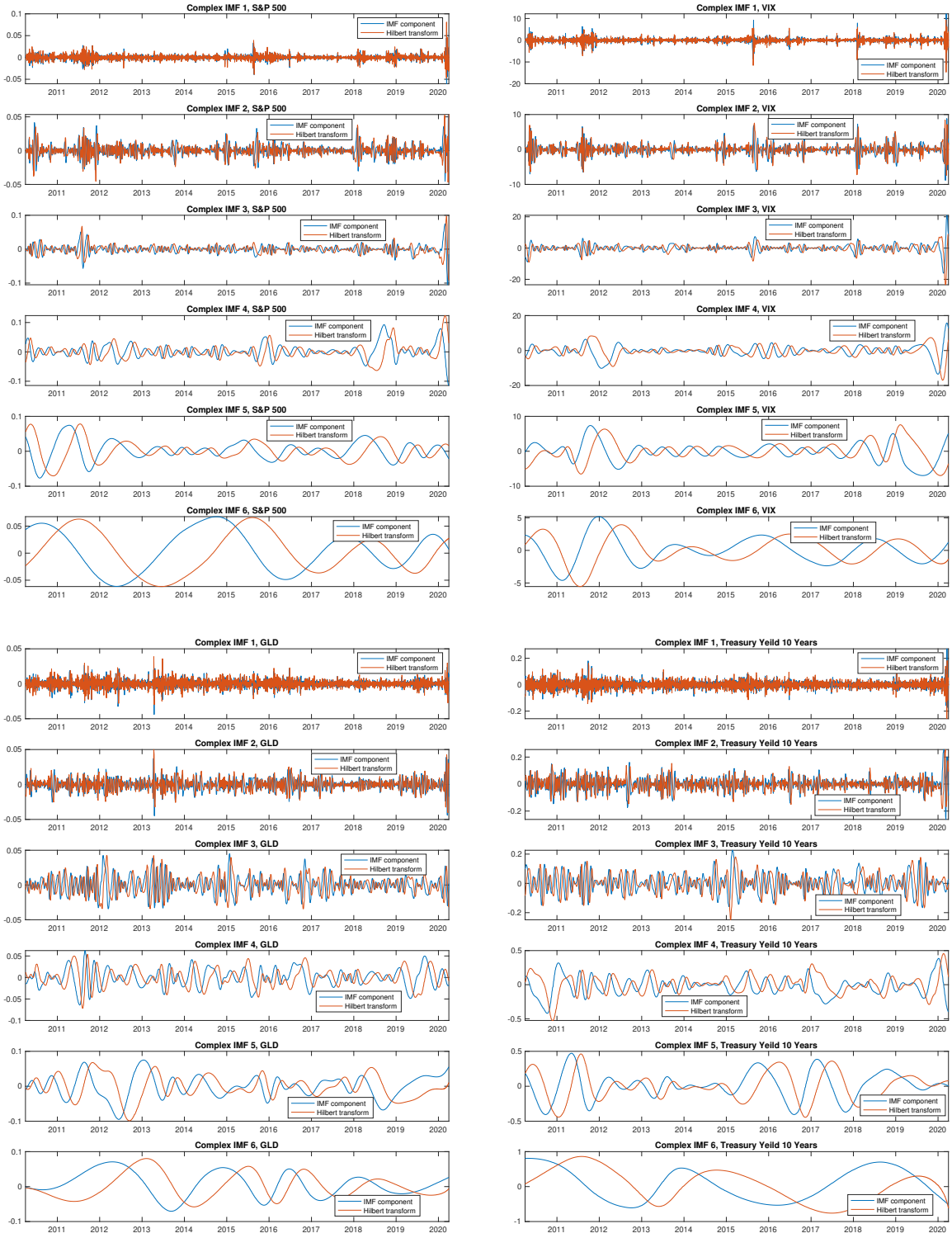

Figure 2.2: Complex IMFs derived from formula (2.6), S&P 500 (top left), VIX (top right), GLD (bottom left), and 10-year treasury yield (bottom right). The real parts (IMF components) are plotted together with their corresponding imaginary parts (Hilbert transforms of IMFs).

The instantaneous frequency $f_j(t)$ from Eq. (2.10) associated with each IMF component $c_j(t)$ of the four financial time series are presented in Fig. 2.3. From the 1st to 6th mode (top to bottom), the fluctuations of instantaneous frequency tend to decrease, and the overall levels of instantaneous frequency are reduced. Fig. 2.4 shows the instantaneous amplitude $a_j(t)$ from Eq. (2.8) associated with each IMF component $c_j(t)$ of the four financial time series. From the 1st to 6th mode (top to bottom), the fluctuations of instantaneous amplitude is reduced significantly. As such, the higher-order modes reflects a clearer long-term trend compared to lower-order modes. The fluctuations of the instantaneous amplitude and frequency are also less rapid than the IMF component itself, satisfying the slow modulation condition.

2.2.5 End Effect of EMD

EMD-based decomposition methods and HHT are adaptive for nonstationary systems. However, the algorithms are also known for the issue of end effect, whereby the decomposition and Hilbert transform have higher error when getting closer to the ends of the time series. This is because the sifting algorithm in Section 2.2.1 uses interpolation in each iteration, requiring an anchor point beyond the time span. The end effect can be relieved when the time series has enough length, and we are only interested in the properties in the interior. However, the error at the end can be a potential problem and may an additional challenge for forecasting.

To characterize the end effect, we need to consider the decomposition with the position in the time frame. Let's define the end effect factor as the position of point t on the decomposition interval $[T_1, T_2]$:

$$\lambda(t) = \frac{2t - (T_1 + T_2)}{T_2 - T_1}. \tag{2.15}$$

Note that $\lambda = 0$ at the middle point of the time frame, and $\lambda = \pm 1$ at the ends. Suppose the time series has the true IMF mode decomposition $x(t) = \sum_{j=1}^{n} c_j^*(t) + r_j^*(t)$, then the decomposition found by the CEEMD will be:

$$(c_1(t), \cdots, c_n(t), r_n(t)) \sim \mathcal{D}(c_1^*(t), \cdots, c_n^*(t), r_n^*(t), \lambda(t)), \tag{2.16}$$

which is a draw from the distribution due to the random nature of the noise-assisted algorithm. Denote the shorthand notice as $\mathbf{c}(t) = (c_1(t), \cdots, c_n(t), r_n(t))$. The error vector of $\mathbf{c}(t)$ is:

$$\mathcal{E}(\mathbf{c}^*(t), \lambda(t)) = \mathcal{D}(\mathbf{c}^*(t), \lambda(t)) - \mathbf{c}^*(t). \tag{2.17}$$

Note that $\mathbf{1}^T \mathcal{E}(\mathbf{c}^*(t), \lambda(t)) = 0$ at any t, due to the exact decomposition in Eq. (2.5), and $Var(\mathcal{E}_j(\mathbf{c}^*, \lambda))$ increase with $|\lambda|$ due to the end effect.

2.2.6 Comparison to Other Methods

We end this section by highlighting the differences among the HHT-based method, Fourier transform, and wavelet transform. As summarized in Table 2.1, HHT allows for nonlinearity, nonstationarity. The HHT bases are determined in adaptive form with respect to the time series, rather than being prespecified as in the Fourier and wavelet transforms. In addition, HHT produces a sparse spectrum with finitely many of modes, which is a useful property for our study of financial time series. The adaptiveness and sparsity also allow for practical feature generation for time series forecasting, which is the main reason for our use of HHT in this study.

	Nonstationarity	Nonlinearity	Basis	Spectrum
Fourier	No	No	A priori	Global; Dense
Wavelet	Yes	No	A priori	Regional; Dense
HHT	Yes	Yes	Adaptive	Local; Sparse

Table 2.1: Comparison between different transforms for time series.

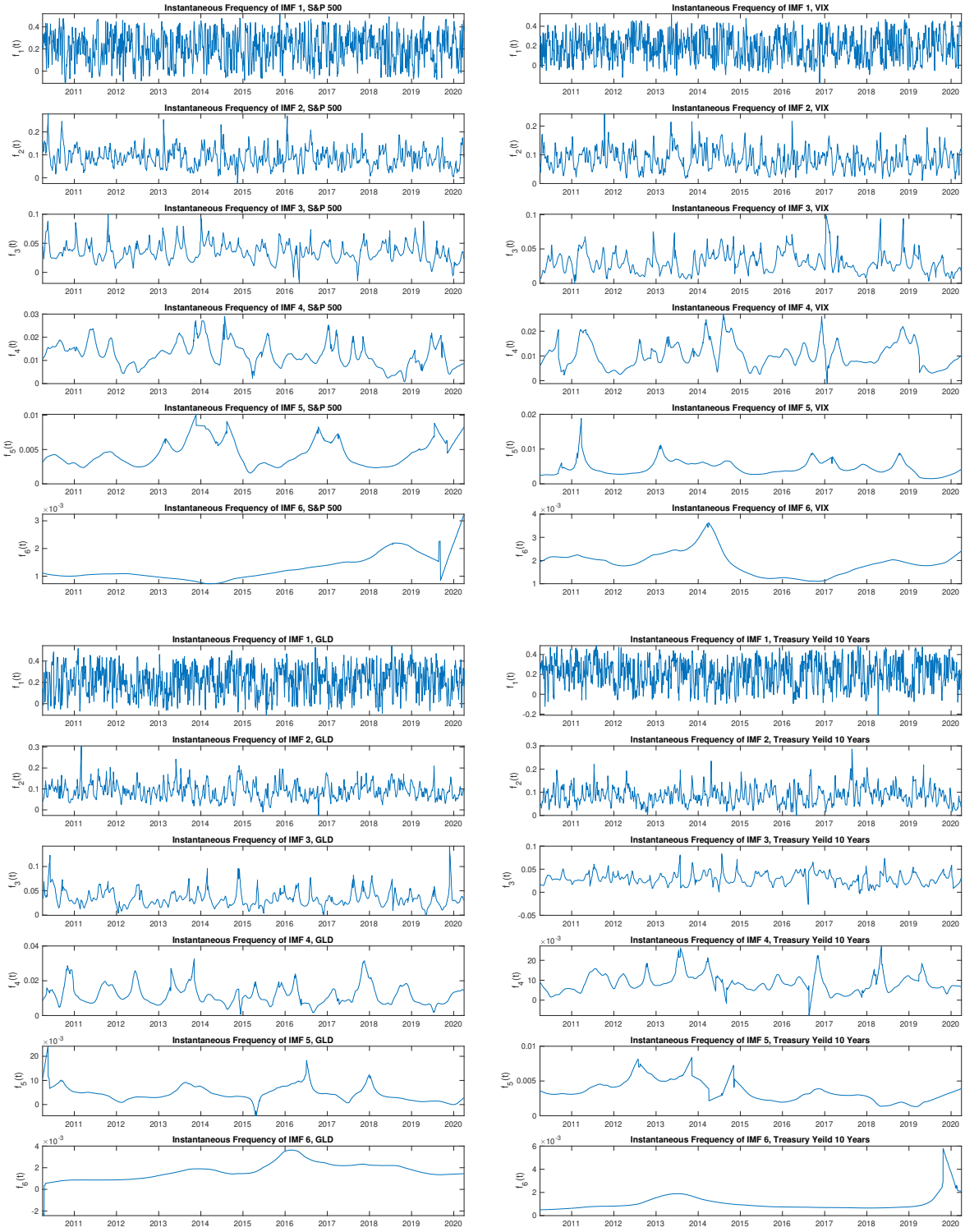

Figure 2.3: The instantaneous frequency associated with each IMF component derived from formula (2.10) of S&P 500, VIX, GLD, and TNX. From top to bottom, the instantaneous frequency correspond to modes with lower frequencies.

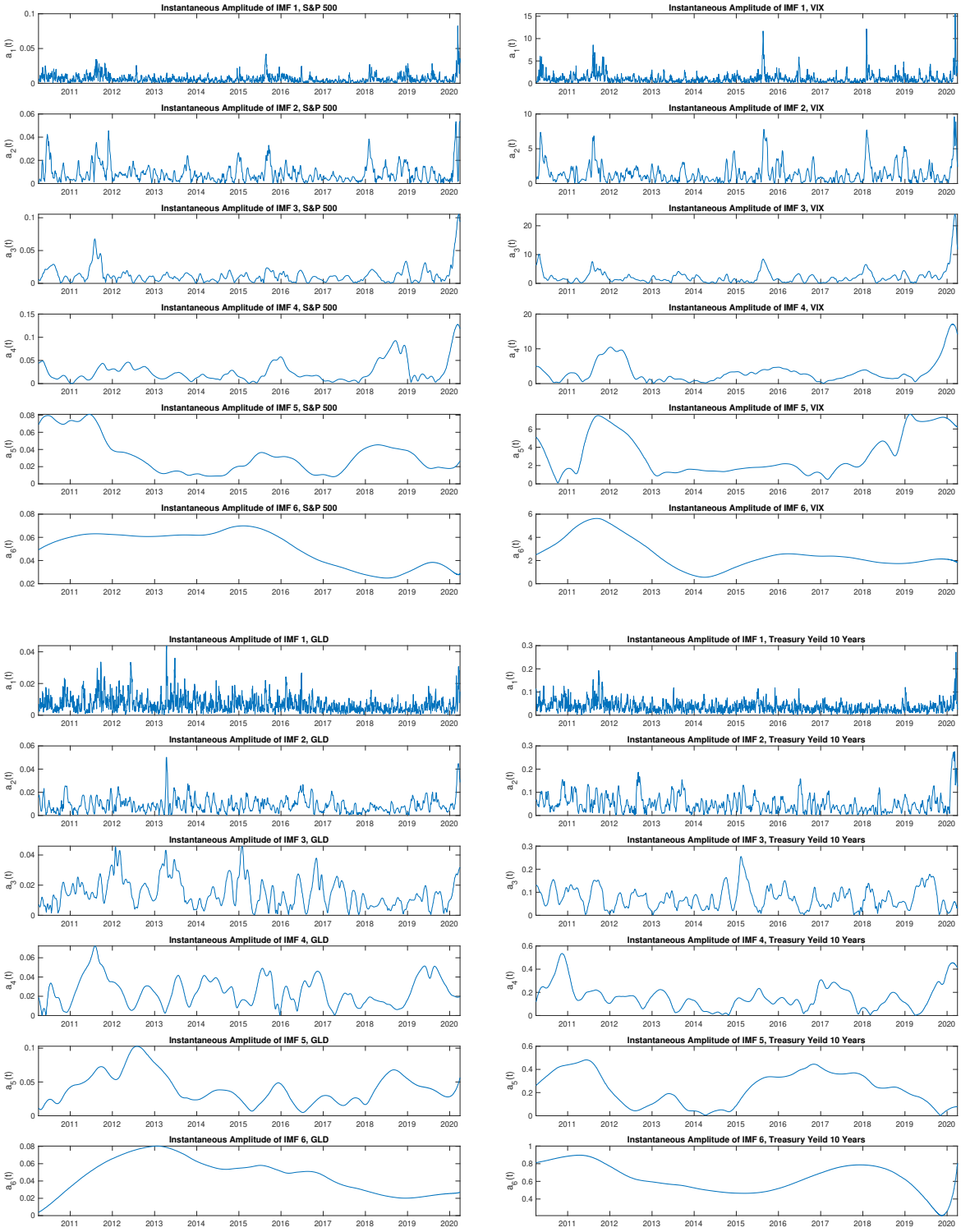

Figure 2.4: The instantaneous amplitude associated with each IMF component derived from formula (2.8) of S&P 500, VIX, GLD, and TNX. From top to bottom, the instantaneous amplitude correspond to modes with lower frequencies.

2.3 Machine Learning Using HHT Features

HHT-enhanced ML prediction involves two steps. In the first step, HHT is applied to the original time series $x(t)$. CEEMD decomposes $x(t)$ into n IMFs $c_1(t), \cdots, c_n(t)$ plus the residual term $r_n(t)$, and then Hilbert transform on the IMFs yields the corresponding imaginary counterparts $\hat{c}_1(t), \cdots, \hat{c}_n(t)$, along with the instantaneous amplitude $a_1(t), \cdots, a_n(t)$, and instantaneous frequency $f_1(t), \cdots, f_n(t)$. These outputs represent the *HHT feature set* derived from the time series $x(t)$ (see Fig. 2.5). The following sections discuss the second step of training and testing machine learning models on the HHT features.

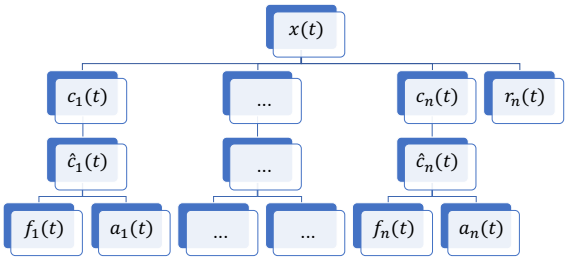

Figure 2.5: The collection of HHT features derived from the original time series.

2.3.1 Training and Testing

After extracting the HHT features, a machine learning model is chosen for time series prediction. Specifically, the goal is to predict the next period change $\Delta x(t+1) := x(t+1) - x(t)$. Traditional time series prediction takes the values of x in the past τ periods $x^{(t,\tau)} := (x(t-\tau+1), \cdots, x(t))$ as input. In contrast, our approach uses *only* the HHT features as predictors. The following shorthand notations are used for convenience:

$$c_j^{(t,\tau)} := (c_j(t-\tau+1), \cdots, c_j(t)), \tag{2.18}$$

$$c_{(j,l)}^{(t,\tau)} := (c_j^{(t,\tau)}, \ldots, c_k^{(t,\tau)}), \tag{2.19}$$

and likewise for other HHT features $\hat{c}, a$ and f.

The machine learning objective is to minimize the expected l_2-loss of $\Delta x(t+1)$ prediction, given information up to time t. This leads to the optimization problem:

$$\min_{g\in\mathcal{G}} \mathbb{E}[(\Delta x(t+1) - g(c_{(1,n)}^{(t,\tau)}, \hat{c}_{(1,n)}^{(t,\tau)}, a_{(1,n)}^{(t,\tau)}, f_{(1,n)}^{(t,\tau)}))^2 | \mathcal{F}_t],$$

where $\mathcal{G}$ is the functional class specified by the ML model so that each g is a prediction function determined within the training period. The filtration of information up to time t is denoted by $\mathcal{F}_t$. For implementation with real data, we do not know the distribution of input features and the target output, so the expectation is estimated by the sample average. Specifically, we split the time series into a training section from $t = 1$ to $t = T_1$, and a testing section after that from $t = T_1 + 1$ to $t = T_1 + T_2$. The ML models learn on the training set by solving

$$\min_{g\in\mathcal{G}} \frac{1}{T_1} \sum_{t=1}^{T_1} (\Delta x(t+1) - g(c_{(1,n)}^{(t,\tau)}, \hat{c}_{(1,n)}^{(t,\tau)}, a_{(1,n)}^{(t,\tau)}, f_{(1,n)}^{(t,\tau)}))^2.$$

The optimal predictive function found, g_*, is then tested on the unseen data. The performance is measured in terms of mean squared error (MSE):

$$\frac{1}{T_2} \sum_{t=T_1+1}^{T_1+T_2} (\Delta x(t+1) - g_*(c_{(1,n)}^{(t,\tau)}, \hat{c}_{(1,n)}^{(t,\tau)}, a_{(1,n)}^{(t,\tau)}, f_{(1,n)}^{(t,\tau)}))^2.$$

2.3.2 Extrapolating Prediction

The training and testing framework discussed above is useful in validating the predictive power of HHT and perform feature selection. However, in real-time forecasting, we are only able to do the decomposition up to the current time t, and the testing is the one-shot extrapolation into the future $t + 1$ where no decomposition is available. The above-mentioned procedure and most of the previous studies suffered from information leakage from the future that enhanced the prediction at the current time. However, when moving towards a truly extrapolating prediction, the end effect inevitably harms the accuracy due to the interpolation property described in Section 2.2.5.

As discussed in Section 2.2.5, the decomposition of $x(t)$ depends on the position of t in the time frame, characterized by the end effect factor $\lambda(t)$ define in Eq. (2.15). Since the distribution of the decomposed component is according to Eq. (2.16), depending on λ, the prediction model then becomes

$$x_{t+1} \sim g(c_{(1,n)}^{(t,\tau)}, \hat{c}_{(1,n)}^{(t,\tau)}, a_{(1,n)}^{(t,\tau)}, f_{(1,n)}^{(t,\tau)}, \lambda(t)), \tag{2.20}$$

which means we need to incorporate λ into the predictors.

Now we are ready to establish a new machine learning framework for extrapolating prediction. Since the traditional train/test split is no longer valid for extrapolating prediction using the HHT features, we propose the following procedure of evaluation:

- For $t = T_1 + 1, \cdots, T_1 + T_2$:
  - Implement CEEMD and HHT on $x(s)$ for $s = t - T, \cdots, t - 1$.
  - Train machine learning model on $t - T, \cdots, t - 1$:

$$\min_{g \in \mathcal{G}} \frac{1}{T} \sum_{s=t-T}^{t-1} (\Delta x(s) - g(c_{(1,n)}^{(s-1,\tau)}, \hat{c}_{(1,n)}^{(s-1,\tau)}, a_{(1,n)}^{(s-1,\tau)}, f_{(1,n)}^{(s-1,\tau)}))^2.$$

  - Test one-shot prediction of $\hat{x}(t) = g_*(c_{(1,n)}^{(t-1,\tau)}, \hat{c}_{(1,n)}^{(t-1,\tau)}, a_{(1,n)}^{(t-1,\tau)}, f_{(1,n)}^{(t-1,\tau)})$.
- Evaluate testing mean square error $\frac{1}{T_2} \sum_{t=T_1+1}^{T_1+T_2} (x(t) - \hat{x}(t))^2$.

Notice that the distribution of the input and output in the time series forecasting task is not homogeneous, which requires an evolving prediction model such the long short-term memory (LSTM) neural network. The generic machine learning models will have poor performance on the testing error as the distribution is different from the training data. Implementation details will be discussed in the following section.

2.3.3 Machine Learning Models

Our approach is generally applicable to all ML models that take multidimensional features as inputs. To illustrate our methodology, we implement several ML models for time series predictions. Specifically, we tested three different types of nonlinear regression models, namely the regression tree ensemble (RTE), the support vector machine (SVM) regression, and the long short-term memory (LSTM) neural network.

The algorithm framework for RTE and SVM are generically applied to classification or regression tasks. Details of the models can be found in [22, 45] for RTE with least square boosting and [36, 117] for SVM for regression. The LSTM model, on the other hand, requires design of the network structure. In this section, we describe the network structure we used for this study, which is visualized in Fig. 2.6

As a type of recurrent neural network (RNN), LSTM learns the long-term dependency in sequenced data [46]. At each time step t, the LSTM block takes in new observations $c_j(t), \hat{c}_j(t), \omega_j(t), a_j(t),\ j = 1, \cdots n$. The new inputs together with the network state information from the last step generate the hidden units $h_1(t), \cdots, h_M(t)$. The LSTM block and the hidden units then propagate to the next LSTM block. For deeper networks, we apply a drop-out layer to the hidden layers and then feed into another layer of LSTM.

Our neural network structure consists of two layers of LSTM each with a drop-out layer. The network then links to two fully-connected layers, each with a ReLU layer for activation. The number of hidden units in the layers are as follows: 100 - 200 for the first LSTM layer, 100 - 200 for the second LSTM layer, 50 - 100 for the first fully-connected layer, and 50 - 100 for the second fully-connected layer. For each data set, we tune the number of hidden units via cross-validation.

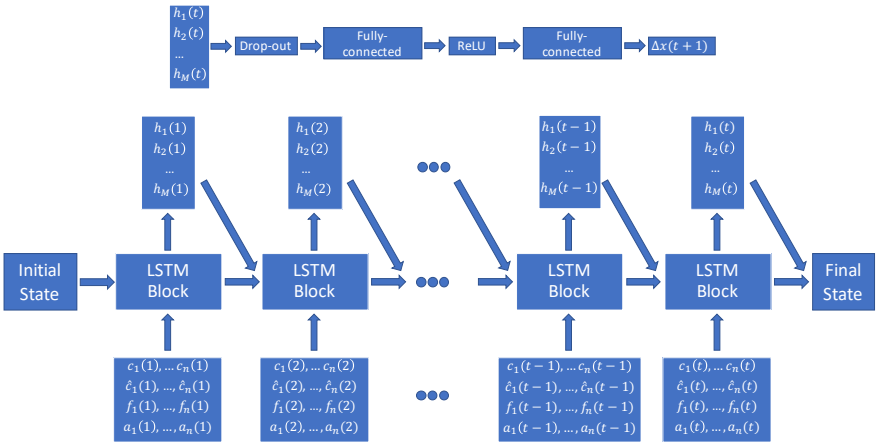

Figure 2.6: The structure of LSTM with HHT features.

For the regression task to predict $y(t) = \Delta x(t+1)$, the hidden units $h(t)$ are connected to the output $y(t)$ through a multi-layer neural network. In our study, we used multiple fully-connected layers with the activation function of rectified linear unit (ReLU). The reason to use ReLU is to create nonlinearity in the prediction and induce sparsity to choose a subset of useful signals. To train the neural network, we use the Adam solver [63] and train multiple times with different randomization. The model with the lowest root mean square error (RMSE) on the training set is used for testing.

2.4 Forecasting Experiments

In this section, we implement the machine learning forecasting experiments on the historical prices of S&P 500 index, CBOE volatility index (VIX), SPDR gold ETF (GLD), and the 10-year treasury yield (TNX). In the first part, we experiment feature selection for the HHT features and IMF components. The prediction follows the standard decomposition and train/test split framework. In Section 2.3.2, we discuss the extrapolation prediction and the associated end effect. The extrapolating prediction using LSTM is implemented.

2.4.1 Features and IMFs Selection

We implement the ML models using HHT features during the 10-year period 4/1/2010–3/31/2020. The data are split into a training set 4/1/2010–3/31/2018 and an unseen test set 4/1/2018–3/31/2020. The goal is short-time prediction for the next day change $\Delta x(t+1)$, based on the features up to time t. Comparison between using the original time series feature $x^{(t,\tau)}$ and with the HHT feature information $c_{(1,n)}^{(t,\tau)}, \hat{c}_{(1,n)}^{(t,\tau)}, a_{(1,n)}^{(t,\tau)}, f_{(1,n)}^{(t,\tau)}$ is the main performance indicator. We set the lookback window $\tau = 5$ for regression tree ensemble and SVM, while LSTM takes all the past features into the recurrent neural network, and each LSTM block controls the past information flow and memory level. Four separate sets of features are used:

1. original time series: $x(t)$,
2. IMF components: $c_1(t), \cdots, c_n(t)$,
3. complex IMF: $c_1(t), \hat{c}_1(t), \cdots, c_n(t), \hat{c}_n(t)$,
4. full HHT features: $c_1(t), \hat{c}_1(t), f_1(t), a_1(t), \cdots, c_n(t), \hat{c}_n(t), f_n(t), a_n(t)$,

For each time series, HHT produces a large set of interpretable features corresponding to different time scales. The feature size is increased by $4n$ times, where n is the number of IMF components. Through our experiments, we determine the most useful set of features for predictions using different ML methods, namely regression tree ensemble (RTE), support vector machine (SVM), and long short-term memory (LSTM) neural network. For performance comparison, we measure the mean squared errors (MSE) of the predictions over the whole horizon, based on the four financial time series S&P 500, VIX, GLD, and 10-year treasury rate.

Tables 2.2 – 2.5 summarize the experimental results from using RTE, SVM, and LSTM based on S&P 500 (log price), VIX index, GLD (log price) and 10-year treasury rate respectively. As benchmarks, we show the MSEs from predictions with the original time series x

as input feature, using the corresponding machine learning model. In all the rest parts of tables, the predictions are made using only HHT features and do not use the original time series x directly. Each row corresponds to using only a subset of HHT features, and each column corresponds to the set of IMFs included. The first and last few components are also referred to as the high-pass (high-frequency) and low-pass (low-frequency) filters respectively.

Tables 2.2–2.5 show that generally the largest feature set (bottom row) lead to much reduced errors. This suggests that HHT is capable of identifying the most important features from a time series. Surprisingly, the feature pair $(c, \hat{c})$, which consists of only the IMF components and the complementary imaginary parts derived from Hilbert transform in Eq. (2.6), yields similar or better performance than a much larger feature sets, e.g. $(c, \hat{c}, f, a)$, which indicates that the most useful information in HHT is included in the complex IMFs. Considering the much smaller feature set, the complex IMFs are more preferable in terms of information criterion. Both feature sets outperform the feature set with only real IMF components, which has been explored in many previous studies of machine learning prediction with EMD.

In terms of IMF selection, all three ML models perform better with all or the first few (high-frequency) IMFs than the last (low-frequency) IMFs. In many cases, the machine learning model performs best when the last IMFs are excluded (first 5). This is intuitive since low-frequency components may not be as useful as high-frequency components for predicting short-term movements, and a machine learning model with high complexity can potentially overfit on the training set with redundant information.

Comparing different machine learning models, we see that LSTM does not perform as good as the other more generic models in some scenarios. This is due to the fact that the prediction framework is under the train/test split within the same decomposition. The generic models lose predictive power compared to LSTM, as we get into extrapolating prediction in the next section.

Model: RTE					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			2.3036		
c	1.2614	1.2185	1.3640	2.1418	2.2442
$c, \hat{c}$	**0.8203**	1.0508	0.9540	2.0625	2.2048
c, f	1.2845	1.1128	1.0747	2.0878	2.2539
c, a	1.2990	1.1378	1.0581	2.1720	2.2481
c, f, a	1.0137	1.3420	1.2366	2.0850	2.2892
$c, \hat{c}, f, a$	0.8331	0.8713	1.0297	2.0815	2.2242
Model: SVM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			2.3313		
c	2.0456	2.0545	2.0646	2.2685	2.3087
$c, \hat{c}$	1.9943	2.0214	1.9987	2.2421	2.2826
c, f	2.1161	**1.9825**	2.0030	2.2913	2.3100
c, a	2.1867	2.1490	2.1105	2.2889	2.3006
c, f, a	2.1842	2.0817	2.1089	2.2953	2.3064
$c, \hat{c}, f, a$	2.1446	2.0589	1.9897	2.2769	2.3023
Model: LSTM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			2.3285		
c	1.1400	1.2738	1.1537	2.0841	2.1510
$c, \hat{c}$	0.6348	0.5938	**0.4876**	1.9315	2.1239
c, f	1.2031	1.4041	1.0702	2.0743	2.1610
c, a	1.4561	1.1516	1.3415	2.0796	2.2130
c, f, a	1.2626	1.4305	1.1992	2.0707	2.1788
$c, \hat{c}, f, a$	0.6834	0.9725	1.0058	1.9217	2.1233

Table 2.2: S&P 500 prediction error (MSE, $\times 10^{-4}$) with different feature sets and components for three machine learning models (RTE (top), SVM (middle), LSTM (bottom)). The benchmark (top row for each model) is the prediction error from using the plain time series x only.

Model: RTE					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			6.2796		
c	2.8104	3.0372	3.9181	5.7601	6.2365
$c, \hat{c}$	1.9488	1.9982	1.8169	5.6598	6.1334
c, f	2.4096	2.4606	2.4603	5.7786	6.2631
c, a	3.7021	3.0241	3.9229	5.8065	6.2387
c, f, a	2.4866	2.7127	2.1297	5.7865	6.2145
$c, \hat{c}, f, a$	2.1461	2.4642	**1.5964**	5.5183	6.1631
Model: SVM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			6.4127		
c	5.9377	6.0057	5.6676	6.2427	6.2916
$c, \hat{c}$	5.9248	5.8020	5.6557	6.1872	6.3019
c, f	5.8231	5.7919	5.5276	6.2204	6.2988
c, a	6.2414	6.0873	5.9045	6.2802	6.2950
c, f, a	5.9618	6.0003	5.6598	6.2615	6.2985
$c, \hat{c}, f, a$	5.8611	5.8534	**5.3969**	6.1940	6.3017
Model: LSTM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			6.3131		
c	3.7682	3.9556	3.4594	5.5977	6.0955
$c, \hat{c}$	2.7004	2.5687	**2.5492**	5.3324	6.1218
c, f	5.4949	5.2656	5.2457	6.4108	6.2548
c, a	4.3976	4.3648	4.4501	6.2045	6.2070
c, f, a	7.0323	5.7723	4.9104	6.2129	6.4870
$c, \hat{c}, f, a$	4.4516	4.6962	3.9247	5.8894	6.2638

Table 2.3: VIX index prediction error (MSE) with different feature sets and components for three machine learning models (RTE (top), SVM (middle), LSTM (bottom)). The benchmark (top row for each model) is the prediction error from using the plain time series x only.

Model: RTE					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			6.9816		
c	2.7139	2.7363	2.5476	5.5666	6.7445
$c, \hat{c}$	1.0644	1.1821	1.0336	5.2963	6.4202
c, f	1.7681	1.9592	1.9780	6.0007	6.8471
c, a	3.1483	2.7417	2.5566	5.8702	6.7719
c, f, a	2.1982	2.0045	2.2453	5.7184	6.6546
$c, \hat{c}, f, a$	0.9626	0.9145	**0.8760**	5.5237	6.5540
Model: SVM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			7.2381		
c	3.8314	3.8783	3.6813	6.2574	7.3537
$c, \hat{c}$	2.6886	2.6979	2.7458	5.8944	6.8402
c, f	3.4269	3.4790	3.3913	6.0189	6.8042
c, a	4.5545	4.4103	4.3258	6.4397	7.0054
c, f, a	4.3807	4.1316	3.8366	6.5134	6.8918
$c, \hat{c}, f, a$	3.0045	2.6902	**2.6795**	6.0785	6.8850
Model: LSTM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)			6.9792		
c	4.8152	3.4164	3.1468	5.7257	6.8414
$c, \hat{c}$	1.4325	1.3451	1.4513	5.1071	6.5997
c, f	2.8661	3.2673	2.8033	6.2854	6.8597
c, a	3.2262	3.1938	3.5799	5.7604	6.8049
c, f, a	3.2504	3.4774	3.6283	5.7533	6.7521
$c, \hat{c}, f, a$	1.3953	1.5691	**1.3248**	5.1094	6.6098

Table 2.4: GLD prediction error (MSE, $\times 10^{-5}$) with different feature sets and components for three machine learning models (RTE (top), SVM (middle), LSTM (bottom)). The benchmark (top row for each model) is the prediction error from using the plain time series x only.

Model: RTE					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)	2.6011				
c	1.4615	1.2206	1.1800	2.2609	2.5347
$c, \hat{c}$	**0.6244**	0.6473	0.6545	2.1302	2.4472
c, f	1.2036	1.1138	1.0990	2.2331	2.5236
c, a	1.3131	1.3719	1.3811	2.2872	2.5832
c, f, a	1.0673	1.0318	1.0657	2.2663	2.5696
$c, \hat{c}, f, a$	0.6299	0.6798	0.8444	2.1907	2.4539
Model: SVM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)	2.7631				
c	1.6064	1.5610	1.5968	2.4099	2.6059
$c, \hat{c}$	1.3893	**1.3023**	1.3737	2.2918	2.5383
c, f	1.9820	1.5531	1.4357	2.4251	2.5881
c, a	1.9156	1.7213	1.6954	2.4124	2.5908
c, f, a	2.0474	1.8175	1.6666	2.4550	2.5877
$c, \hat{c}, f, a$	1.7609	1.4761	1.3948	2.3599	2.5489
Model: LSTM					
Features	All IMFs	First 5	First 4	Last 5	Last 4
x (original time series)	2.6071				
c	1.1932	1.1545	1.1304	2.2402	2.4243
$c, \hat{c}$	0.4184	0.4348	0.5390	1.9995	2.4374
c, f	1.2942	1.1846	1.1972	2.2446	2.4780
c, a	1.1745	1.1426	1.0941	2.2503	2.4284
c, f, a	1.2464	1.1732	1.1668	2.2991	2.5692
$c, \hat{c}, f, a$	0.4633	**0.3816**	0.4190	2.0137	2.4758

Table 2.5: 10-year treasury yield prediction error (MSE, $\times 10^{-3}$) with different feature sets and components for three machine learning models (RTE (top), SVM (middle), LSTM (bottom)). The benchmark (top row for each model) is the prediction error from using the plain time series x only.

2.4.2 Extrapolation and End Effect

Table 2.6 shows the extrapolating prediction results with the end effect correction described in 2.3.2. We see that the different HHT feature sets all improved the prediction accuracy from using the plain time series and from the benchmark naive prediction $x_{T+1} = x(T)$. Furthermore, the complex IMF components and the full HHT feature set not only achieved lower mean square error, but also required less hidden units in the network and less training time. This shows the HHT features are better at explaining the dynamics and presenting the data.

Fig. 2.7 shows the 1-month rolling MSE of LSTM prediction using different feature sets. The experiment is implemented on S&P 500 GLD, VIX and TNX from 1/1/2020 - 10/29/2020. The HHT feature sets consistently outperform the benchmark prediction with plain time series as feature (dashed lines).

2.5 Concluding Remark

In this chapter we presented a fully data-driven approach for multiscale analysis of non-stationary financial time series. The key outputs of this method are the series of intrinsic mode functions, along with the time-varying instantaneous amplitudes and instantaneous frequencies. The method is then used to generate a collection of unique features that can be integrated into machine learning models, such as regression tree ensemble, support vector machine (SVM), and long short-term memory (LSTM) neural network. A novel technique is proposed to attend the end-effect in training and evaluation. Through a series of examples with empirical financial data, we showed that multiscale features can enhance the machine learning models in terms of time series forecasting performance.

S&P 500	**Benchmark**: 5.5770		
Features	MSEs	Avg. # hidden units	Avg. train time (s)
x (original time series)	5.5794	500.0	6.6166
c	5.0029	553.3	7.9682
$c, \hat{c}$	5.1114	373.1	5.3871
$c, \hat{c}, f, a$	5.1444	322.6	4.1803
GLD	**Benchmark**: 1.5698		
Features	MSEs	Avg. # hidden units	Avg. train time (s)
x (original time series)	1.5781	550.0	7.7274
c	1.3497	552.0	7.9372
$c, \hat{c}$	1.3304	348.1	5.4421
$c, \hat{c}, f, a$	1.3872	327.4	4.1814
VIX	**Benchmark**: 87.141		
Features	MSEs	Avg. # hidden units	Avg. train time (s)
x (original time series)	86.828	400.0	6.8851
c	79.581	401.0	5.6356
$c, \hat{c}$	79.265	399.5	5.6866
$c, \hat{c}, f, a$	75.450	515.7	8.3581
TNX	**Benchmark**: 37.299		
Features	MSEs	Avg. # hidden units	Avg. train time (s)
x (original time series)	37.691	400.0	7.3313
c	35.749	504.8	8.8135
$c, \hat{c}$	34.044	380.7	6.2190
$c, \hat{c}, f, a$	31.954	431.4	7.1112

Table 2.6: Out-of-sample prediction error (MSE, $\times 10^{-4}$) and computational cost of LSTM with different features. The benchmark is the error of naive prediction $x_{T+1} = x_T$. The first rows are the prediction with the plain time series.

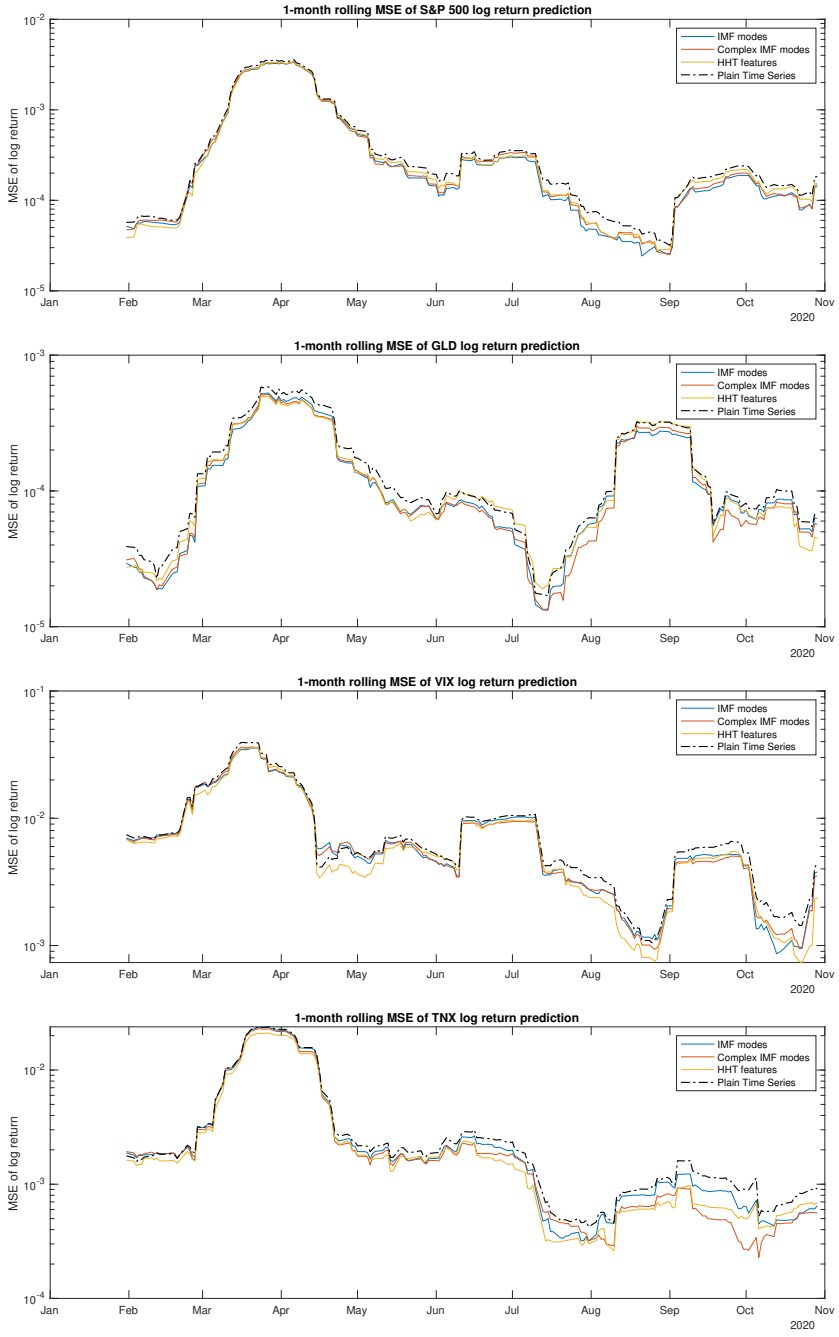

Figure 2.7: 1-month rolling MSE of LSTM extrapolating prediction using HHT features. The benchmark (dashed line) is the prediction with plain time series as the feature.

Chapter 3

ADAPTIVE MULTISCALE ANALYSIS FOR NONSTATIONARY TIME SERIES

3.1 Overview

In this chapter we delve deeper into the multiscale behaviors of nonstationary time series, taking cryptocurrency price as our study objective. We propose the adaptive complementary ensemble empirical mode decomposition (ACE-EMD) to deal with the highly nonstationary property of cryptocurrency prices. We then proceed to study the cryptocurrency price dynamics at long-term and short-term using ACE-EMD as an adaptive scale separation filter. Lastly the energy-frequency spectrum is used to show how different timescales are glued together and evolve over time.

3.2 Adaptive Multiscale Decomposition

In this section, we propose the adaptive complementary ensemble empirical mode decomposition (ACE-EMD) for multiscale decomposition of highly nonstationary time series. Implementation is conducted using Bitcoin and Etherium historical prices as candidates for cryptocurrency price dynamics.

3.2.1 Adaptive Complementary Ensemble EMD

We have seen in the previous chapter how the empirical mode decomposition (EMD) decomposes any time series into a sequence of intrinsic mode functions (IMFs). Implementing on nonstationary and nosy financial times series often leads to the undesirable mode mixing phenomenon, and the noise-assisted ensemble methods (EEMD and CEEMD) introduced in Chapter 2.2.2 were developed to handle the problem.

However, new problem comes when the time series is of high degree of nonstaionarity and noise, such as crypto currency prices. Both EEMD and CEEMD apply white noise with a *constant variance* over time. Most financial time series, however, have nonstationary or time-varying volatility (see Fig. 3.3 for example). This phenomenon is commonly observed among cryptocurrencies. Many cryptocurrencies tend to have a very low volume after inception or initial coin offering (ICO), and some may experience huge rise in trading volume and volatility at some point. A constant noise level will decrease the signal to noise ratio when the time series is of low volatility, even resulting in a meaningless signal there.

In order to capture the time-varying volatility of cryptocurrencies, we consider the *adaptive complementary ensemble empirical mode decomposition* (ACE-EMD), where the noise level of $w_i(t)$ is proportional to the intrinsic noise level. Inspired by [55], a pilot sifting process is first implemented on $x(t)$, taking the first IMF component $c_p(t)$ as an adaptive noise estimation. The full algorithm is summarized as follows:

- Implement a pilot sifting process on $x(t)$ and extract the first IMF component $c_p(t)$.
- Estimate the local maxima of $c_p(t)$ and interpolate it with cubic spline $a_p(t)$. According to the symmetry definition of IMF, the upper envelope $a_p(t)$ should equal to the amplitude of the pilot mode $c_p(t)$.
- For $i = 1, \cdots, N$:
  - Generate noise $w_i(t)$ from a distribution (e.g. Gaussian noise) such that

$$\mathbb{E}[w_i(t)] = 0,\ Var[w_i(t)] = \sigma^2 a_1^2(t),\ w_i(s) \perp\!\!\!\perp w_i(t) \text{ for } s \neq t. \tag{3.1}$$

  - Implement EMD on $x(t)+w_i(t)$ and decompose it into IMF components $c_{i1}^+(t), \cdots, c_{in}^+(t)$ plus a residual term $r_{in}^+(t)$.

  - Implement EMD on $x(t)-w_i(t)$ and decompose it into IMF components $c_{i1}^-(t), \cdots, c_{in}^-(t)$ plus a residual term $r_{in}^-(t)$.

- Compute the decomposition of $x(t)$ as the ensemble mean:

$$c_j(t) = \frac{1}{2N}\sum_{i=1}^{N}\left(c_{ij}^+(t) + c_{ij}^-(t)\right), \quad j = 1, \cdots, n,$$
$$r_n(t) = \frac{1}{2N}\sum_{i=1}^{N}\left(r_{in}^+(t) + r_{in}^-(t)\right).$$

Applying these to Eq. (2.5), we conclude that the components from the ensemble mean sum up to the original time series, so ACE-EMD ensures an exact decomposition.

It remains to determine the only parameter in the algorithm, the noise level σ. To that end we choose σ to optimize the orthogonality and separability of the decomposition (see e.g. [123]). Following [54], the *orthogonality index* of the decomposition is defined as

$$OI = \frac{1}{T}\int_0^T \sum_{j\neq k} \frac{c_j(t)c_k(t)}{x^2(t)}dt. \tag{3.2}$$

The summation is over all the mode pairs, including the residual term $r_n(t)$ as the $(n+1)$-th mode. Similarly, the separability is defined as the root-mean-square of the pairwise correlation between the modes (see [123]). The strategy is to choose σ minimizing the separability while keeping the absolute value of orthogonality index small (e.g. $|OI| < 0.05$).

3.2.2 Implementation on Cryptocurrencies

In this chapter we implement the algorithms on the daily price of a collection of cryptocurrencies from 1/4/2016 to 3/29/2021. Fig. 3.1 shows the decomposition from ACE-EMD for the log prices of BTC and ETH. The top row in each plot shows the original time series $x(t)$, followed by the IMF $c_j(t)$'s with decreasing frequencies, ended with the residual term. The number of IMF components $n = 5$, so along with the residual term, there are 6 components

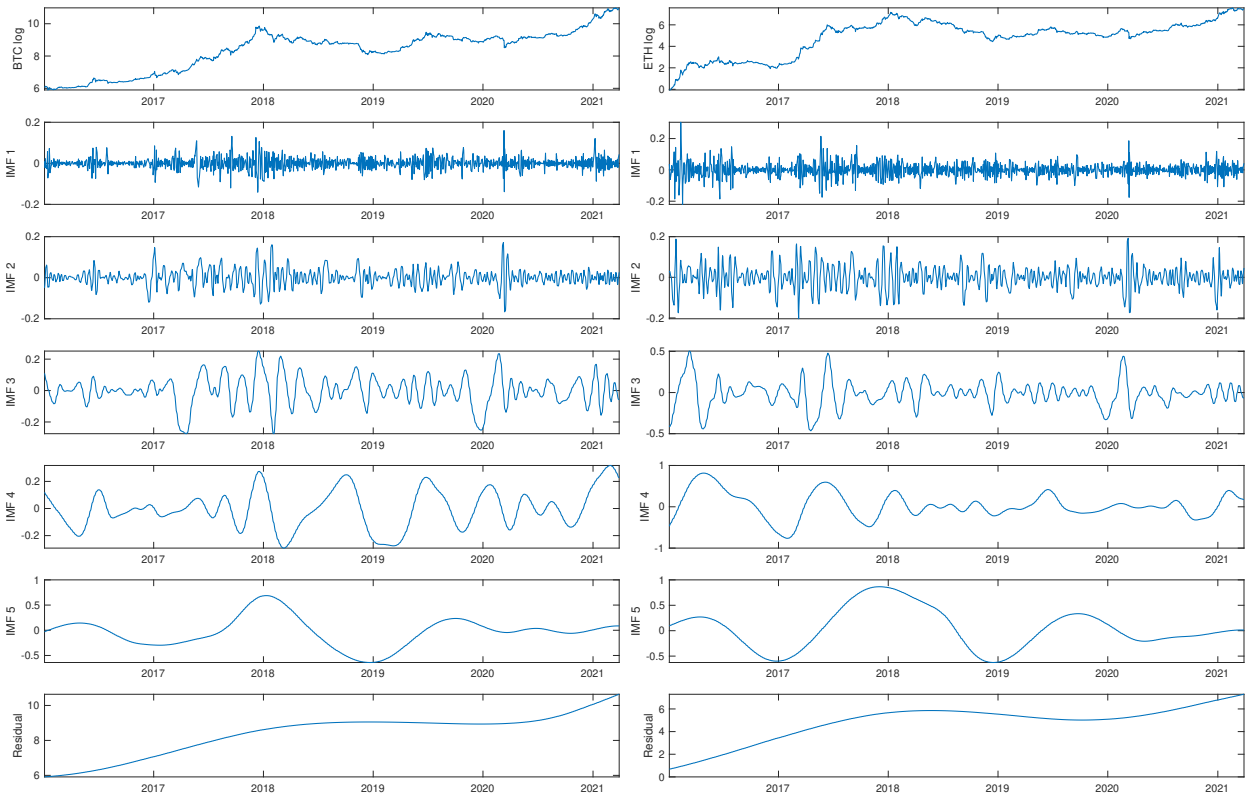

Figure 3.1: IMFs and residual terms extracted from ACE-EMD of BTC (left) and ETH (right), from 1/4/2016 to 3/29/2021. The top row in each plot shows the original time series (in log prices). The subsequent rows show the IMFs of the corresponding time series, except for the bottom row which shows the residuals.

in total. As we can see, the first IMF exhibits the highest frequency of fluctuation, whereas the smooth residual term reflects the overall trend.

3.3 Timescale Separation with ACE-EMD

Financial time series often exhibit characteristics on different timescales. The short-term scale explains more of the volatility while the long-term scale reflects the trend. Separating a time series onto different timescales provides extra insights of the dynamics. In this section we use ACE-EMD as an adaptive filter, showing low-frequency reconstruction and stationarity at high frequency. Interesting volatility behavior is observed under high-pass filtering.

3.3.1 ACE-EMD Filtering

By construction of the EMD algorithm, the iterative sifting process identifies the finest structures, and then extract longer and longer scales. As a result, the first few components have higher frequency which are more noisy, and the last few components have lower frequency representing long term structure. In fact, [55] showed that EMD and its variations can be used as a dyadic filter for time series. Here we use ACE-EMD to be the time series filter.

In the reconstruction of the original time series using the IMF components, we can choose a subset of modes as a filter for desired information. By removing the first few high frequency components, we create a low-pass filter; that is,

$$x_L^{(m_l)}(t) = x(t) - \sum_{j=1}^{n-m_l+1} c_j(t). \tag{3.3}$$

This reconstruction using only the last few components can serve as a smoothing of the time series. Similarly, we can also build a high-pass filter with

$$x_H^{(m_h)}(t) = \sum_{j=1}^{m_h} c_j(t), \tag{3.4}$$

which captures the high-frequency local behaviors corresponding to noise and volatility.

In each case, the number of components (including the residual term) equals to m_l and m_h respectively. We use $x_L^{(m_l)}(t)$ and $x_H^{(m_h)}(t)$ to denote the low-pass and high-pass filtered reconstruction of $x(t)$ with m_l and m_h components. Note that the low-pass filter and the high-pass filter are complementary to each other when $m_l + m_h = n + 1$, where $n + 1$ is the total number of components (including the residual term).

Fig. 3.2 illustrates the low-pass filtered reconstruction of BTC and ETH prices. This involves applying Eq. (3.3) using different collections of components. Specifically, we have used the last 4, 3, and 2 components including the residual term, i.e. $x_L^{(m_l)}(t)$ with $m_l = 4, 3, 2$. The

reconstructed log prices are taken exponential to approximate the original price data. We can see that, with more components included in the reconstruction, the resulting time series resembles the original time series on a finer timescale. Compared to some other time series smoothing techniques, such as moving average, the ACE-EMD low-pass filter achieves smoothing without lags.

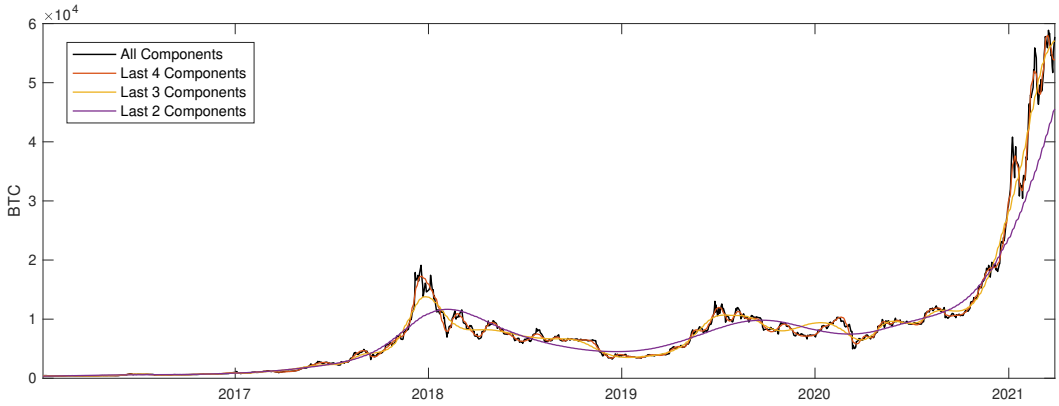

Figure 3.2: Reconstruction of BTC price (in $) time series from 1/4/2016 to 3/29/2021. The decomposition is implemented on the log price using the low-pass filter in Eq. (3.3), and then exponentiated to show the reconstruction in US dollar.

While the low-pass filter with ACE-EMD exhibits smoothed trends in the time series, the high-pass filter removes nonstationarity thus making the time series ready for statistical analysis. Table 3.1 shows the stationarity test results for BTC, ETH, XRP and LTC, before and after high/low-pass filtering. We experimented with three stationarity (unit-root) tests, namely the Augmented Dickey-Fuller (ADF) test [24], the Kwiatkowski, Phillips, Schmidt, and Shin (KPSS) test [70], and the Phillips-Perron (PP) test [103]. The results show that the highly nonstationary cryptocurrency prices become stationary after high-pass filtered with ACE-EMD. The three stationarity tests suggest consistently that a high-pass filter with one or two ACE-EMD components is favorable when doing timescale separation.

BTC	$x_H^{(1)}$	$x_H^{(2)}$	$x_H^{(3)}$	$x_H^{(4)}$	$x_H^{(5)}$	x	$x_L^{(5)}$	$x_L^{(4)}$	$x_L^{(3)}$	$x_L^{(2)}$	$x_L^{(1)}$
ADF test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0172**	0.9985	0.9990	0.9990	0.9990	0.9990	0.9990
KPSS test	**0.1000**	**0.1000**	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
PP test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0172**	0.9985	0.9990	0.9990	0.9990	0.9990	0.9990
ETH	$x_H^{(1)}$	$x_H^{(2)}$	$x_H^{(3)}$	$x_H^{(4)}$	$x_H^{(5)}$	x	$x_L^{(5)}$	$x_L^{(4)}$	$x_L^{(3)}$	$x_L^{(2)}$	$x_L^{(1)}$
ADF test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0168**	0.9840	0.9990	0.9990	0.9990	0.9990	0.9990
KPSS test	**0.1000**	**0.1000**	0.0104	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
PP test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0168**	0.9840	0.9990	0.9990	0.9990	0.9990	0.9990
XRP	$x_H^{(1)}$	$x_H^{(2)}$	$x_H^{(3)}$	$x_H^{(4)}$	$x_H^{(5)}$	x	$x_L^{(5)}$	$x_L^{(4)}$	$x_L^{(3)}$	$x_L^{(2)}$	$x_L^{(1)}$
ADF test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0131**	**0.0263**	**0.0012**	**0.0010**	**0.0010**	**0.0010**	**0.0010**
KPSS test	**0.1000**	**0.1000**	**0.1000**	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
PP test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0131**	**0.0263**	**0.0012**	**0.0010**	**0.0010**	**0.0010**	**0.0010**
LTC	$x_H^{(1)}$	$x_H^{(2)}$	$x_H^{(3)}$	$x_H^{(4)}$	$x_H^{(5)}$	x	$x_L^{(5)}$	$x_L^{(4)}$	$x_L^{(3)}$	$x_L^{(2)}$	$x_L^{(1)}$
ADF test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0235**	0.9325	0.9971	0.9990	0.9990	0.9990	0.9990
KPSS test	**0.1000**	**0.1000**	**0.1000**	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100	0.0100
PP test	**0.0010**	**0.0010**	**0.0010**	**0.0010**	**0.0235**	0.9325	0.9971	0.9990	0.9990	0.9990	0.9990

Table 3.1: The p-values from stationarity tests of the low/high-pass filtered time series of four major cryptocurrency prices. Results from the high-pass filtered time series are on the left with increasing number of components included; results from the original unfiltered time series x are placed in the middle column, and results from the low-pass filtered time series are on the right with decreasing number of components included. The null hypotheses of ADF test and PP test are nonstationary (unit-root), while the null hypothesis of KPSS test is stationary (no unit-root). The bold-highlighted values indicate stationarity under 0.05 significant level.

3.3.2 Volatility Dynamics at Different Scales

Applying Eq. (3.3) and (3.4) to filter the original time series into low-pass and high-pass components, we are able to analyze the long-term and short-term behaviors through filtered statistics such as mean and volatility. As the base setting, let $s(t)$ be the asset price, and $x(t) = \log s(t)$ be the log price. Implement ACE-EMD on $x(t)$ and apply Eq. (3.3) and (3.4) to get $x_L^{(m_l)}(t)$ and $x_H^{(m_h)}(t)$, with m_l and m_h being the number of components in the

low-pass and high-pass filtering respectively. Define the low-pass and high-pass log returns:

$$r_L^{(m_l)}(t) = x_L^{(m_l)}(t) - x_L^{(m_l)}(t-1), \tag{3.5}$$

$$r_H^{(m_h)}(t) = x_H^{(m_h)}(t) - x_H^{(m_h)}(t-1). \tag{3.6}$$

The low-pass and high-pass volatility are the standard deviation of the low-pass and high-pass returns, defined by the unbiased estimators

$$\sigma_L^{(m_l)} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(r_L^{(m_l)}(t) - \mu_L^{(m_l)}\right)^2}, \tag{3.7}$$

$$\sigma_H^{(m_h)} = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}\left(r_H^{(m_h)}(t) - \mu_H^{(m_h)}\right)^2}, \tag{3.8}$$

where

$$\mu_L^{(m_l)} = \frac{1}{T}\sum_{t=1}^{T} r_L^{(m_l)}(t), \qquad \mu_H^{(m_h)} = \frac{1}{T}\sum_{t=1}^{T} r_H^{(m_h)}(t), \tag{3.9}$$

are the mean values of low-pass and high-pass log returns.

To obtain annualized volatility, multiply $\sigma_L^{(m_l)}$ and $\sigma_H^{(m_h)}$ by $\sqrt{252}$, where 252 is the number of trading days in a year. The annualized volatility of filtered time series is estimated for BTC, ETH and S&P 500 from 4/1/2016 to 3/31/2020. Fig. 3.3 shows the 3-month rolling volatility computed using all components, high-pass filtered data, and low-pass filtered data. Recall from Fig. 3.1 that the time series are decomposed into 6 components including the residual terms. We use $m_l = 4$ for low-pass filtering and $m_h = 2$ for high-pass filtering. Notice that the high-pass filtered time series captures most of the volatility residing in the original times series. Therefore, we consider using the stationary high-pass filtered data in the following statistical analysis.

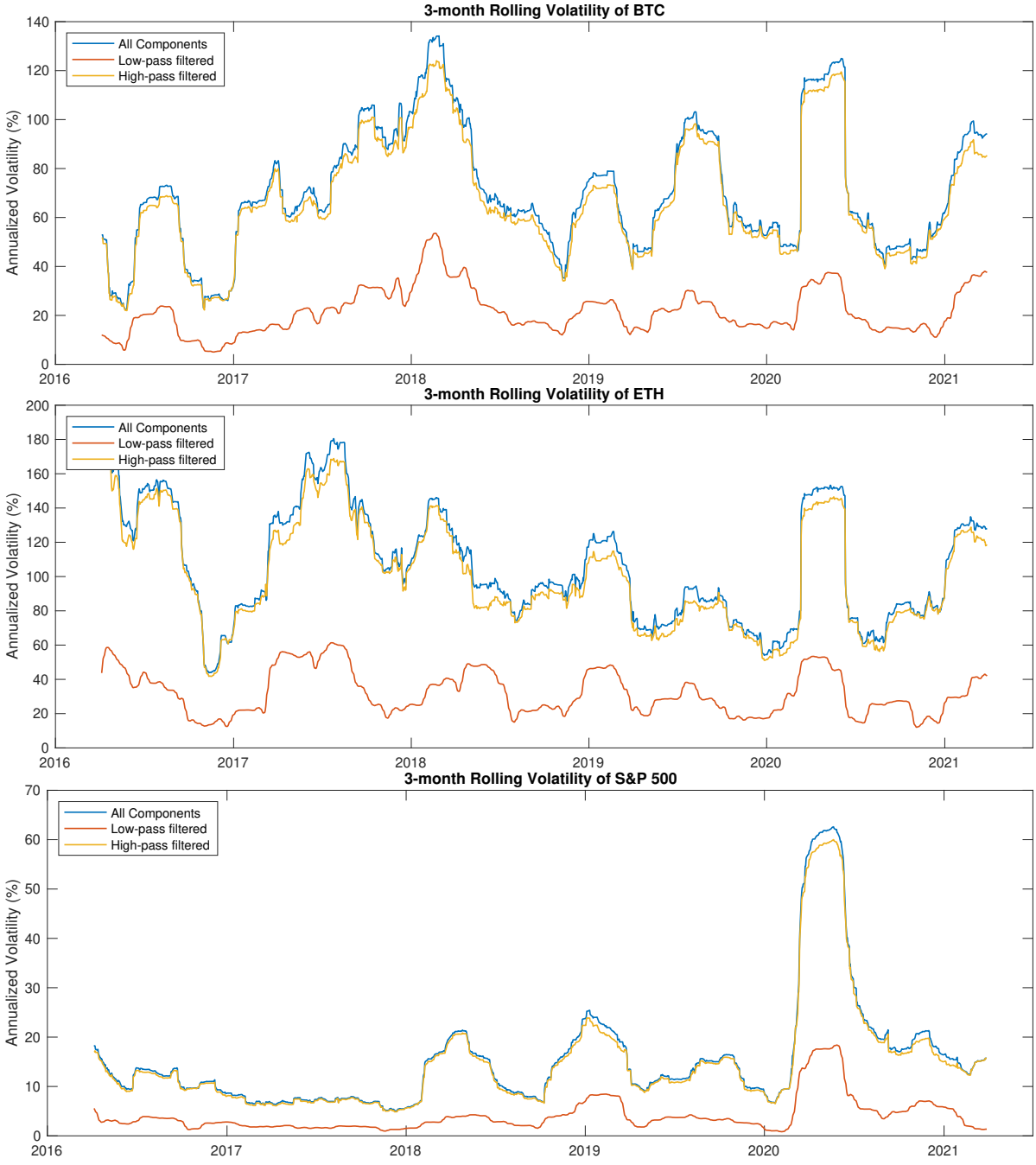

Figure 3.3: 3-month rolling volatility of BTC, ETH, and S&P 500 from 1/4/2016 to 3/29/2021. Values are computed using all components, high-pass ($m_h = 2$) filtered data, and low-pass ($m_l = 4$) filtered data.

3.3.3 Asymmetric Volatility

It has been observed in the financial market that the volatility of asset returns is usually asymmetric, i.e. the volatility is higher in a downside market than that in an upside market. Following [16], asymmetry can be captured by looking at the conditional volatility. Specifically, we examine if the asymmetry exists in the price dynamics. To that end, we define the conditional volatilities based on the ACE-EMD high-pass filter as follows:

$$\sigma_{+H}^{(m_h)} = \sqrt{Var(r_H^{(m_h)}(t)|r_H^{(m_h)}(t-1) > \mu_H^{(m_h)})}, \tag{3.10}$$

$$\sigma_{-H}^{(m_h)} = \sqrt{Var(r_H^{(m_h)}(t)|r_H^{(m_h)}(t-1) < \mu_H^{(m_h)})}. \tag{3.11}$$

In essence, $\sigma_{+H}^{(m_h)}$ and $\sigma_{-H}^{(m_h)}$ capture the high-pass volatilities conditioned on an upside movement and a downside movement respectively.

Fig. 3.4 shows the 3-month rolling estimation of the conditional volatility using high-pass filtered data of BTC, ETH, and S&P 500. In the high-pass components, we see that S&P 500 shows asymmetric volatility with a larger downside volatility most of the time. The cryptocurrencies, however, exhibit both directions of asymmetry, where upside and downside movements trigger high volatility alternately during different periods of time. This phenomenon suggests a distinct behavior of the cryptocurrencies comparing with the traditional equity indices.

Define the events of upside volatility asymmetry and downside volatility asymmetry as:

$$A^+ = (\sigma_{+H}^{(m_h)} - \sigma_{-H}^{(m_h)}) > \epsilon\sigma_H^{(m_h)} \tag{3.12}$$

$$A^- = (\sigma_{+H}^{(m_h)} - \sigma_{-H}^{(m_h)}) < -\epsilon\sigma_H^{(m_h)} \tag{3.13}$$

In Fig. 3.5, we plot the frequency of event A^+ as defined in Eq. (3.12), against the frequency of event A^- as defined in Eq. (3.13). The dashed line corresponds to $\mathbb{P}(A^+)+\mathbb{P}(A^-) = 1$, The closeness to the dashed line indicates the level of volatility asymmetry in general. Towards the upper left direction the time series are upside volatility asymmetry dominated, while towards

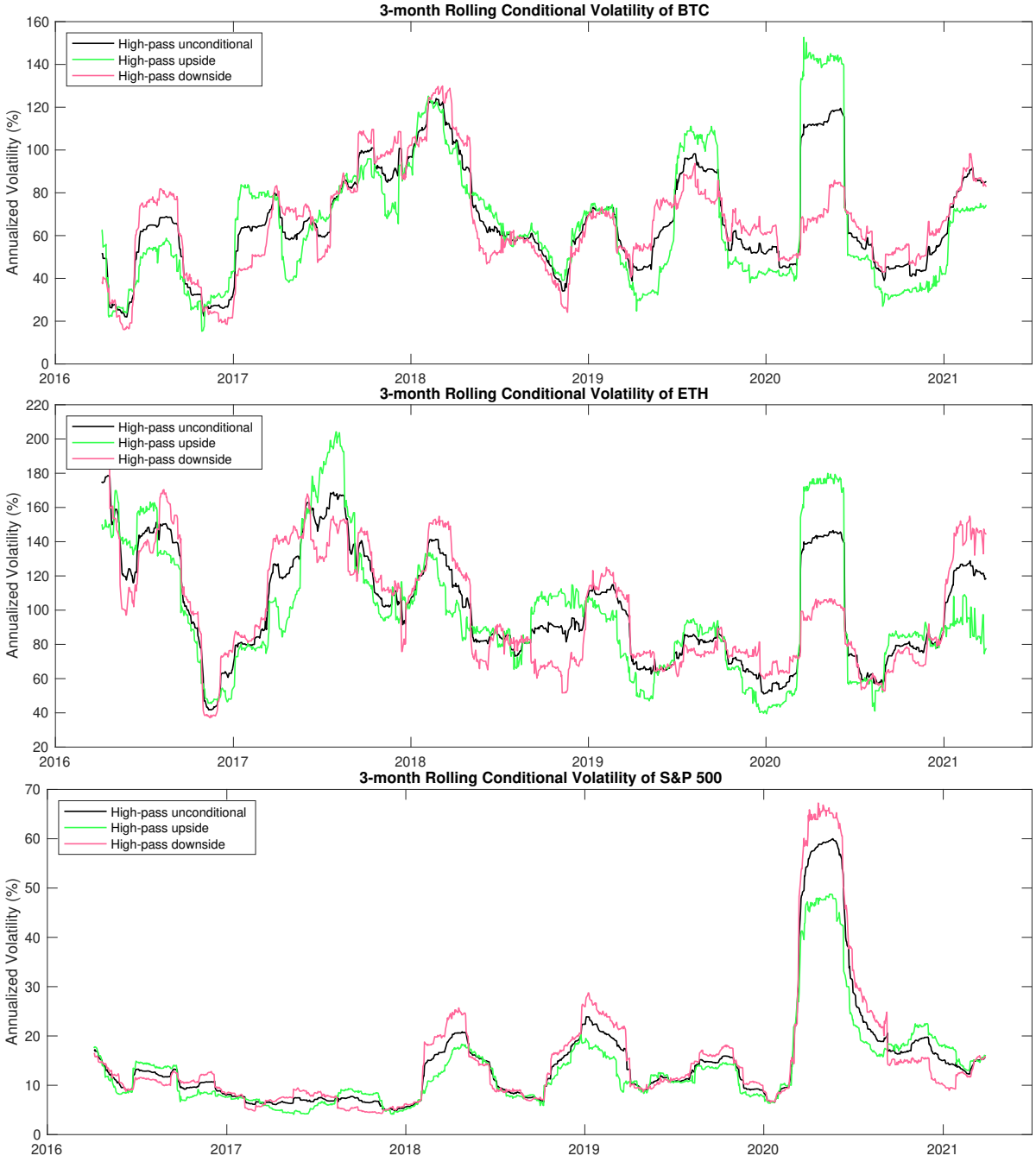

Figure 3.4: 3-month rolling estimation of the conditional volatility using low-pass and high-pass filtered data of BTC, ETH, and S&P 500 from 1/4/2016 to 3/29/2021. Values are computed using high-pass ($m_h = 2$) filtered data.

the lower right direction the time series are downside volatility asymmetry dominated. We can see there is a clear gap separating the assets into two clusters of upside and downside asymmetry. The three equity indices, S&P 500, DJI, and NASDAQ, are all on the downside volatility asymmetry end. The top cryptocurrencies like BTC and ETH are also in that region. In contrast, the gold ETF (GLD) is in the upside volatility asymmetry region. Several cryptocurrencies, such as XRP, TRX, XLM, and DOGE, are also found in the upside volatility asymmetry region.

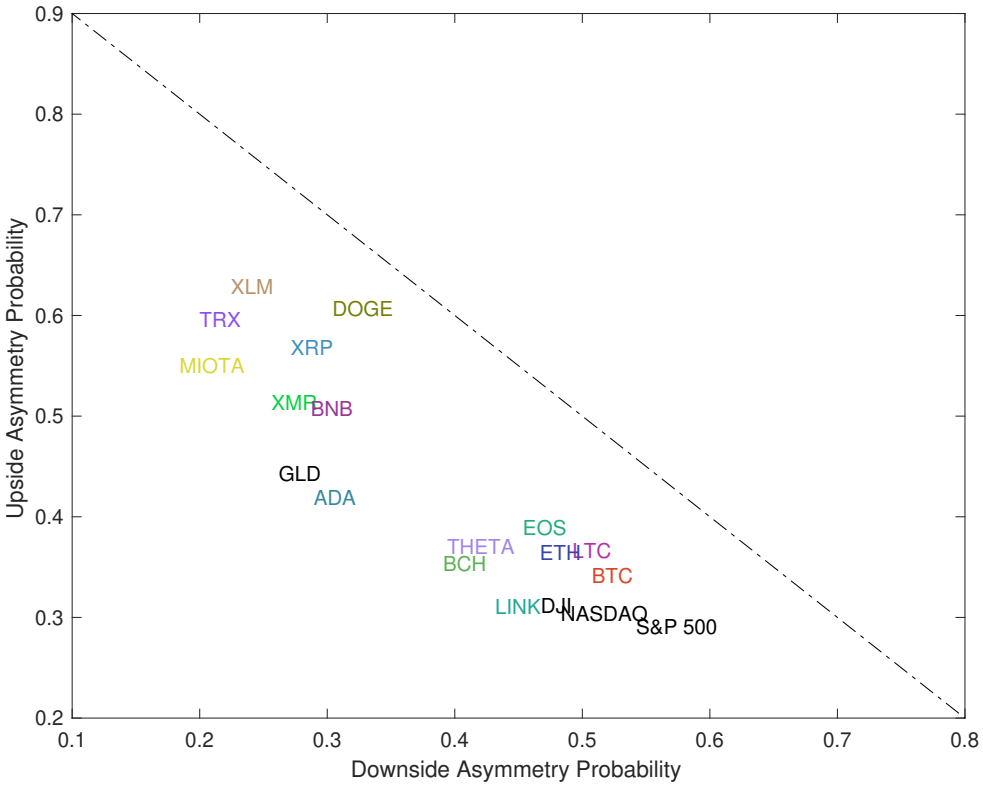

Figure 3.5: Asymmetric volatility effect for cryptocurrencies and traditional financial securities, estimated using prices from 1/4/2016 to 3/29/2021. Points closer to the lower right corner are more dominated by downside volatility, while the upper left region shows more upside volatility. Points closer to the dashed line show more volatility asymmetry in general.

3.4 Energy-Frequency Spectrum

In Chapter 2.2.3 we have shown how Hilbert spectral analysis recovers the instantaneous frequency and energy for each IMF. Instead of studying the evolution in time, spectral analysis looks at the energy distribution on different scales, providing an alternative aspect from the frequency domain [64]. In this section, we compute and analyze the energy-frequency spectra for a wide collection of cryptocurrencies.

3.4.1 Central Frequency and Power Spectrum

For each time series, we can obtain the instantaneous frequency $f_j(t)$ from Eq. (2.10) and instantaneous energy $E_j(t)$ from Eq. (2.14), corresponding to mode $j = 1, \cdots, n$. This allows us to derive the instantaneous energy-frequency spectrum as shown in Fig. 3.6 for BTC, ETH, LTC, XRP and two traditional financial market indices S&P 500 and GLD. Each point on the plots is a pair of $(f_j(t), E_j(t))$, for mode $j = 1, \cdots, n$, and time $t = 1, \cdots, T$. We see that for each mode j, the instantaneous energy-frequency pairs $(f_j(t), E_j(t)), t = 1, \cdots, T$ form a cluster of points. We define the *central frequency* and *central energy* of mode j during the time period $[0, T]$ as follows:

$$\bar{f_j} = \exp\left(\frac{1}{T}\int_0^T \log f_j(t)dt\right), \tag{3.14}$$

$$\bar{E_j} = \exp\left(\frac{1}{T}\int_0^T \log E_j(t)dt\right). \tag{3.15}$$

While the instantaneous frequency and energy are time-varying, they are typically fluctuating or orbiting around the central points. [35] assume a central frequency in each mode and used the concept to derive the variational mode decomposition (VMD). Intuitively, the central frequency and central energy capture the overall spectral properties of the time series [120].

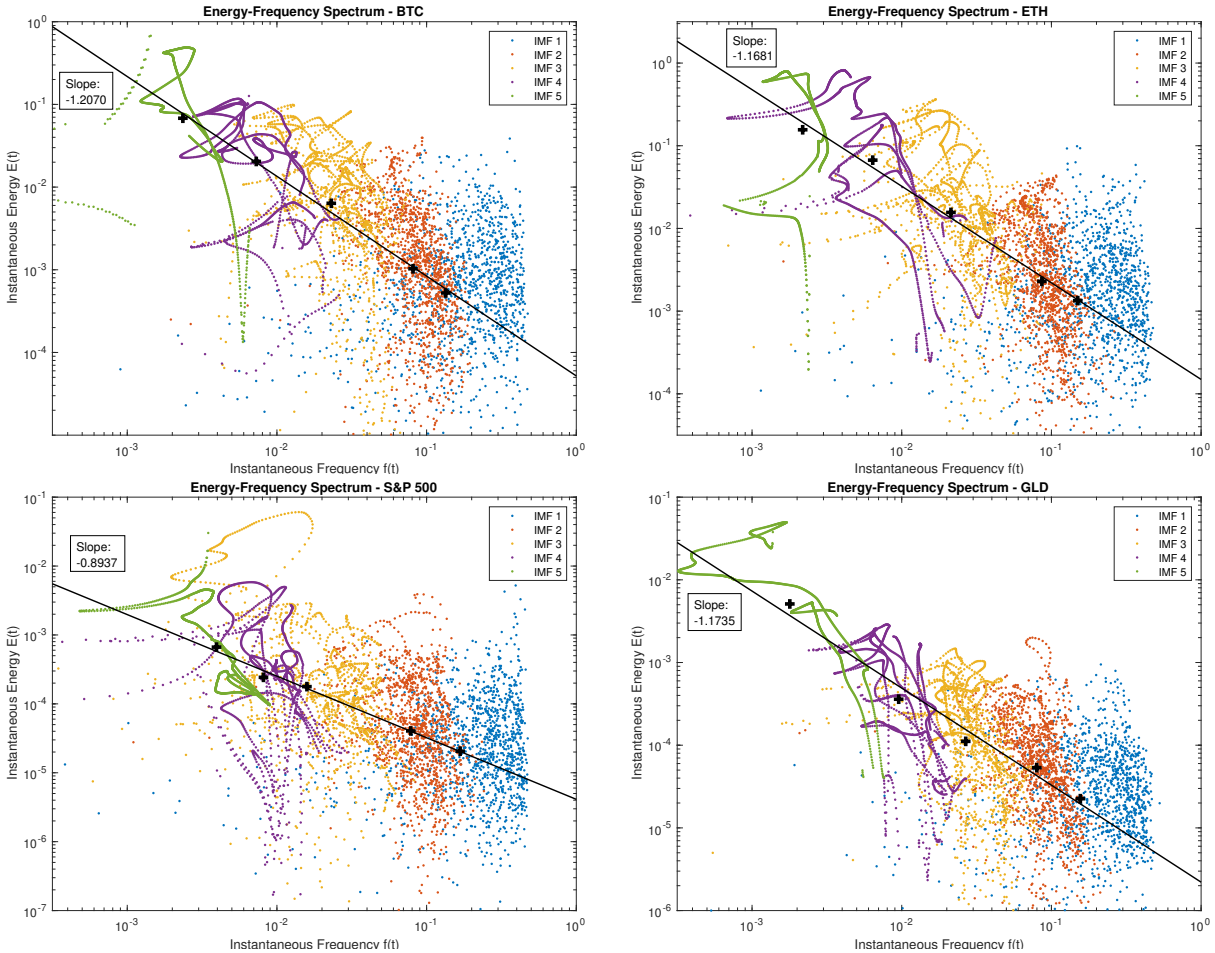

Figure 3.6: Instantaneous energy-frequency spectrum for two major cryptocurrencies BTC , ETH and two traditional financial market indices S&P 500 and GLD for comparison. The estimation are done on the time period from 1/4/2016 to 3/29/2021. The black cross in each plot marks the mean of the cluster of points in log space, representing the central frequency for each mode. A linear regression is run on the central frequencies to estimate the slope of the spectrum.

From Fig. 3.6 we also observe a clear linear relationship in the log space of central frequency and energy pair $(\bar{f}_j, \bar{E}_j)$, which are marked as the black crosses. This indicates a power spectrum relation

$$E(f) \sim \frac{1}{f^{\alpha}}. \tag{3.16}$$

The power spectrum exponent α controls how fast the energy decays from lower to higher frequency, and is key to the property of the spectrum and the associated time series. It has

been long observed that many time series in nature have α close to one, well-known as the $1/f$ spectrum or "pink noise". In Fig. 3.6 the solid line in each plot is obtained from linear regression of $(\log(\bar{f}_j), \log(\bar{E}_j))$. The negative slope of the line estimates the power spectrum exponent α for the time series.

Comparing the financial time series, the cryptocurrencies exhibit a more rapid dissipation of energy with respect to frequency than that of S&P 500 and GLD. This reveals that the rapid fluctuations in the cryptocurrencies tend to have relatively smaller magnitude than those for S&P 500, which contradicts the notion that the cryptocurrencies are volatile with rapid movements. In terms of power spectrum, S&P 500 has exponents closest to the $1/f$ spectrum, while the cryptocurrencies and GLD generally have $\alpha > 1$. In Table 3.2, we computed the α exponents for the top crytocurrencies in market capitalization, and compared against the results from three traditional financial market indices. The results show consistently that cryptocurrencies and GLD mostly have $\alpha > 1$, while the equity indices typically have α below and close to 1.

Crypto	α	R^2	Crypto	α	R^2	Crypto	α	R^2
BTC	1.2070	0.9911	ETH	1.1681	0.9898	BNB	1.2460	0.9333
ADA	1.3294	0.9865	XRP	1.2307	0.9844	LTC	1.1832	0.9994
THETA	1.1371	0.9591	LINK	0.9433	0.9200	BCH	1.0574	0.9887
XLM	1.1545	0.9964	DOGE	1.1759	0.8701	TRX	1.1160	0.9621
MIOTA	1.0950	0.9860	XMR	1.0613	0.9809	EOS	1.0811	0.9817
Index	α	R^2	Index	α	R^2	Index	α	R^2
S&P 500	0.8937	0.9924	DJI	0.8776	0.9191	GLD	1.1735	0.9669

Table 3.2: Power spectrum exponents α of top 30 cryptocurrencies by market capitalization, excluding stable coins and coins with less than 3 years of history. At the bottom of the table are equity indices (S&P 500 and Dow Jones) and gold ETF (GLD) for comparison. Estimations are implemented on the time period 1/4/2016–3/29/2021 (or start from inception dates or earliest available records).

3.4.2 Frequency Synchronization

To further investigate the spectral difference between cryptocurrencies and traditional financial market, we compare between each two time series the central frequencies $\bar{f}_j,\ j = 1, \cdots, n$. In Fig. 3.7, we show the log-log scatter plots of the instantaneous frequencies of four asset pairs: BTC vs ETH, BTC vs XRP, BTC vs S&P 500, and ETH vs S&P 500. Each point on the plot is a pair of instantaneous frequencies of the two time series recorded at the same time, and the central frequencies are marked as the black crosses. The solid straight line of unit slope shows the reference line for identical frequencies. We observe that the cryptocurrencies, e.g. BTC vs ETH and BTC vs XRP, share very similar frequency profiles. The central frequencies of all the IMF components are very close between two cryptocurrencies. On the other hand, the central frequencies deviate from the identical line at low frequency modes for BTC vs S&P 500 and ETH vs S&P 500. More specifically, the cryptocurrencies show slower dynamics in the longer term components, while the fast modes of both cryptocurrencies and S&P 500 have very similar mean frequencies.

The central frequencies of the IMF components show consistent similarity within the cryptocurrency market and common deviation from the traditional stock market. The interesting identical frequency profile within the cryptocurrency market suggests synchronization, which is a typical phenomenon in nonlinear dynamics with interaction (see e.g. [105]). In order to quantify the frequency synchronization level between two time series x_1 and x_2, we define the associated *frequency deviation* as follows:

$$D(x_1, x_2) := \sum_{j=1}^{n} \log^2 \left(\frac{\bar{f}_j^{(1)}}{\bar{f}_j^{(2)}} \right) = \| \log(\bar{f}^{(1)}) - \log(\bar{f}^{(2)}) \|^2. \tag{3.17}$$

A lower frequency deviation value represents higher synchronization level. In particular, we have $D(x_1, x_2) = 0$ if and only if the central frequencies of all the IMF components are identical for x_1 and x_2, meaning the two time series are fully synchronized.

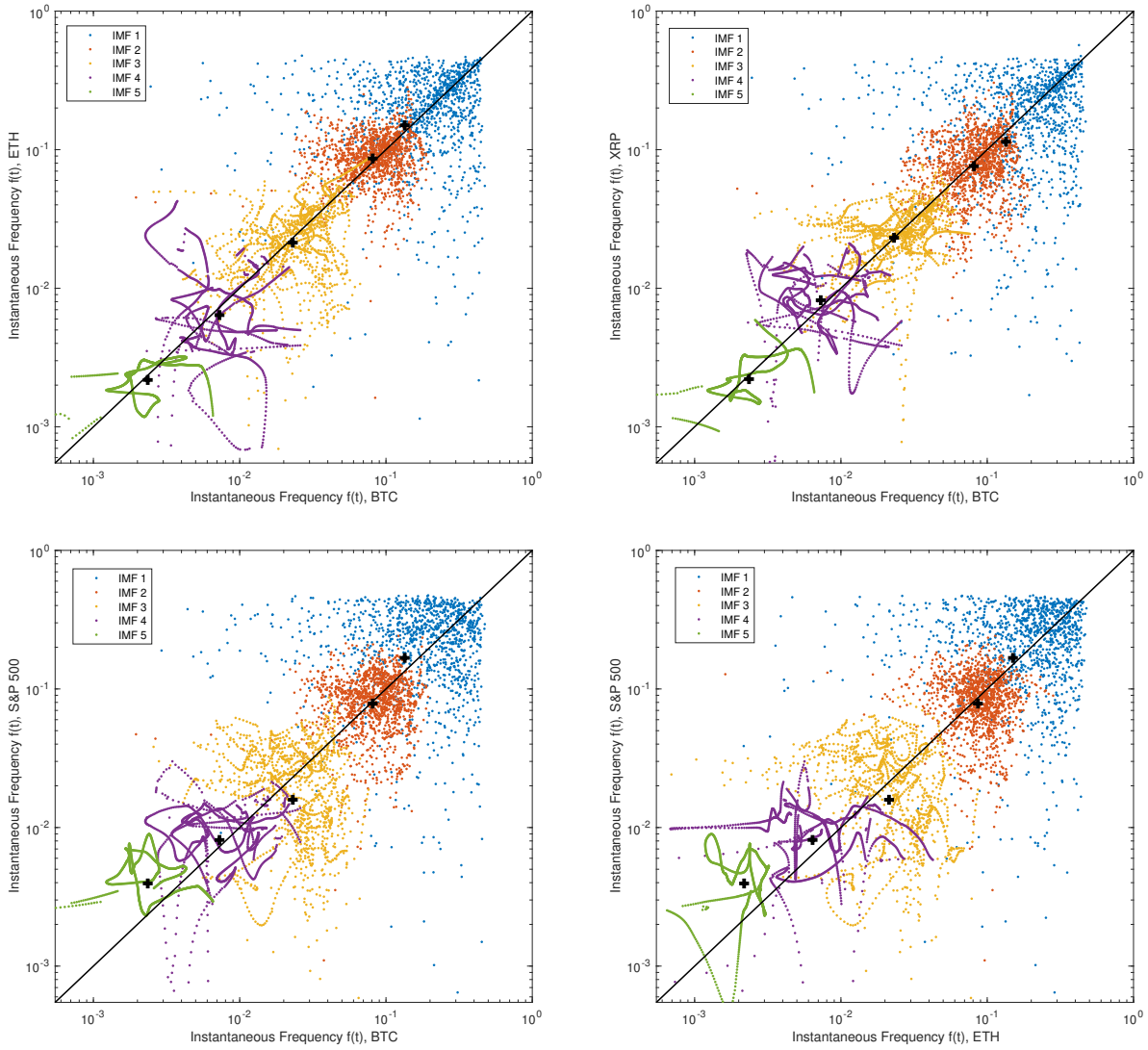

Figure 3.7: Instantaneous frequency: BTC v.s. ETH (top left), BTC v.s. XRP (top right), BTC v.s. S&P 500 (bottom left), and ETH v.s. S&P 500 (bottom right). The estimations are done on the time period from 1/4/2016 to 3/29/2021.

It is of interest to see how the cryptocurrencies have been evolving compared against the stock market represented by S&P 500. In Fig. 3.8, we plot the estimated power spectrum exponent α based on a 2-year rolling window, for the cryptocurrencies BTC, ETH, LTC, XRP, and the benchmark S&P 500. We see that the power spectrum exponents α of cryp-

tocurrencies are moving closer to that of S&P 500. Furthermore, in the second plot, we see that the frequency deviation from S&P 500 defined in Eq. (3.17) has decreased from July 2017 till July 2020 for all the four cryptocurrencies. These observations are evidence of synchronization over a relatively long period of time and suggest an underlying interaction between cryptocurrencies and the stock market.

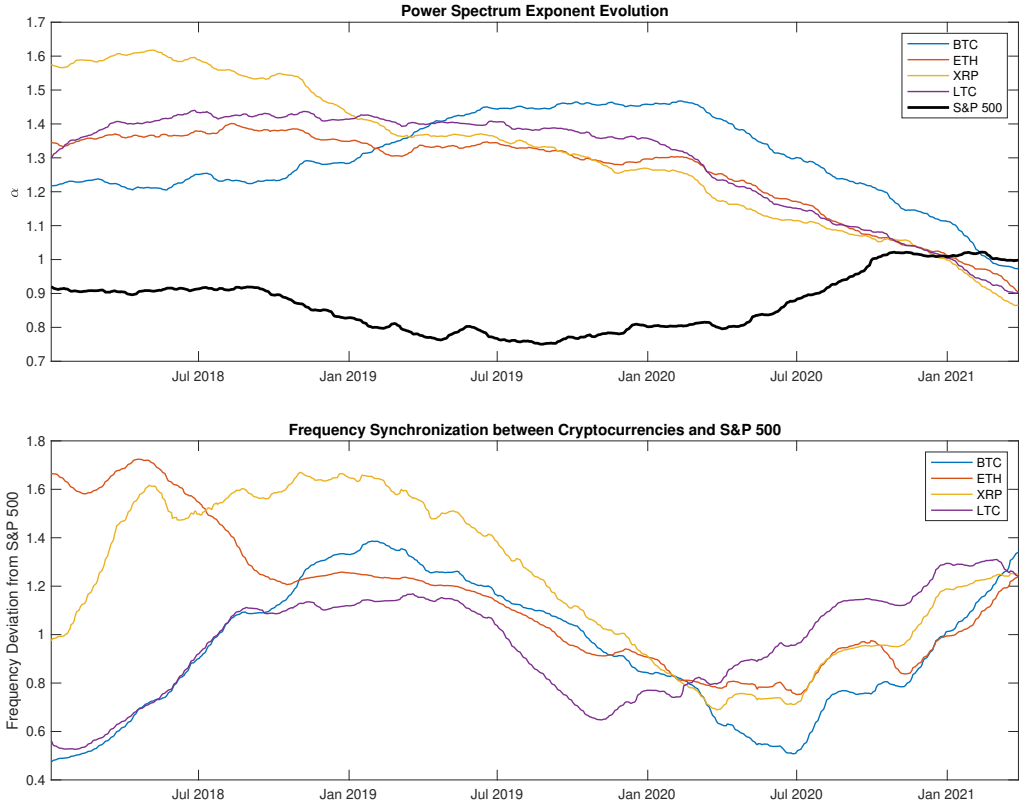

Figure 3.8: Top: The evolution of the power spectrum exponents for four major cryptocurrencies {BTC, ETH, XRP, LTX}, compared against S&P 500. Bottom: The frequency deviations from S&P 500 for the 4 cryptocurrencies. In both plots, results are computed based on 2-year rolling windows from 1/4/2016 to 3/29/2021.

3.5 Concluding Remark

We have presented the ACE-EMD method for multiscale analysis of nonstationary financial time series with a focus on cryptocurrency prices. The key outputs of this method are the series of intrinsic mode functions, along with the time-varying instantaneous amplitudes and instantaneous frequencies. Different combinations of modes allow us to reconstruct the financial time series based on different timescales. This allows us to better understand cryptocurrency price movements due to short-term fluctuations vs. long-term trends. It also sheds light on the multiscale properties of cryptocurrency volatilities that set them apart from traditional equities. In the same spirit, we apply Hilbert spectral analysis to compute the associated instantaneous energy-frequency spectrum for a collection of cryptocurrencies. In particular, the power spectrum exponents of cryptocurrencies have been significantly higher than that of S&P 500, but are seen to be converging in recent months.

Chapter 4

INTRADAY MULTISCALE VOLATILITY

4.1 Overview

In this chapter we discuss the multiscale behaviors of intraday price movement for high-frequency data. The focus is on how univariate risk measures such as variance and volatility change with timescale. Starting with definitions around multiscale volatility, we show for a wide range of typical price processes how they should behave. Then we introduce multiscale models for high-frequency intraday price, using fractional Brownian motion and microstructure noise as the building blocks. The proposed noisy fractional Brownian motion model is shown to possess different volatility behavior. Algorithms for numerical estimation from time series observations are then introduced, with new Hurst exponent estimator proposed for the noisy fractional Brownian motion model. Lastly we show experimental results on real-world high-frequency price data for a collection of U.S. stocks and ETFs and compare then to theoretical results from the models.

4.2 Multiscale Volatility

As the basic setting, the price of a financial asset at time t is denoted as P_t. The standard (arithmetic) return over the period from t to $t+\tau$ is defined as $P_{t+\tau}/P_t - 1$. While the arithmetic return is convenient for calculating wealth change in a portfolio, it is not additive when dealing with multiple periods. For multiscale study which mainly considers price movements over periods of time, it is more convenient to use the logarithmic return $\log(P_{t+\tau}/P_t)$. The arithmetic return is well approximated by the logarithmic return at small price movement which is usually true especially for the intraday scenario. For simplicity of notation let's directly analyze the log price defined as follows.

Definition 1. Let P_t be the random process of an asset price. Its log price is defined as $X_t := \log(P_t)$. The log return over the time interval from t to $t+\tau$ is defined as

$$r^X_{t,\tau} = X_{t+\tau} - X_t. \tag{4.1}$$

In the later part of the thesis, we will mainly work on the log price, as it has good properties when considering multiple time periods, thus a better choice for multiscale study. In most of the following discussions, the term "return" will just refer to the log return for simplicity.

4.2.1 Definition

$r_{t,\tau}$ is a time series of t in the time domain. However, under stationary increments assumption, the distribution of $r_{t,\tau}$ depends on τ only, which is the main focus of this study. (Often times this is not true, but a good approximation for studying intraday return behaviors.) The two most important properties of a return distribution are its expectation and variance. Since expectation is a linearly additive operator, the expected return will just scale proportionally with τ, i.e. $\mathbb{E}[r^X_{t,\tau}] \propto \tau$. On the other hand, the the second moment behaviors will be much more interesting. The scaling properties of variance can be described by the following:

Definition 2. Let the log price $X_t \in \mathbb{R}$ be a stochastic process with stationary increments on $0 \le t \le T$. Define the following functions of $\tau \in \mathbb{R}^+$:

- Multiscale variance:

$$\mathcal{V}[X](\tau) := \mathbb{V}[r^X_{t,\tau}] = \mathbb{E}[(r^X_{t,\tau})^2] - (\mathbb{E}[r^X_{t,\tau}])^2, \tag{4.2}$$

- Multiscale volatility:

$$\mathcal{S}[X](\tau) := \sqrt{\frac{\mathcal{V}[X](\tau)}{\tau}}. \tag{4.3}$$

- Growth rate:

$$\Lambda[X](\tau) := \frac{1}{2}\frac{d(\log \mathcal{V}[X](\tau))}{d(\log \tau)}. \tag{4.4}$$

Remark. While $\mathcal{V}[X]$ and $\mathcal{S}[X]$ are functions describing the variance and volatility at different scales, the growth rate Λ is an unit-less operator controlling the speed of risk accumulation on every scales. We will see how it can be used to adjust noisy variance in the next section.

The following property is useful for the rest of the thesis:

Property 2.1 (Summation of processes)**.** *Let $X_t^{(1)}, \cdots, X_t^{(n)}$ be a sequence of independent processes, then the multiscale variance is linearly additive:*

$$\mathcal{V}\left[\sum_{i=1}^{n} X^{(i)}\right](\tau) = \sum_{i=1}^{n} \mathcal{V}[X^{(i)}](\tau). \tag{4.5}$$

Proof. The proof is straightforward using the property of linear additive variance for independent random variables:

$$\mathcal{V}\left[\sum_{i=1}^{n} X^{(i)}\right](\tau) = \mathbb{V}\left[\sum_{i=1}^{n} X_{t+\tau}^{(i)} - X_t^{(i)}\right] = \sum_{i=1}^{n} \mathbb{V}[X_{t+\tau}^{(i)} - X_t^{(i)}] = \sum_{i=1}^{n} \mathcal{V}[X^{(i)}](\tau).$$

□

4.2.2 Common Behaviors

The multiscale functions can actually take on many forms (not even always increasing, e.g. $X_t = \sin(kt + 2\pi u)$, with $u \sim \mathcal{U}(0,1)$), however, for some special but wide classes of processes, the multiscale functions have certain behaviors as shown in the following:

Proposition 1 (Uncorrelated increments)**.** *Suppose X_t has stationary increments that are uncorrelated (random walk), i.e.*

$$\mathbb{E}[(X_{t_2} - X_{t_1})(X_{t_3} - X_{t_2})] = \mathbb{E}[X_{t_2} - X_{t_1}] \cdot \mathbb{E}[X_{t_3} - X_{t_2}], \quad \forall t_1 < t_2 < t_3.$$

Then there must be

$$\mathcal{V}[X](\tau) \propto \tau, \quad \mathcal{S}[X]'(\tau) = 0, \quad \Lambda[X](\tau) = \frac{1}{2}.$$

Proof. For any $\tau_1, \tau_2 > 0$, we can show that $\mathcal{V}[X](\tau_1 + \tau_2) = \mathcal{V}[X](\tau_1) + \mathcal{V}[X](\tau_2)$. Then

$$\begin{aligned}
\mathcal{V}[X](\tau_1+\tau_2) =& \mathbb{E}[(X_{\tau_1+\tau_2} - X_0)^2] - (\mathbb{E}[X_{\tau_1+\tau_2} - X_0])^2 \\
=& \mathbb{E}[((X_{\tau_1+\tau_2} - X_{\tau_1}) + (X_{\tau_1} - X_0))^2] - (\mathbb{E}[(X_{\tau_1+\tau_2} - X_{\tau_1}) + (X_{\tau_1} - X_0)])^2 \\
=& \mathbb{E}[(X_{\tau_1+\tau_2} - X_{\tau_1})^2] + \mathbb{E}[(X_{\tau_1} - X_0)^2] + 2\mathbb{E}[(X_{\tau_1+\tau_2} - X_{\tau_1})(X_{\tau_1} - X_0)] \\
& - (\mathbb{E}[X_{\tau_1+\tau_2} - X_{\tau_1}])^2 - (\mathbb{E}[X_{\tau_1} - X_0])^2 - 2\mathbb{E}[X_{\tau_1+\tau_2} - X_{\tau_1}]\mathbb{E}[X_{\tau_1} - X_0] \\
=& \mathcal{V}[X](\tau_2) + \mathcal{V}[X](\tau_1).
\end{aligned}$$

Then there must be $\mathcal{V}[X](\tau) \propto \tau$.[1] According to Definition 2, we can derive that $\mathcal{S}[X](\tau)$ is a constant, and $\Lambda[X](\tau) = \frac{1}{2}$. □

Remark. Typically the requirement for a random walk is independent increments. Here we loosen the condition to be uncorrelated increments, which guarantees a larger class of process than that restricted by stochastic independence, while still satisfying the scale-independent volatility property. The following is one example.

Example (Stochastic volatility)**.** Let B_t and W_t be independent standard Brownian motions. Define the stochastic volatility processes:

$$dX_t = \mu dt + \sqrt{\nu_t} dB_t, \quad \nu_t = \alpha_t dt + \beta_t dW_t.$$

B_t, W_t are independent Brownian motions. Even though X_t does not have independent increments (volatility clustering), its increments are uncorrelated:

$$\mathbb{E}\left[\left(\int_{t_1}^{t_2} \sqrt{\nu_t} dB_t\right)\left(\int_{t_2}^{t_3} \sqrt{\nu_t} dB_t\right)\right] = 0.$$

Therefore, X_t enjoys the multiscale properties in Proposition 1.

---

[1] Rigorously speaking, we can only conclude from additive-ness to linearity for τ on a rational grid. Usually $\mathcal{V}[X](\tau)$ is considered to be a continuous function to justify the claim. Detailed discussion will not be the main focus of this thesis.

4.3 High-frequency Multiscale Models

As we have seen from the previous section, scaling behavior is prescribed by the type of random process. In this section we discuss multiscale volatility in the intraday high-frequency setting. We first review the classic fractional Brownian motion model proposed in [84], which is widely used in finance to model long-range dependency. The market microstructure noise model is introduced later for intraday high-frequency data. The combination of the two creates interesting new scaling behavior, which will be the core of discussion in this section.

4.3.1 Fractional Brownian motion

The fractional Brownian motion model proposed in [84] is defined as follows:

Definition 3. Fractional Brownian motion (fBm) $B_t^H,\ t \geq 0$, is a continuous-time Gaussian process satisfies that $\mathbb{E}[B_t^H] = 0,\ \forall t \geq 0$, and has the following covariance function:

$$\mathbb{E}[B_t^H B_s^H] = \frac{1}{2}(t^{2H} + s^{2H} - |t-s|^{2H}),$$

where $t, s \geq 0$, and $H \in (0,1)$ is called the Hurst exponent.

When $H = \frac{1}{2}$, fBm reduces to the standard Brownian motion with independent increments. When $H \in (0, 1/2)$, the process is anti-persistent or mean-reverting, with increments negatively correlated. When $H \in (1/2, 1)$, the process is persistent or trending, with increments positively correlated.

Fractional Brownian motion admits various forms of stochastic integral representation which can be found in Chapter 1.2 of [17]. In this study we use the following representation to construct a fractional Brownian motion from an underlying standard Brownian motion:

$$B_t^H = \mathcal{B}(W; H)_t := \frac{1}{\Gamma(H+1/2)} \int_{-\infty}^{t} \left((t-s)^{H-1/2} - (-s)_+^{H-1/2} \right) dW_s, \tag{4.6}$$

where $W_t,\ t \in \mathbb{R}$ is the underlying Brownian motion. For the rest of this thesis, we will be using this representation to define correlated fractional Brownian motions, and generate sample paths of fractional Brownian motion (pairs).

Definition 4 (Self-similarity)**.** An univariate stochastic process $X_t \in \mathbb{R},\ t \geq 0$ is self-similar if there exist $H \in (0, 1)$, s.t.

$$X_{\lambda t} \overset{dist.}{=} \lambda^H X_t, \quad \forall t \geq 0. \tag{4.7}$$

One can check that a fractional Brownian motion has stationary increments and satisfies the self-similar property, with the Hurst exponent serving as the scaling exponent H. (In fact sometimes fBm is defined in such way.) In this study we are mostly interested in the second moments scaling. Following Definition 2, it is easy to show that a fractional Brownian motion with Hurst exponent H has

$$\mathcal{V}[B^H](\tau) = \tau^{2H}, \quad \mathcal{S}[B^H](\tau) = \tau^{H-1/2}, \quad \Lambda[B^H](\tau) = H. \tag{4.8}$$

When $H+1/2$, the functions reduce to the form prescribed by Proposition 1. When $H < 1/2$, the volatility decreases with the scale, and increase with τ in the $H > 1/2$ case. Also note that the growth rate Λ is a constant for any self-similar process, which can serve as an important indicator for pure self-similarity..

4.3.2 Microstructure noise

In the study of intraday high frequency financial time series, there have been a variety of studies on the market microstructure noise effect. The market microstructure noise has been recognized to play an important role in finance, with earlier research date back to [19], and resent studies on various types of assets such as the cryptocurrency [21]. At any time point, the asset price is the result from a trading activity. Various frictions in the trading process (e.g bid-ask spread, the discreteness of price, etc) may lead to "noise" in the price process [60]. Several models have been proposed to formulate the market microstructure noise problem in different ways. In the recent work by [28], the noise is implicitly modeled by the discrepancy between stochastic volatility and realized volatility, and the authors showed its effect on estimating the Hurst exponent of the volatility process.

To avoid any confusion on the concept of "market microstructure noise", we fix our modeling to be the same as the form in [4, 126, 106], among others. It worth pointing out that, in most of these studies, the aim is to improve realized volatility estimation, and the "noise" is regarded as some deviation from the "true price". However, a price observed is the price that was actually traded at, thus even the "noisy" price is financially meaningful. Therefore, the "noise" is not only something to be removed for better estimation of other quantities, but also worth careful study itself. The philosophy in this study is to consider the "noise" as an integral part of the price dynamics itself. In this section we are going to establish its multiscale behaviors.

Definition 5. Let $X_t \in \mathbb{R}$, $t \geq 0$ be some latent stochastic process. The noisy price process is defined as

$$P_t = \exp(Y_t), \quad Y_t = X_t + \epsilon_t, \quad \forall t \geq 0,$$

where ϵ is the microstructure noise s.t. $\epsilon \perp X$, $\mathbb{E}[\epsilon_t] = 0$, $\mathbb{V}[\epsilon_t] = \sigma_\epsilon^2$, and ϵ_t i.i.d., for $\forall t \geq 0$.

Remark. The noise can be viewed as an effect acting on the price whenever there is a trade. Unfortunately, the noisy price process defined in this way will be nowhere continuous. However, in the market microstructure view the price will also be a non-continuous time process, as the trades happen at discrete time points. This is the reason that we should always keep $\tau > 0$. Detailed discussion will be beyond the scope of this thesis.

Directly apply Property 2.1, and use the fact that

$$\mathcal{V}[\epsilon](\tau) = \mathbb{V}[\epsilon_{t+\tau} - \epsilon_t] = 2\sigma_\epsilon^2, \quad \forall \tau > 0,$$

we can derive the multiscale functions in Definition 2 for the noisy price process:

$$\mathcal{V}[Y](\tau) = \mathcal{V}[X](\tau) + 2\sigma_\epsilon^2, \tag{4.9}$$

$$\mathcal{S}[Y](\tau) = \sqrt{\mathcal{S}^2[X](\tau) + \frac{2\sigma_\epsilon^2}{\tau}}. \tag{4.10}$$

Note that the volatility $\mathcal{S}[Y](\tau) \to \infty$ as $\tau \to 0^+$, for any $\forall \sigma_\epsilon > 0$. The undefined volatility at $\tau = 0$ is the result from non-continuity of the noisy price path. Unlike the sensitiveness of volatility to the noise, the following proposition shows a conservation law for the variance and the growth rate.

Proposition 2. *Define the growth-adjusted variance operator*

$$\tilde{\mathcal{V}} := \mathcal{V} \cdot \Lambda \tag{4.11}$$

Then for any noisy process $Y_t = X_t + \epsilon_t$ as defined in Definition 5,

$$\tilde{\mathcal{V}}[Y] \equiv \tilde{\mathcal{V}}[X]. \tag{4.12}$$

Proof. Using the result in Eq. (4.9), we can derive

$$\begin{aligned}
\Lambda[Y](\tau) &= \frac{1}{2}\frac{d}{d\log(\tau)}\log(\mathcal{V}[X](\tau) + 2\sigma_\epsilon^2) \\
&= \frac{1}{2(\mathcal{V}[X](\tau) + 2\sigma_\epsilon^2)}\frac{d}{d\log(\tau)}(\mathcal{V}[X](\tau) + 2\sigma_\epsilon^2) \\
&= \frac{1}{2\mathcal{V}[Y](\tau)}\frac{d\mathcal{V}[X](\tau)}{d\log(\tau)}.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\tilde{\mathcal{V}}[Y](\tau) &= \mathcal{V}[Y](\tau) \cdot \Lambda[Y](\tau) \\
&= \mathcal{V}[Y](\tau) \cdot \frac{1}{2\mathcal{V}[Y](\tau)}\frac{d\mathcal{V}[X](\tau)}{d\log(\tau)} \\
&= \frac{1}{2}\frac{d\mathcal{V}[X](\tau)}{d\log(\tau)}.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\tilde{\mathcal{V}}[X](\tau) &= \mathcal{V}[X](\tau) \cdot \Lambda[X](\tau) \\
&= \mathcal{V}[X](\tau) \cdot \frac{1}{2}\frac{d\log(\mathcal{V}[X](\tau))}{d\log(\tau)} \\
&= \mathcal{V}[X](\tau) \cdot \frac{1}{2\mathcal{V}[X](\tau)}\frac{d\mathcal{V}[X](\tau)}{d\log(\tau)} \\
&= \tilde{\mathcal{V}}[Y](\tau),
\end{aligned}$$

which holds for any $\tau > 0$. □

4.3.3 Noisy fractional Brownian motion

Based on definitions from the previous sections, we can model the asset log price at high frequency as a noisy fractional Brownian motion defined as follows:

Definition 6. For Hurst exponent $H \in (0,1)$, drift $\mu \in \mathbb{R}$, volatility parameter $\sigma \in \mathbb{R}^+$, and initial value $Y_0 \in \mathbb{R}$, the noisy fractional Brownian motion is defined as:

$$Y_t = Y_0 + \mu t + \sigma B_t^H + \epsilon_t, \tag{4.13}$$

where ϵ_t is the microstructure noise with variance $\mathbb{V}(\epsilon_t) = \sigma_\epsilon^2$.

Property 6.1. *The multiscale functions of the noisy fractional Brownian motion Y_t are*

$$\begin{aligned}
\mathcal{V}[Y](\tau) &= \sigma^2(\tau^{2H} + 2\eta), \\
\mathcal{S}[Y](\tau) &= \sigma \cdot \sqrt{\tau^{2H-1} + \frac{2\eta}{\tau}}, \\
\Lambda[Y](\tau) &= H \cdot \frac{\tau^{2H}}{\tau^{2H} + 2\eta},
\end{aligned}$$

where $\eta := \sigma_\epsilon^2/\sigma^2$ is the noise ratio.

Proof. Directly application of Eq. (4.9) and (4.10) to the multiscale functions of fractional Brownian motion in Eq. (4.8) yields the variance and volatility functions. For growth rate, use Proposition 2 to derive

$$\Lambda[Y](\tau) = \frac{\Lambda[B^H](\tau) \cdot \mathcal{V}[B^H](\tau)}{\mathcal{V}[Y](\tau)} = H \cdot \frac{\tau^{2H}}{\tau^{2H} + 2\eta}.$$

□

Remark. The effect of the microstructure noise on the multiscale functions is only through the noise ratio η. One should be careful that the ratio is *not* unit-less. The reason is that while σ_ϵ^2 is the variance of the noise, σ^2 is *not* the variance of the latent process itself, but contributes in the form of $\sigma^2\tau^{2H}$. The unit of η is thus $[\text{Time}]^{2H}$, depending on H as well.

4.3.4 Asymptotic Behaviors

In this section let's discuss the asymptotic behaviors of the multiscale functions for noisy fractional Brownian motion. The results will be useful when comparing against numerical estimation on real world data.

- $\tau \to \infty$:

$$\begin{aligned}
&\mathcal{V}[Y](\tau) = \sigma^2 \tau^{2H} + \mathcal{O}(1),\\
&\mathcal{S}[Y](\tau) = \sigma \tau^{H-1/2} + \mathcal{O}(\tau^{-1/2}),\\
&\Lambda[Y](\tau) \to H.
\end{aligned}$$

  The asymptotic behavior under big scale limit is approximating the latent fractional Brownian motion itself, because the microstructure noise effect dissipates at long range. Note that the big O terms are dominated by the asymptotic terms by rates controlled by $H > 0$. The larger the Hurst exponent, the faster the convergence rate.

- $\tau \to 0$:

$$\begin{aligned}
&\mathcal{V}[Y](\tau) = 2\sigma_\epsilon^2 + \mathcal{O}(\tau^{2H}),\\
&\mathcal{S}[Y](\tau) \to \infty,\\
&\Lambda[Y](\tau) \to 0.
\end{aligned}$$

  We see that the variance will not vanish to zero when then scale goes to zero, but to a constant determined by the microstructure noise level. As discussed before, the volatility is not well-defined at zero. At very small scale the microstructure noise dominates the process, making the path very rough. It also worth note that even for a plain fractional Brownian motion without noise, the volatility is not well-defined at zero for any $H \in (0, 1/2)$. The variance growth is also "overwhelmed" by the noise at small scale, making the growth rate close to zero.

For Hurst exponent $H \in (1/2, 1)$, the volatility goes to infinity both at zero and at infinity. The multiscale volatility function is not monotonic and there exist a critical timescale minimizing the volatility. The optimality is achieved with

$$\tau^* = \left(\frac{\eta}{H-1/2}\right)^{\frac{1}{2H}}, \quad \mathcal{S}[Y](\tau^*) = \sigma \cdot \sqrt{2H} \left(\frac{\eta}{H-1/2}\right)^{\frac{H-1/2}{2H}}.$$

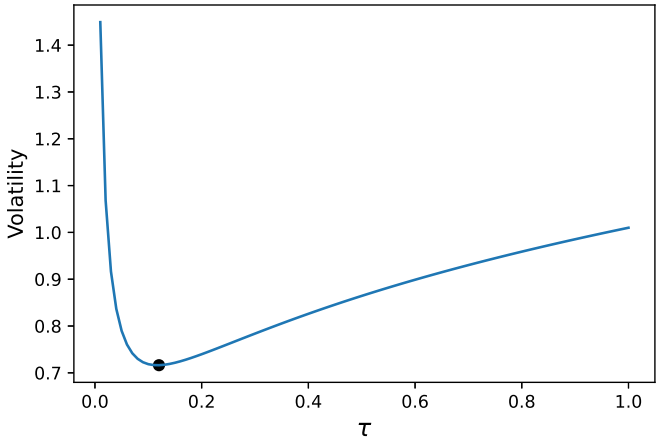

Figure 4.1: Volatility curve of a noisy Fractional Brownian motion with $H = 0.75, \sigma = 1, \eta = 0.01$. The minimal volatility is marked with the black dot.

We end this section with a discussion on the question: what is the role of the microstructure noise played in the multiscale behavior of price? In the previous chapters of the thesis, multiscale analysis is through components on different scales, from short-term fluctuation to long-term trend. While the noise term be also be viewed as some form of "component", the question can be restated as: what is the scale of the noise effect? From the asymptotic analysis, we see that the noise gradually dominates as $\tau \to 0$. We may refer to the noise component as "scale 0". Its effect, though vanishes at long range, has impact on all other scales.[2]

[2]An interesting comparison can be made with the mode mixing effect and the noise-assisted ensemble solution discussed in Sect. 2.2.2. Even though noise is believed to reside in the highest frequency term, it can potentially affect some of the lower frequency modes.

4.4 Numerical Estimation

In reality, we are only able to observe a sequence of price at discrete time points $t_1, t_2, \cdots, t_n$. At the highest frequency, these time points are the instants a trade happens, which may not be nicely distributed with equal length intervals. For simplicity, in this thesis we assume the time series is evenly sampled, which is a good approximation when the scale is not too small (the experimental data to be used in the next section is at 3 seconds).

For the rest of this chapter, we stick to the setting that one sample path of price $P_1, P_2, \ldots, P_n$ is observed discretely at equal-spaced time points $\delta t, 2\delta t, \cdots, n\delta t$. The rest of this section discusses numerical estimation of the quantities describing multiscale return behaviors.

4.4.1 Multiscale volatility

Let's first convert the observations to the log price time series

$$X_i = \log(P_i), \quad i = 1, \cdots, n.$$

For $m \in \mathbb{N}^+ < n$, we can estimate the multiscale functions in Definition 2 at scale $\tau = m\delta t$ as follows. Let $\mathcal{P}_m \subset \{(i,j) | 1 \le i, j \le n,\ j - i = m\}$ be a subset of index pairs spanning a range of length m.

- Variance:

$$V_m = \frac{1}{|\mathcal{P}_m|} \sum_{(i,j)\in\mathcal{P}_m} (X_j - X_i - m\mu)^2. \tag{4.14}$$

  where $\mu = \frac{X_n - X_1}{n-1}$ is the estimated drift for detrending the process.

- Volatility:

$$S_m = \sqrt{\frac{V_m}{m\delta t}}. \tag{4.15}$$

After computing a sequence of V_m values, the growth rate can be estimated by

$$\Lambda_m = \frac{\log(V_{m+1}/V_m)}{\log(1 + 1/m)}. \tag{4.16}$$

Remark. The index pair set $\mathcal{P}$ is made flexible. In practice, there are some common choices. The continuous rolling window $\{(1, m+1), (2, m+2), \cdots, (n-m, n)\}$ is a popular one, which forms the largest possible set. In some cases people use non-overlapping windows $\{(1, m+1), (m+1, 2m+1), \cdots, ((k-1)m+1, km+1)\}$.

Remark (Bias-ness). The estimator of the variance in Eq. (4.14) is biased, even if the index pair set $\mathcal{P}$ consists of non-overlapping windows. The reason is that the log return on disjoint windows can be correlated. However, if the true value of the drift μ is know, Eq. (4.14) is an unbiased estimator of the variance. In practice there are many ways to detrend the process, such as fitting an ordinary least square, or using EMD introduced in Chapter 2. Nevertheless, even if we have an unbiased estimator of the variance, S_m is still biased because of the concave square root function.

4.4.2 Estimating Hurst exponent

The Hurst exponent is the key parameter in self-similar processes and the noisy fractional Brownian motion model. For the past decades, several estimation methods were proposed, including the original Rescale Range (R/S) analysis [57] and the popular Detrended Fluctuation Analysis (DFA) [102]. The Generalized Hurst exponent was proposed in Eq. (1) and (2) of [12] to estimate the scaling exponent for each order of moment. Because moment scaling is an important tool to understand the properties of financial time series, there have been numerous applications of the generalized Hurst exponent to finance [32, 83, 41].

For the same reason, we focus on modification of the generalized Hurst exponent algorithm in this study. We first review the original Rescaled Range (R/S) method, as it is the standard when comparing different algorithms. The review of generalized Hurst exponent estimation comes next, followed by discussion on how the estimator can be biased under market microstructure noise and how to remove the bias. Before proceeding, we have to point out that no method is perfect, and they can even disagree with each other significantly [119].

Rescaled range analysis

The original rescaled range (R/S) analysis algorithm is described below. The algorithm is adapted to the log price observation scenario:

Rescaled Range For time series observation $X_1, X_2, \cdots, X_n$, take a consecutive subsequence : $X_k, X_{k+1}, \cdots, X_{k+m}$, and apply the following operations:

- Estimate the drift $\mu = \frac{1}{m}(X_{k+m} - X_k)$.
- Detrend the partial time series: $Z_t = X_{k+t} - \mu t$, for $t = 1, 2, \cdots, m$.
- Compute the range R at scale m:

$$R(m) = \max(Z_1, Z_2, \cdots, Z_m) - \min(Z_1, Z_2, \cdots, Z_m).$$

- Compute the return standard deviation S at scale m:

$$S(m) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_{k+i} - X_{k+i-1} - \mu)^2}.$$

- Estimate the rescaled range at length m as $(R/S)_m = R(m)/S(m)$, and average over all non-overlapping length-m sub-sequences.

Following the same procedure, we can estimate the rescaled range at different values of m. The Hurst exponent H is then estimated by fitting $(R/S)_m \sim m^H$. Usually an ordinary least square is fitted on the log-log plot $\log(R/S)_m \sim \log m$, taking the slope as the estimator.

The R/S algorithm, however, will produce biased result when the time series is noisy. Here we briefly show it by considering how the range (R) and the standard deviation (S) change under the noise. Denote the noisy time series as $\tilde{X}_t = X_t + \epsilon_t$. It is easier to consider the return standard deviation first:

$$\tilde{S}(m) \approx \sqrt{S^2(m) + 2Var(\epsilon)} \approx S(m) + \frac{Var(\epsilon)}{S(m)}.$$

The return standard deviation will be larger. However, it is the scaling property of $S(m)$ that is important. In that sense, because all $S(m)$'s are computed from the finest timescale returns, its expectation can be assume to be the same if the process is stationary. The noise just makes the same normalization factor smaller, thus won't affect the final scaling behavior from this part.

On the other hand, the range $R(m)$ depends on m in most cases, and will experience a change that due to the noise. $R(m)$ is the range of the detrended log price path over m consecutive intervals. Imagine a white noise is added to the path of length m, and a new maximal value and a new minimal value are obtained. The longer the length m, the higher the chance for the maximal value to be higher and the minimal value to be lower. Therefore, the range $R(m)$ will increase under the noise, and the level of increment depends on m, causing potential bias to the slope estimation. A detailed analytical analysis of the bias amount and direction will be over-complicated, so we only provide numerical evaluation of the deviation in Table 4.1.

Generalized Hurst exponent

By fitting the moments of increments against the scale, the original Hurst exponent can be generalized to be order-based [12]. Generalized Hurst exponent, denoted by H_q, studies the scaling property of the increment distribution at the q-th order moment. For a time series observation $X_i,\ n = 1, 2, \cdots, n$, compute the q-th order structure function as

$$G_q(m) := \langle |X_{i+m} - X_i|^q \rangle_i ,$$

where the $\langle \cdot \rangle_i$ operator takes average over all feasible index i. H_q is obtained by fitting

$$G_q(m) \sim m^{qH_q}.$$

For self-similar processes, H_q is the same for all order q. In our study, we are mostly interested in the scaling of the second moment, thus fixing $q = 2$. The algorithm is to compute

$$G_2(m) := \left\langle (X_{i+m} - X_i)^2 \right\rangle_i \sim m^{2H_2}.$$

For simplicity, in the later part of the chapter we will just use $H = H_2$. Notice that this is actually the same as the multiscale variance function V_m estimator in Eq. (4.14) (with data centered). The formal variance analysis algorithm is as follows.

Variance Analysis

- Let $\mathcal{M} \subset \mathbb{N}^+$ be the set of scales to estimate variance.
- For $m \in \mathcal{M}$, compute V_m using Eq. (4.14).
- Fit ordinary least square linear regression of $\log(V_m) \sim \log(m)$. Denote the slope as α.
- Estimator $\hat{H} = \alpha/2$.

Similar to the original rescaled range method, the Hurst exponent is estimated by taking log of both V_m and m, and fitting least square linear regression to find the slope. However, as we have shown in Eq. (4.9), there will be a constant term in the structure function if the data is subject to microstructure noise. Therefore, directly taking log and fit linear regression will cause inevitable bias in the estimation of H. The following proposition provides a analytical estimate of the deviation from the true value.

Proposition 3. *Let X_t be a self-similar process with Hurst exponent H and volatility σ. The noisy observation $Y_t = X_t + \epsilon_t$ has noise level σ_ϵ. Suppose we know the variance $\mathcal{V}[Y](\tau)$ for $\tau \in \mathcal{T} \subset \mathbb{R}^+$ on a finite set of scales. Then the Hurst exponent estimation $\hat{H}$ by fitting $\log(\mathcal{V}[Y]) \sim \log(\tau)$ will be*

$$\left(1 - \frac{2\eta}{\tau_{min}^{2H} + 2\eta}\right) \cdot H \leq \hat{H} \leq \left(1 - \frac{2\eta}{\tau_{max}^{2H} + 2\eta}\right) \cdot H,$$

where $\eta := \sigma_\epsilon^2/\sigma^2$ is the noise ratio, and $\tau_{min} := \inf \mathcal{T}$, $\tau_{max} := \sup \mathcal{T}$.

Proof. The slope of the OSL regression is bounded by the range of the derivative

$$\frac{d\log(\mathcal{V}[Y](\tau))}{d\log(\tau)} = 2\Lambda[Y](\tau).$$

Use the growth rate function in Property 6.1, and note that $\Lambda[Y](\tau)$ is a monotonically increasing function, we can get

$$\begin{aligned} 2H \cdot \frac{\tau_{min}^{2H}}{\tau_{min}^{2H} + 2\eta} \leq &\alpha \leq 2H \cdot \frac{\tau_{max}^{2H}}{\tau_{max}^{2H} + 2\eta}, \\ \left(1 - \frac{2\eta}{\tau_{min}^{2H} + 2\eta}\right) \cdot H \leq &\hat{H} \leq \left(1 - \frac{2\eta}{\tau_{max}^{2H} + 2\eta}\right) \cdot H. \end{aligned}$$

□

Remark. The proposition shows that the estimated Hurst exponent will be strictly lower than the true value, if there exist microstructure noise. The deviation amount, as calculated of the form $\frac{2\eta}{\tau^{2H}+2\eta}H$, is determined by the noise level only through the noise ratio η. Choosing $\mathcal{T}$ at large timescales can reduce, but not completely removing the bias. Lastly, the Hurst exponent itself can also affect the amount of deviation, with smaller H suffering greater bias.

Growth-Adjusted Variance Analysis

In order to remove the microstructure noise bias, we propose the Growth-Adjusted (GrAd) Variance Analysis algorithm to estimate the Hurst exponent. Instead of directly fitting the variance, the growth-adjusted variance function defined in Proposition 2 is used as it has good conservation property. Notice that under the noisy fractional Brownian motion model, the growth-adjusted variance recovers the variance function of the latent process, up to a constant of H (use Proposition 2 and Eq. (4.8)). The formal algorithm is as follows:

Growth-Adjusted Variance Analysis

- For $m = 1, 2, \cdots M$, compute V_m using Eq. (4.14).
- Compute growth rate Λ_m using Eq. (4.16).

- Compute the growth-adjusted variance:

$$\tilde{V}_m = \Lambda_m \cdot \frac{V_m + V_{m+1}}{2}. \tag{4.17}$$

- Fit least square linear regression of $\log \tilde{V}_m \sim \log(m + 1/2)$. Denote the slope as α.

- Estimator $\hat{H} = \alpha/2$.

Because we approximate differentiation by finite difference when calculating the growth rate, all other values are adapted to the average between scales m and $m + 1$ for better accuracy.

4.4.3 Estimating Noise Level

For the noisy fractional Brownian motion model, we can further estimate the noise level σ_ϵ.

Brownian motion assumption Even though the base assumption is the noisy fractional Brownian motion model, it is very common for people to hold the assumption that the latent process is a Brownian motion. In this case the noise level still need evaluation. Using Property 6.1 and fix $H = 0.5$, we propose the following estimator for the noise level:

$$\min_{\beta_0 \geq 0, \beta_1 \in \mathbb{R}} \quad \sum_{m \in \mathcal{M}} (\beta_1 m + \beta_0 - V_m)^2. \tag{4.18}$$

Then the noise level is estimated by $\hat{\sigma}_\epsilon = \sqrt{\beta_0/2}$, and the volatility is estimated by $\hat{\sigma} = \sqrt{\beta_1}$. The noise ratio can be computed as $\hat{\eta} = \beta_0/(2\beta_1)$. $\mathcal{M} \subset \mathbb{N}^+$ is a pre-specified scale set, and V_m's are from Eq. (4.14). This is actually an OLS with the constraint on the intercept to be non-negative, because the variance of the noise $\sigma_\epsilon^2 \geq 0$.

Fractional Brownian motion assumption In the general case the latent process is assumed to be a fractional Brownian motion with $H \in (0, 1)$. In this case we first need to use the Grow-adjusted Variance analysis to estimate $\hat{H}$, and then solve

$$\min_{\beta_0 \geq 0, \beta_1 \in \mathbb{R}} \quad \sum_{m \in \mathcal{M}} (\beta_1 m^{2\hat{H}} + \beta_0 - V_m)^2. \tag{4.19}$$

Then the noise level is estimated by $\hat{\sigma}_\epsilon = \sqrt{\beta_0/2}$.

Remark. In practice the scale set $\mathcal{M}$ can be chosen to consist of only very small m's, because V_m is most sensitive to the noise at small scale. On the other hand, the error of the variance estimator becomes larger as m increase. Usually even $\mathcal{M} = \{1, 2\}$ can be a good choice.

4.4.4 Monte Carlo Simulation

To evaluate the performance of the proposed algorithm, we test on simulated noisy fractional Brownian motion sample paths. Three scenarios with Hurst exponent $0.5, 0.4, 0.6$ are examined with noise level $\sigma_\epsilon = 2 \times 10^{-5}$ and volatility parameter $\sigma = 1 \times 10^{-2}$. To mimic the 3-second intraday time series to be used in the next section, sample paths were generated on $t \in [0, 1]$ with 7800 equal-spaced points.

In Table 4.1 below we show the simulation results of Hurst exponent estimation, using the proposed Growth-Adjusted (GrAd) Variance analysis, in comparison to the standard Variance analysis and the original Rescaled Range analysis. Under three different scenarios with Hurst exponent corresponding to random walk, mean-reverting and trending behaviors, we see that the proposed algorithm provides consistently unbiased results, and has the lowest mean absolute deviation from the ground truth values. Algorithms not handling microstructure noise can not produce unbiased estimates for all of the scenarios. The histograms of the simulation result are in Fig. 4.2 below. The red vertical line in each histograms indicates the ground truth value. We can see the distribution of the GrAd variance analysis estimator is centered around the ground truth values, while the other two algorithms yield biased results.

In Table 4.2 below we show the noise level estimation results on simulated noisy fractional Brownian motion data, with parameters same as above. From the result we can see that the noise estimation is most accurate when H is large and the fractional Brownian motion path is smoother. The accuracy is low when H is small. The reason might be the fractional Brownian motion path at $H < 0.5$ is rough and non-continuous, making the separation of noise extremely difficult.

H=0.5 (random walk)			
Algorithm	Mean	+- 1 mean std	Mean abs. dev.
GrAd Variance	0.5003	(0.4998, 0.5009)	0.01474
Variance analysis	0.4846	(0.4843, 0.4848)	0.01558
Rescaled range	0.5302	(0.5294, 0.5311)	0.03410
H=0.4 (mean-reverting)			
Algorithm	Mean	+- 1 mean std	Mean abs. dev.
GrAd Variance	0.4057	(0.4050, 0.4064)	0.01799
Variance analysis	0.4192	(0.4189, 0.4194)	0.01923
Rescaled range	0.4518	(0.4510, 0.4526)	0.05202
H=0.6 (trending)			
Algorithm	Mean	+- 1 mean std	Mean abs. dev.
GrAd Variance	0.6035	(0.6030, 0.6041)	0.01373
Variance analysis	0.5070	(0.5068, 0.5073)	0.09296
Rescaled range	0.6077	(0.6067, 0.6087)	0.02673

Table 4.1: Hurst exponent estimation on synthesized noisy fractional Brownian motion ($\sigma = 1 \times 10^{-2}, \sigma_\epsilon = 2 \times 10^{-5}$). The proposed Growth-Adjusted (GrAd) Variance algorithm is compared against the two standard algorithms, Variance analysis and Rescaled Range analysis under three different scenarios: $H = 0.5$ (random walk), $H = 0.4$ (mean-reverting), and $H = 0.6$ (trending). For each scenario, 1000 sample paths are generated to evaluate the mean, confidence interval corresponding to plus/minus one standard deviation of mean estimator, and mean absolute deviation from the ground truth value.

(10^{-5})	$H = 0.5$	$H = 0.4$	$H = 0.6$
Mean	1.8109	0.7962	2.1046
St. Dev.	0.9877	1.6801	0.1194

Table 4.2: Noise estimates on simulated noisy fractional Brownian motion data. The ground truth value is 2×10^{-5}. Each scenario is estimated with 1000 sample paths.

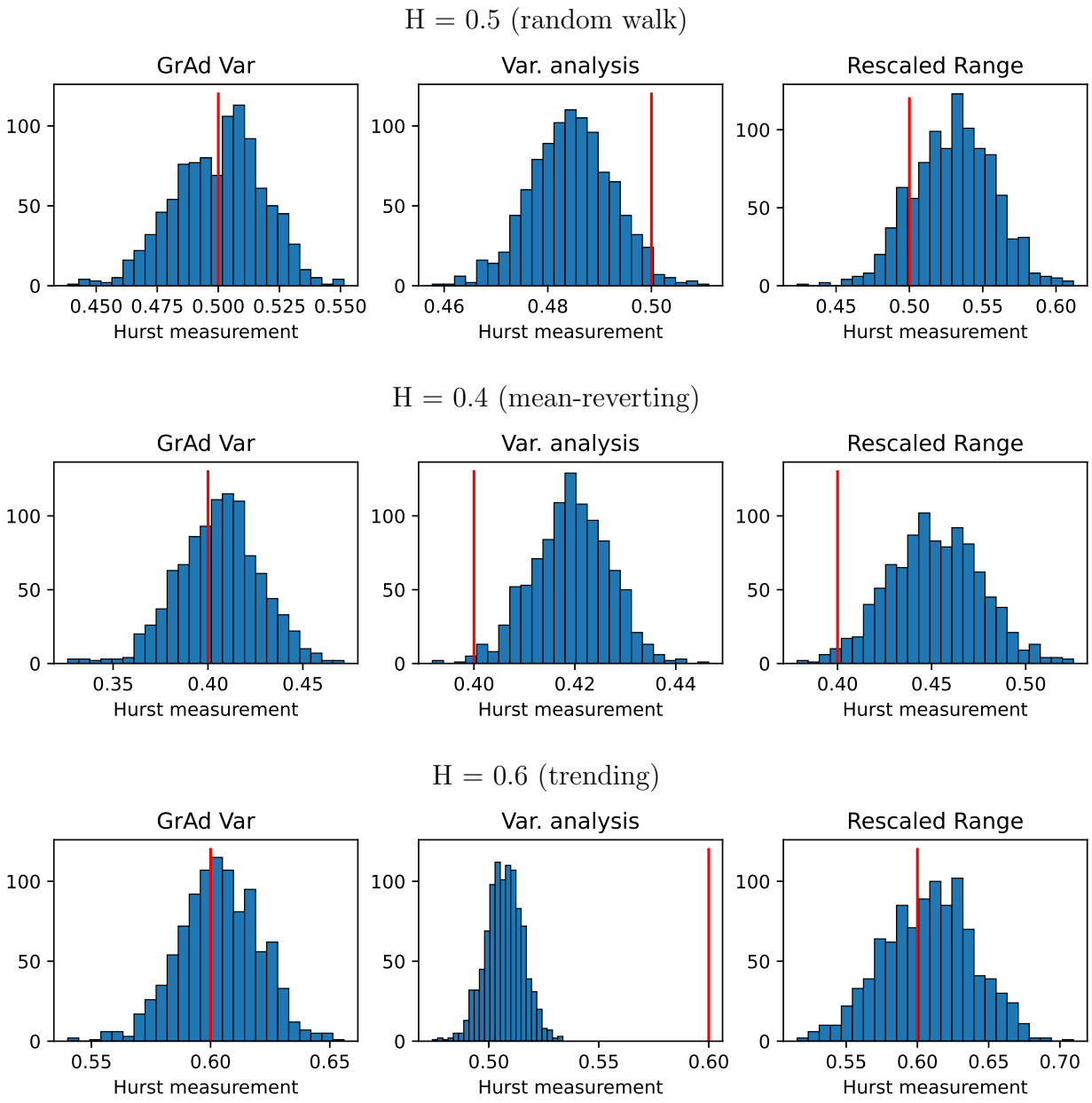

Figure 4.2: Histograms of Hurst exponent measurement using different algorithms. Three different scenarios are tested with ground truth value $H = 0.5$ (random walk), $H = 0.4$ (mean-reverting), and $H = 0.6$ (trending). The red vertical line in each histograms indicates the ground truth value.

4.5 Experiments on Intraday Data

In the previous sections we proposed high-frequency models and derived what their multiscale functions should look like. Using the algorithms introduced in 4.4, we show experimental results on real-world data. We use 3-second high frequency intraday price data for a collection of exchange traded funds (ETFs) and stocks (tickers SPY, IWM, QQQ, XLK, AAPL, and MSFT). The dates range from Jan. 2020 to May 2022. Prices are recorded on each day from open to close spanning 6.5 hours, making 7800 equal-spaced time points. We choose a time frame such that $t = 1$ corresponds to 1 minute (i.e. $\delta t = 0.05$). The interpretation is that any volatility estimates will reflect equivalently the standard deviation of 1 minute price change.

4.5.1 Ensemble Multiscale Curves

Let's start with showing the multiscale behaviors of real-world high-frequency price data. Fig. 4.3 shows the plots of the multiscale functions estimated on 3-second intraday time series, averaged over all available dates in the dataset. Starting with the top two plots showing the average variance curves, we see that for all tickers, the variance scales in a approximately linear form, with very slight concavity. This is close to many base models in finance such as Brownian motion or other types of random walk with uncorrelated increments, as shown in Proposition 1. The log-log plot shows that the variance scales with a approximately constant growth rate, which looks like the behavior of self-similar processes (see Eq. (4.8)). The slopes of the lines on the log-log plot can be used to estimate the Hurst exponents.

However, as we move on to the two bottom plots showing finer details of the curves, there are observations unexplained by a standard model. Starting with the bottom left plot showing the volatility curves, we see that the volatility is actually not constant, violating the form given by Proposition 1. Although the slope down is very small, the trend is consistent across all six tickers. Moreover, for many of the tickers, especially the two stocks AAPL and MSFT,

we see a clear sharp drop of volatility when the timescale is very small. This phenomenon matches the asymptotic behavior of the noisy fractional Brownian motion model, showing clear evidence of the microstructure noise. Finer sampling frequency may show larger effect of the noise.

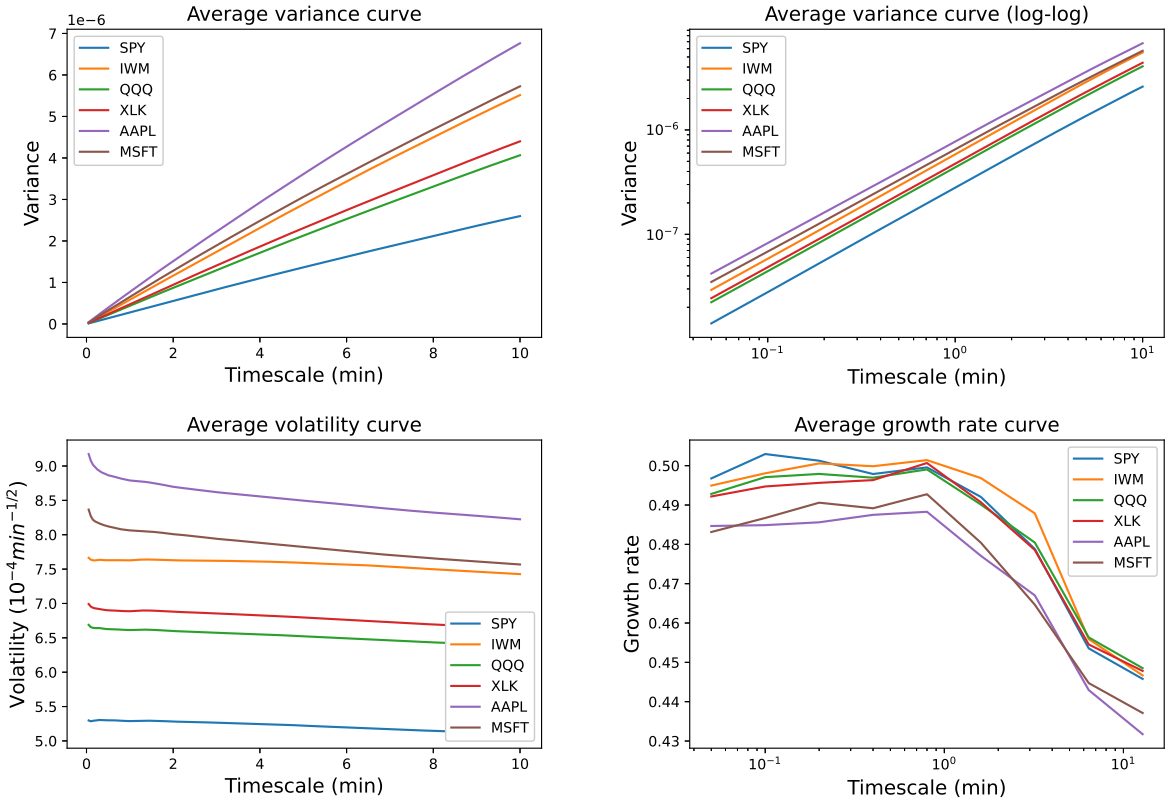

Figure 4.3: Average variance, volatility, and growth rate curves estimated on the 3-second intraday dataset. For the intraday time series on each day, one variance curve is estimated for each asset using Eq. (4.14). The top two plots shows the average variance curves over all dates in the dataset. The bottom two plots are volatility curves and growth rate curves derived from the *average variance curves*, using Eq. (4.15) and (4.16), respectively.

Going to the lower right plot, we show the growth rate curves calculated from average variance functions. Even though the upper right plot indicates that the prices follow self-similar processes with seemingly constant growth rates, the bottom right plot shows that they are

not. For all of the tickers, if we look at the timescale within the 1-minute range, the growth rate stays on an almost constant level, with very slight increase. This is explainable by the noisy fractional Brownian motion model as derived in Property 6.1. However, the growth rate declines obviously when going beyond the 1-minute threshold, which means that self-similarity no longer holds. This is also not explained by the microstructure noise, which only contributes to an increasing growth-rate. This phenomenon suggests that there might be some other dynamics in the mid-frequency region.

We then proceed to estimate the parameters in the noisy fraction Brownian motion setting, using the average variance curves in Fig. 4.3. Because we have seen in the growth rate plot that the model assumption may not be valid beyond the 1 minute limit, we only take the V_m for $m = 1, \cdots, 20$ (from 3 seconds to 1 minute). The GrAd variance analysis algorithm is used to fit Hurst exponent H, with results shown in Table 4.3. Since the Hurst exponents are very close to 1/2, we take the Brownian motion assumption for noise fitting (Eq. (4.18)). Noise level σ_ϵ, volatility σ, and noise ratio $\eta : \sigma_\epsilon^2/\sigma^2$ estimations are also shown in Table 4.3.

	SPY	IWM	QQQ	XLK	AAPL	MSFT
H	0.4974	0.5002	0.4975	0.4974	0.4875	0.4913
σ_ϵ (10^{-5})	1.2696	0.9985	2.2494	2.6386	6.2366	5.1041
σ $(10^{-4}\ min^{-\frac{1}{2}})$	1.1828	1.7055	1.4778	1.5381	1.9591	1.7980
η $(10^{-2}\ min)$	1.1521	0.3427	2.3170	2.9430	10.1341	8.0589

Table 4.3: Hurst exponent H, noise level σ_ϵ, volatility σ, and noise ratio $\eta := \sigma_\epsilon^2/\sigma^2$ estimations on the average variance curves in Fig. 4.3. Eq. (4.14) estimates variance curves for each intraday time series. The average over all dates is taken as the V_m in the GrAd variance analysis algorithm to fit the Hurst exponent H. Noise level σ_ϵ, volatility σ, and noise ratio η are estimated using the Brownian motion assumption estimator in Eq. (4.18), also using the averages V_m.

4.5.2 Daily Hurst Exponent

Although the ensemble multiscale behavior suggests that the price process is very close to random walk, the dynamics of each intraday path is unknown. It is possible that some days are more mean-reverting, while others more trending. In this section we use the proposed Growth-adjusted Variance analysis to estimate Hurst exponents for each intraday price paths. The time series on each day for each ticker yield one Hurst exponent estimate.

Fig. 4.4 shows the histograms of daily Hurst exponent results, compared against the distribution of estimating Hurst exponent from Brownian motion simulation. Although the Hurst exponent of the intraday data is centered around 1/2, we can clearly see that its distribution is not the same as Brownian motion. H estimated on the real-world data is in general distributed to the left of a Brownian motion. The kernel density curves show much wider spread distribution, and also indicates long tail behavior. There are significant portion of days that have very low or high Hurst exponent, which is unlikely if the Brownian motion assumption holds. The message is that the price dynamics can be different on different days. There are days when the price movement is mean-reverting, and on other days the movements are independent or trending.

Even though there are Hurst exponents estimated to be lower or higher than 0.5, we haven't provided their statistical significance yet. Indeed, the debates over whether a small or large Hurst exponent estimated from price data is able to reject the Brownian motion null hypothesis has been going on for a long time. Even though the statistical analysis of H estimates is complicated, there have been researches on its estimation error, including the error bounds of H estimator on standard Brownian motions [29]. For our new algorithm, we are going to use a more empirical way to determine whether the result is statistically significant to reject the null hypothesis $H = 1/2$.

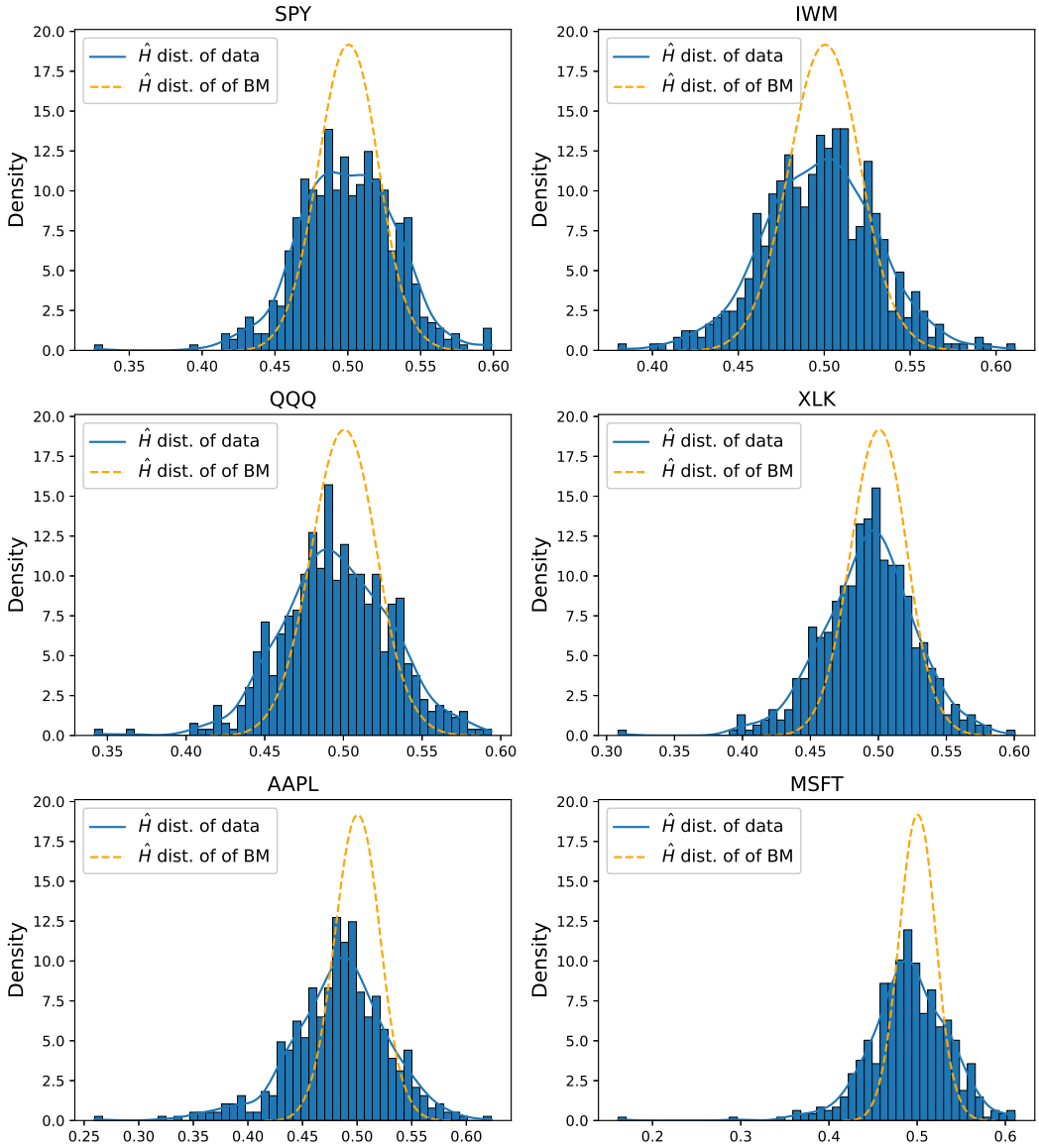

Figure 4.4: Hurst exponent estimated on 3-second intraday data from 2020 to 2022. The intraday time series on each date for each ticker yield one Hurst exponent estimation. This figure shows the histogram of the estimated values over all dates. The orange dash-line shows the distribution of Hurst exponent estimator $\hat{H}$ from a standard Brownian motion in comparison to that from the data set.

For any algorithm estimating $\hat{H}$, evaluate on simulations of standard Brownian motions. Estimate μ_0 as the mean of the null hypothesis, and σ_0 as the standard deviation of the null hypothesis. For any new $\hat{H}$ estimated on real-world data, compute

$$t = \frac{\hat{H} - \mu_0}{\sigma_0}$$

as the t-statistics. One can check if the t-statistics is below or above a certain significance threshold to decide if the Brownian motion null hypothesis should be rejected.

In Table 4.4 we show the distributional statistics of daily Hurst exponent values, with the last three columns showing the percentage of days when the estimated value shows mean-reverting, trending, or random walk values, at 5% significance level. We can see that the portion is much larger that could be from standard Brownian motion. Similar to Table 4.3, the overall behavior the estimated Hurst exponents are slightly below 0.5. Also note that the standard deviation is much larger than that from a (fractional) Brownian motion, which we have seen from Table 4.1 in the previous section.

	Mean	St.Dev	mean-reverting (%)	trending (%)	random walk (%)
SPY	0.4994	0.0329	11.8868	12.8302	75.2830
IWM	0.4976	0.0326	13.9623	10.0000	76.0377
QQQ	0.4955	0.0344	16.6038	10.9434	72.4528
XLK	0.4927	0.0340	18.1132	8.4906	73.3962
AAPL	0.4833	0.0446	29.2453	10.3774	60.3774
MSFT	0.4899	0.0452	23.5849	13.9623	62.4528

Table 4.4: Hurst exponent estimated on 3-second intraday data from 2020 to 2022. The intraday time series on each date for each ticker yield one Hurst exponent estimation. This table shows statistics of the estimates over all dates. The mean and the standard deviation of H are in the first two columns, and the last three columns show the percentage of dates with Hurst exponent estimated to be mean-reverting, trending, or random walk values, with threshold at 5% significance level.

4.5.3 Intraday Evolution

One latent assumption throughout the previous part of this chapter is stationary of time series. However, usually the price dynamics can be different at different time of day. There have been numerous evidence showing patterns and seasonality of return distribution during the trading hours [6]. One of the well-known observations is the U-shape volatility [9, 1], stating that the volatility is highest near market open and close, and lowest at midday.

To test if the multiscale parameters are constant, we show intraday rolling Hurst exponent, noise level, and noise ratio in Fig. 4.5. The result shows varying Hurst exponent at different time of day, i.e. the scaling exponent is not constant. For all of the tickers, the exponent is highest at market open, and decrease right after. The market index ETFs SPY and IWM have smallest values in the middle of the day, showing mean-reverting behavior. Their Hurst exponents at market open and in the last hours are close to 0.5, indicating week dependency. The "U" shape of their Hurst exponents can be compared to the U-shape volatility to answer the question: how is the volatility pattern formed? In the middle of the day, the price movement is more mean-reverting, making the variability "locked" in small timescale. Even though the volatility is lower at midday, the smaller Hurst exponents indicate that the return can be even more "noisy".

Unlike the index ETFs, the two stocks AAPL and MSFT, have Hurst exponents always below 0.5, and continue decreasing in the afternoon. Near the last hour before market close, their Hurst exponents notably increase towards 0.5, which might be due to increasing trading activities. The decreasing Hurst exponent can be compared with the increasing kurtosis result from [6], suggesting that the more long-tail movement towards the end of the day actually may happen at small timescales. For QQQ and XLK, which have larger correlation with the tech stocks, their behavior is in between SPY, IWM, and AAPL, MSFT. Different Hurst exponent patterns clearly show different trading dynamics for ETFs and stocks.

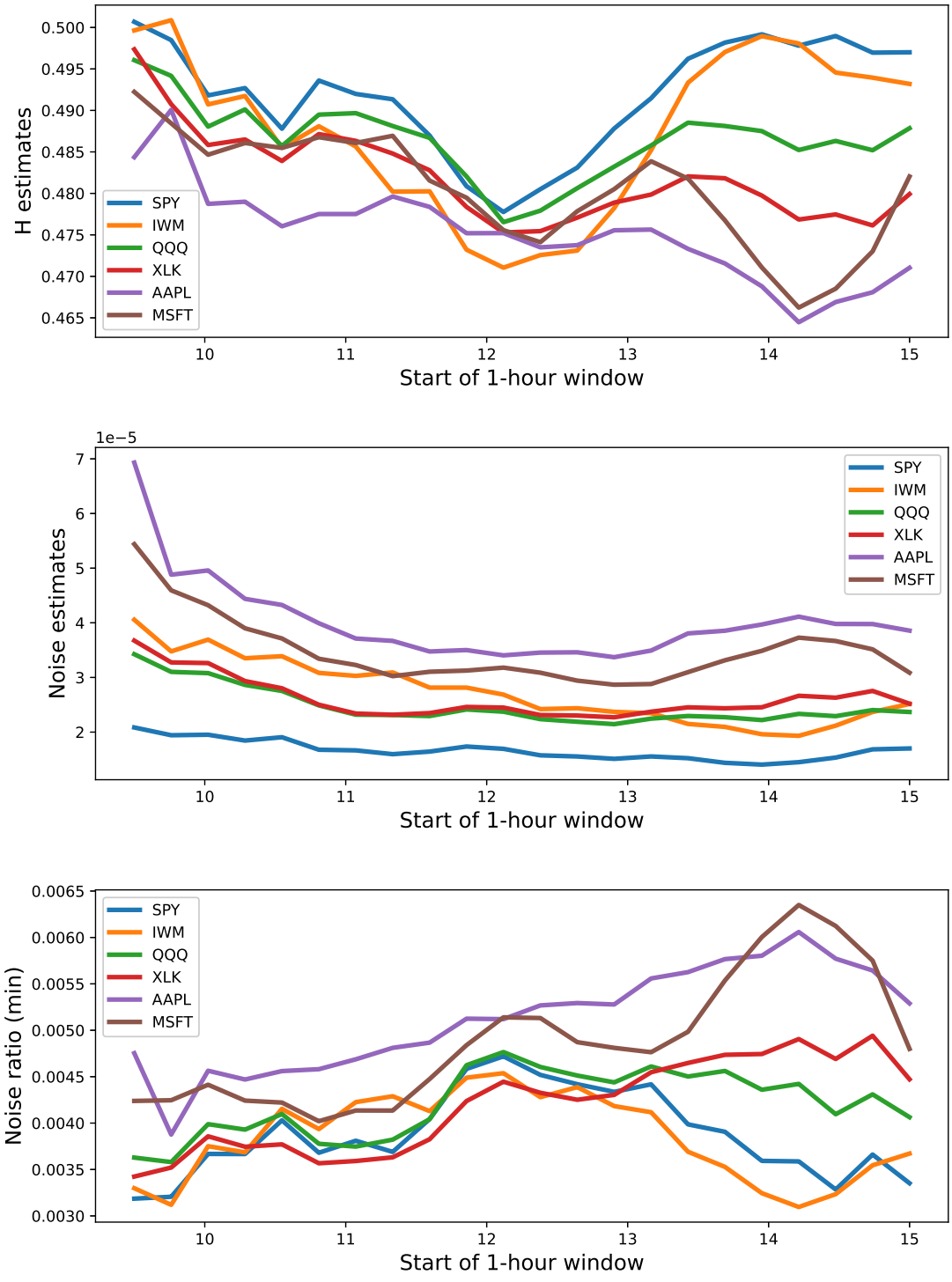

Figure 4.5: Hurst exponent H, noise level σ_ϵ, and noise ratio $\eta := \sigma_\epsilon^2/\sigma^2$ on a 1-hour rolling window. Starting point of the window shifts on a 15-min basis. For each day one rolling Hurst curve is estimated for each asset. The plot shows the average over all dates in the dataset.

The noise level in the noisy fractional Brownian motion model provides another lens to inspect the intraday volatility pattern. The middle plot in Fig. 4.5 shows the microstructure noise estimation on a 1-hour rolling window. We see that all of the assets have highest noise value at market open, and decrease or flat out into the day. The two stocks AAPL and MSFT have sharpest trend. The bottom plot in Fig. 4.5 shows the rolling noise ratio defined as $\eta := \sigma_\epsilon^2/\sigma^2$. From the plot, we see that the two market index ETFs SPY and IWM have noise ratios increase in the morning then decrease in the afternoon. In comparison, the two stocks AAPL and MSFT have increasing noise ratios throughout the day. Different noise ratio patterns may indicate different trading activities.

4.5.4 Evolution of Parameters

From previous discussion we saw the Hurst exponent can be different on different days. We close this chapter with time series of the parameters evolving over the dates. In Fig. 4.6 we show the 60-day rolling average of Hurst exponent from 2020 to 2022, in comparison with the realized volatility and noise level change. We can see that at the beginning of the pandemic, most assets have Hurst exponent close to the 0.5 level, corresponding to random walk. The trend moved towards mean-reverting values, and experienced a few up and downs. In the most recent time the ETFs moved up and reached above 0.5, becoming more trending, while the two individual tech stocks AAPL and MSFT remained fluctuating below the 0.5 level.

In terms of volatility level, we see that the noise pattern move together with the realized volatility time series. The difference between the noise among different assets is smaller than that of the volatility. We can also notice that the two stocks, AAPL and MSFT, have the microstructure noise higher than that in the ETFs, and deviated from the ETFs for a long period from early 2020 to early 2021. Even though all of the tickers have different volatility levels, the noise level difference within the stock group and the ETF group is very small, suggesting two distinct types of trading activities.

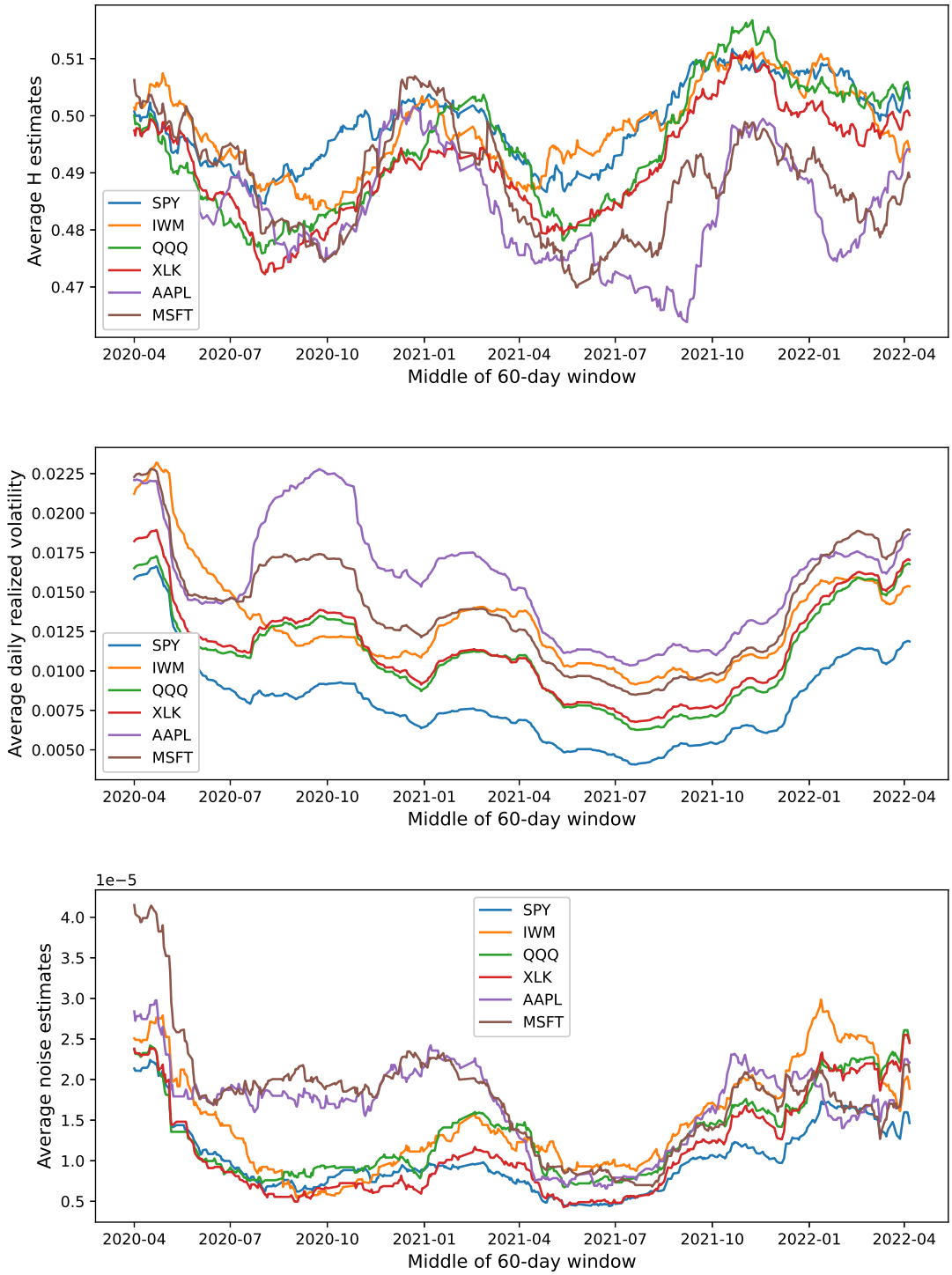

Figure 4.6: Time series evolution of Hurst exponent, realized volatility, and noise level from 2020 to 2022. Values are estimated on 3-second time series for each date and each ticker, and averaged over 60-day rolling windows.

Chapter 5

INTRADAY MULTISCALE CORRELATION

5.1 Overview

Continuing the discussion on intraday multiscale risk, let's move on to the multivariate case. With the scaling properties of univariate price processes already established in the previous chapter, correlation between returns of multiple assets is the main focus. With definitions for multiscale correlation set up, a family of multivariate high-frequency models are proposed. We show how their multiscale correlation properties are different from standard models. Lastly we show experimental results on intraday high-frequency price data, comparing the real-world correlation behavior with our model outcomes.

5.2 Multiscale Correlation

Denote the vector of prices for a collection of p financial assets at time t as $\vec{P}_t \in \mathbb{R}^p$. As in the univariate case, it is more convenient to work on the log prices $\vec{X}_t := \log(\vec{P}_t)$. The log return vector over the time interval from t to $t+\tau$ is defined as

$$\vec{r}^X_{t,\tau} = \vec{X}_{t+\tau} - \vec{X}_t. \tag{5.1}$$

All operations are applied element-wise. As a generalization from the multiscale variance in Definition 2, the multiscale covariance matrix can be defined as:

$$\mathcal{V}[\vec{X}][\tau] := \mathbb{E}[(\vec{r}^X_{t,\tau} - \mathbb{E}[\vec{r}^X_{t,\tau}])(\vec{r}^X_{t,\tau} - \mathbb{E}[\vec{r}^X_{t,\tau}])^T]. \tag{5.2}$$

Suppose $\vec{X}_t \in \mathbb{R}^p$ is a stationary multivariate process, $\mathcal{V}[\vec{X}][\tau] \in \mathbb{R}^{p\times p}$ is a matrix depending on τ only. It is more convenient to study the pair-wise covariance and correlation represented by each entry of the matrix. Let's start with the following definitions.

5.2.1 Definition

Definition 7. Let the pair of log prices $X_t \in \mathbb{R}, Y_t \in \mathbb{R}$ be stationary processes on $0 \leq t \leq T$. For $\tau > 0$, define their multiscale covariance as

$$\mathcal{C}[X,Y](\tau) := \mathbb{E}[r^X_{t,\tau} r^Y_{t,\tau}] - \mathbb{E}[r^X_{t,\tau}] \cdot \mathbb{E}[r^Y_{t,\tau}], \tag{5.3}$$

and their multiscale correlation as:

$$\mathcal{R}[X,Y](\tau) := \frac{\mathcal{C}[X,Y](\tau)}{\sqrt{\mathcal{V}[X](\tau) \cdot \mathcal{V}[Y](\tau)}}. \tag{5.4}$$

Like volatility, the multiscale correlation behavior is limited for a wide class of processes.

Proposition 4 (Uncorrelated increments)**.** *For any pair of stationary random processes X_t, Y_t that has jointly uncorrelated increments, i.e.*

$$\begin{aligned}
\mathbb{E}[(X_{t_2} - X_{t_1})(X_{t_3} - X_{t_2})] &= \mathbb{E}[X_{t_2} - X_{t_1}] \cdot \mathbb{E}[X_{t_3} - X_{t_2}], \\
\mathbb{E}[(Y_{t_2} - Y_{t_1})(Y_{t_3} - Y_{t_2})] &= \mathbb{E}[Y_{t_2} - Y_{t_1}] \cdot \mathbb{E}[Y_{t_3} - Y_{t_2}], \\
\mathbb{E}[(X_{t_2} - X_{t_1})(Y_{t_3} - Y_{t_2})] &= \mathbb{E}[X_{t_2} - X_{t_1}] \cdot \mathbb{E}[Y_{t_3} - Y_{t_2}], \\
\mathbb{E}[(Y_{t_2} - Y_{t_1})(X_{t_3} - X_{t_2})] &= \mathbb{E}[Y_{t_2} - Y_{t_1}] \cdot \mathbb{E}[X_{t_3} - X_{t_2}],
\end{aligned}$$

for any $0 \leq t_1 < t_2 < t_3$. Then $\mathcal{R}[X,Y](\tau)$ must be a constant in τ.

Proof. We first show that the covariance $\mathcal{C}[X,Y](\tau)$ scales linearly with τ:

$$\mathcal{C}[X,Y](\tau) \propto \tau.$$

For any $\tau_1, \tau_2 > 0$, let's show additive-ness $\mathcal{C}[X,Y](\tau_1 + \tau_2) = \mathcal{C}[X,Y](\tau_1) + \mathcal{C}[X,Y](\tau_2)$. Consider $0 < \tau_1 < \tau_1 + \tau_2$, then

$$\begin{aligned}
\mathcal{C}[X,Y](\tau_1+\tau_2) =& \mathbb{E}[(X_{\tau_1+\tau_2}-X_0)(Y_{\tau_1+\tau_2}-Y_0)] - \mathbb{E}[X_{\tau_1+\tau_2}-X_0]\cdot\mathbb{E}[Y_{\tau_1+\tau_2}-Y_0] \\
=& \mathbb{E}[((X_{\tau_1+\tau_2}-X_{\tau_1})+(X_{\tau_1}-X_0))\cdot((Y_{\tau_1+\tau_2}-Y_{\tau_1})+(Y_{\tau_1}-Y_0))] \\
& - \mathbb{E}[(X_{\tau_1+\tau_2}-X_{\tau_1})+(X_{\tau_1}-X_0)]\cdot\mathbb{E}[(Y_{\tau_1+\tau_2}-Y_{\tau_1})+(Y_{\tau_1}-Y_0)] \\
=& \mathbb{E}[(X_{\tau_1+\tau_2}-X_{\tau_1})(Y_{\tau_1+\tau_2}-Y_{\tau_1})] + \mathbb{E}[(X_{\tau_1}-X_0)(Y_{\tau_1}-Y_0))] \\
& - \mathbb{E}[X_{\tau_1+\tau_2}-X_{\tau_1}]\cdot\mathbb{E}[Y_{\tau_1+\tau_2}-Y_{\tau_1}] - \mathbb{E}[X_{\tau_1}-X_0]\cdot\mathbb{E}[Y_{\tau_1}-Y_0] + 0 \\
=& \mathcal{C}[X,Y](\tau_2)+\mathcal{C}[X,Y](\tau_1).
\end{aligned}$$

Then $\mathcal{C}[X,Y](\tau) \propto \tau$.[1] Using Property 1, the multiscale variance of the two processes also satisfy $\mathcal{V}[X] \propto \tau$ and $\mathcal{V}[Y] \propto \tau$. Therefore, the correlation must be constant at any scale:

$$\mathcal{R}[X,Y](\tau) \propto \frac{\tau}{\sqrt{\tau\cdot\tau}} = const.$$

□

Example (Stochastic volatility)**.** Let B_t and W_t be independent standard Brownian motions. For $\rho \in [-1,1]$, $\rho_\sigma \in [-1,1]$, define the pair of correlated stochastic volatility processes:

$$\begin{aligned}
dX_t^{(1)} &= \mu_1 dt + \sqrt{\nu_t^{(1)}}dB_t^{(1)}, \\
dX_t^{(2)} &= \mu_2 dt + \sqrt{\nu_t^{(2)}}\left(\rho dB_t^{(1)} + \sqrt{1-\rho^2}dB_t^{(2)}\right),
\end{aligned}$$

where the volatility processes are

$$\begin{aligned}
\nu_t^{(1)} &= \alpha_t^{(1)}dt + \beta_t^{(1)}dW_t^{(1)}, \\
\nu_t^{(2)} &= \alpha_t^{(2)}dt + \beta_t^{(2)}\left(\rho_\sigma dW_t^{(1)} + \sqrt{1-\rho_\sigma^2}dW_t^{(2)}\right).
\end{aligned}$$

$B_t^{(1)}, B_t^{(2)}, W_t^{(1)}, W_t^{(2)}$ are independent Brownian motions. Even though $(X_t^{(1)}, X_t^{(2)})$ does not have independent increments (volatility clustering), their increments are jointly uncorrelated because

$$\mathbb{E}\left[\left(\int_{t_1}^{t_2}\sqrt{\nu_t^{(k)}}dB_t^{(i)}\right)\left(\int_{t_2}^{t_3}\sqrt{\nu_t^{(l)}}dB_t^{(j)}\right)\right] = 0,$$

[1] Again, we can only conclude from additive-ness to linearity for τ on a rational grid. For continuous $\mathcal{C}[X,Y](\tau)$ function we can justify the claim. Detailed discussion will not be the main focus of this thesis.

for any $i, j, k, l \in \{1, 2\}$. Therefore, the correlation between $X_t^{(1)}$ and $X_t^{(2)}$ is scale-independent.

Proposition 5. *Let $X_t^{(1)}, \cdots, X_t^{(n)}$ be a sequence of independent processes, and $Y_t^{(1)}, \cdots, Y_t^{(n)}$ be another sequence of independent processes. Further assume that $X^{(i)} \perp Y^{(j)}, \ \forall i \neq j$. Denote the sequence summations as $S_X = \sum_{i=1}^n X^{(i)}, S_Y = \sum_{i=1}^n Y^{(i)}$ Then the multiscale correlation between the summation of the two sequences is:*

$$\mathcal{R}[S_X, S_Y](\tau) = \sum_{i=1}^{n} \mathcal{R}[X^{(i)}, Y^{(i)}](\tau) \cdot \frac{\sqrt{\mathcal{V}[X^{(i)}](\tau) \cdot \mathcal{V}[Y^{(i)}](\tau)}}{\sqrt{\left(\sum_{j=1}^n \mathcal{V}[X^{(j)}](\tau)\right) \cdot \left(\sum_{j=1}^n \mathcal{V}[Y^{(j)}](\tau)\right)}}. \tag{5.5}$$

Proof. Direct computation of the covariance gives:

$$\begin{aligned}
\mathcal{C}[S_X, S_Y](\tau) &= \mathbb{E}\left[\left(\sum_{i=1}^n r_{t,\tau}^{X^{(i)}}\right)\left(\sum_{i=1}^n r_{t,\tau}^{Y^{(i)}}\right)\right] - \mathbb{E}\left[\sum_{i=1}^n r_{t,\tau}^{X^{(i)}}\right]\mathbb{E}\left[\sum_{i=1}^n r_{t,\tau}^{Y^{(i)}}\right] \\
&= \sum_{i=1}^n \mathbb{E}[r_{t,\tau}^{X^{(i)}} r_{t,\tau}^{Y^{(i)}}] + \sum_{i\neq j}^n \mathbb{E}[r_{t,\tau}^{X^{(i)}} r_{t,\tau}^{Y^{(j)}}] - \sum_{i=1}^n \mathbb{E}[r_{t,\tau}^{X^{(i)}}]\mathbb{E}[r_{t,\tau}^{Y^{(i)}}] - \sum_{i\neq j}^n \mathbb{E}[r_{t,\tau}^{X^{(i)}}]\mathbb{E}[r_{t,\tau}^{Y^{(j)}}] \\
&= \sum_{i=1}^n \mathcal{C}[X^{(i)}, Y^{(i)}](\tau) + 0 \\
&= \sum_{i=1}^n \mathcal{R}[X^{(i)}, Y^{(i)}](\tau) \cdot \sqrt{\mathcal{V}[X^{(i)}](\tau) \cdot \mathcal{V}[Y^{(i)}](\tau)}.
\end{aligned}$$

The denominator can be easily computed using Property 2.1. □

Remark. Note that in Eq. (5.5), even if $\mathcal{R}[X^{(i)}, Y^{(i)}](\tau)$ is constant for all $i = 1, \cdots, n$, the correlation of the two summations can still be scale-dependent if the variance scaling behavior is different among the pairs $(X^{(i)}, Y^{(i)}), \ i = 1, \cdots, n$.

5.2.2 Numerical Estimation

Here we briefly discuss how to estimate the multiscale correlation function from discrete observations. Following the same principles as in Chapter 4.4, let's consider a pair of time series, (X_i, Y_i), for $i = 1, \cdots, n$. The multiscale correlation can be estimated as follows.

For $m = 1, 2, \cdots, M$:

- Compute variance $V_{X,m}, V_{Y,m}$ using Eq. (4.14).

- Compute covariance

$$C_{XY,m} = \frac{1}{n-m} \sum_{i=1}^{n-m} (X_{i+m} - X_i - \mu_X m)(Y_{i+m} - Y_i - \mu_Y m).$$

- Compute correlation

$$R_{XY,m} = \frac{C_{XY,m}}{\sqrt{V_{X,m} \cdot V_{Y,m}}}. \tag{5.6}$$

where $\mu_X = \frac{X_n - X_1}{n-1}$, and $\mu_Y = \frac{Y_n - Y_1}{n-1}$ are estimated drifts for detrending the processes.

5.3 Multivariate High-frequency Models

In this section we extend the fractional Brownian motion and the microstructure noise model to the multivariate case. Multiscale correlation behaviors of the models are shown. We then propose the multivariate noisy fractional Brownian motions model and show the theoretical result of its correlation. Numerical model evaluation and fitting framework will be discussed at the end of this section.

5.3.1 Correlated Fractional Brownian Motions

Correlation between Brownian motions has been well-established for a long time. Like any random walk with stationary and independent increments, Brownian motion also has scale-independent correlation as shown in Proposition 4. Correlation between fractional Brownian motions was studied more recently in the setting of multivariate fractional Brownian motion (mfBm) [8, 7, 26, 27]. To begin with, let's first define a pair of correlated fractional Brownian motions using Eq. (4.6) and study its correlation property.

Proposition 6. *For any Hurst exponent pair $H_1, H_2 \in (0,1)$, and for any correlation coefficient $\rho \in [-1,1]$, define the pair of correlated fractional Brownian motions*

$$B_t^{H_1} = \mathcal{B}(W^{(1)}; H_1)_t,$$
$$B_t^{H_2} = \mathcal{B}(\rho W^{(1)} + \sqrt{1-\rho^2} W^{(2)}; H_2)_t,$$

where $W^{(1)}, W^{(2)}$ are independent standard Brownian motions, and the $\mathcal{B}(\cdot)$ operator is defined in Eq. (4.6). Then their multiscale correlation function

$$\mathcal{R}[B^{H_1}, B^{H_2}](\tau) = \rho$$

Proof. Without loss of generality, let's take $t = 0$ in Eq. (5.3) and analyze the increments from 0 to τ. The covariance between B^{H_1} and B^{H_2} is

$$\begin{aligned}
\mathcal{C}[B^{H_1}, B^{H_2}](\tau) &= \mathbb{E}[B_\tau^{H_1} B_\tau^{H_2}] - \mathbb{E}[B_\tau^{H_1}] \cdot \mathbb{E}[B_\tau^{H_2}] \\
&= \mathbb{E}[\mathcal{B}(W^{(1)}; H_1)_\tau \cdot \mathcal{B}(\rho W^{(1)} + \sqrt{1-\rho^2} W^{(2)}; H_2)_\tau] - 0 \\
&= \rho \cdot \mathbb{E}[\mathcal{B}(W^{(1)}; H_1)_\tau \cdot \mathcal{B}(W^{(1)}; H_2)_\tau] + \sqrt{1-\rho^2} \cdot \mathbb{E}[\mathcal{B}(W^{(1)}; H_1)_\tau \cdot \mathcal{B}(W^{(2)}; H_2)_\tau] \\
&= \rho \cdot \mathbb{E}[\mathcal{B}(W^{(1)}; H_1)_\tau \cdot \mathcal{B}(W^{(1)}; H_2)_\tau].
\end{aligned}$$

We've used linearity of the $\mathcal{B}(\cdot)$ operator. Now we need to compute the expectation of the product of two stochastic integrals. Using (generalized) Ito isometry,

$$\begin{aligned}
&\mathbb{E}\left[\left(\int_{-\infty}^{\tau} \left((\tau - s)^{H_1 - 1/2} - (-s)_+^{H_1 - 1/2}\right) dW_s\right)\left(\int_{-\infty}^{\tau} \left((\tau - s)^{H_2 - 1/2} - (-s)_+^{H_2 - 1/2}\right) dW_s\right)\right] \\
&= \int_{-\infty}^{\tau} \left((\tau - s)^{H_1 - 1/2} - (-s)_+^{H_1 - 1/2}\right)\left((\tau - s)^{H_2 - 1/2} - (-s)_+^{H_2 - 1/2}\right) ds.
\end{aligned}$$

Use change of variable $u := s/\tau$, we can write

$$\begin{aligned}
&\int_{-\infty}^{\tau} \left((\tau - s)^{H1 - 1/2} - (-s)_+^{H_1 - 1/2}\right)\left((\tau - s)^{H_2 - 1/2} - (-s)_+^{H_2 - 1/2}\right) ds \\
&= \int_{-\infty}^{1} \tau^{H_1 - 1/2} \left((1 - u)^{H_1 - 1/2} - (-u)_+^{H_1 - 1/2}\right) \tau^{H_2 - 1/2} \left((1 - u)^{H_2 - 1/2} - (-u)_+^{H_2 - 1/2}\right) \tau du \\
&= \tau^{H_1 + H_2} \cdot I(H_1, H_2),
\end{aligned}$$

where $I(H_1, H_2)$ is the integral

$$I(H_1, H_2) := \int_{-\infty}^{1} \left((1-u)^{H_1-1/2} - (-u)_+^{H_1-1/2}\right)\left((1-u)^{H_2-1/2} - (-u)_+^{H_2-1/2}\right) du.$$

Thus we have the covariance

$$\begin{aligned}\mathcal{C}[B^{H_1}, B^{H_2}](\tau) &= \rho\tau^{H_1+H_2} \cdot \frac{I(H_1, H_2)}{\Gamma(H_1+1/2)\cdot\Gamma(H_2+1/2)} \\ &= \rho\tau^{H_1+H_2}.\end{aligned}$$

Here we refer to [88] for cancelling the integral $I(H_1, H_2)$ with the normalizing factor. The variance terms as given in Eq. (4.8) can be verified by taking $H_1 = H_2$:

$$\begin{aligned}\mathcal{V}[B^{H_1}](\tau) &= \mathbb{E}[\mathcal{B}(W^{(1)}; H_1)_\tau^2] = \tau^{2H_1}, \\ \mathcal{V}[B^{H_2}](\tau) &= \mathbb{E}[\mathcal{B}(\rho W^{(1)} + \sqrt{1-\rho^2}W^{(2)}; H_2)_\tau^2] = \tau^{2H_2}.\end{aligned}$$

By definition of correlation,

$$\mathcal{R}[B^{H_1}, B^{H_2}](\tau) = \rho.$$

□

More general multivariate fractional Brownian motion can be defined through self-similarity in vector form. Its covariance structure is given in Theorem 2.1 of [71]. One can show that the pair-wise correlation will still be constant. In the following, we will show scale-independent correlation for the more general case, with multivariate fractional Brownian motion being a special example. To begin with, let's define self-similarity formally in the multivariate scenario.

Definition 8 (Self-similarity)**.** A p-dimensional multivariate random process $\vec{X}_t \in \mathbb{R}^p,\ t \geq 0$ is self-similar if there exist a vector $\vec{H} \in (0,1)^p$, s.t.

$$\left(X_{\lambda t}^{(1)}, \cdots, X_{\lambda t}^{(p)}\right) \overset{dist.}{=} \left(\lambda^{H_1} X_t^{(1)}, \cdots, \lambda^{H_p} X_t^{(p)}\right), \quad \forall t \geq 0. \tag{5.7}$$

Proposition 7. *For any p-dimensional multivariate random process that is self-similar, the pair-wise multiscale correlation $\mathcal{R}[X^{(i)}, X^{(j)}](\tau)$ between any $1 \leq i, j \leq p$ must be a constant.*

Proof. We prove by showing for any $\tau_1 \neq \tau_2 \in \mathbb{R}^+$, $\mathcal{R}[X^{(i)}, X^{(j)}](\tau_1) = \mathcal{R}[X^{(i)}, X^{(j)}](\tau_2)$. Due to self-similarity, the variance

$$\mathcal{V}[X^{(i)}](\tau_2) = \left(\frac{\tau_2}{\tau_1}\right)^{2H_i} \mathcal{V}[X^{(i)}](\tau_1), \quad \mathcal{V}[X^{(j)}](\tau_2) = \left(\frac{\tau_2}{\tau_1}\right)^{2H_j} \mathcal{V}[X^{(j)}](\tau_1).$$

The covariance

$$\begin{aligned}
\mathcal{C}[X^{(i)}, X^{(j)}](\tau_2) &= \mathbb{E}[(X^{(i)}_{t+\tau_2} - X^{(i)}_t) \cdot (X^{(j)}_{t+\tau_2} - X^{(j)}_t)] \\
&= \mathbb{E}[X^{(i)}_{\tau_2} X^{(j)}_{\tau_2}] \\
&= \mathbb{E}\left[\left(\frac{\tau_2}{\tau_1}\right)^{2H_i} X^{(i)}_{\tau_1} \cdot \left(\frac{\tau_2}{\tau_1}\right)^{2H_j} X^{(j)}_{\tau_1}\right] \\
&= \left(\frac{\tau_2}{\tau_1}\right)^{H_i+H_j} \mathbb{E}\left[X^{(i)}_{\tau_1} X^{(j)}_{\tau_1}\right] \\
&= \left(\frac{\tau_2}{\tau_1}\right)^{H_i+H_j} \mathcal{C}[X^{(i)}, X^{(j)}](\tau_1).
\end{aligned}$$

Then we have

$$\begin{aligned}
\mathcal{R}[X^{(i)}, X^{(j)}](\tau_2) &= \frac{\mathcal{C}[X^{(i)}, X^{(j)}](\tau_2)}{\sqrt{\mathcal{V}[X^{(i)}](\tau_2)\mathcal{V}[X^{(j)}](\tau_2)}} \\
&= \frac{(\tau_2/\tau_1)^{H_i+H_j}\mathcal{C}[X^{(i)}, X^{(j)}](\tau_1)}{\sqrt{(\tau_2/\tau_1)^{2H_i}\mathcal{V}[X^{(i)}](\tau_1) \cdot (\tau_2/\tau_1)^{2H_j}\mathcal{V}[X^{(j)}](\tau_1)}} \\
&= \mathcal{R}[X^{(i)}, X^{(j)}](\tau_1).
\end{aligned}$$

□

Example (Multivariate fractional Brownian motion)**.** A multivariate fractional Brownian motion (mfBm) is a p-dimensional Gaussian process with stationary increments and satisfies self-similarity with $\vec{H} \in (0,1)^p$. Applying Proposition 7, one can show that it has scale-independent pair-wise correlation.

5.3.2 Microstructure Noise in Correlated Prices

Following the discussion of microstructure noise in the previous chapter, let's further generalize the definition into the multivariate case.

Definition 9. Let $\vec{X}_t \in \mathbb{R}^p$, $t \geq 0$ be some underlying p-dimensional stochastic process. The multivariate noisy price process is defined as

$$\vec{P}_t = \exp(\vec{Y}_t), \quad \vec{Y}_t = \vec{X}_t + \vec{\epsilon}_t, \quad \forall t \geq 0,$$

where $\vec{\epsilon}$ is the independent noise s.t. $\vec{\epsilon} \perp X$, and $\mathbb{E}[\vec{\epsilon}_t] = \vec{0}$, $Cov(\epsilon_t) = \Sigma_\epsilon$, ϵ_t i.i.d., for $\forall t \geq 0$. The covariance matrix Σ_ϵ is a positive semi-definite matrix.

Property 9.1. *For any p-dimensional noisy price process $\vec{Y}_t = \vec{X}_t + \vec{\epsilon}_t$, the pair-wise multi-scale correlation between i and j, $i \neq j$ is*

$$\mathcal{R}[Y_i, Y_j](\tau) = \frac{\mathcal{R}[X_i, X_j](\tau)\sqrt{\mathcal{V}[X_i](\tau)\mathcal{V}[X_j](\tau)} + 2\Sigma_{\epsilon,ij}}{\sqrt{\left(\mathcal{V}[X_i](\tau) + 2\sigma^2_{\epsilon_i}\right) \cdot \left(\mathcal{V}[X_j](\tau) + 2\sigma^2_{\epsilon_j}\right)}}.$$

Proof. The multiscale covariance in the noise is

$$\begin{aligned}
\mathcal{C}[\epsilon_i, \epsilon_j] &= \mathbb{E}[(\epsilon_{i,t+\tau} - \epsilon_{i,t})(\epsilon_{j,t+\tau} - \epsilon_{j,t})] \\
&= \mathbb{E}[\epsilon_{i,t+\tau}\epsilon_{j,t+\tau}] + \mathbb{E}[\epsilon_{i,t}\epsilon_{j,t}] \\
&= 2\Sigma_{\epsilon,ij}.
\end{aligned}$$

The rest is to apply Proposition 5 and Eq. (4.9). □

Remark. Note that even when $R_{X_iX_j}(\tau)$ is scale-independent, $R_{Y_iY_j}(\tau)$ can still be scale dependent, as long as the multiscale variance of the underlying process is not constant (which is almost always the case unless the price is pure noise). Therefore, for almost all price processes, its noisy process will have scale-dependent correlation.

Remark. The noise cross term $\Sigma_{\epsilon,ij}$ depends on the correlation between microstructure noise. Based on various assumptions of the source of noise, this term can sometime to be assumed zero. However, as the study in microstructure noise is not settled, we keep this term for mathematical completeness.

5.3.3 Noisy fractional Brownian motion

Based on definitions from the previous sections, we can model asset log prices as a pair of noisy fractional Brownian motion defined as follows:

Definition 10. For Hurst exponents $H_1, H_2 \in (0,1)$, drift coefficients $\mu_1, \mu_2 \in \mathbb{R}$, volatility parameters $\sigma_1, \sigma_2 \in \mathbb{R}^+$, initial values $Y_0^{(1)}, Y_0^{(2)} \in \mathbb{R}$, and correlation coefficient $\rho \in [-1,1]$, define the pair of noisy fractional Brownian motion:

$$Y_t^{(1)} = Y_0^{(1)} + \mu_1 t + \sigma_1 \mathcal{B}(W^{(1)}; H_1)_t + \epsilon_t^{(1)}, \tag{5.8}$$

$$Y_t^{(2)} = Y_0^{(2)} + \mu_2 t + \sigma_2 \mathcal{B}(\rho W^{(1)} + \sqrt{1-\rho^2} W^{(2)}; H_2)_t + \epsilon_t^{(2)}, \tag{5.9}$$

where $W_t^{(1)}, W_t^{(2)}$ are independent Brownian motions, and $\epsilon_t^{(1)}, \epsilon_t^{(2)}$ are microstructure noises with variance $\mathbb{V}(\epsilon_t^{(1)}) = \sigma_{\epsilon,1}^2$, $\mathbb{V}(\epsilon_t^{(2)}) = \sigma_{\epsilon,2}^2$, and correlation $\rho_\epsilon \in [-1,1]$.

Correlation Curve

Proposition 6 and Property 9.1 together give the multiscale correlation of noisy fractional Brownian motions. Denote $\eta_i := \frac{\sigma_{\epsilon,i}^2}{\sigma_i^2},\ i = 1,2$ as the noise ratios. As a general case let's first consider the microstructure noise $\epsilon_t^{(1)}, \epsilon_t^{(2)}$ to be correlated, i.e. the noise correlation $\rho_\epsilon \neq 0$. The multiscale correlation will be:

$$\mathcal{R}[Y_1, Y_2](\tau) = \frac{\rho \tau^{H_1+H_2} + 2\rho_\epsilon \sqrt{\eta_1 \eta_2}}{\sqrt{(\tau^{2H_1} + 2\eta_1)(\tau^{2H_2} + 2\eta_2)}}. \tag{5.10}$$

If we assume the microstructure noise to be independent, the formula can be simplified to

$$\mathcal{R}[Y_1, Y_2](\tau) = \rho \cdot \frac{\tau^{H_1+H_2}}{\sqrt{(\tau^{2H_1} + 2\eta_1)(\tau^{2H_2} + 2\eta_2)}}. \tag{5.11}$$

We can see from Eq. (5.11) the noisy correlation is just the underlying correlation times a scaling factor, which depends on the noise ratios and the Hurst exponents of the underlying

processes. In the special case where $H_1 = H_2 = 1/2$, i.e. the log price processes are noisy Brownian motions, we have

$$\mathcal{R}[Y_1, Y_2](\tau) = \rho \cdot \frac{\tau}{\sqrt{(\tau + 2\eta_1)(\tau + 2\eta_2)}}. \tag{5.12}$$

Asymptotic Behavior

Next we consider the asymptotic behavior of multiscale correlation. Results are derived for the general case assuming the microstructure noises can be possibly correlated, followed by the special case of independent noise.

- $\tau \to \infty$,

  $$\mathcal{R}[Y_1, Y_2](\tau) \to \rho.$$

  The correlation converges to the correlation between the two underlying fractional Brownian motions.

- $\tau \to 0$,

  $$\mathcal{R}[Y_1, Y_2](\tau) \to \rho_\epsilon.$$

  Unlike volatility, the limit of correlation exists when the timescale is approaching zero. The intercept depends on the correlation between the noises, which can be used to determine if there are correlated noise. Also note that the asymptotic behavior of correlation does not depend on the Hurst exponent H at both ends, which is different from the volatility function. The Hurst exponent only affects the rate correlation increases to the underlying value.

In order to better understand the speed of correlation scaling, let's further look at the derivative of the correlation function. We can derive

$$\begin{aligned}\frac{d}{d\tau}\mathcal{R}[Y_1, Y_2](\tau) =& \rho \cdot \frac{2\tau^{H_1+H_2}(2\eta_1\eta_2(H_1 + H_2) + \eta_1 H_1 \tau^{2H_2} + \eta_2 H_2 \tau^{2H_1})}{\tau((2\eta_1 + \tau^{2H_1})(2\eta_2 + \tau^{2H_2}))^{3/2}} \\ & - \rho_\epsilon \cdot \frac{2\sqrt{\eta_1\eta_2}\left(2\eta_2 H_1 \tau^{2H_1} + 2\eta_1 H_2 \tau^{2H_2} + (H_1 + H_2)\tau^{2(H_1+H_2)}\right)}{\tau((2\eta_1 + \tau^{2H_1})(2\eta_2 + \tau^{2H_2}))^{3/2}}.\end{aligned}$$

The form is rather complicated with one fBm correlation term and one noise correlation term. Let's look at its asymptotic behaviors under certain conditions.

- $\tau \to \infty$,
  - Case $H_1 \neq H_2$:
    $$\frac{d}{d\tau}\mathcal{R}[Y_1, Y_2](\tau) \approx \rho \frac{2\eta_{i_-} H_{i_-}}{\tau^{1+2H_{i_-}}},$$
    where i_- as the minimizer of $H_i, i = 1, 2$. We can see that the noise correlation term doesn't affect the asymptotic behavior at large scale, and the limiting behavior is controlled by the smaller Hurst exponent.
  - Case $H_1 = H_2 = H$:
    $$\frac{d}{d\tau}\mathcal{R}[Y_1, Y_2](\tau) \approx \frac{4H}{\tau^{1+2H}} \left(\frac{\eta_1 + \eta_2}{2}\rho - \sqrt{\eta_1 \eta_2}\rho_\epsilon \right).$$
    The function form is the same as the $H_1 \neq H_2$ case, however, it is surprising that the noise correlation term is playing a part in the asymptotic behavior even at very large scale. Positive correlation in noise will actually decrease the speed of correlation convergence.

- $\tau \to 0$,
  - Case $H_1 \neq H_2$:
    $$\frac{d}{d\tau}\mathcal{R}[Y_1, Y_2](\tau) \approx -\frac{\rho_\epsilon H_{i_-}}{2\eta_{i_-}} \tau^{2H_{i_-}-1}.$$
    Again, the smaller Hurst exponent controls the limiting behavior. Also note that positive noise correlation leads to negative derivative in correlation at small scale.
  - Case $H_1 = H_2 = H$:
    $$\frac{d}{d\tau}\mathcal{R}[Y_1, Y_2](\tau) \approx \left(\frac{\rho}{\sqrt{\eta_1 \eta_2}} - \rho_\epsilon \frac{\eta_1 + \eta_2}{2\eta_1 \eta_2} \right) H \tau^{2H-1}.$$
    As in the large scale limit scenario, equal Hurst exponent makes the derivative contribution of fBm correlation and noise correlation balanced throughout the whole correlation curve.

In general, the contribution of the fBm and the noise terms to correlation derivative is sensitive to the difference between the Hurst exponents. If the Hurst exponents are different, it is always the smaller one dominating the asymptotic behavior, at both small and large scale. The dominant term is different at the two ends though. When the two processes have identical Hurst exponent, the fBm and the noise terms are balanced at any scale, weighted according to the arithmetic mean and the geometric mean of the noise ratios.

Fig. 5.1 below shows the correlation and derivative curves of the fractional Brownian motion model. The parameters are: $\rho = 0.8, \rho_\epsilon = 0, \eta_1 = 10^{-5}, \eta_2 = 2\times 10^{-5}$. Different combinations of Hurst exponents are shown in the legends.

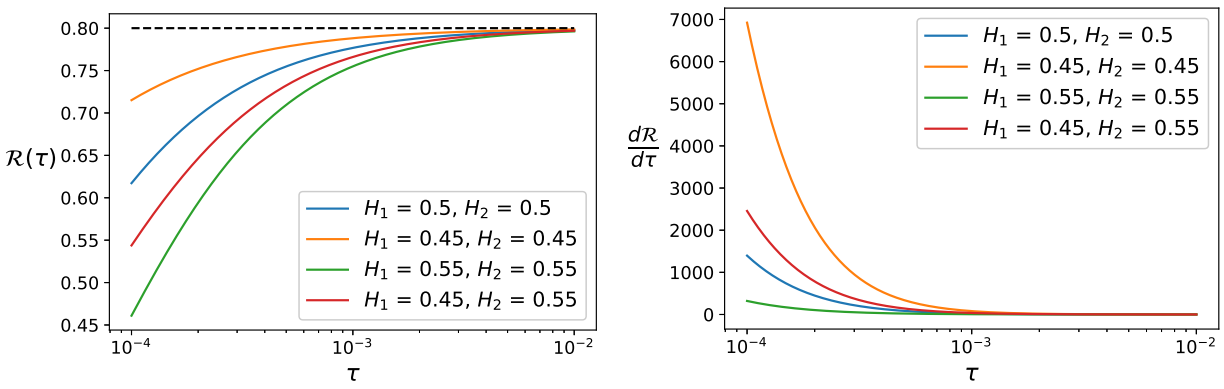

Figure 5.1: Multiscale correlation of the noisy Brownian motion model with different Hurst exponents. Left: correlation curves. Right: derivative of correlation curves.

Evaluation and Fitting

The correlation curve given by Eq. (5.10) can be rather complicated with six parameters, making it hard to evaluate against observation from real-world data. In practice it is well reasonable to hold independent noise and Brownian motion assumptions, reducing the correlation form to be as in Eq. (5.12). Starting from this expression with only three parameters ρ, η_1, η_2, let's do the following operation:

$$\frac{1}{(\mathcal{R}[Y_1, Y_2](\tau))^2} = \frac{1}{\rho^2}\frac{(\tau+2\eta_1)(\tau+2\eta_2)}{\tau^2} = \frac{1}{\rho^2}\left(1+\frac{2\eta_1}{\tau}\right)\left(1+\frac{2\eta_2}{\tau}\right).$$

Define the frequency variable $\omega := 1/\tau$, and assume $\eta_1, \eta_2 \ll 1$, we have

$$\frac{1}{\mathcal{R}[Y_1, Y_2]^2} = \frac{1}{\rho^2}(1 + 2(\eta_1+\eta_2)\omega + 4\eta_1\eta_2\omega^2) \tag{5.13}$$

$$\approx \frac{1}{\rho^2} + \frac{2(\eta_1+\eta_2)}{\rho^2}\omega. \tag{5.14}$$

As we can see from Eq. (5.14), the inverse of correlation squared can be approximated by a linear function in frequency $\omega : 1/\tau$. The intercept is determined by ρ^2, and the slope is positive and depends on the summation of the two noise ratios $\eta_1 + \eta_2$.

Consider a collection of p noisy Brownian motions $Y_1, \cdots, Y_p$, with noise ratios $\eta_1, \cdots, \eta_p$ correspondingly. Suppose the correlation coefficient between the latent processes of Y_i and Y_j is ρ_{ij}, we can evaluate the parameters following the following procedure:

- Compute $R_{ij,m}$, $1 \le i < j \le p$ using Eq. (5.6), and $\omega_m = \frac{1}{m\delta t}$, for $m = 1, \cdots, M$.
- Fit linear regressions $\frac{1}{R^2_{ij,m}} = \alpha_{ij} + \beta_{ij}\omega_m$, for $1 \le i < j \le p$.
- Estimator of correlation $\hat{\rho_{ij}} = \text{sgn}(R_{ij,M})\frac{1}{\sqrt{\alpha_{ij}}}$.
- Define the index mapping function $\xi(i,j) = \sum_{l=1}^{i-1}(p-l) + (j-i)$, for $1 \le i < j \le p$. The values of ξ are $1, \cdots, p(p-1)/2$.
- Construct vector $\vec{b} \in \mathbb{R}^{p(p-1)/2}$, s.t. $b_{\xi(i,j)} = \frac{\beta_{ij}}{2\alpha_{ij}}$, for $1 \le i < j \le p$.
- Construct matrix $A \in \mathbb{R}^{p(p-1)/2 \times p}$, s.t. $A_{k,i} = 1$ if there exist $j > i$ s.t. $k = \xi(i,j)$, or $j < i$ s.t. $k = \xi(j,i)$, otherwise $A_{k,i} = 0$.
- Estimator of the noise ratio vector $\hat{\eta} = A^{\dagger}\vec{b}$.

$A^{\dagger}$ is the pseudo-inverse of A. We need at least $p \ge 3$ for $\hat{\eta}$ to be solvable. When $p = 3$, A is invertible. When $p > 3$, the matrix is over-determined.

5.4 Experiments on Intraday Data

In this section we discuss experimental results on real-world high-frequency data, using the same 3-second intraday dataset as in Chapter 4.5. Again, we chose a time frame such taht $t = 1$ corresponds to 1 minute, i.e. $\delta t = 0.05$ for the 3-second interval. In total there are 530 intraday time series with 7800 points for each ticker.

Using Eq. (5.6), we computed correlation curves for different asset pairs. Fig. 5.2 shows the correlation curves for all of the pairs among the six tickers SPY, IWM, QQQ, XLK, AAPL and MSFT, averaged over all dates in the dataset. From the plots we can see that all of the correlation curves show a concave increasing shape, converging up to certain levels. This is clear evidence that the correlation between intraday price movements is scale-dependent. As we have shown in the previous sections, none of Brownian motion, fractional Brownian motion, and stochastic volatility models yields such scale-dependent correlation. The shape of the curves are very similar to that of the noisy (fractional) Brownian motion model shown in Fig. 5.1. Let's proceed to verifying the fitness of the model.

To further evaluate whether the real-world data match the correlation function of a noisy fractional Brownian motion model, we transformed the estimated curves to check if the data fit the linear form in Eq. (5.14). Fig. 5.3 shows the plots of $1/\mathcal{R}^2$ against $\omega : 1/\tau$. From the figures we can see approximately linear relation for all of the ticker pairs, indicating that the model can be a good fit. Nevertheless, we should also notice the bumpy shape when ω is very small, i.e. at large timescale. Even though they only show very small parts on the plots, we should keep in mind that $\omega := 1/\tau$ is not evenly distributed on the x-axis. Therefore, there might be unexplained patterns at large timescale. Even though the noisy fractional Brownian motion model may not be a perfect fit for the whole dynamics of the data, it provides a theoretical framework to explain the correlation behavior existing in real-world high frequency price.

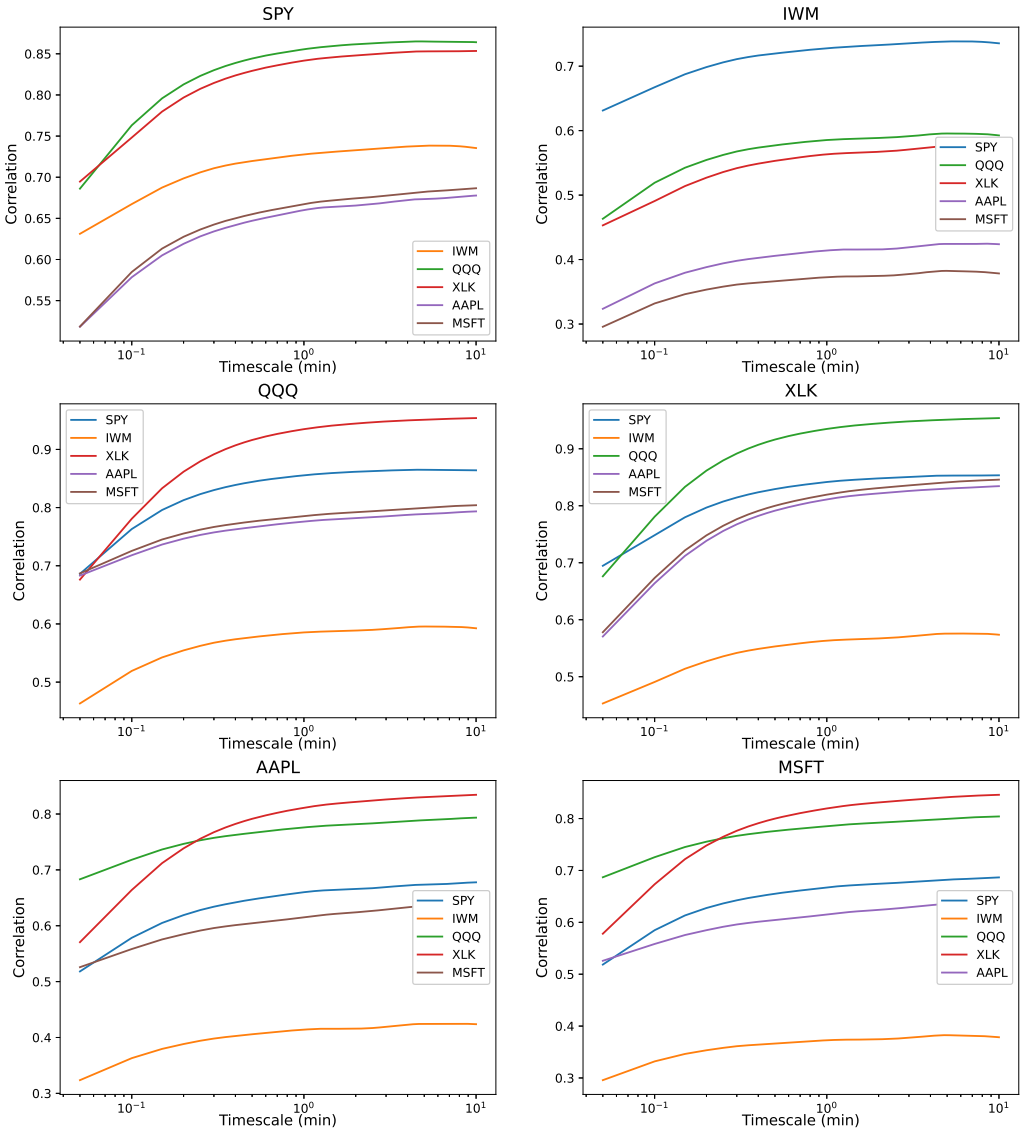

Figure 5.2: Correlation curves estimated on the 3-second intraday dataset. Algorithm see section 5.2.2. For each day one curve is estimated for each asset pair. The average curves over all dates are shown in the plot.

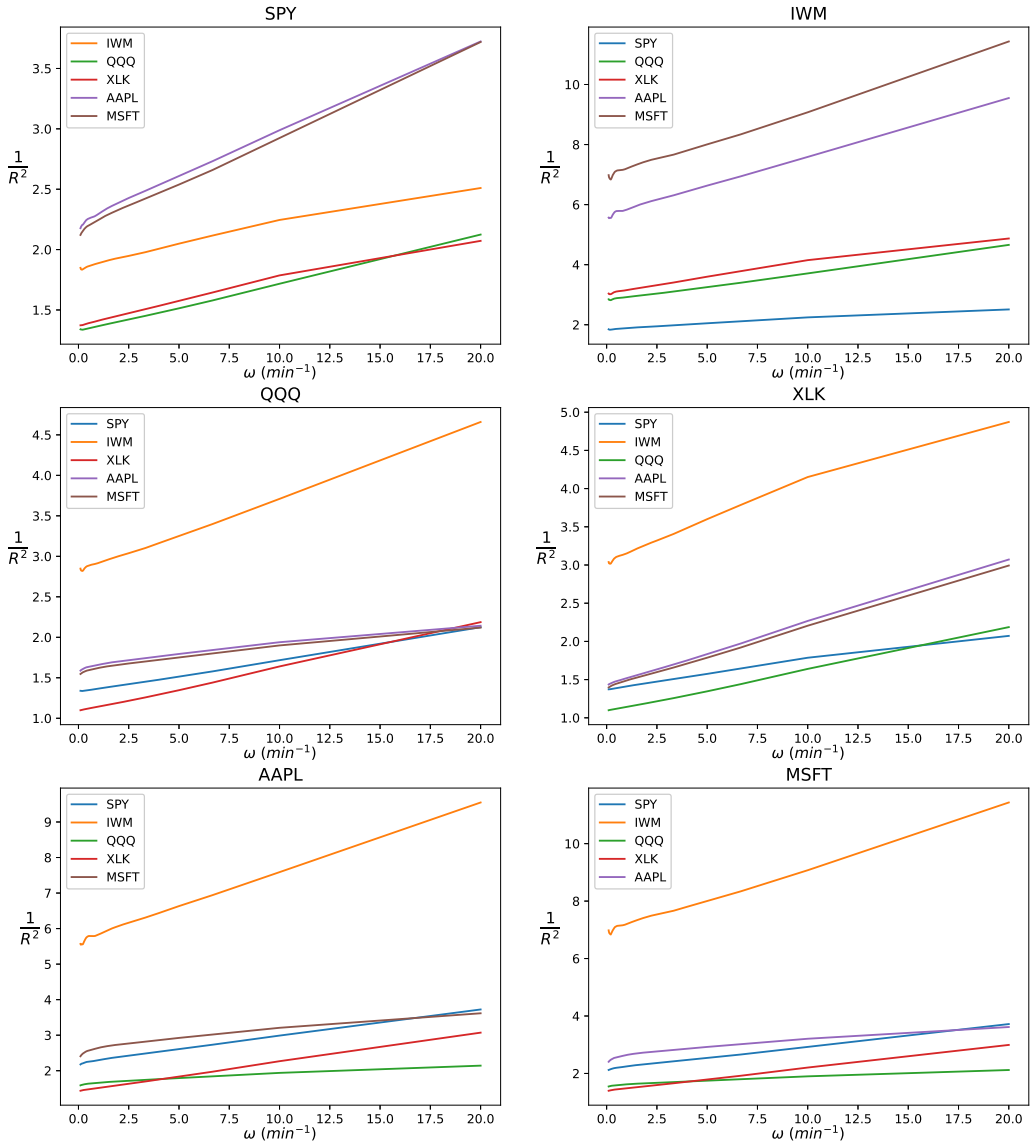

Figure 5.3: $1/\mathcal{R}^2$ v.s. $\omega := 1/\tau$ plots for the 3-second intraday dataset. The transform using Eq. (5.14) is applied to the average correlation curves over all dates.

Following the fitting procedure for Eq. (5.14), we provide numerical estimation of the correlation and noise ratio parameters in Table 5.1. Compared to the noise ratio in Table 4.3 estimated using variance information, the values are much smaller. One possibility is that the Hurst exponents are smaller than 1/2, but we took the Brownian motion assumption when fitting both curves. According to Fig. 5.1, smaller H gives flatter correlation change, making the noise ratio underestimated. On the other hand, fitting the variance curve with smaller Hurst exponent makes the intercept smaller, thus the noise ratio in Table 4.3 could be overestimated. In conclusion, the discrepancy can be explained by Hurst exponent smaller than 1/2, and we need a parameter estimation framework for more general cases.

	η $(10^{-3}\ min)$	SPY	IWM	QQQ	XLK	AAPL	MSFT
SPY	5.666800	1	0.732389	0.869289	0.850049	0.66915	0.678656
IWM	6.110624	0.732389	1	0.59579	0.569363	0.420207	0.378528
QQQ	5.992276	0.869289	0.59579	1	0.961054	0.778732	0.789037
XLK	15.389514	0.850049	0.569363	0.961054	1	0.83515	0.843776
AAPL	7.875992	0.66915	0.420207	0.778732	0.83515	1	0.617018
MSFT	7.738115	0.678656	0.378528	0.789037	0.843776	0.617018	1

Table 5.1: Estimation of the parameters in multivariate noisy Brownian motion model using the correlation curves estimated on the intraday dataset. The first column shows the noise ratio estimator $\eta_i := \sigma^2_{\epsilon,i}/\sigma^2_i$. The right columns show the estimated latent correlation matrix.

In Fig. 5.4 and 5.5 we show the 60-day rolling average of correlation estimated at 3-second, 30-second, and 10-minute scales. In comparison we provide rolling correlation of daily close return in the same window. We can see that the intraday correlation at 30 seconds is already close to daily correlation, and moves in the same pattern. However, the correlation gap from 3 seconds to 30 seconds was significantly large in 2020 for SPY v.s. QQQ, AAPL v.s. SPY, and AAPL v.s. XLK. The 3-second correlation for the other pairs always stayed parallel to the others. This may indicate a change in high-frequency trading for some of the tech stocks.

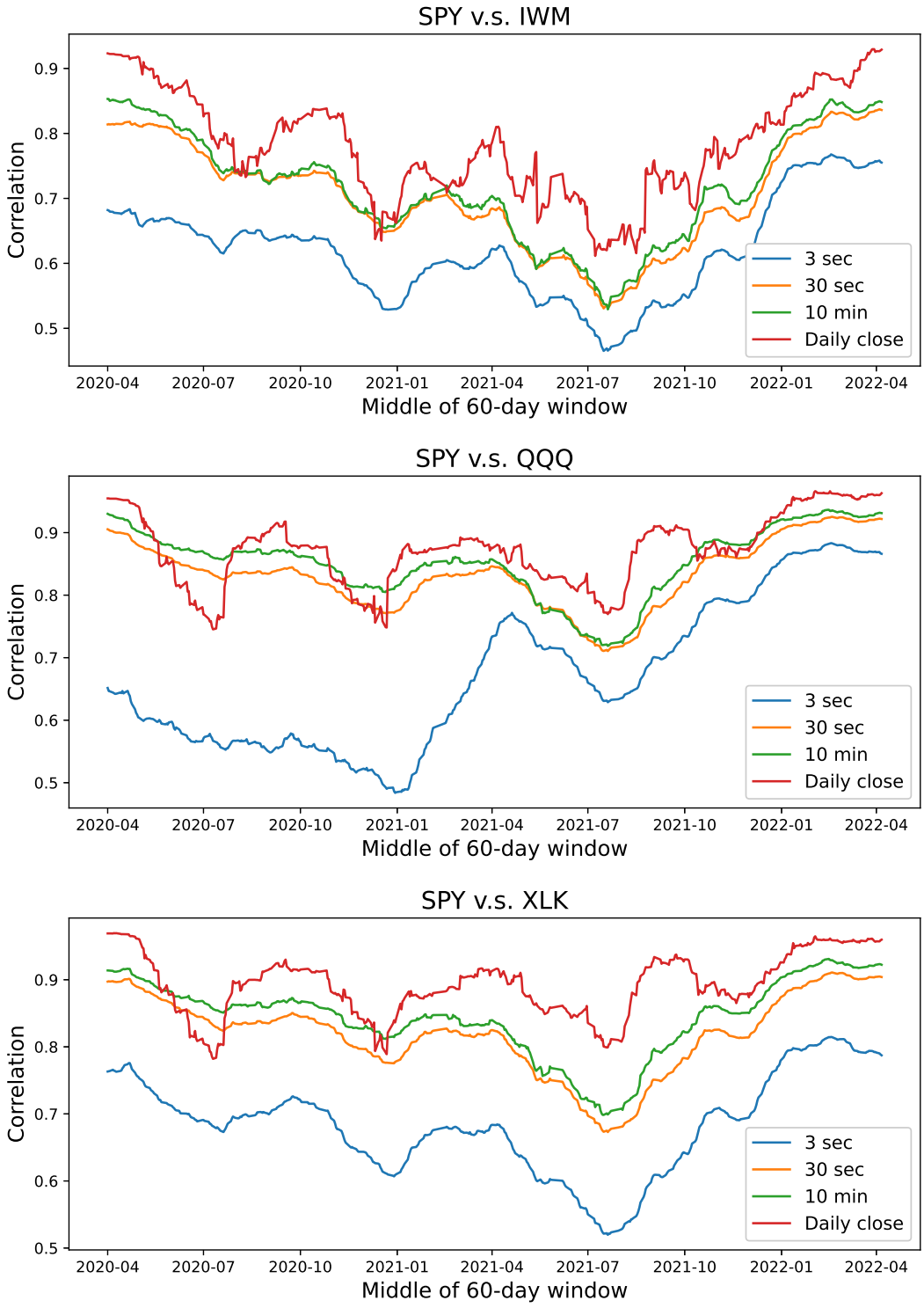

Figure 5.4: Multiscale correlation with SPY estimated on 60-day rolling window. Intraday return correlation at different scales are estimated on each day, and averaged over all dates in the rolling window. As a comparison, the return correlation for daily close price estimated in the same window is shown.

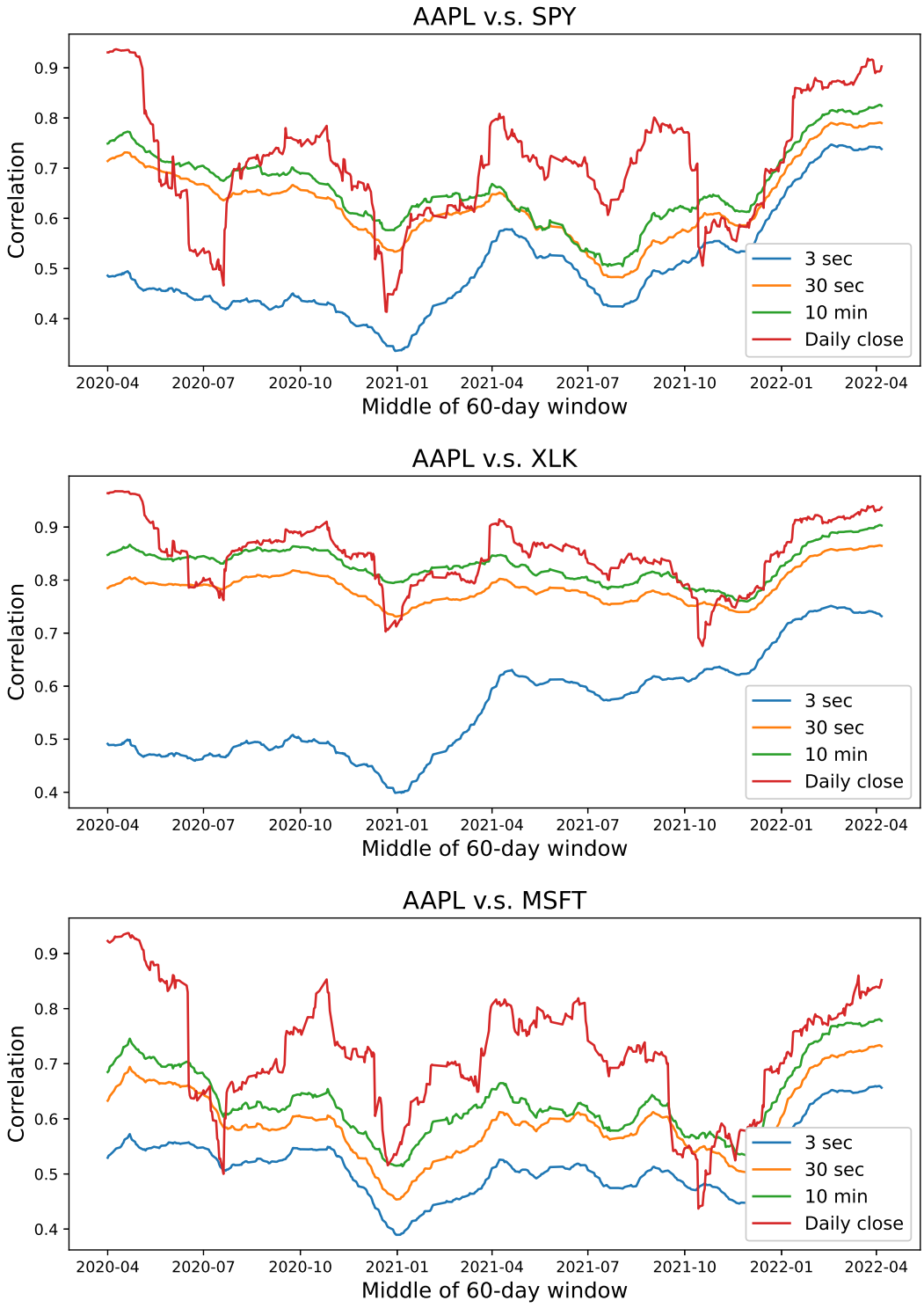

Figure 5.5: Multiscale correlation with AAPL estimated on 60-day rolling window. Intraday return correlation at different scales are estimated on each day, and averaged over all dates in the rolling window. As a comparison, the return correlation for daily close price estimated in the same window is shown.

BIBLIOGRAPHY

[1] Anat R Admati and Paul Pfleiderer. A theory of intraday patterns: Volume and price variability. *The review of financial studies*, 1(1):3–40, 1988.

[2] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[3] Yacine Aït-Sahalia, Jianqing Fan, and Dacheng Xiu. High-frequency covariance estimates with noisy and asynchronous financial data. *Journal of the American Statistical Association*, 105(492):1504–1517, 2010.

[4] Yacine Ait-Sahalia, Per A Mykland, and Lan Zhang. How often to sample a continuous-time process in the presence of market microstructure noise. *The review of financial studies*, 18(2):351–416, 2005.

[5] E Alessio, A Carbone, G Castelli, and V Frappietro. Second-order moving average and scaling of stochastic time series. *The European Physical Journal B-Condensed Matter and Complex Systems*, 27(2):197–200, 2002.

[6] Romain Allez and Jean-Philippe Bouchaud. Individual and collective stock dynamics: intra-day seasonalities. *New Journal of Physics*, 13(2):025010, 2011.

[7] Pierre-Olivier Amblard and Jean-François Coeurjolly. Identification of the multivariate fractional brownian motion. *IEEE Transactions on Signal Processing*, 59(11):5152–5168, 2011.

[8] Pierre-Olivier Amblard, Jean-François Coeurjolly, Frédéric Lavancier, and Anne Philippe. Basic properties of the multivariate fractional brownian motion. *arXiv preprint arXiv:1007.0828*, 2010.

[9] Torben G Andersen and Tim Bollerslev. Intraday periodicity and volatility persistence in financial markets. *Journal of empirical finance*, 4(2-3):115–158, 1997.

[10] Torben G Andersen, Tim Bollerslev, and Francis X Diebold. Some like it smooth, and some like it rough: Untangling continuous and jump components in measuring, modeling, and forecasting asset return volatility. *Modeling, and Forecasting Asset Return Volatility (September 2003)*, 2003.

[11] Federico M Bandi and Jeffrey R Russell. Microstructure noise, realized volatility, and optimal sampling. *Unpublished paper, Graduate School of Business, University of Chicago*, 2003.

[12] Albert-László Barabási and Tamás Vicsek. Multifractality of self-affine fractals. *Physical Review A*, 44(4):2730, 1991.

[13] Aurelio F Bariviera, María José Basgall, Waldo Hasperué, and Marcelo Naiouf. Some stylized facts of the Bitcoin market. *Physica A: Statistical Mechanics and its Applications*, 484:82–90, 2017.

[14] Ole E Barndorff-Nielsen and Neil Shephard. Econometric analysis of realized volatility and its use in estimating stochastic volatility models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2):253–280, 2002.

[15] Edward Bedrosian. A product theorem for Hilbert transforms. *Proceedings of the IEEE*, 51(5):868–869, 1963.

[16] Geert Bekaert and Guojun Wu. Asymmetric volatility and risk in equity markets. *The review of financial studies*, 13(1):1–42, 2000.

[17] Francesca Biagini, Yaozhong Hu, Bernt Øksendal, and Tusheng Zhang. *Stochastic calculus for fractional Brownian motion and applications*. Springer Science & Business Media, 2008.

[18] Markus Bibinger, Nikolaus Hautsch, Peter Malec, and Markus Reiß. Estimating the quadratic covariation matrix from noisy observations: Local method of moments and efficiency. *The Annals of Statistics*, 42(4):1312–1346, 2014.

[19] Fischer Black. Noise. *The journal of finance*, 41(3):528–543, 1986.

[20] Elie Bouri, Luis A Gil-Alana, Rangan Gupta, and David Roubaud. Modelling long memory volatility in the bitcoin market: Evidence of persistence and structural breaks. *International Journal of Finance & Economics*, 24(1):412–426, 2019.

[21] Elie Bouri, Ladislav Kristoufek, Tanveer Ahmad, and Syed Jawad Hussain Shahzad. Microstructure noise and idiosyncratic volatility anomalies in cryptocurrencies. *Annals of Operations Research*, pages 1–27, 2022.

[22] Leo Breiman. *Classification and Regression Trees*. Routledge, 2017.

[23] Enrico Capobianco. Multiscale stochastic dynamics in finance. *Physica A: Statistical Mechanics and its Applications*, 344(1-2):122–127, 2004.

[24] Yin-Wong Cheung and Kon S Lai. Lag order and critical values of the augmented dickey–fuller test. *Journal of Business & Economic Statistics*, 13(3):277–280, 1995.

[25] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979.

[26] Jean-François Coeurjolly, Pierre-Olivier Amblard, and Sophie Achard. On multivariate fractional brownian motion and multivariate fractional gaussian noise. In *2010 18th European Signal Processing Conference*, pages 1567–1571. IEEE, 2010.

[27] Jean-François Coeurjolly, Pierre-Olivier Amblard, and Sophie Achard. Wavelet analysis of the multivariate fractional brownian motion. *ESAIM: Probability and Statistics*, 17:592–604, 2013.

[28] Rama Cont and Purba Das. Rough volatility: fact or artefact? *Available at SSRN*, 2022.

[29] Michel Couillard and Matt Davison. A comment on measuring the hurst exponent of financial time series. *Physica A: Statistical Mechanics and its Applications*, 348:404–418, 2005.

[30] J. Cvitanić, Z. Wiener, and F. Zapatero. Analytic pricing of employee stock options. *Review of Financial Studies*, 21(2):683–724, 2008.

[31] Richard A Davis, Keh-Shin Lii, and Dimitris N Politis. Remarks on some nonparametric estimates of a density function. In *Selected Works of Murray Rosenblatt*, pages 95–100. Springer, 2011.

[32] Tiziana Di Matteo. Multi-scaling in finance. *Quantitative finance*, 7(1):21–36, 2007.

[33] Tiziana Di Matteo, Tomaso Aste, and Michel M Dacorogna. Scaling behaviors in differently developed markets. *Physica A: Statistical Mechanics and its Applications*, 324(1-2):183–188, 2003.

[34] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[35] Konstantin Dragomiretskiy and Dominique Zosso. Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544, 2013.

[36] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, pages 155–161, 1997.

[37] Eugene F Fama. The behavior of stock-market prices. *The journal of Business*, 38(1):34–105, 1965.

[38] Jianqing Fan and Yazhen Wang. Multi-scale jump and volatility analysis for high-frequency financial data. *Journal of the American Statistical Association*, 102(480):1349–1362, 2007.

[39] Jean-Pierre Fouque, George Papanicolaou, Ronnie Sircar, and Knut Solna. Multiscale stochastic volatility asymptotics. *Multiscale Modeling & Simulation*, 2(1):22–42, 2003.

[40] Andrés García-Medina and Graciela González Farías. Transfer entropy as a variable selection methodology of cryptocurrencies in the framework of a high dimensional predictive model. *PloS one*, 15(1), 2020.

[41] AZ Górski, S Drożdż, and Jülich Speth. Financial multifractality and its subtleties: an example of dax. *Physica A: Statistical Mechanics and its Applications*, 316(1-4):496–510, 2002.

[42] MA Sánchez Granero, JE Trinidad Segovia, and J García Pérez. Some comments on hurst exponent and the long memory processes on capital markets. *Physica A: Statistical Mechanics and its applications*, 387(22):5543–5551, 2008.

[43] Paolo Guasoni, Yuliya Mishura, and Miklós Rásonyi. High-frequency trading with fractional brownian motion. *Finance and stochastics*, 25(2):277–310, 2021.

[44] Paolo Guasoni, Zsolt Nika, and Miklós Rásonyi. Trading fractional brownian motion. *SIAM journal on financial mathematics*, 10(3):769–789, 2019.

[45] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[47] Thomas Y Hou and Zuoqiang Shi. Adaptive data analysis via sparse time-frequency representation. *Advances in Adaptive Data Analysis*, 3(01n02):1–28, 2011.

[48] Thomas Y Hou and Zuoqiang Shi. Data-driven time-frequency analysis. *Appl. Comput. Harmon. Anal*, 35(2):284–308, 2013.

[49] Thomas Y Hou, Mike P Yan, and Zhaohua Wu. A variant of the EMD method for multi-scale data. *Advances in Adaptive Data Analysis*, 1(04):483–516, 2009.

[50] Albert S Hu, Christine A Parlour, and Uday Rajan. Cryptocurrencies: Stylized facts on a new investible instrument. *Financial Management*, 48(4):1049–1068, 2019.

[51] Boqiang Huang and Angela Kunoth. An optimization based empirical mode decomposition scheme. *Journal of Computational and Applied Mathematics*, 240:174–183, 2013.

[52] Norden Huang and Steven Long. Normalized Hilbert transform and instantaneous frequency. *NASA Patent Pending GSC*, 14:673–680, 2003.

[53] Norden E Huang, Zheng Shen, and Steven R Long. A new view of nonlinear water waves: the Hilbert spectrum. *Annual Review of Fluid Mechanics*, 31(1):417–457, 1999.

[54] Norden E Huang, Zheng Shen, Steven R Long, Manli C Wu, Hsing H Shih, Quanan Zheng, Nai-Chyuan Yen, Chi Chao Tung, and Henry 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*, 454(1971):903–995, 1998.

[55] Norden E Huang, Man-Li Wu, Wendong Qu, Steven R Long, and Samuel SP Shen. Applications of Hilbert–Huang transform to non-stationary financial time series analysis. *Applied Stochastic Models in Business and Industry*, 19(3):245–268, 2003.

[56] Norden Eh Huang. *Hilbert-Huang Transform and Its Applications*, volume 16. World Scientific, 2014.

[57] Harold Edwin Hurst. Long-term storage capacity of reservoirs. *Transactions of the American society of civil engineers*, 116(1):770–799, 1951.

[58] Francis In and Sangbae Kim. *An Introduction to Wavelet Theory in Finance*. World Scientific, Singapore, 2012.

[59] Francis In and Sangbae Kim. *An introduction to wavelet theory in finance: a wavelet multiscale approach*. World scientific, 2013.

[60] Jean Jacod, Yingying Li, and Xinghua Zheng. Statistical properties of microstructure noise. *Econometrica*, 85(4):1133–1174, 2017.

[61] Sang Hoon Kang, Ron P McIver, and Jose Arreola Hernandez. Co-movements between bitcoin and gold: A wavelet coherence analysis. *Physica A: Statistical Mechanics and its Applications*, 536:120888, 2019.

[62] Paraskevi Katsiampa. Volatility estimation for bitcoin: A comparison of garch models. *Economics Letters*, 158:3–6, 2017.

[63] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ArXiv Preprint arXiv:1412.6980*, 2014.

[64] Lambert H Koopmans. *The spectral analysis of time series*. Elsevier, 1995.

[65] Ladislav Kristoufek. What are the main drivers of the bitcoin price? evidence from wavelet coherence analysis. *PloS one*, 10(4), 2015.

[66] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[67] Viktor Grigor'evich Kurbatskii, Denis Nikolaevich Sidorov, Vadim Aleksandrovich Spiryaev, and Nikita V Tomin. On the neural network approach for forecasting of nonstationary time series on the basis of the Hilbert-Huang transform. *Automation and Remote Control*, 72(7):1405–1414, 2011.

[68] V Kurbatsky, D Sidorov, V Spiryaev, and N Tomin. Using the Hilbert-Huang transform for ANN prediction of nonstationary processes in complex power systems. In *8 th World Energy System Conference "WESC*, pages 106–110, 2010.

[69] Victor G Kurbatsky, Denis N Sidorov, Vadim A Spiryaev, and Nikita V Tomin. Forecasting nonstationary time series based on Hilbert-Huang transform and machine learning. *Automation and Remote Control*, 75(5):922–934, 2014.

[70] Denis Kwiatkowski, Peter CB Phillips, Peter Schmidt, Yongcheol Shin, et al. Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of econometrics*, 54(1-3):159–178, 1992.

[71] Frédéric Lavancier, Anne Philippe, and Donatas Surgailis. Covariance function of vector self-similar processes. *Statistics & probability letters*, 79(23):2415–2421, 2009.

[72] T. Leung and T. Zhao. Financial time series analysis and forecasting with hht feature generation and machine learning. *Applied Stochastic Models in Business and Industry*, 2021.

[73] Tim Leung and Xin Li. *Optimal Mean Reversion Trading: Mathematical Analysis and Practical Applications*. Modern Trends in Financial Engineering. World Scientific Publishing Company, March 2016.

[74] Tim Leung and Hung Nguyen. Constructing cointegrated cryptocurrency portfolios for statistical arbitrage. *Studies in Economics and Finance*, 36(3):581–599, 2019.

[75] Tim Leung and Theodore Zhao. Financial time series analysis and forecasting with hilbert–huang transform feature generation and machine learning. *Applied Stochastic Models in Business and Industry*, 37(6):993–1016, 2021.

[76] Jing Li, Cai Liu, Zhaofa Zeng, and Lingna Chen. GPR signal denoising and target extraction with the CEEMD method. *IEEE Geoscience and Remote Sensing Letters*, 12(8):1615–1619, 2015.

[77] Aktham I. Maghyereh, Basel Awartani, and Hussein Abdoh. The co-movement between oil and clean energy stocks: A wavelet-based analysis of horizon associations. *Energy*, 169:895–913, 2019.

[78] Paul Malliavin and Maria Elvira Mancino. Fourier series method for measurement of multivariate volatilities. *Finance and Stochastics*, 6(1):49–61, 2002.

[79] Paul Malliavin, Maria Elvira Mancino, et al. A Fourier transform method for nonparametric estimation of multivariate volatility. *The Annals of Statistics*, 37(4):1983–2010, 2009.

[80] Maria Elvira Mancino, Maria Cristina Recchioni, and Simona Sanfelici. *Fourier-Malliavin volatility estimation: Theory and practice*. Springer, 2017.

[81] Benoit Mandelbrot. New methods in statistical economics. *Journal of political economy*, 71(5):421–440, 1963.

[82] Benoit Mandelbrot and Richard L Hudson. *The Misbehavior of Markets: A fractal view of financial turbulence*. Basic books, 2007.

[83] Benoit B Mandelbrot. *Fractals and scaling in finance: Discontinuity, concentration, risk. Selecta volume E*. Springer Science & Business Media, 2013.

[84] Benoit B Mandelbrot and John W Van Ness. Fractional brownian motions, fractional noises and applications. *SIAM review*, 10(4):422–437, 1968.

[85] Benoit B Mandelbrot and James R Wallis. Noah, joseph, and operational hydrology. *Water resources research*, 4(5):909–918, 1968.

[86] José AO Matos, Sílvio MA Gama, Heather J Ruskin, Adel Al Sharkasi, and Martin Crane. Time and scale hurst exponent analysis for financial markets. *Physica A: Statistical Mechanics and its Applications*, 387(15):3910–3915, 2008.

[87] Walid Mensi, Mobeen Ur Rehman, Muhammad Shafiullah, Khamis Hamed Al-Yahyaee, and Ahmet Sensoy. High frequency multiscale relationships among major cryptocurrencies: portfolio management implications. *Financial Innovation*, 7(1):1–21, 2021.

[88] I.S. Mishura, I.S. Mishura, J.S. Mišura, Y. Mishura, and Û.S. Mišura. *Stochastic calculus for fractional Brownian motion and related processes*, volume 1929. Springer Science & Business Media, 2008.

[89] Raffaello Morales, Tiziana Di Matteo, Ruggero Gramatica, and Tomaso Aste. Dynamical generalized hurst exponent as a tool to monitor unstable periods in financial time series. *Physica A: statistical mechanics and its applications*, 391(11):3180–3189, 2012.

[90] Ulrich A Müller, Michel M Dacorogna, Richard B Olsen, Olivier V Pictet, Matthias Schwarz, and Claude Morgenegg. Statistical study of foreign exchange rates, empirical evidence of a price change scaling law, and intraday analysis. *Journal of Banking & Finance*, 14(6):1189–1208, 1990.

[91] Saif Nalband, Amalin Prince, and Anita Agrawal. Entropy-based feature extraction and classification of vibroarthographic signal using complete ensemble empirical mode decomposition with adaptive noise. *IET Science, Measurement & Technology*, 12(3):350–359, 2017.

[92] Saif Nalband, RR Sreekrishna, and A Amalin Prince. Analysis of knee joint vibration signals using ensemble empirical mode decomposition. *Procedia Computer Science*, 89:820–827, 2016.

[93] Saif Nalband, CA Valliappan, A Amalin Prince, and Anita Agrawal. Time-frequency based feature extraction for the analysis of vibroarthographic signals. *Computers & Electrical Engineering*, 69:720–731, 2018.

[94] Noemi Nava, Tiziana Di Matteo, and Tomaso Aste. Dynamic correlations at different time-scales with empirical mode decomposition. *Physica A: Statistical Mechanics and its Applications*, 502:534–544, 2018.

[95] Noemi Nava, Tiziana Di Matteo, and Tomaso Aste. Financial time series forecasting using empirical mode decomposition and support vector regression. *Risks*, 6(1):7, 2018.

[96] Chun-Xiao Nie. Correlation dynamics in the cryptocurrency market based on dimensionality reduction analysis. *Physica A: Statistical Mechanics and its Applications*, page 124702, 2020.

[97] Mingfei Niu, Yufang Wang, Shaolong Sun, and Yongwu Li. A novel hybrid decomposition-and-ensemble model based on CEEMD and GWO for short-term PM2.5 concentration forecasting. *Atmospheric Environment*, 134:168–180, 2016.

[98] A.H. Nuttall and Edward Bedrosian. On the quadrature approximation to the Hilbert transform of modulated signals. *Proceedings of the IEEE*, 54(10):1458–1459, 1966.

[99] Bernt Øksendal. Fractional brownian motion in finance. *Preprint series. Pure mathematics http://urn. nb. no/URN: NBN: no-8076*, 2003.

[100] Maury FM Osborne. Brownian motion in the stock market. *Operations research*, 7(2):145–173, 1959.

[101] Emanuel Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[102] C-K Peng, Sergey V Buldyrev, Shlomo Havlin, Michael Simons, H Eugene Stanley, and Ary L Goldberger. Mosaic organization of dna nucleotides. *Physical review e*, 49(2):1685, 1994.

[103] Peter CB Phillips and Pierre Perron. Testing for a unit root in time series regression. *Biometrika*, 75(2):335–346, 1988.

[104] Ross C Phillips and Denise Gorse. Cryptocurrency price drivers: Wavelet coherence analysis revisited. *PloS one*, 13(4), 2018.

[105] Arkady Pikovsky, Jurgen Kurths, Michael Rosenblum, and Jürgen Kurths. *Synchronization: a universal concept in nonlinear sciences*, volume 12. Cambridge university press, 2003.

[106] Mark Podolskij and Mathias Vetter. Estimation of volatility functionals in the simultaneous presence of microstructure noise and jumps. *Bernoulli*, 15(3):634–658, 2009.

[107] Bo Qian and Khaled Rasheed. Hurst exponent and financial market predictability. In *IASTED conference on Financial Engineering and Applications*, pages 203–209. Proceedings of the IASTED International Conference Cambridge, MA, 2004.

[108] Roberto Reno. Nonparametric estimation of the diffusion coefficient of stochastic volatility models. *Econometric Theory*, 24(5):1174–1206, 2008.

[109] Gabriel Rilling, Patrick Flandrin, Paulo Goncalves, et al. On empirical mode decomposition and its algorithms. In *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, volume 3, pages 8–11, 2003.

[110] L Chris G Rogers. Arbitrage with fractional brownian motion. *Mathematical Finance*, 7(1):95–105, 1997.

[111] Manoj Kumar Saxena, SDVS Jagannadha Raju, Rajesh Arya, Ram Bilas Pachori, SVG Ravindranath, Sanjay Kher, and Shrikant M Oak. Empirical mode decomposition-based detection of bend-induced error and its correction in a raman optical fiber distributed temperature sensor. *IEEE Sensors Journal*, 16(5):1243–1252, 2015.

[112] Ling Tang, Wei Dai, Lean Yu, and Shouyang Wang. A novel CEEMD-based EELM ensemble learning paradigm for crude oil price forecasting. *International Journal of Information Technology & Decision Making*, 14(01):141–169, 2015.

[113] Edward C Titchmarsh. *Introduction to the theory of Fourier integrals*. Clarendon Press, 1948.

[114] Aviral Kumar Tiwari, Ibrahim Dolapo Raheem, and Sang Hoon Kang. Time-varying dynamic conditional correlation between stock and cryptocurrency markets using the copula-adcc-egarch model. *Physica A: Statistical Mechanics and its Applications*, 535:122295, 2019.

[115] Spilios Tzouras, Christoforos Anagnostopoulos, and Emma McCoy. Financial time series modeling using the hurst exponent. *Physica A: Statistical Mechanics and its Applications*, 425:50–68, 2015.

[116] Vladimir Vapnik. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.

[117] Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2013.

[118] Jie Wang and Jun Wang. Forecasting stochastic neural network based on financial empirical mode decomposition. *Neural Networks*, 90:8–20, 2017.

[119] Rafał Weron. Estimating long-range dependence: finite sample properties and confidence intervals. *Physica A: Statistical Mechanics and its Applications*, 312(1-2):285–299, 2002.

[120] Zhaohua Wu and Norden 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*, 460(2046):1597–1611, 2004.

[121] Zhaohua Wu and Norden E Huang. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in Adaptive Data Analysis*, 1(01):1–41, 2009.

[122] Muhammad Yahya, Atle Oglend, and Roy Endré Dahl. Temporal and spectral dependence between crude oil and agricultural commodities: A wavelet-based copula approach. *Energy Economics*, 80:277–296, 2019.

[123] Albert C Yang, Chung-Kang Peng, and Norden E Huang. Causal decomposition in the mutual causation system. *Nature communications*, 9(1):1–10, 2018.

[124] Jia-Rong Yeh, Jiann-Shing Shieh, and Norden E Huang. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Advances in Adaptive Data Analysis*, 2(02):135–156, 2010.

[125] Jize Zhang, Tim Leung, and Aleksandr Aravkin. Sparse mean-reverting portfolios via penalized likelihood optimization. *Automatica*, 111:108651, 2020.

[126] Lan Zhang, Per A Mykland, and Yacine Aït-Sahalia. A tale of two time scales: Determining integrated volatility with noisy high-frequency data. *Journal of the American Statistical Association*, 100(472):1394–1411, 2005.