Commodity Futures Market Prices: Decomposition Approach

by

ANTWI EMMANUEL (16023590)

submitted in fulfillment of the
requirements for the degree of
Doctor of Philosophy (Statistics)
in the
Faculty of Science, Engineering and Agriculture
Department of Mathematical and Computational Sciences

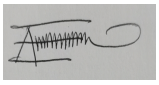
at the
UNIVERSITY OF VENDA

Supervisor: Prof. Kyei K. A.
Co-supervisor: Prof. Gyamfi E. N.
Co-supervisor: Prof. Gill R.

Submitted on July 2023

Declaration of Authorship

I, Emmanuel Antwi, hereby declare that the thesis for the Doctor of Philosophy (PhD - Statistics) degree at the University of Venda, hereby submitted by me, has not previously been submitted for a degree at this or any other university, and that it is my own work in design and execution and that all reference material contained therein has been duly acknowledged. The university is authorised to make copies of a thesis and to distribute such copies as it deems fit.

Signed (Student):  Date: 05/07/203

Abstract

Financial investments on commodity markets have attracted many investigations due to its importance to the global economy, and worldwide trade as a whole. The radical price changes in commodity market prices, especially agricultural, energy and industrial metal products have significant consequences on consumers and producers of economic activities. It is very crucial to accurately estimate and predict volatility in commodity futures market prices, since continuous price fluctuations have dire consequences for investors, portfolio managers, dealers and policymakers in taking prudent and sustainable decisions. Commodity price component determination and forecasting are challenging due to remarkable price volatility, uncertainty, and complexity in the futures market. As a result, commodity futures price series is nonlinear and nonstationary. Various studies are reported in the literature, in an attempt to develop models to study the persistent changes in the commodity futures price series, but these models have failed to account for the inherent complexity in the commodity futures price series. This study aims to use decomposition techniques, combined with back-propagation neural network (BPNN) and autoregressive integrated moving average (ARIMA) models to address difficulties in studying commodity futures market prices.

As said earlier, this study utilized the decomposition methods, Empirical Mode Decomposition (EMD) and Variational Mode Decomposition (VMD), to analyze the daily real price series of three commodity futures market prices of: corn from agricultural products, crude oil from energy, and gold from industrial metal, using the data from 4th May 2016 to 30th April 2021.

In the first part of the study, we explored the descriptive and statistical properties of the data. It was found that the three commodities market futures prices series were nonstationary and nonlinear. Subsequently, we performed an EMD-Granger causality test to establish the spillover effects among the three commodities' markets. It was revealed that there exists a strong mutual relationship among the three commodity markets price series, which implies that the price movement of one market can be used to explain the price fluctuations of the other markets.

In the second part, the EMD and VMD methods were applied to decompose the daily data of each commodity price from different periods and frequencies to their respective individual intrinsic mode functions. First, we used the Hierarchical Clustering Method and Euclidean Distance Approach to classify the IMFs, residue, and modes into high-frequency, low-frequency, and trend. Next, applying statistical measures, particularly, the Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation coefficient, we observed that the trend and low-frequency parts of the market prices are the main drivers of commodity futures markets prices' fluctuations and that special events caused the low frequency. In essence, commodity futures prices are affected by economic development rather than short-lived market variations caused by ordinary supply-demand disequilibrium.

The third part compared the EMD and the VMD- based models using three forecasting performance evaluation criteria and statistical measures, such as, mean absolute error (MAE), root mean square error (RMSE), and mean percentage error (MAPE) to compare the capabilities of the suggested models. We also introduced Diebold Mariano (DM) test in selecting the optimal models for each commodity, since

MAE, RMSE and MAPE have some shortcomings. The combined models outperformed the individual back propagation neural network (BPNN) and autoregressive integrated moving average (ARIMA) models in forecasting the series of corn and crude oil's futures prices. At the same time, BPNN emerged as the optimal model for predicting gold futures prices' series. In addition, variational mode decomposition emerged as the ideal data pre-treatment method and contributed to enhancing the predicting ability of the BPNN and the ARIMA models. The empirical results showed that models combined with decomposition methods predict commodity futures prices accurately and can easily capture the volatility in commodity futures prices.

By utilizing the decomposition-based models in studying commodity market prices, the study filled the following gap in the existing literature as follows: the pre-treatment effect of the EMD and VMD can be compared horizontally, in decomposing commodity market price series and studying the underlying components that cause the above mentioned commodity markets price fluctuations is a novel approach in studying commodity market prices. In addition, utilizing Hierarchical Clustering and Euclidean Distance Approaches, the IMFs, residue and modes were classified into their distinctive frequencies, namely, high-frequency, low-frequency, and trend units. The effect of these frequencies and trends on commodity market price fluctuation is the first of its kind in the literature. Furthermore, applying statistical measures such as Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation coefficient to evaluate the contribution of the IMFs, residue, and modes to the net variance of the volatility of crude oil, corn, and gold markets price fluctuations, is an innovative approach to studying financial times series. The EMD-Causality technique proposed to study the causal relationship of corn, crude oil, and gold futures prices movement, is novel in the financial market. This new approach to study price movement of commodity markets, will provide a vital information about one commodity market to explain the other commodity market price fluctuations in various markets. Also, Decomposition of financial data before forecasting have high forecasting precision accuracy in commodity futures price prediction. Additionally, using decomposition techniques in agriculture, energy, and industrial metal commodities futures markets, effectively, minimizes the prediction complexity. Furthermore, using econometric and machine learner models incorporated with decomposition methods can capture the price series information up to acceptable degrees. Finally, decomposition-based predicting techniques can effectively raise the predicting performance capability of BPNN and ARIMA models and reduce errors, thus, the proposed novel combination method can statistically improve forecast accuracy. This study, therefore, may assist in arresting the agricultural, energy, and industrial commodities markets trends and estimate volatility risk factors accurately, consequently serving as a guide for investors, governments policymakers and related sectors such as agriculture, energy, and metal industry to take prudent and sustainable planning and investment decisions.

The suggested decomposition strategy, particularly VMD-based is robust in analyzing the determinants, modeling, and forecasting commodity futures market prices fluctuations, thereby, improving forecasting precision accuracy. Remarkably, in using the decomposition approach in estimating compositions of commodity prices data series separately, different predicting strategies can be explored. For instance, based

on the features of decomposed IMFs or modes, a suitable predicting technique can be considered to forecast each IMF or mode; for example, the residue can be estimated by utilizing a polynomial function, while Fourier transform can be considered in predicting low-frequency IMFs or modes, hence, it is recommended that researchers, institutions, investors, and policymakers interested in studying commodity price movements should consider using this novel technique to achieve better results. It is further suggested that the decomposition approach could be utilized in other fields of study to prove the approach's generality.

Finally, further study can extend the proposed methodology by considering other decompositions techniques rather than just EMD and VMD and evaluate their robustness in studying financial markets, as EMD approach has the problem of mode mixing and endpoint effects. Eventually, we propose that a new model or consolidated predicting technique should be investigated to cater for special events' influences on commodity market prices since no one can predict the time and the place they will occur.

Keywords: *Empirical Mode Decomposition, Commodity price, Forecasting, Granger, Euclidean Distance, Hierarchical Clustering*.

Acknowledgements

I would like to express my heartfelt gratitude to Almighty God for His blessings and knowledge given me to go through this research successfully. This thesis would not have been possible if it were not for the tireless guidance and support I got from my supervisors - Prof. K. A. Kyei, Prof. E.N.Gyamfi and Prof. R. Gill, I say 'thank you' for your professional directions. Also, I am very grateful to Dr. Phyllis Kaburise for taking her time to edit this research. I would like to also show my gratitude to all anonymous reviewers of the journals that published the chapters for their constructive criticisms, which assisted in molding up this thesis.

To all other members of the Department of Mathematical and Computational Sciences, I owe you my deepest gratitude for your support and assistance; you stood by me every step of the way, during the period of thesis write-up. I am also indebted to my colleagues; your presence provided a source of warmth and gave me a source of hope in the challenging and difficult periods throughout the research. I would like to express my sincere gratitude to Derrick Owusu Asamoah for his unwavering support and encouragement. Sincere appreciation is also extended to all my friends, especially Prof. Anokye Mohammed Adam, Jerry Boano-Danquah and Lawrence Opoku Kwasi Nkansah.

Autobiography

EMMANUEL ANTWI has -

- a certificate in Education (Mathematics and Science) from University of Cape Coast, Ghana;
- a Bachelor of Science Honors degree in Mathematics;
- a Master of Business Administration (MBA-Logistics and Supply Chain Management) from Kwame Nkrumah University of Science and Technology, Kumasi, Ghana,
- a Master of Science degree in Statistics from the University of Venda, South Africa.

He is an educator at Department of Education, Limpopo, South Africa. Emmanuel's study interests are - applied financial econometrics and financial markets. He is affiliated to the South African Statistical Association (SASA), American Statistical Association (ASA), and is a member of the South African Democratic Teachers Union (SADTU). Has published four articles in the accredited international journals including the following two which were previously published from master's work:

- Antwi E., Gyamfi E. N., & Kyei K. A. (2019): Modeling and forecasting Ghana's inflation rate under threshold models. Journal of Developing Areas, Vol.53, No: 3
- Kyei K. A. Antwi E. (2019): Determining the level and trend of fertility in four provinces in South Africa: International Conference on Social Science and Economics (ICSSE). Paper code: II-SSEMONT-270220-17310.

Reviewer

Currently appointed as a reviewer to accredited international journals, Resources Policy, Institute of Electrical and Electronics Engineers (IEEE Access), Fluctuation and Noise Letters, and Mathematical Problems in Engineering (Hindawi) and has since reviewed the following articles:

- Manuscript Number: JRPO-D-23-00081 Natural Resources Extraction and Financial Inclusion: Linear and Non-Linear Effect of Natural Resources on Financial Sector.
- Manuscript Number: JRPO-D-22-00865R1 Booms in commodities price: assessing disorder and similarity over economic cycles.
- Manuscript Number: JRPO-D-22-01303 Examining the Role of Sustainability and Natural Resources Management in Improving Environmental Quality
- Manuscript Number JRPO-D-22-01829 Performance of Different Models in Iron Ore Price prediction during the time of commodity price spike.
- Manuscript ID: Access-2022-32551: Predicting Chinese Commodity Futures Price: An EEMDHurst-LSTM Hybrid Approach
- Manuscript ID: Access-2022-34356: Attentional Multi-Channel Convolution with Bidirectional LSTM Cell toward Hate Speech Prediction
- Manuscript Number: FNL-D-22-00204: A Hybrid Model of Primary Ensemble Empirical Mode Decomposition and Quantum Neural Network in Financial Time Series Prediction
- Manusript ID: 6300740: A novel hybrid model for stock index price forecasting.

Summary

The thesis is based on the following published articles in accredited journals. These have made original contribution to the modeling of financial (commodity future market price fluctuations). A summary of the specific contributions can be found in Chapter seven of the thesis.

- E. Antwi, E. N. Gyamfi, K. A. Kyei, R. Gill and A. M. Adam,(2022) "Modeling and Forecasting Commodity Futures Prices: Decomposition Approach," in IEEE Access, vol. 10, pp. 27484-27503, 2022, doi: 10.1109/ACCESS.2022.3152694.
- Antwi E, Gyamfi E. N, Kyei K, Gill R, Adam A. M. (2021) Determinants of Commodity Futures Prices: Decomposition Approach. Mathematical Problems in Engineering, 2021, Oct. 4, 2021, Volume 2021 — Article ID 6032325 — https://doi.org/10.1155/2021/6032325.

Price fluctuation is a paramount concern since commodity prices are associated with livelihood and the economy of a nation; the significant impact of any price fluctuations in the futures market, show that

forecasts in commodities are an essential venture. The difficulties in predicting commodity prices are due to the unpredictability of the world's financial issues, fiscal dispensation, the speculative markets exacerbate, and several other elements. As a result, commodity market prices are nonstationary and nonlinear. This thesis aimed to determine a model and forecast the market price of commodity futures. We applied decomposition techniques, empirical mode decomposition (EMD), and variational mode decomposition (VMD) to three commodities: corn, crude oil, and gold, over the commodity spot market prices.

In Chapter One, the general overview of the thesis, the underlying factors that cause commodity market price series fluctuations and how it could be modeled accurately were considered. The importance of modeling these financial data in relation to the global market was detailed. The background, the rationale of the study, problem statement, objectives, expected outcomes, the scope of the study, and the research questions of the thesis were also discussed.

Chapter Two presented the literature reviews of empirical and theoretical studies related to this work. The various forms of models were debated upon in an attempt to solve the problem of non-linearity in financial data. Introduction of the EMD and VMD concepts and their effectiveness in handling non-linear and non-stationary dynamics in financial data generation were also investigated.

Chapter Three introduced the theoretical background of the suggested methodology, the EMD and VMD in studying the commodity market futures prices. Back-propagation neural network (BPNN) and autoregressive integrated moving average (ARIMA) were also introduced, which served as benchmark and comparative models, respectively.

In chapter Four, a descriptive analysis and the statistical properties of the three commodities markets were examined. It was observed that corn and gold price series were skewed to the right, hence these two commodities market prices have right tail. On the other hand, the crude oil price series were skewed to the left, and that means, the price series of crude oil distribution was left tail. It was also revealed that the daily spot prices of the three commodities were leptokurtic. The Jarque-Bera test revealed that these three commodities price series did not follow a normal distribution. We also tested for the non-linearity and non- stationarity of corn, crude oil, and gold price series using Keenan, Tsay, and ADF tests. The ADF test results revealed that the corn, crude oil, and gold price series were not stationary while the results of Keenan and Tsay tests on corn, crude oil, and gold markets prices were nonlinear. Finally, the EMD-Granger Causality analysis was introduced to examine the causal relationships among the three commodities markets' prices. The results showed strong mutual relationships among the three commodities markets. This was why crude oil market prices have persistently affected the agricultural and industrial-based metal commodities' markets.

Chapter Five is the analysis of the primary components that cause crude oil, corn, and gold futures markets prices' fluctuations using proposed EMD and VMD strategies. The EMD and VMD approaches were employed to disintegrate day-to-day closing prices of corn, crude, and gold from the Bloomberg commodity index, which span from May 2016-April 2021 into their distinctive IMFs or modes and a residue to discover the elements that influence commodity futures market prices' variations. The Hierarchical

Clustering Method and Euclidean Distance Technique were employed into categorizing the IMFs and modes into their distinctive frequencies-high-frequency, low-frequency, and trend. Subsequently, statistical measures, namely, the Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation coefficient were applied to each IMF, mode, and residue to access the correlation between the IMFs and the actual data. We also evaluated the variance of each IMF, mode, and residue contribution to the total volatility of each commodity market future price variation. It was revealed that the trend and low-frequency components are the main factors responsible for the commodity futures markets prices' variations; in addition, special events cause low frequencies. Overall, commodity market futures prices are influenced by economic growth rather than short-lived market fluctuations originating from normal supply-demand imbalances.

In Chapter Six, we presented two decomposition techniques (EMD and VMD methods), combined with BPNN and ARIMA, to forecast the futures prices of the three types of commodities mentioned earlier selected across the commodities markets. The EMD and VMD were utilized to decompose the three commodities futures prices series into IMFs, residue and modes to predict commodity futures prices. To improve the forecasting ability and accuracy of BPNN and ARIMA, we compared the forecasting capability of the techniques using the three predicting performance rating criteria - MAE, RMSE, and MAPE. We also performed the Diebold Mariano test to select the optimal model since the MAE, RMSE, and MAPE have specific limitations. The combined models proposed outperformed the standard models; BPNN and ARIMA, in predicting futures prices series of corn and crude oil but failed in forecasting the gold futures prices' series. The forecasting performance of combined models methods is more acceptable performance than the ARIMA model in every instance of this study. The forecasting capability of the VMD-BPNN model is more robust than the EMD-BPNN model, the EMD-ARIMA model, and the VMD-ARIMA model, suggesting that the VMD method is more suitable for the prices data pre-treatment since the process raises the forecasting precision.

Chapter Seven presented the general conclusions, contributions, recommendation, limitations and suggestions for further research. It was found that the VMD-ARIMA model emerged as the optimal model for predicting corn future price series, while the EMD-ARIMA model was the best predicting model for crude oil future price series. The decomposition-based models failed to improve the forecasting ability of gold future price series. The BPNN model outperformed all model combinations suggested for forecasting gold future price series. Using a decomposition-based model method, market participants, government, policymakers, and businesses, therefore, can make sustainable decisions to reduce loss and increase profit.

Decomposition-based techniques can effectively improve the forecasting performance ability of BPNN and ARIMA models and reduces errors, Thus, the proposed novel combination method can statistically improve forecast accuracy and robustness. This study, therefore, may assist to arrest the agricultural, energy, and industrial commodities markets trends and estimate volatility risk factors accurately, consequently serving as a guide for investors, governments, policymakers and related sectors such as agricultural, energy, and industrial sectors to take prudent and sustainable planning and investment decisions.

Contents

List of Figures

List of Tables

Abbreviations

The following table describes the significance of various abbreviations and acronyms used throughout the thesis. The page on which each one is defined or first used is also given. Nonstandard acronyms that are used in some places to abbreviate the names of certain white matter structure are not included in this list.

Abbreviation	Meaning
ADMM	Alternate direction method of multipliers
MI	Machine intelligence
ARMA	Autoregressive moving average
BPNN	Back-propagation neural network
COMEX	Commodity Exchange Inc
EEMD	Ensemble empirical mode decomposition
EMD	Empirical mode decomposition
EWT	Empirical wavelet transform
GA	Genetic algorithm
GDP	Gross domestic product
IMF	Intrinsic mode functions
LSSVR	Least squares support vector regression
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MRS	Markov regime-switching
MS-GARCH	Markov switching generalized autoregressive conditional heteroskedasticity
MSR	Markov switching regressive
NN	Neural network

Abbreviation	Meaning
RBFNN	Radial basis function neural network
SSA	Singular spectrum analysis
SST	Synchrosqueezed transform
SVR	Support vector regression
VMD	Variational mode decomposition
VAR	Vector autoregressive
JB	Jarque-Bera
Skew	Skewdeness
Min	Minimum
Max	Maximum
Sd	Standard deviation
Kurt	Kurtosis
DM	Diebold Mariano
DM-AE	Diebold Mariano test based on absolute error loss
DM-SE	Diebold Mariano test based on square error loss
DM-PE	Diebold Mariano test based on percentage error loss
SE	Square-error
AE	Absolute-error
PE	Percentage error
EMH	Efficient market hypothesis
ARCH	Autoregressive Conditional Heteroscedasticity
OPEC	Organization of the Petroleum Exporting Countries
DEMD	Differential empirical mode decomposition
DWT	Discrete wavelet transform
MW	Moving window
ICA	Independent component analysis
GRUNN	Gate recurrent unit neural network
ADF	Augmented Dickey-Fuller
IEA	International Energy Agency
M	Mode
Abbreviation	Meaning
ADF	Augmented Dickey-Fuller
ACF	Autocorrelation Function
AIC	Akaike Information Creterion

Abbreviation	Meaning
ARIMA	Autoregressive Integrated Moving Average
AR	Autoregressive
BET	Bucharest Stock Exchange
BoG	Bank of Ghana
BIC	Bayesian Information Creterion
BVAR	Bayesian Vector Auto Regressive
COICOP	classification of individual consumption by purpose
CPI	Consumer Price Index10
DIS	Databank Stock Index
DJ-AIGCI	Dow Jones AIG Commodity Index
DJIA	Dow Jones Industrial Average
EGARCH	Exponential Generalized Auto Regressive Conditional Heteroscedastic
ERP	Economic Recovery Programme
EWMA	Risk Metric Exponential Weighted Moving Average
GARCH	Generalized Auto Regressive Conditional Heteroscedastic
GDP	Gross Domestic Product
GJR-GARCH	G Glosten-Jagannathan Runkle Generalized Auto Regressive Conditional Heteroscedastic
GSE	Ghana Stock Exchange
GSS	Ghana Statistical Service
GSSNL	Ghana Statistical Service News Letter
IGARCH	Integrated Generalized Auto Regressive Conditional Heteroscedastic
i.i.d.	independent and identically distributed
IT	Inflation Targeting
LSTAR	Logistics Smooth Threshold Autoregressive
MA	Moving Average
MAPE	Mean Absolute Error
MOFEP	Ministry of Finance and Economic Planning
MSE	Mean Square Error
MT	Monetary Targeting
PACF	Partial Autocorrelation Function

Abbreviation	Meaning
PARCH	Power Auto Regressive Conditional Heteroscedastic
RW	random walk
SAP	structural Adjustment Programme
SAR	Simple Auto Regressive
SARIMA	Seasonal Auto Regressive Integrated Moving Average
SETAR	Selt-Exciting Threshold Autoregressive
S and P	Standard and Poors
SVAR	structural Vector Auto Regressive
VAR	Vector Auto Regressive
VAT	Value Added Tax
QTM	Quantum Theory of Money

Dedication

This work is dedicated to my lovely wife, Olivia, my children, Rachel, Samuel and Emmanuel junior, my sweet mother, Akua Badu and also to my siblings, Adu, Mary, Ernest, Patrick and Samuel for their love, care and support.

Chapter 1

General introduction

Chapter summary

This chapter provides a general overview of the thesis, which focused on the underlying factors that cause commodity market prices series' fluctuations, how the process can be modeled accurately and the necessity for modeling these financial data for the global market. The background, the rationale of the study, problem statement, objectives, expected outcomes, the scope of the study, and the research questions of the thesis are also discussed.

1.1 Background information

Globally, crude oil, corn, and gold futures prices have great influence on countries' economy. These commodities are prone to price fluctuations, and have decisive impact on international economic activities, and for both consumers and producers (Zhang et al., 2018, Wang and Wei, 2021). Price fluctuation is a primary concern, as investors anticipate making satisfactory profit from their investments in the commodity futures market exchange; this means that studies on the prices of crude oil, corn and gold can supply macroeconomic information to investors so as to minimize the risks of loss and maximize profit. The futures market price determinants of these commodities are not affected by demand-supply only, but also influenced by several components, such as financial policies, nature of the economy, exacerbation of market speculation and special events such as, eruption of war, outbreak of pandemics, the September 11, 2001 gruesome attacks in the USA, and the 2008 and 2012 global debt crises) (Zhao et al., 2016). The noticeable price volatility in the futures market of crude oil, corn, and gold makes it difficult to develop

models to analyze the price movements of the aforementioned commodities. Prudent economic and investment decision-making, which can assist investors and policymakers to reduce the danger posed by the ongoing price fluctuations, therefore, can only be achieved by accurately analyzing the futures market.

The United States is the leading consumer of corn in the world. In 2019/2020, the United States consumed about 12.30 billion bushels of corn, followed by China which consumed about 10.98 billion bushels of corn in the same year. The European Union is the third largest consumer of corn globally (Zou et al., 2007). These countries are regarded as the world's leading economies, which implies that this agricultural commodity plays a significant role in the daily planning of the economy of these three countries. According to the International Energy Agency (IEA) 2019 report, United States and China are also the leading consumers of crude oil worldwide, at about 19.4 million barrels per day and 14 million barrels per day, respectively. Gold as a precious metal is used as an indicator to hedge against inflation and measure wealth in the world. It is a metal which fascinates everyone and has maintain its value over centuries. Unlike crude oil, gold price is not determined by any organization, but mostly depends on the cost of extraction and how much people are willing to pay for it. Gold is expensive due to the high cost associated with its production; it is exceedingly popular and can maintain constant weight all the time, which makes it price to rise more often. As stated by the futures price discovery mechanism, these commodities are extremely affected by macroeconomic policies, therefore, they are useful in providing price information to spot markets. Discovering the drivers of these commodities, hence, is expected not only to minimize the unreliability and decrease the danger in commodity markets, but can also help to make strategic and reliable blueprints for government.

Crude oil, corn and gold were chosen across the commodity price market to conduct this study, which aimed at identifying the underlying components that drive the futures prices fluctuations in energy, agriculture and industrial metal, because these commodities are regarded as the most volatile, interactive and complex in the commodity price market and very sensitive to macroeconomic policies (Zou et al., 2007). These commodities play a vital role in the world economy, hence, the futures prices of these commodities market price data are very important for any future development plan, since there is strong mutual relationship between price, supply and demand.

A commodity market is a place where primary products are traded-off. Unlike an equity market, commodity market deals with buying and selling of commodities.

It is on the market that many products, such as hard commodities (like metals and energy), and soft commodities (like agriculture and livestock) are exchanged. It is crucial, therefore, to create a robust, functional, and liquid commodity market for the smooth running of any economy (Kaur and Rao, 2010). This would enable investors to hedge their commodities' risks, use speculative conditions in commodities,

and utilize arbitrage changes in the market (Arendas, 2017). There are several participants in the commodity markets. The major ones include - commodity market producers/consumers, processors, traders, financial institutions/dealers, and investors - however, the set-up of the commodity market participants and their activities are more complicated than in other asset markets (Ali and Gupta, 2011).

According to the efficiency market hypothesis (EMH), no market participant has full control in forecasting a profit on a stock price, since everyone has access to the same information (Fama and French, 2020). Market efficiency provides a platform for explaining price behavior in financial bonds and demonstrates the capability to process timely and quality information. A market is efficient, if the asset prices give all the necessary information (Fama, 1965). In other words, in an inefficient market, asset prices are imprecise with regards to present market information; this makes it difficult for market participants to make above normal returns on their investments.

Market efficiency may be classified as either - weak, semi-strong, or strong (Fama, 1970). A market efficiency is said to be in weak form, if the present market prices give all the necessary information in historical prices. Inefficient markets as the weak form can be easily predicted and investors can use indicators to analyze the market, thereby, make above-normal profits (Arora et al., 2017). Semi-strong market efficiency is one in which present market prices reflect past prices and available information, openly (Aktan et al., 2018). In this type of market, participants cannot use the available information to predict the market and make supernormal returns. Strong-form market efficiency combines all published, known information, significant information not yet published, such as insider information in analyzing the market (Aktan et al., 2018). It is difficult for investors to predict this type of market; even insiders cannot secure supernormal returns.

In inefficient commodity futures markets, the information that the present futures prices deliver on spot prices in the future is efficient, which makes it difficult for participants to gain more profit utilizing hedging approaches (Chhabra and Gupta, 2020).

Commodity prices are complex, uncertain, volatile and independent of each other. As a result, the price series of commodity markets successively keeps on fluctuating and does not follow a random behavior, that is, they are non-linear and non-stationary (Malkiel, 2003).

The non-linearity and non-stationarity features in a financial data, like the commodity market price series make the linear modeling approach unsuitable.

Many empirical studies have suggested models, in an attempt to account for the complexity in financial data. These models include - singular spectrum analysis (SSA) Wang and Li (2018), wavelet Boubaker

and Raza (2017), vector autoregressive (VAR) Maghyereh (2006), autoregressive integrated moving average (ARIMA) Salisu and Oloko (2015), Markov regime-switching (MRS) Zhang et al., (2015), neural network Kristjanpoller and Minutolo (2015), support vector machine (SVM) Zhang et al., (2018), and genetic algorithm (GA) Motlaghi et al., (2008). Due to various setbacks of each of these approaches in dealing with unstable behavior in financial data, they turn to produce false results. For instance, singular spectrum analysis (SSA) and wavelet methods are inefficient in studying noisy, nonlinear, nonstationary, and periodic times series. Additionally, these techniques need a presumptive selection, which lacks economic interpretations. The conventional methods such as ARIMA, SVM, VAR and MRS fail to explicitly bring out the embedded components in the data structures (Bacon, 1991).

Despite these limitations mentioned above, these approaches have been largely used in studying financial data, such as commodity futures market prices series, which, however, leads to biased predictions and affects the validity of the analysis and its conclusions.

Inspired by the aforementioned reasons, we proposed an enhanced signal detector, EMD and VMD techniques, to decompose price series of crude oil, corn, and gold to establish the components that drive commodity futures spot markets prices. This study, therefore, utilizes EMD and VMD to analyze the components of futures prices of corn (from agricultural commodities), crude oil (from energies), and gold (from industrial metals).

1.2 Problem Statement

With the current global economic uncertainties, the recent Covid-19 pandemic, the ongoing Russian-Ukraine invasion, and exchange rate market fluctuations, commodity prices have become more volatile than ever. This has made some of the existing methods in predicting commodity market prices unreliable, therefore, there is a need to modify the existing approaches in literature in forecasting commodity market prices, since accurate volatility forecasting have tremendous impact on investors, policy-makers, and the economy in general. Accurate forecasts, hence, enable investors, policy makers, and traders to take prudent and informed decisions about their investments.

In an attempt to solve the problem of non-linearity and non-stationary characteristics in financial data, Huang et al., (1998) proposed the Empirical Mode Decomposition (EMD) method which has attracted much research attention, lately. EMD addresses full frequency content that a single Hilbert transform cannot explain, since several hidden modes may be uncovered when analyzing financial data within its time domain or bulk state. EMD, as a data pre-processing tool, can bring out inherent modes from the actual data and can present each embedded mode in the data as an intrinsic mode function (IMF) (Wang

et al., 2012). In addition, it differs from the other decomposition methods because the process depends on the distinct characteristic time scale of the data and allows only extrema in the sifting process, therefore, it is robust, highly efficient, and adaptive (Addison et al., 2009). EMD requires no presumptive base functions' selection, but despite these merits, EMD encounters limitations, such as extreme points effect and mode-mixing, which affect the analysis of the extracted IMFs from the signal.

Another class of decomposition method was introduced as an improvement on EMD to overcome the problems of extreme points effect and mode-mixing. The variational mode decomposition (VMD) was suggested by Dragomiretskiy and Zosso (2013) to address the limitations of the EMD method. It is a non-recursive and bandwidth-limited maximization decomposition technique that uses the Wiener filtering and Hilbert transform. VMD is capable of decomposing time series data for any desirable modes. The technique assumes that each mode has a center frequency which is a limited bandwidth Dragomiretskiy and Zosso (2013), therefore, each mode decomposed by the VMD method can be compressed around a "pulsation", which is determined along with the decomposition process. In addition, it is good for decomposing non-linear, non-stationary, and noise-sensitive data, thus, it is more robust than the EMD approach.

It can be seen clearly that the EMD and VMD techniques can provide better analysis and meaning in studying financial time series, however, few empirical studies have considered these novel approaches in analyzing commodity futures markets. With this background, this thesis seeks to evaluate the ability of this class of decomposition methods in modeling and forecasting futures commodities markets' prices of corn, crude oil, and gold, selected across commodity markets.

1.3 Rationale of the Study

Commodity markets as a type of financial investment have attracted many investigations due to its importance to the worldwide economy and global trade. The drastic price fluctuations of commodity market prices, particularly, agricultural, energy, and industrial metal products, have a decisive impact on consumers and producers of economic activities (Zhang et al., 2018, Wang and Wei, 2021). As a result, it is crucial to accurately estimate and predict volatility in commodity futures market prices, since continuous price fluctuations have serious negative effects on investors, portfolio management, dealers, and policy-makers' ability to take prudent and sustainable decisions. To better understand commodity market price movements, this thesis used decomposition-based models, EMD and VMD combined with back-propagation neural network (BPNN) and autoregressive integrated moving average (ARIMA) to disclose the embedded dynamics in the commodity futures market.

1.4 Objectives of the Study

This thesis aimed to model commodity futures market prices' volatility, using decomposition-based approach, thereby, to examine their effectiveness and superiority in predicting commodity prices. In addition, the study aimed at identifying the drivers of commodity market price fluctuations and predict a one day out-of-sample forecast of corn, crude oil, and gold futures market prices, utilizing decomposition-based forecasting models.

1.5 Broad objective of the study

The following broad objectives were taken into consideration:

1. To evaluate the ability of a category of empirical mode decomposition techniques in decomposing commodity futures market prices.
2. To assess the performance of decomposition-based models in forecasting commodity futures market prices.

1.6 Specific Objectives of the Study

The study examined the following specific objectives:

1. To examine the causal relationships among three commodities - corn, crude oil, and gold markets.
2. To compare EMD and VMD methods horizontally in examining the fundamental components of the commodity futures market that causes price fluctuations.
3. To forecast one-day ahead out-of-sample commodity futures markets prices of corn, crude oil, and gold, using decomposition-based models.

1.7 Expected contributions from the study

The study, it is anticipated, would augment the existing literature on - commodity futures markets prices' determinants, market price volatility predictions, and relationships among commodity markets. In addition, this study aspires to strengthen empirical studies that support decomposition and hybrid modeling of

commodity futures markets prices and provide a comprehensive analysis of intrinsic dynamics in commodity futures markets' prices. Furthermore, the study intends to serve as an alternative method for modeling and predicting commodity futures markets prices fluctuations and help commodity market participants and policymakers make informed decisions.

1.8 Scope of the study

The conceptual framework of the study was a combined one, with the intention of establishing a framework for empirical decomposition modeling of commodity futures markets' prices. The model's strength rests on the choice of the decomposition technique, in this instance, VMD.

The data used in the study was limited to three selected daily-closing spot prices across commodity markets, from May 2016 to April 2021. This time frame was due to data unavailability and inconsistency of other commodities. The selection of these commodities, hence, was based on the accessibility of consistent and reliable data, for efficient analysis.

1.9 Research questions

The proposed study sought to answer the following questions:

1. What causal relationships exist among corn, crude oil, and gold markets prices?
2. Which decomposition methods effectively decompose commodity futures markets prices data accurately?
3. Which models best predict commodity market futures prices accurately?
4. What are the similarities among the corn, crude oil, and gold futures markets' prices?

1.10 Conclusion

This chapter presented the importance of the commodity market to investors, portfolio managers, policymakers', and the global economy in general. A brief discussion of efficient and inefficient commodity markets were discussed, while the uniqueness of the proposed method to study financial data was highlighted. In the deliberations, the 'road map' of the thesis was established. The justification of the study,

problem statement, research objectives, (comprehensive and specific), the scope of the study, possible contributions of the study to the existing research, and research questions were also detailed.

Chapter 2

Literature Review

Chapter Summary

This chapter presents the literature reviews of empirical and theoretical studies related to this study. Reviewed the various forms of models developed to solve the problem of non-linearity in financial data. We discussed the EMD and VMD concepts and their effectiveness in handling non-linear and non-stationary dynamics in financial data generation.

2.1 Introduction

Globally, commodity markets are heterogeneous, undefined, volatile, and co-dependent, hence, commodity price data are asymmetric and non-stationary (Zhu et al., 2019). Corn, crude oil, and gold markets are considered as the largest market in the commodity markets' industry, consequently, discovering the volatility factors of these commodities markets is rigorous (Jin et al., 2021). Considerable amount of studies have looked into the commodity market prices and the common techniques applied can be categorized into two classes: data-driven techniques and structure-modeling approaches (Dees et al., 2007). Data-driven technique is made up of linear regression models such as - Autoregressive Moving Average (ARMA), Autoregressive Conditional Heteroscedasticity (ARCH) type models (Morana, 2001, Sadorsky, 2002); and nonlinear models such as Back Propagation Neural Network (BPNN), Radial Basis Function Neural Network (RBFNN) (Mirmirani and Li, 2004). The structure technique approaches outline market

prices and then schedule supply-demand equilibrium to access the commodity price fluctuation (Bacon, 1991).

The structure technique models have proved to be an effective approach in analyzing commodity market futures prices, but the approach is demanding in its application as a result of unique characteristics of the commodity futures market. In particular, oil supply prediction is challenging because it is distributed by two self-sufficient entities - the Non-OPEC and OPEC countries. The Non-OPEC nations have no influence on oil prices determination, while the OPEC nations control the price and production levels, applying different components aside storage level (Dees et al., 2007). Additionally, the unstable international commodity market environment is another contributor in modeling commodity market futures prices. The data-driven methods have demonstrated superiority in short-term forecasting of commodity market futures prices but lack economic analysis, as these methods fail to identify the underlying factors influencing commodity price fluctuation.

2.2 Empirical literature review of the study

A couple of studies in literature have reported on attempts to trying to develop methods to analyze the price instability in commodity futures spot market; include those proposed by (Zhu et al., 2015, Wang et al., 2017, Miao et al., 2017, Boubaker and Raza, 2017, Wang and Li, 2018). Wang and Li (2018) utilized singular spectrum analysis (SSA) and wavelet integrated with neural network models, namely, backpropagation neural network, radial basis function, and wavelet neural network, to estimate commodity futures market prices. The SSA and wavelet were employed to break down crude oil, corn, and gold prices' series. The results demonstrated that the neural network models integrated with SSA shows superior performance over the benchmark models.

In the quest to understand the spillover effects of oil price on the BRICS stock exchange markets, Boubaker and Raza (2017) proposed decomposition and model combination method, VARM-GARCH-DCC and proxy by mean and variance of oil prices at separate time horizons. They engaged wavelet technique to disintegrate oil price series at different time periods. The results from the study confirmed that the oil market price uncertainty has serious impact on oil prices and stock exchange market prices, although,they have absolutely no impact on the volatilities of other prices. It was revealed that the suggested VARM-GARCH-DCC integrated with wavelet decomposition, forecasts the oil price precisely, however, utilizing SSA and wavelet approaches as tools for data pre-processing has some disadvantages. SSA and wavelet methods have high tendency of extracting incorrect cycles as part of the series because the function basis

and the decomposition levels selections are based on individual assumptions; the wavelet technique is non-adaptive.

The Empirical Mode Decomposition (EMD) and Variational Mode Decomposition (VMD) are objective data analysis methods (Huang et al., 1998, Dragomiretskiy and Zosso, 2013). These two decomposition techniques can be utilized to overcome the difficulties in modeling and forecasting and they are devoid of wrongful economic explanations in the commodity market futures prices.

The EMD method is evidence-based, intuitive, direct, and robust data pre-processing approach proposed for pre-treating non-linear and non-stationary data, such as financial time series. The function of EMD is to decompose the actual data into a distinct number of intrinsic mode functions (IMFs) in such a way that the data is near periodic based on distinctive features called the "extrema", thus, the distance between two succeeding extrema, and the economic meaning and the interpretations depend on the IMF scale. For example, an IMF with a scale of three months in financial time series is regarded as a special event or seasonal event. Investigating data's components intrinsically is crucial to comprehend economic explanation because data is considered as the only link to real life situation.

Investigators have used EMD in studying geo-economic and financial data. The EMD has been utilized, for instance, in predicting agricultural products, electricity prices , exchange rates , gold prices, crude oil prices, and on carbon prices (Wu and Chen, 2007, Zhang et al., 2008, Jian-Hui and Wei, 2012, Lin et al., 2012, Premanode and Toumazou, 2013, An et al., 2013, Abadan and Shabri, 2014, Plakandaras et al., 2015, Hua and Jiang, 2015, Zhu et al., 2015, Xiong et al., 2015, Chen et al., 2016, Wang et al., 2017, Crosato et al., 2018, Meng et al., 2020).

Premanode and Toumazou (2013) applied differential empirical mode decomposition (DEMD) to enhance exchange rates forecasting employing support vector regression (SVR). They confirmed that when DEMD combined with the SVR model it produces robust results as compared to the benchmark models of Markov switching generalized autoregressive conditional heteroskedasticity (MS-GARCH) and Markov switching regressive (MSR).

In assessing the impact of economic sections on the ruble exchange rate in Ukraine, Korotin et al., (2019) applied EMD strategy and combined it with the Hurst exponent to study the ruble's market exchange rate. Based on the effective market theory it was affirmed that economic sanctions from international bodies have no direct effects on the ruble exchange rate, rather the Ukraine's foreign exchange market has a long memory.

Zhu et al., (2018) suggested a completely adaptive predict strategy to examine China's export policy. The EMD technique was adopted and integrated with the counterfactual model to evaluate the consequences

of China's export policy on the price of tin. Four different decomposition methods were employed in the study and the EMD method emerged as the best decomposition technique. The results further disclosed that there is no significant effects of China's export policy on the tin market price. Nevertheless, the tin supply increased globally as a result of the termination of export policy, which in turn caused a downward trend in the tin prices, in the long run; this also resulted in a short-term fluctuation in the international tin market, demonstrating that market policy and price are sensitive to trade policy.

In a similar work, Zhu et al., (2020) applied the EMD method to investigate the spillover effects between the carbon and electricity markets. EMD was applied to break down the carbon and electricity prices into their respective individual modes with different frequencies. This was followed by a decomposition model, conditional value at risk (CVaR), to identify the impacts of the spillover effects between the carbon and electricity market. The results indicated that there was positivity spillover effects of carbon market over the electricity market, but the electricity market has negative spillover effects on the carbon market. For high and low frequency modes, the carbon market has positive spillover effects on the electricity market and vice versa. For the case of intermediate frequency, there exists a negative bidirectional spillover effect between the two markets.

Likewise, Wei and Cheng (2012) utilized decomposition and ensemble technique model, the EMD-BPNN, to predict a short-term commuter movement in public transportation network using travellers' historical data; a short-lived commuters' data was decomposed into several IMF units using EMD. They reported that the decomposition and ensemble model, EMD-BPNN strategy outperforms the sole BPNN model, which did not undergo decomposition process.

Similarly, Ngo (2011) embraced EMD and wavelet decomposition approach to classify and forecast the arrival data of venture clusters and reported that the EMD is capable of extracting the embedded features within the non-linear data. Again, the EMD-based model emerged as the best predicting method over the wavelet-based models in terms of minimizing the forecasting errors.

In the same vein, Antwi et al., (2021) capitalized on the advantages of EMD and VMD techniques in investigating the drivers of price fluctuations in three commodities futures markets - corn, crude oil, and gold. EMD and VMD were utilized to disintegrate the price data of each commodity into distinctive IMFs and mode. They reported that the decomposition method could reveal the intrinsic factors that cause volatility in the commodity futures market prices.

Despite the advantages of using EMD as non-linear data pre-treatment tool, it is still has some shortcomings, such as mode-mixing and extreme points issue. To address these problems, Dragomiretskiy and Zosso (2013) formulated VMD. The VMD method is non-iterative optimization decomposition approach

that utilizes Wiener filtering and Hilbert transformation in grouping the extracted modes. It has the ability to decompose time series data to any acceptable modes. The method is premised on the fact that each and every mode extracted has a center frequency which is bandwidth-limited. The center frequency can reduce evaluated bandwidth of each decomposed mode to be equal to the original series; additionally, the method is suitable for breaking down non-linear and non-stationary time series data.

Recently, economist and financial researchers have used the VMD method in analyzing various commodity markets. For instance, in finding the factors affecting commodity futures prices' instabilities, Antwi et al., (2021) applied VMD to decompose corn, crude oil, and gold price data into individual modes. They adopted the Hierarchical Clustering and Euclidean Distance Approaches to group the extracted IMFs and modes into their respective frequencies, namely, high-frequency, low-frequency, and trend. This was followed by the application of statistical measures, such as, the Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation coefficient. The investigation affirmed that the trend and low-frequency components are the main causes of commodity futures markets prices' volatilities.

Lahmiri (2016) conducted a similar study using the novel VMD method to estimate stock market prices. A new multiresolution approach, VMD-BPNN model was suggested to predict the daytime market. The VMD technique was applied in breaking down price data into several modes. The decomposed modes were utilized for training BPNN. Particle swarm optimization (PSO) was initially introduced to enhance BPNN model. The results from six different stock markets show that the hybrid VMD–PSO–BPNN method has high predictive ability in contrast to the standard PSO–BPNN model.

Aneesh et al., (2015) exploited the opportunities of the VMD in evaluating the qualities of six distinctive powers and compare their performance with the empirical wavelet transformation (EWT). Empirically, the results illustrated that the VMD method performed better than EWT regarding the data distinctive segregation and identification accuracy.

Seo et al., (2018) utilized hybrid machine learning models (MLMs) to boost the forecasting of daily overland flow. VMD was combined with least squares support vector regression (LSSVR) and extreme learning machine (ELM) to form hybrid models - VMD-LSSVR and VMD-ELM respectively. The VMD was utilized in decomposing the actual series into its simplest units called modes. The modes extracted by VMD were used as inputs for LSSVR and ELM to develop models for rainfall-runoff. The forecasting performance of VMD-LSSVR and VMD-ELM were evaluated using the efficiency indices, and effectiveness index respectively. In addition the predictive performances of VMD-LSSVR and VMD-ELM were compared to the following decomposition models - artificial neural network (VMD-ANN), discrete

wavelet transform (DWT-ELM), (DWT-LSSVR), and (DMT-ANN), as well as the individual MLMs models, ELM, LSSVR, and ANN. The results revealed that VMD-based MLMs methods performed better than DWT-based MLMs in terms of efficiency and effectiveness.

The price movement of stock market depicts volatile features and unique patterns, hence accurately forecasting the stock price index is a pivotal for investors and researchers. Niu et al., (2020) embraced VMD approach to forecast the stock market price volatility. VMD was employed to pre-treat the actual stock market price series into different modes. The decomposed modes were forecasted by using VMD-LSTM and EMD-based models. It was reported that the VMD-LSTM model raises the stock market price index predictions; in addition, the VMD-based model was superior to the EMD-based model.

Besides stock market prediction, the VMD has been adopted to forecast natural rubber market. Zhu et al., (2019) developed VMD- based models to evaluate rubber's futures market prices' fluctuations and was utilized to decompose the rubber market price into sub-series of distinct frequencies. The VMD was combined with the BiGRU model to predict short-term rubbers' futures market prices on the Shanghai Future Exchange. The outcome of the study proved that the suggested technique optimal the natural rubber market price prediction and could reveal the embedded features of rubber market price disparities.

In addition, Jianwei et al., (2019) combined independent component analysis (ICA), gate recurrent unit neural network (GRUNN) and VMD to analyze gold future market price variations. The VMD technique was applied in disintegrating the price series of gold to identify the continuous price fluctuation of the gold market. According to the findings, the VMD-ICA-GRUNN model achieves superior forecasting accuracy as compare to individual ARIMA, RBFNN, LSTM, GRUNN, and ICA-LSTM models.

He et al., (2018) welcomed the effectiveness of the VMD strategy and combined it with quantile regression neural Network (QRNN) to predict the inconsistency in the crude oil markets, in particular the West Taxes Intermediate (WTI) market, Brent, and the OPEC markets. The VMD was proposed to form decomposition model, VMD-QRNN. The study reported that the combination method suggested achieved high predicting accuracy for three commodities markets instability and improved forecasting precision of QRNN.

Li et al., (2019) carried out an hydroacoustic investigation making use of radar in detecting physical and biological signals of underwater plants and animals. First, the researchers utlized the VMD to separate the hydroaccoustic signals in water into its respective modes and frequencies, followed by formation of VMD-SVM model to distinguish various underwater acoustic signals. The study found that the proposed approach could recognize and perfectly group underwater signals for smooth detection.

Fianu (2022) employed a categories of decomposition-based models and combined them with extreme

machine learning, namely; Complete Ensemble Empirical Mode Decomposition with Adaptive Noise-based ELM Model (CEEMDAN-ELM), Ensemble Empirical Mode Decomposition-based ELM Model (EEMD-ELM), and Empirical Mode Decomposition Based ELM Model (EMD-ELM) to examine the energy commodity prices volatility of crude oil, Japan gas, US gas, and coal markets. The overall findings indicated that all decomposition-based extreme learning machines are superior to the benchmark model.

Daoui et al., (2022) utilized the VMD approach to study the co-movement of European energy power industries, Nordic power exchange (Nord Pool) and Mercado Ibérico de Electricidade, Iberian power market (MIBEL) energy exchanges, during the COVID-19 pandemic. Their findings indicated that there are similarities among European energy power industries over long- term and medium-term investment horizons. In addition, there was significant connectivity between Nord Pool and MIBEL during the crisis period.

The greenhouse gases emission is regarded as the paramount source of global warming crisis (Sun and Xu, 2022). Carbon emission is identified as the main contributor of global warming and its production needs to be regulated. One of such measures is open-carbon trading market to reduce carbon emission globally. In estimating accurate carbon price prediction, Sun and Xu (2022) proposed decomposition-based models using VMD to break down the actual carbon price series into several modes, followed by dynamic adaptive inertial factor particle swarm optimization (DAIFPSO) to augment the neural network prediction. The results reveal that the model is stable in predicting carbon price by extracting the intrinsic features in the carbon market price series.

To obtain accurate prediction of crude oil price movements, Li et al., (2021) developed multiscale hybrid models to estimate the prices of crude oil. The study took advantage of VMD in decomposing the price data of crude oil into simple units. Data feature analysis was conducted to evaluate the different units of the price series. The empirical study proved that the suggested model can attain superior predicting results.

Liu et al., (2021) adopted VMD decomposition approach to prognosticate market price series of gold and decompose it into their respective modes and frequencies. The ANN method was used to predict each component independently, and then each predicted component was integrated. The results emerged that the VMD-ANN considerably enhanced the predicting accuracy of gold price as juxtapose to the state-of-art approaches.

Load-forecasting technique is critical for distribution of power supply and demand. To achieve an accurate predicting power supply and demand, Zhang et al., (2022) used VMD and stacking model approach to study short-term load supply and demand. The VMD algorithm was employed to decompose real-time

power dispatch to different IMF, and then followed by the application of Approximate Entropy (ApEn) to calculate corresponding new IMF components. The results signified that the component evaluation is firmly fused when applying the Stacking model fusion technique.

Accurately forecasting energy prices has a consequences on a country's energy reliability and environmental blueprint. To establish the precise forecast for energy prices, Lin et al., (2022) developed a VMD model to predict energy prices' fluctuations. The novel VMD was utilized in disintegrating the actual energy price data into sub-series with different frequencies. A low frequency component was forecasted using the autoregression model (AR), while the high frequency component was predicted using Elman neural network (ELMAN). The predicted sub-series with different models were reconstructed and ensembled as the final forecasted values. The novel VMD-based model surpassed other comparative models.

As noted by Fianu (2022), the price volatility of crude oil, corn, and gold does influence energy, food, and industrial metal prices, which have direct consequences on the separate sectors of the global economy. This, hence, give rise to accurate price volatility forecasts of these commodities price predictions for valuable and reliable operational security of energy, food and metal markets. It is critical, thus, to analyzing all these three commodities since they play crucial role in the energy, food, and industrial consumption, in addition to individual consumers.

In view of this, we employed decomposition-based models, Empirical Mode Decomposition and Variational Mode Decomposition combined with back-propagation neural network and autoregressive integrated moving average, to examine commodity futures prices' market in respect of their predicting abilities.

2.3 Theoretical literature review

The forecasting theory is formulated on the assumption that present and previous knowledge can be utilized to make estimations about what will happen in the future, particularly for times series. In times series analysis, it is possible to link the trend followed by the past values, and effectively forecasting future values. In practice, forecasting methods are unique if they have a link to an issue to be addressed. A hypothesis can be formulated by identifying the necessary characteristics of the problem, subsequently, the theoretical outcomes can enhance practice; from this conception, predicting theories are proposed as predicting techniques and models. Predicting methods are explained as predetermined series of steps that yields predict at subsequent time ahead. Numerous predicting models, but not all, have their equivalent stochastic models that can give similar point predicts. In addition, stochastic method gives a data-generating process that can be utilized to forecast intervals and the complete forecast distributions. Each

and every stochastic method provides presumptions about the procedure and the underlying probability distributions. Even when a predicting model has basic stochastic method, it is still not exceptional model. For instance, the simple exponential smoothing method has many stochastic methods, as well as models that may or may not be a homoscedastic method. Model-integration-forecasting method has been proven to be an efficient forecasting technique. The amalgamation of stochastic methods, if they work together, constitute another method. Effective forecasts can be achieved by a method that combines new and established predicting models; even these approach also form forecasting models.

Selection of the type of parameters and their participation in the predicting procedure is vital. In univariate predicting, the models are formulated for a individual time series by applying the past knowledge from the previous values of the time series itself. While in multivariate predicting, other times series variables are required as time series regression, in producing the forecasts. The data for time series is generated in different forms depending on the type of predicting model to be considered, for instance, a study in economics normally includes several variables that influence one another.

Prior to employing a forecasting technique, the data may need pre-treatment like checking for validity and missing data points; this ensures that the data meet basic features, like checking for validity and missing data points. Other pre-treatment issues might follow the choice of a particular forecasting approach, for example, management of seasonality in the data. Some forecasting models need de-seasonalized time-series data, while others check seasonality within the models.

Finally, it is exceeding crucial to assess the success of a predicting model; a good forecasting model provides guideline on how to test its validity. There are many loss functions that have been suggested to measure the variation in observed data point and the forecasted point, for example, using evaluation tools such as prediction intervals and percentiles, procedures, and metrics. The remaining sections, in this chapter, focus on the forecasting theories relevant to this topic.

2.4 Martingale process

Initially, the martingale theory in probability was proposed by Levy (1935), although he did not give it a name. The name 'martingale' was later introduced by Ville (1939) and expanded the concept to cater for continuous martingales. Much of the original concept was extended by Doob (1971) to show the unfeasibilty of successful betting approaches in a fair game.

The martingale process is a stochastic technique whereby the conditional prediction of its future value, is the same as its present value, given all the information up to now. In other words, the future price of an

asset is expected to remain the same at all time, given the asset past information.

Let $C_0, C_1, C_2, C_3, ...$ be a squence of random stock prices and C_n the information available on the stock at time n. Surpose P is any random price, let $E(P|\Omega_n)$ represent conditional probability of P given information available at time n. Let assume that the available information at time n comprise exactly $C_0, C_1, C_2, ..., C_n$, then

$$E(P|\Omega_n) = E(P|C_0, C_1, C_2, ..., C_n). \tag{2.4.0.1}$$

However, in applying equation (2.4.0.1), it could be noticed that there are other information known in addition to time n. For instance, if C_n corresponds to the price of an asset C on the n^{th} day and K_n denotes the price of an asset K on n^{th} on day, and particular n day the following sequence of information is known about both assets. Then $E(P|\Omega_n)$ is our new expectation of P after taking into consideration the access of information about the two sequences at time n.

If C_n is a stochastic process, then C_n is said to be a martingale if

$E(|C_n|)) < \infty \forall n$ and $E(C_{n+1}|\Omega_n) = C_n \forall n$. That is, if $C_0, C_1, C_2, ...$ is a martingale if this condition is true:

$$E(C_{n+1}|\Omega_n) = C_n \forall n \tag{2.4.0.2}$$

From the definition in equation (2.4.0.2), suppose C_n denotes a stock price on day n. In this case, the martingale concept states that the price of the stock tomorrow is expected to be equal to the price of the stock today', taking into account all available information on the stock today. It is not ridiculous to proclaim that on a particular day the stock prices should approaching this property presuming there is no internal information and no returns have been shared today or tomorrow. For instance, if the price of a stock today is 70 and investors anticipated it to be 75 tomorrow, then it is easy to make profit, that is, "buy today and sell tomorrow", however, if the general public had access to the same internal information , then other venture capitalists would take advantage to enter into the market by buying the stock today at its current price, and the stockholder holding the stock would be unwilling to sell it out for its current price, hence, this will expect the price to rise up to 75 today. This is not applicable in efficient market where the present stock prices contains all previous prices information. Therefore, market is said to be efficient when stock market prices are randomly changed and unpredictable. In financial market, investors take risk and anticipating to gain profit in the near future, however, martingale process puts constraint on anticipated profit, and eleminate risk, which make a martingale process insufficient in determining the stock prices. Notwithstanding, the martingale concept has been used in the contemporary theories in determing stock prices (Campbell, 2004). In theory, if the risk is well adjusted in the stock market, then

the investors are expected to gain some profit from their investment, and this interns make the martingale concept invalid. This give rise to proposition of a Random process.

2.5 Random Walk proces

A random walk model is associated with the martingale technique. It is utilized to evaluate whether returns can be predicted. Random walk theory proposes that fluctuations in stock prices have identical distribution and do not dependent on each other, hence, it supposes that the previous market price of a stock cannot be utilized to forecast its future trend. In other words, random walk theory demonstrates that market stocks are randomly generated and indeterminate such that all techniques of forecasting stock prices become pointless eventually. A random Walk process is given by:

$$C_{n+1} = \mu + C_n + \varepsilon_{n+1}, \tag{2.5.0.1}$$

where C_{n+1} is the future price of an asset, μ is the average prices change of expected returns by holding the asset over a period of time, C_n represents the current price of the asset, and ε_{n+1} is unpredicted error term. The Random Walk method can be changed to martingale process by putting constraints on the error component ε_n. The characteristic of the error term is very crucial, and putting constraints on ε_n leads to three different Random Walk methods, namely Random Walk I- identical independent distribution increments, Random Walk II- independent increments, and Random Walk III- uncorrelated increments(Campbell, 2004). For details of the three Random Walk process, see the work of (Campbell, 2004).

2.6 Efficient Market Hypothesis (EMH)

The efficient market hypothesis (EMH) suggests that stock prices completely contain all accessible information and expectancies, therefore present prices are the finest estimation of a firm's inherent value. This ensures that price changes follow a random process and caused by unpredicted events which prevent anyone making abnormal returns (Jones and Netter, 2008). There are three forms of market efficiency - Weak form, Semi-Strong form, and Strong form. The weak form EMH presumes that the all previous stock prices are reflected in the current stock prices. It further assumes that future stock prices are independent of the past information of the stock prices. Unlike weak form, semi-strong form EMH assumes that stock prices completely and equitably contain all public accessible information as well as the past information. It further explains that not anyone can use elementary nor technological analysis to predict the stock future prices to gain superior returns (Fama, 1970). The strong form EMH refers to a market form whereby

stock prices wholly and impartially reflect all public, past, and private accessible information about the market. In this type of market, though, the information available does not give the investors advantage to gain super normal returns from their investments.

2.7 Exponential smoothing models

Exponential smoothing is considered one of the best models in financial forecasting and despite numerous advances in the field, exponential smooting still has a remarkable forecasting performance. The development of exponential smoothing dates back to 1944, when Robert G. Brown, through a mechanical computing device evaluated key variables for fire-control on the location of submarines (Gardner Jr, 2006). The work of Gardner (2006) provides more details about the development in exponential smoothing.

The exponential smoothing method depends on the weighted average of past observations, whereby the weight diminishes exponentially as one observation moves away from the current observations. A suitable exponential smoothing technique relies on the elements that emerge in the time series, for example, in a situation where there is no clear trend or a seasonal pattern is detected, the Simple or Single Exponential Smoothing (SES) is adequate. The Simple or Single Exponential Smoothing method is given by:

$$y_{t+1} = \diagdown x_t(1 - \diagdown)y_t, \tag{2.7.0.1}$$

where y_1 is the initial forecast and $\diagdown$ is the exponential smoothing parameter. The SES' expression is sometimes referred to as "Exponentially Weighted Moving Average" (Harvey, 1990). The equation for SES can be obtained by minimizing the discounted least squares error function and writing the entire expression in a recursive form (Harvey, 1990). If the observations differ in weight, the ordinary least squares is not applicable, rather the recursive form is appropriate. The SES can be used by first estimating the initial forecast y_t and exponential smoothing parameter $\diagdown$. Generally, the initialization can be achieved by using the huerlastic method (Hyndman et al., 2008). The initial forecast and the optimal smoothing parameter can be calculated by means of minimizing the sum of squares of the one-step ahead forecast errors

2.8 Autoregressive integrated moving average (ARIMA)model

Autoregressive integrated moving average is most often used for forecasting in time series models (Box et al., 1976). The ARIMA(p, d, q) model for a time series x, is given by:

$$(1 - \psi_1 L - ... - \psi_p L^p)(1 - L)^d x_t = n + (1 + \vartheta_1 L + ... + \vartheta_q L^p) + \varepsilon_t, \tag{2.8.0.1}$$

where L represents the lag operator and is defined by $L^k x_t = x_t - k$. ε_t is the error term with uncorrelated zero-mean and common variance σ_ε^2. In instances where the ETS models assume that $d = 1$ or $d = 0$, some exponential smoothing model can be expressed in ARIMA form. For instance, exponential smoothing model is the same as ARIMA(0, 1, 1) when $\vartheta_1 = \beta - 1$.

Maximum Likelihood can be used to estimate the parameters in the ARIMA model, while an Ordinary Least Squares is used to evaluate the ARIMA (p, d, 0).

The iterative model-building process can be used to determine the values of p, d, and q (Franses, 2014). Autocorrelation function and partial autocorrelation function can be utilized to estimate the values of p and q, in instances where p and q values are small. In practice, the well established information criteria, such as, AIC and BIC, can be usd to evaluate p and q. An R statistical software package that contains the function auto.arima has been developed for comparing models using information criteria, and has been proved to be efficient and widely utilized in ARIMA modeling (Hyndman et al., 2008).

In general, it is easy to use ARIMA models to predict, and at the same time, they are flexible to calculate forecast intervals. For instance given ARIMA(1, 0, 1) model: $x_t = n + \psi_1 x_{t-1} + \varepsilon_t + \vartheta_1 \varepsilon_{t-1}$. The one-step-head forecast from origin m is given by: $f_m + \frac{1}{m} = n + \psi_1 x_m + \vartheta_1 \varepsilon_m$ as the expected value $E(\varepsilon_m + 1) = 0$. The forecast error is expressed as $x_{t=1} - f_m + \frac{1}{m} = \varepsilon_{m+1}$, therefore, the forecast error variance is σ_ε^2. Similarly, two-step-ahead forecast from the origin m can be expressed as: $F_m + \frac{2}{m} = n + \psi_1 \varepsilon_{m+1}$ and forecast error variance $(1 + \psi_1^2)\sigma_\varepsilon^2$. These two expressions indicate that the establishment of forecasts and errros directly follow from the model expressions, and therefore,can be automatically evaluated.

The ARIMA model depends largely on the choice of d. If $d = 0$, the ARIMA model is generated at the levels of time series, thus, x_t. If $d = 1$, $(1 - L)x_t$ is the model and the data must be differenced before fitting an ARIMA.

2.9 Autoregressive conditional heteroscedasticity (ARCH) and the generalised autoregressive conditional heteroscedasticity (GARCH) models

Volatility has been identified as the main component of measuring risks and uncertainties in commodity forecasting (Markowitz, Sharpe, 1964, Taylor et al., 2009, Gneiting, 2011). Evaluating future volatility for estimating the uncertainty of predicts is crucial for probabilistic predicting, yet, choosing the right period to compute future volatility has been challenging as too long a volatility period will make frivolous forecast horizon, while too short a volatility period produces too much noise (Engle, 2004). To avert this issue, Engle (2004) proposed a dynamic volatility models - the autoregressive conditional heteroscedasticity (ARCH), and the generalised autoregressive conditional heteroscedasticity (GARCH). The ARCH model makes use of the weighted average of the previous squared forecast error while the GARCH model generalizes the ARCH model by further adopting previous squared conditional volatilities. The GARCH model is an amalgamation of: (1) a constant volatility, which evaluates the long-run mean, (2) the volatility forecast (s) in the final stages, and (3) the new information collected in the final stages. The weightings of these elements are evaluated using maximum likelihood. The models take a residual distribution form to produce density forecasts. One of the advantages of the GARCH model is its ability to model heteroscedasticity in the time series, thus, the volatility grouping features of time series (Mandelbrot, 1963). Volatility grouping occurs when new information is added and a certain time interval is needed for the time series to be steady as the new information is initially identified as “a shock”.

2.10 Markov switching (MS) models

In the late 1980s, Markov switching (MS) models were applied in modeling and forecasting macroeconomics and finance data. MS models are dynamic econometric modeling and forecasting methods that accommodate regime shifts. The work of MS is to link the parameters of standard dynamics, such as systems of regressions, vector autoregressions (VAR), and vector error corrections to one or more unobserved state variables. This assumes that, V_t is MS model and can take N values and capture the notion of systems passing through regimes, and follow discrete stochastic process and the variables do not dependent on the shocks of the model.

A standard dynamic AR(1) model, for example, can be expanded as: $s_t = \psi_0, v_t + \psi_1, v_t s_{t-1} + \sigma_{v_t}\varepsilon_t$, where all the parameters, including the variance of the shocks may be assumed to take different values and can be estimated as function of v_t. Comparably, in N-regime MS VAR(p), the vector of intercepts and the p autoregressive matrices may be assumed to depend on v_t . Furthermore, the covariance matrix

of the system shocks depend on some state variable, either the same as the mean parameters (v_t) or an additional, specific one (k_t), and depend on lags of v_t. The expansion of MS VAR model to include exogenous regressor is a special case of MS regression.

Multivariate MS models suffer from over-parameterisations and must be checked, yet it is still powerful in fitting complex non-linearities and can be used to approximate many distributions (Wand, 1992). In addition, MS models have ability to capture main features of time series, such as difference in conditional means across the regime passing through high moments like kurtosis, variance, and skewness (Timmermann, 2000).

2.11 Neural networks (NNs)

The Artificial Neural Networks (ANNs) or Neural Networks (NNs) are data-driven mathematical methods motivated by the work of biological neurons. They have ability to model non-stationary, nonlinear and complex datasets. This characteristics have put NNs in the frontline of research in most fields of science and econometrics (Zhang et al., 1998, Dees et al., 2007).

A classical NN is made up of three layers - input, hidden and output - and each is made up of nodes. The input layer is the first layer of every NN and the number of nodes corresponds to the number of explanatory variables (inputs). The last layer is the output layer and the number of nodes equal to the number of response variables or forecasts. In between the input and the output layers, is the hidden layers. The hidden layer determines the amount of nodes the model is capable of fitting. Most NN models have an extra node between the input and the first hidden layer, defined as the "bias node". The bias node is fixed and performs a function similar to the traditional regression models' intercepts; each and every node in one layer has weights with all of the nodes of the next layer.

NNs process a given information as follow: the explanatory variables are contained in the input nodes; these variables are weighted by the input and the first hidden nodes, and the information are linked to the hidden nodes as a weighted sum of the inputs. In the hidden nodes, there is a non-linear function such as the sigmoid, which transmits the information received. This process is repeated until the information reaches the output layer as forecasts.

The most common NN process, is the back-propagation. The simplest and most common NN model, is the Multilayer Forward Perceptron (MLP). In MLP, the hidden nodes contain the sigmoid function and the information moves from the inputs to the output nodes. Another well-known NN model where the information moves also, only in a forward direction, is the Radial Basis Function NN (RBF). The hidden

neurons estimate the Euclidean distance of the test case from the neuron's centre point and then applies the Gaussian function to this distance using the spread values. Recurrent Neural Networks (RNNs) are NN models that permit past outputs to be used as inputs while having hidden states. The information moves both forwards and backwards. RNNs have short-term memory and inputs are taken potentially from all previous values. MLPs, RBFs and RNNs are universal function approximators Hornik (1991), Park and Sandberg (1991), Schafer and Zimmermann (2006), however, the amount of NN complexity in terms of hidden layers and nodes to reach this property, might make the NN model computationally difficult to train.

2.12 Threshold models

Studies have shown that financial and economic time series often depict non-linear features, such as irregular structure, and appear in the form of repeated regimes in model parameters. This structural instability is continuous, forms stochastic process, and can be predicted. Modeling economic and financial irregularity has gain popularity for econometricians since the 1970s. Threshold models are also used to model and forecast volatility, for example, the GJR-GARCH model by (Glosten et al., 1993).

One of the oldest and most used models is the threshold autoregressive (TAR) model proposed by Tong (1990). A TAR model is an autoregressive model for the time series f_t in which the parameters are driven by a state variable S_t, and is itself a random variable with K distinct integer values, that is, $(S_t = k, k = 1, ..., K)$. The S_t depends on the value of the threshold variable q_t when compared to $K - 1$ threshold levels, q_k, for example, if only two regimes exist, they are $S_t = 1$ if $q_t \leq q_1$and $S_t = 2$ otherwise. The threshold variable q_t can be exogenous or can be a lagged value of f_t. In another instance, we talk of self-exciting threshold autoregressive (SETAR) models. Other choices of q_t include linear Chen et al., (2003), Chen and So (2006), Gerlach et al., (2006) or non-linear Chen (1995), Wu and Chen (2007) combinations of the lagged dependent variable or of exogenous variables.

The TAR model has also been extended to account for different specifications of the conditional mean function, leading to the development of the threshold moving average (TMA) De Gooijer (1989), Tong (1990), Ling (1999) and the threshold autoregressive moving average (TARMA) Amendola et al., (2006), Ling (1999) models. Those models are similar in their operations, but their parameters depend on the regime K.

The set-back of TAR models is the utilization of conditional moment function, which fails to be continuous. To arrest this situation, Chan and Tong (1986) developed the smooth transition autoregressive (STAR) model. The difference between TAR and STAR models is that, TAR forces a sudden shift from

one regime to the others at any time that the threshold variable exceeds or is below a certain amount, whereas the STAR model changes gradually among regimes.

An STAR model is a two-regime model in which the dependent variable f_t is evaluated as the weighted average of two autoregressive (AR) models, thus,

$f_t = \Sigma + j = 1^p \varphi_{j,1} f_{t-j} p(s_t = 1; g(x_t)) + \Sigma_{j=1}^{p} \varphi_{j,2} f_{t-j} p(s_t = 2; g(x_t)) + \varepsilon_t$ wherex_t represents the transition variable and g denotes the transformation of the transition variable x_t. Regime probabilities are assigned through the transition function $Y(k; g(x_t))$, with Y depicting cumulative density function of choice; the transition variable x_t can be the lagged.

2.13 Bootstrap method

The bootstrap technique has been widely used in many areas of study, as well as time series analysis. The bootstrap process is a well-known methodology for independent data due to its flexibility and good effects (Efron and Tibshirani, 1986). It is a computer-exhaustive technique that displays solutions in states where the traditional methods are difficult to use. Efron and Tibshirani (1986) have, however, disclosed itself ineffective in term of dependent data, in the case of time series, where the dependence structure arrangement has to remain the same throughout the resampling process.

In resampling for dependent data, well-defined blocks are considered, such that the dependence structure within each are maintained. Blocks are classified according to the way they are constructed, these include: the nonoverlapping block bootstrap Carlstein (1990), the moving block bootstrap Kunsch (1989), the circular block bootstrap Politis and Romano (1994), and the stationary block bootstrap (Politis and Romano, 1994). If the time series process is guided by iid revolutions, another method of resampling, such as sieve bootstrap can be considered (Hlmann, 1997). The sieve bootstrap method is based on first fitting parametric models rescaling from the residuals. These models include, for instance, the linear regression Friedman (1981) and autoregressive time series (Efron and Tibshirani, 1986). These techniques differ from the previous bootstrap methods for dependent data; the sample bootstrap is stationary and the constructed structure does not depict dependence. Another different feature is that the sieve bootstrap sample is not a subsample from the original data, as in the previous approaches; even if the sieve bootstrap is dependent on a parametric model, it is still nonparametric in its operations.

The sieve bootstrap method was utilized in estimating forecast intervals (Zagda, 2001, Andre'es et al., 2002). These methods were extended by (Cordeiro and Neves, 2006, 2009, 2010) to evaluate point forecasts. The main idea of their study was to insert an exponential smoothing model to the time series, select

the residuals and then use the sieve bootstrap with the residuals, however, the bootstrap type for maximum or minimum fails to converge to external limit law in extreme value (Bickel and Doksum, 1981, Friedman, 1981, Angus, 1992).

A study conducted by Zelterman (1993) pointed out that "to resample the data for approximating the distribution of the k largest observations would not work because the 'pseudo-samples' would never have values greater than $X_{n:n}$ ". Unfortunately, these methods have been extensively used in modeling and forecasting commodity market prices volatility but they are unable to reveal the dynamic features of the data and cover the frequency domain of the price movement of commodity markets. The nature of commodity market data is such that utilizing the approaches discussed above is inadequate for a better understanding. We need, therefore, to decompose the commodity price data into different frequencies to completely understand the dynamic characteristics of the data The current thesis is based on the assumption that a decomposition and ensemble framework is necessary to fully appreciate the intrinsic structure of commodity price data.

2.14 Conclusion

In this chapter, some of the relevant literature on empirical mode decomposition and variational mode decomposition were reviewed, including the areas in which these novel approaches had been applied. We also compared the performance of EMD and VMD with its competing methods, such as SSA, wavelet transforms, and DWT, in terms of their uniqueness in extracting the embedded intrinsic factors in the data, flexibility, empirical, and efficiency in decomposing nonlinear and nonstationary data. We also discussed the theoretical literature framework of this current study.

Chapter 3

Methodology

Chapter summary

This chapter presents the theoretical background of the methodology used to model and forecast the commodity market futures prices, including the EMD and VMD methods. Back-propagation neural network (BPNN) and autoregressive integrated moving average (ARIMA) were also introduced, which served as the benchmark and comparative models respectively.

3.1 Introduction

There is a consensus among academics and practitioners that commodities compared to other assets can be considered, in a portfolio context, as an asset class of their own (Fabozzi et al., 2008). An asset class in that context consists of assets that show a high correlation in their risk or return ratio and a different ratio compared to other assets. Commodities, as an exception, possess intra-class heterogeneity. In other words, every commodity has its specific characteristics.

Commodities are generally grouped under two different categories: soft commodities and hard commodities (Fabozzi et al., 2008). Hard commodities are products from energy, precious and industrial metals, and soft commodities are usually perishables from the agricultural sector. Figure 3.1 shows the components of each of these categories:

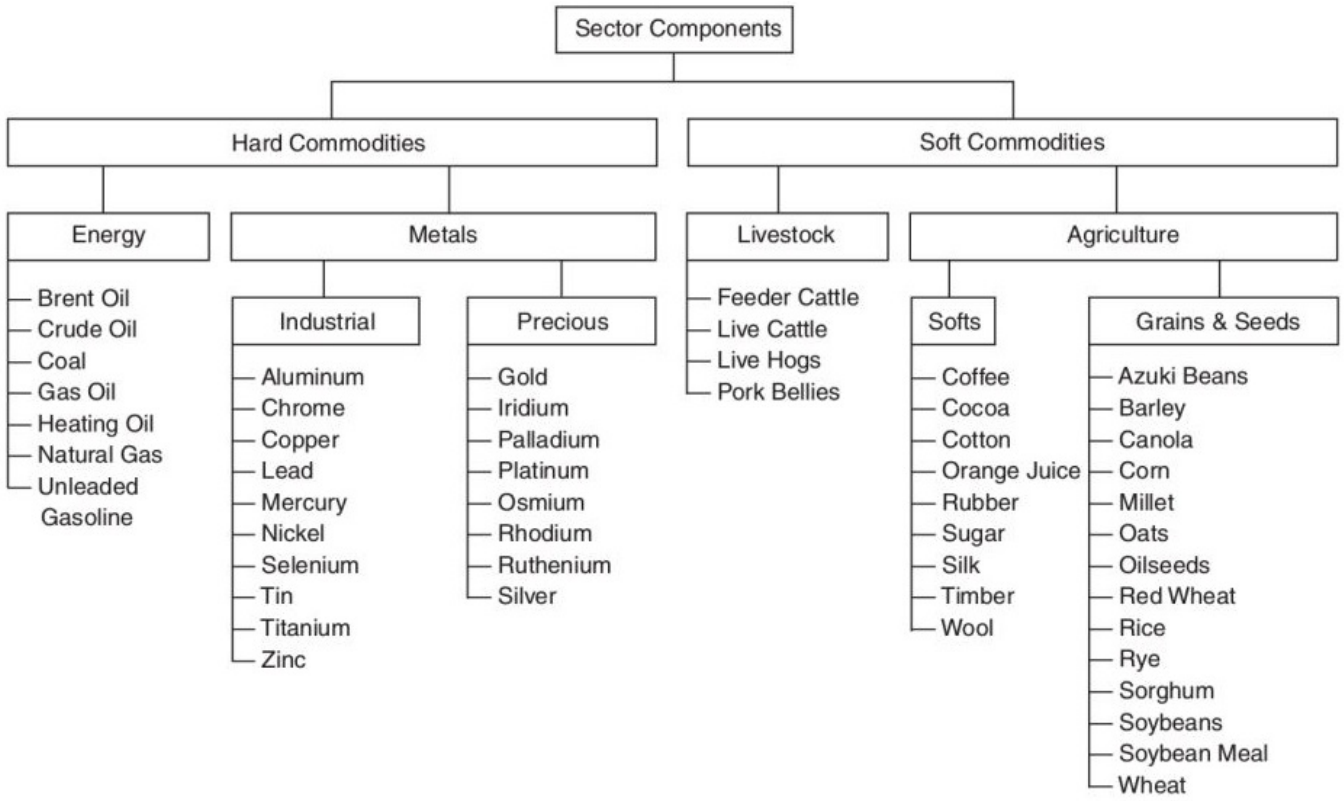


Figure 3.1: Categories of commodity

Commodities are traded in exchanges in commodity futures contracts, which is an agreement to buy or sell a certain amount of a specific commodity at a specific price and date. Fluctuations in the price and traded volume of these contracts generate historical information in the form of time series, used for econometric studies and building portfolios. The raw value for a certain amount of a commodity can also be used as a time series.

3.2 Pre-treatment data

Box-Cox transformations

In practice, the variable of interest is transformed in forecasting models by using transformation as suggested by (Box and Cox, 1964). The Box and Cox (1964) transformation is given as:

$$h^{\Upsilon} = \begin{cases} \dfrac{(h^{\Upsilon} - 1)}{\Upsilon}, \Upsilon \neq 0 \\ log(h), \Upsilon = 0. \end{cases}$$

The transformation range depends on the sign of Υ, hence, Bickel and Doksum (1981) suggested the following modification to allow for wider range of Υ

$$h^{\Upsilon} = \begin{cases} \dfrac{(|h|^{\Upsilon} sign(h_i) - 1)}{\Upsilon}, \Upsilon \neq 0 \\ log(h), \Upsilon = 0, \end{cases}$$

$\Upsilon \in (-\infty, \infty)$ for any value of Υ. The (Box and Cox, 1964) transformation ensures that data conform to presumptions of constant variance and normality that are requisite for statistical models to cater for interference in the model. There is a natural continuum between additive and multiplication models when Υ lies between 0 and 1.

The Box and Cox technique divides the series into blocks, estimates the coefficient of variation within each block and then re-estimates the coefficient of variation between these blocks. The block with minimum Υ is chosen. Choosing a Υ is extensive procedure; this process of estimating the Υ in forecasting is given by (Guerrero, 1993). Transformation of data using Box and Cox method is monotonic. It should be noted though, that prediction intervals that are symmetric in terms of the transformed data will not be symmetric in terms of the original data. In a similar vein, back transformation of the forecast median of the transformed data returns the forecast median in terms of the original data.Transformed data do not yield the forecast mean of the original data, due to the non-linearity of the transformation, consequently forecasts on the original scale of the data will be biased unless a correction is used.Examples of bias correction techniques are the works of (Granger and Newbold, 1976, Taylor, 1986, Pankratz and Dudley, 1987, Guerrero, 1993).

3.3 Time series decomposition

Time series decomposition is an essential building block for various forecasting approaches and a crucial tools for statistical agencies. Seasonal decomposition is a way to present a time series as a function of other time series - components. Commonly-used decompositions are additive and multiplicative, where such functions are summation and multiplication respectively. If logs can be applied to time series, any additive decomposition method can serve as multiplicative after applying log transformation to the data.

The simplest additive decomposition of a time series, with single seasonality comprises three components: trend, seasonal component, and the “remainder”. It is assumed that the seasonal component has a repeating pattern (thus, sub-series corresponding to every season are smooth or even constant); the trend component describes the smooth underlying mean and the remainder component is small and contains noise.

The first attempt to decompose time series into trend and seasonality, is dated 1847, when Buys-Ballot (1847) performed decomposition between trend and seasonality - modeling the trend by a polynomial and the seasonality by dummy variables. Then, Poynting (1884) proposed price averaging as a tool for eliminating trend and seasonal fluctuations. Later, his approach was extended by (Anderson, 1914, Hooker, 1901, Spencer, 1904). Copeland (1915) was the first who attempted to extract the seasonal component, and Macaulay (1931) developed a method which is currently considered "classical".

The main idea of this method comes from the observation that averaging a time series with window size of the time series' seasonal period leaves the trend almost intact, while effectively removing the seasonal and random components. For the next step, subtracting the estimated trend from the data and averaging the result for every season gives the seasonal component.

3.4 Empirical Mode Decomposition (EMD)

Financial data such as commodity prices are complex due to multiple factors and often exhibit intermittent behavior, which occurs at different time intervals. Furthermore, the frequencies of the factors that cause these sporadic characteristics vary with time. These behavioral patterns in financial data are readily detected by EMD analysis, however, they cannot be identified by SSA, wavelet transform, and Fourier analysis.

The Empirical Mode decomposition introduced by Huang et al., (1998) is an adaptive method that simplifies nonlinear and non-stationary data as a small number of orthogonal modes, defined as IMF. Each IMF has asymmetric envelope - local maximums and minimums - such that its mean is zero.

Comparing EMD to its competing models such as Fourier transform, SSA, and wavelet transform, EMD presents time series as a sum of harmonics with different frequencies and amplitudes. It works perfectly for non-stationary non-linear processes when the harmonic frequencies are not always available. A Fourier transform can represent time series as harmonic functions, however, it only works well with series with fixed frequencies and amplitudes and is less efficient in dealing with non-periodic or non-stationary series with acute peaks. Under such circumstances, wavelet transform and SSA are often used, although they have setbacks, such as basis functions, adding false cycles to the series, and an inability to determine local changes.

Unlike Fourier, wavelet, and SSA, the EMD method changes with time to adapt to the observed data changeability. EMD uses information from the actual series instead of a priori basis functions, thus, it is empirical, local, and efficient (Pegram et al., 2008). EMD is a typical data-driven decomposition approach

Huang et al., (1998); these researchers specify that, IMF extracted in the EMD process must conform to these basic principles:

- In the actual data point, the number of extrema and the number of zero-crossings must either equal or differ at most by one.
- At any period, the mean value of the envelope is specified by local maxima and the envelope specified by the local minima is zero.

The above-stated assumptions guarantee that, the derived IMF is a harmonic function and carries real facts about the signal associated with the formation of the investigation of the non-stationary input signal, $v(t)$. At the same time, the IMFs are not fixed and orthogonal to each other, but are flexible and based on the kind of basic signal. The process of obtaining the IMFs is defined as a sifting process.

We follow Zhu et al., (2015) in describing the EMD process:

(a) identify the maxima and minima of the prices data and represent it as $v(t)$;

(b) use cubic spline interpolation to create the upper and lower envelopes, and;

(c) calculate the mean of the two envelopes using the formula.

$$h(t) = \frac{e_{min(t)+e_{max(t)}}}{2} \tag{3.4.0.1}$$

(d) subtract the envelope mean, $h(t)$ from the price series, $v(t)$ and obtain

$$d(t) = v(t) - h(t) \tag{3.4.0.2}$$

(e) verify the characteristics of $d(t)$;

- If $d(t)$ satisfy the conditions of an IMF, we denoted $d(t)$as i^{th} IMF and replaced $v(t)$ with residual

$$r(t) = v(t) - d(t) \tag{3.4.0.3}$$

The i^{th} IMF is denoted by $c_i(t)$ and the $c_i(t)$ is called its index;

- If $d(t)$ does not satisfy an IMF conditions, replaced $v(t)$with $d(t)$.

(f) Repeat steps (a) to (e) until the residuals become either a constant, a monotonic function or have only one extremum. For information about stopping conditions, refer to the work of (Zhu et al., 2015). The

commodity price series, $v(t)$ at the end of EMD process can be reconstructed as:

$$v(t) = \sum_{i=1}^{n} c_i(t) + r_n t \tag{3.4.0.4}$$

where n is IMFs produced, $r_n(t)$ is the last residue, and $c_i(t)$ is the i^{th} IMF.

In the sifting process, c_1 is the first IMF element, and contains the finest IMF of the price series, followed by the residue with a longer period, consequently, EMD sorts and classifies the IMFs into their respective frequencies, thus, high and low-frequency modes and residue. In practice, the classification of IMFs is based on the principle defined as 'fine-to-coarse' reconstruction.

EMD has the following attractive properties which make it suitable for decomposing data - EMD can break down non-stationary and non-linear signals into individual intrinsic mode function, the decomposition is based on the local characteristic time scale of the data and only extrema take part in the sifting process, hence, it is local, adaptive, effective, and unique Guo et al., (2012), and the IMFs decomposed by EMD has an instantaneous frequency based on phase functions, such that the Hilbert transform can be applied to the IMFs.

3.5 Variational mode decomposition (VMD)

In spite of the advantages of EMD over other decomposition methods, it is still suffers from a problem called 'mode mixing' and 'extreme point effect'. This is a situation where by one IMF comprises extensively different scales. To overcome the problem of mode-mixing and extreme point effect in EMD, Dragomiretskiy and Zosso (2013) introduced VMD as mode-supported technique to improve EMD. VMD is a non-repetitive signal procedure. It is utilized to break down time series data into individual numbers of band-limited sub-signals, defined as "modes y_k,' with some sparse features. The decomposed modes can be decreased to a single center called 'pulsation w_t,' and it is accompanied by the decomposition process. We followed the steps below in estimating the bandwidth:

1. Apply the Hilbert transform to each decomposed mode, y_k, to obtain the center frequency spectrum;
2. Adjust the mode's frequency spectrum to the baseband by variegating the center frequency and the exponential tuned;
3. Evaluate the bandwidth of individual mode, y_k by applying Gaussian smoothness, H^1.

A constrained variational problem can be expressed as: suppose $g(t)$ is the original signal of the data and y_k is the k^{th} of the original signal of the data, then

$$g(t) = \sum_{k=1}^{m} y_k \tag{3.5.0.1}$$

We reduced the constrained variation as follows:

$$min_{(y_k,\omega_k)}\{\Sigma_{k=1}^{K}||\delta_t[(\delta(t)+\frac{j}{\lambda t})\otimes y_k(t)]e^{-j\omega_k t||_2^2}\}s.t.\Sigma_{k=1}^{K}y_k = g(t) \tag{3.5.0.2}$$

where $g(t)$ is the actual series, y_k, is the k^{th} element of the actual signal, w_k is the center frequency of y_k, $\delta(t)$ is the Dirac distribution, and $\otimes$ act as convolution operator; m is decomposed number of modes and t express the time script. Given the penalty term and Lagrangian multiplier, λ, we can change the constrained problem to unconstrained one as:

$$L(y_k,\omega_k,\lambda) = \kappa\Sigma_{k=1}^{K}||\delta_t[(\delta(t)+\frac{1}{\pi t})\otimes y_k(t)]e^{jw_k t}||_2^2 + ||g(t)-\Sigma_{t=1}^{k}y_k(t)||_2^2 + [\lambda(t), g(t)-\Sigma_{k=1}^{k}y_k(t)] \tag{3.5.0.3}$$

where κ represent constraint stabilizing parameter , L denotes augmented Lagrangian. The augmented Lagrangian L can be estimated in the equation (3.5.0.3) and its associated saddle point in the iterative series. Sub-optimization of L and its minimax level can be obtained through the alternate direction method of multipliers (ADMM). ADMM optimization method assumes that upgrading the original signal, y_k, and center frequency,ω_k, in two different directions helps to secure a good VMD results. For a full explanation of ADMM process, see (Dragomiretskiy and Zosso, 2013).

The values of y_k and ω can be estimaed as:

$$\hat{y}_k^{n+1} = \frac{\hat{f}(\omega) - \Sigma_{i=k}\hat{y}_i(\omega) + \frac{\hat{\lambda}(\omega)}{2}}{1+2\kappa(\omega-\omega_k)^2} \tag{3.5.0.4}$$

$$\omega_t^{n+1} = \frac{\int_0^\infty \omega|\hat{y}_k^{n+1}(\omega)|^2 d_w}{\int_0^\infty |\hat{y}_k^{n+1}(\omega)|^2 d_\omega} \tag{3.5.0.5}$$

3.6 Back propagation neural network (BPNN)

The ANN is a data-driven representative model that can adjust very big non-linear continuous functions to acceptable correctness (Hamza E, 2008). Among all the neural networks, Feed-Forward is the most widely used in commodity-markets price forecasting due to its simplicity and numerous studies backing up its efficiency (Khandelwal et al., 2015). BPNN is a well-known feed-forward artificial neural network built

on the back-propagation process and it has been used in several fields of study. One exceptional edge of BPNN over other models is that it can estimate any non-linear continuous function to any desirable level compared to the conventional statistic methods (Zhang and Qi, 2005). In general, a BPNN comprises one input level, single or additional concealed levels, and one output level. This study considered one concealed level, such that a standard three-level, p x p x r BPNN model was obtained, as illustrated in Figure 3.2.

Recently, the application of BPNN to predict commodity markets prices has gained much attention. BPNN can detect the relationship among past, current, and future prices. Using the present price data, the BPNN approach can trace past price patterns and predict a price pattern. It does not need any priori function, for instance, in forecasting a day-head electricity price, Reddy et al., (2016) employed the BPNN technique to reconstruct the electricity market price of Pennsylvania, New Jersey and Maryland. They reported that BPNN efficiently revealed the underlying factors of price fluctuations in electricity commodity markets.

Lasheras et al., (2015) used the BPNN model to study copper spot price and compared its forecasting ability to the ARIMA model using New York Commodity Exchange (COMEX) data. The empirical results revealed that the BPNN model over-performed the ARIMA model.

Wang et al., (2011) utilized a wavelet De-noising-based back propagation neural network (WDBPNN) to predict stock markets indices, using monthly closing price data from Shanghai Composite Index. The empirical analysis showed that the combined WDBPNN model forecasts stock market prices accurately, hence, can effectively reveal the stock market prices.

Jo (2013) adopted the BPNN to predict multivariate time series. It was empirically validated that BPNN improves Virtual Term Generation (WTG) schemes effectively, therefore, the BPNN has been used as a forecasting model since it can fit nonlinear series better than linear models. Nevertheless, the BPNN model requires more training sets to boost its robustness, as it does not work well for short length times series.

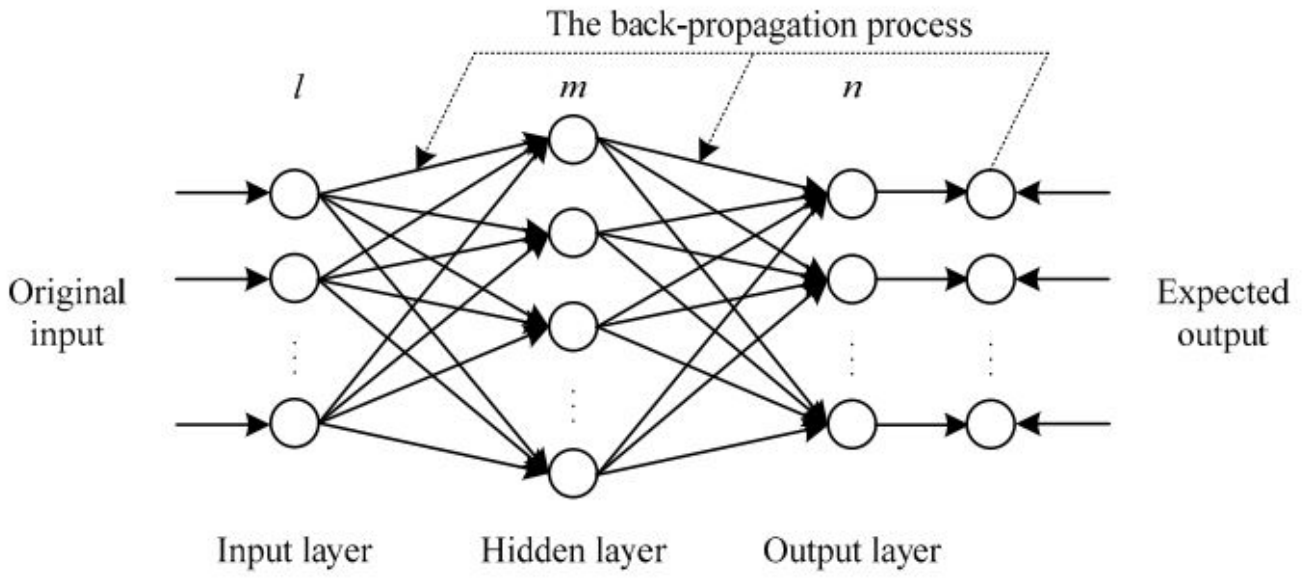


Figure 3.2: Three-level back propagation neural network

3.7 Back-propagation process

We followed Wang et al., (2017) in presenting the BPNN algorithm. A standard three-level BPNN contains - input point, a single concealed point, and an output point. Every point contains several components; the number of components is obtained by using an experimental approach when the BPNN is linked. Each component serves as an input to all the components in the next forward point, and no weight returns to the preceding point output component. Each link has a bias, θ_j, and several weights ω_{ij}, which link the component $i$in the preceding point, denoted by v_i, to component j, hence, we calculated the output, y_j^h in the hidden point and output point using the formula:

$$y_j^h = f(\Sigma_{i=1}^{l}\omega_{ji}v_i + \theta_j) \tag{3.7.0.1}$$

where ω_{ji} denotes the connected weight from i^{th} input to node to j^{th} hidden node, v_i represents thei^{it} input data, θ_j is defined as bias ofj^{th} concealed neuron, and $f(.)$ represents the non-linear transfer function of the concealed level, which is normally an activation function.

We computed the output of the neural network as:

$$y_k^o = \rho(\Sigma_{j=1}^{m}\omega_{kj}y_j^k + v_k) \tag{3.7.0.2}$$

where ω_{kj} denotes the weight linking j^{th} concealed node to k^{th} output node, v_k defines the bias of the k^{th} output neuron, and $\rho(.)$ represents the output point transfer function, and is always linear.

Generally, the BPNN model reduces the mean square error (MSE), E and is estimated as:

$$E = \frac{1}{n}\Sigma_{t=1}^{n}\Sigma_{k=1}^{n}(Y_Y - y_k^o)^2 \tag{3.7.0.3}$$

Where n represent the inputs number , Y_k denotes the k^{th} predicted output data.

3.8 Autoregressive Integrated Moving Average Process (ARIMA)

The ARIMA proposed by Box et al., (1976) is the most popular and widely used statistical method for time series analysis among the models. ARIMA works on the principle that future values of the time series are generated from the linear function of the previous observations and error terms (Box et al., 2015). An ARIMA (p, d, q) model is given by:

$$\psi A1 - A^c y_t = \vartheta A\varepsilon_t, \tag{3.8.0.1}$$

where y_t and ε_t represents the actual observation and error term at time t, respectively; $\psi A = 1 - \psi_1 A - \psi_2 A^2 - ... - \psi_p A^p$,

$\vartheta A = 1 + \vartheta_1 A + \vartheta_2 A^2 + ... + \vartheta_q A^q$ are lag polynomials, and A denotes the lag operator, such that $Ay_t = y_{t-1}$. p, q are the order of model, whereas $\psi_1, \vartheta_j i = 1, 2, ..., pj = 1, 2, ..., q$ are parameters of the model. c is the degree of ordinary differencing applied to ensure that the series is stationary. The correct orders of the ARIMA (p, d,q) model are obtained through the Box-Jenkins model building process (Zhang, 2003). Owing to the limitations of ARIMA model, such as linearity restriction, an ARIMA model is not suitable for modeling time series, like commodity market price series.

3.9 Conclusion

This chapter presented the theoretical background of the suggested methods used in this study - the EMD and the VMD techniques. The Back-propagation neural network and autoregressive integrated moving average were introduced as benchmark and comparative models. It was revealed that BPNN could estimate any non-linear continuous function to any desirable level, compared to the traditional methods.

Chapter 4

Data and Statistical Properties of the Three Commodities Futures Markets' Prices

Chapter summary

This chapter presents the procedure adopted in selecting the data for this study, a descriptive analysis of the data, and the statistical properties of the three commodities markets prices' series. As mentioned in the previous chapters, the commodity market price data is probably non-linear and non-stationary because the commodity market is extremely mixed and noisy, therefore, the statistical properties of commodity market price series is vital in its modeling. Furthermore, the commodity market price series exhibit non-linear and non-stationary characteristics that affect the type of model to be considered when modeling its volatility. The descriptive analysis of the data, stationarity and linearity tests of corn market price series, crude oil market price series, and gold market price series are investigated. Causality analysis among the three commodities is also explored.

4.1 Data and Data Source

Day-to-day market prices of corn, crude oil, and gold obtained from the Bloomberg commodities index were employed to carry out this study. The Bloomberg commodities' market index which is regarded as the finest commodity markets index globally, was utilized in analyzing futures prices of the three commodities

markets in this thesis. The data point consists of 1277 observations of each commodity market price from May 2016 to April 2021. The data of the three markets under consideration were secured from (http//: www.bloomberg.com.).

The corn and gold markets price series exhibit positive skewness, hence, the distributions of the two commodities market prices have a right tail. The crude oil market price was negatively skewed, which suggested that the price series of crude oil possess a left tail, as appeared in Table 4.1. The maximum and the minimum values of the date points and the time series representation validate the results of the market prices of the selected commodities as represented by Figures 4.1-4.3. The three markets series revealed a Kurtosis values of 11.98, 4.05, and 2.59 respectively, suggesting that the distribution of day-to-day commodity market is leptokurtic. The soaring Jarque-Bera values illustrate that the distribution of commodity market prices is not normal, thereby, substantiating the fact that the corn and the gold markets price series depict right skewness, while the crude market price series show left skewness, as illustrated by the time series graph in Figures 4.1-4-3.

Commodity	Mean	Std dev	Skew	Kurt.	JB	Min	Max
Corn	3.8000	0.5800	2.6900	11.9800	5830.3000	3.0200	7.5300
Crude oil	52.8000	10.6200	-0.6100	4.0500	137.1700	10.0100	76.4100
Gold	1429.0000	232.8900	0.9900	2.5900	215.6600	1128.0000	2069.0000

Table 4.1: Descriptive statistics of the three commodities

Commodity	Time	Sample size	Training	Testing
Corn	2016-2021	1277	1022	255
Crude oil	2016-2021	1277	1022	255
Gold	2016-2021	1277	1022	255

Table 4.2: Time and sample size of the three commodities futures prices

4.2 Time series plots of the three commodities

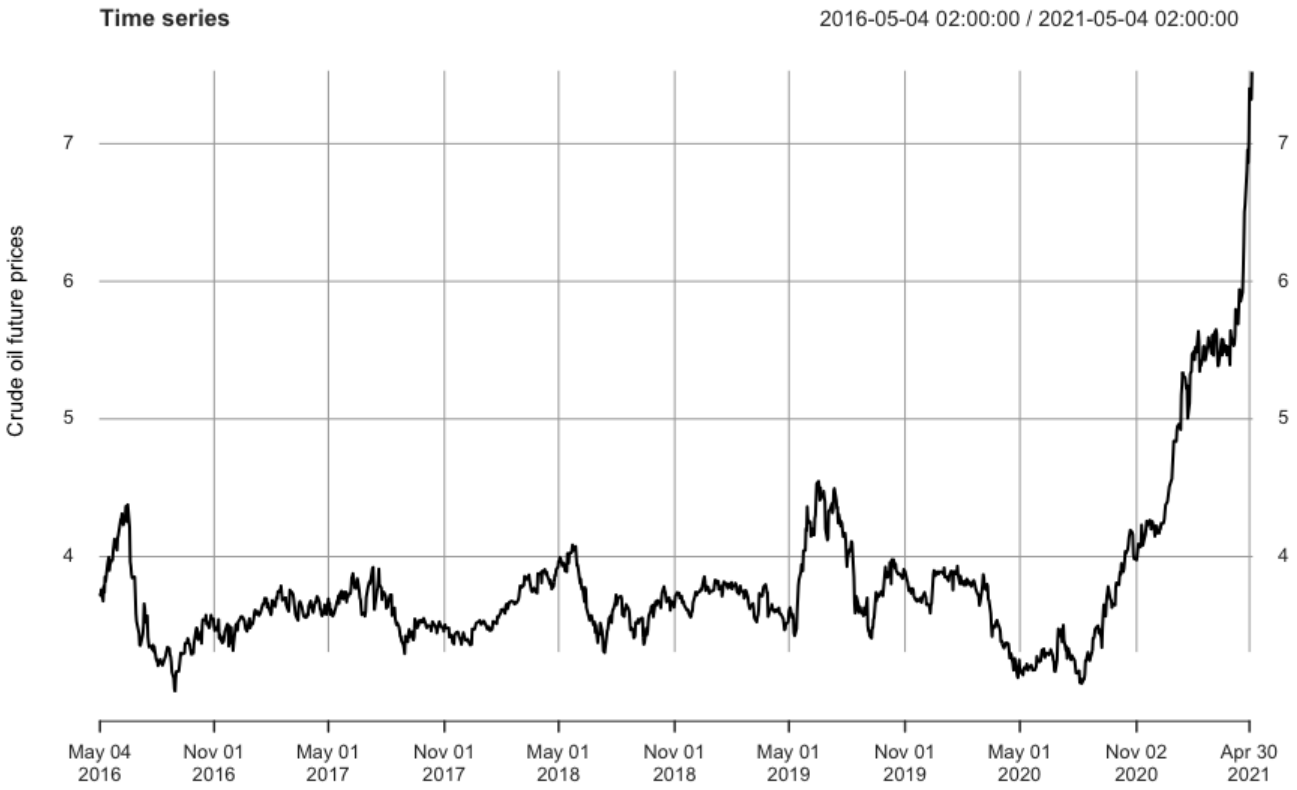


Figure 4.1: Futures prices series of corn (2016-2021)

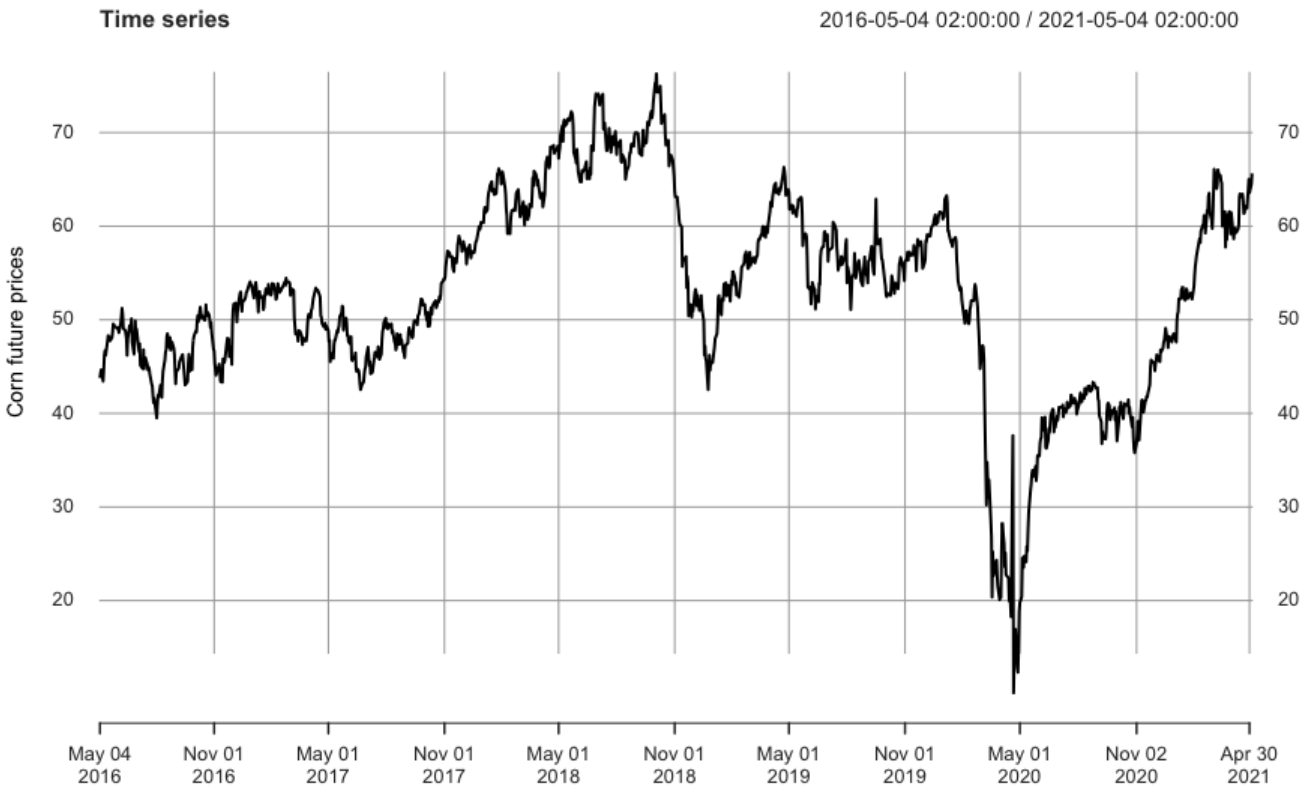


Figure 4.2: Futures prices series of crude oil (2016-2021)

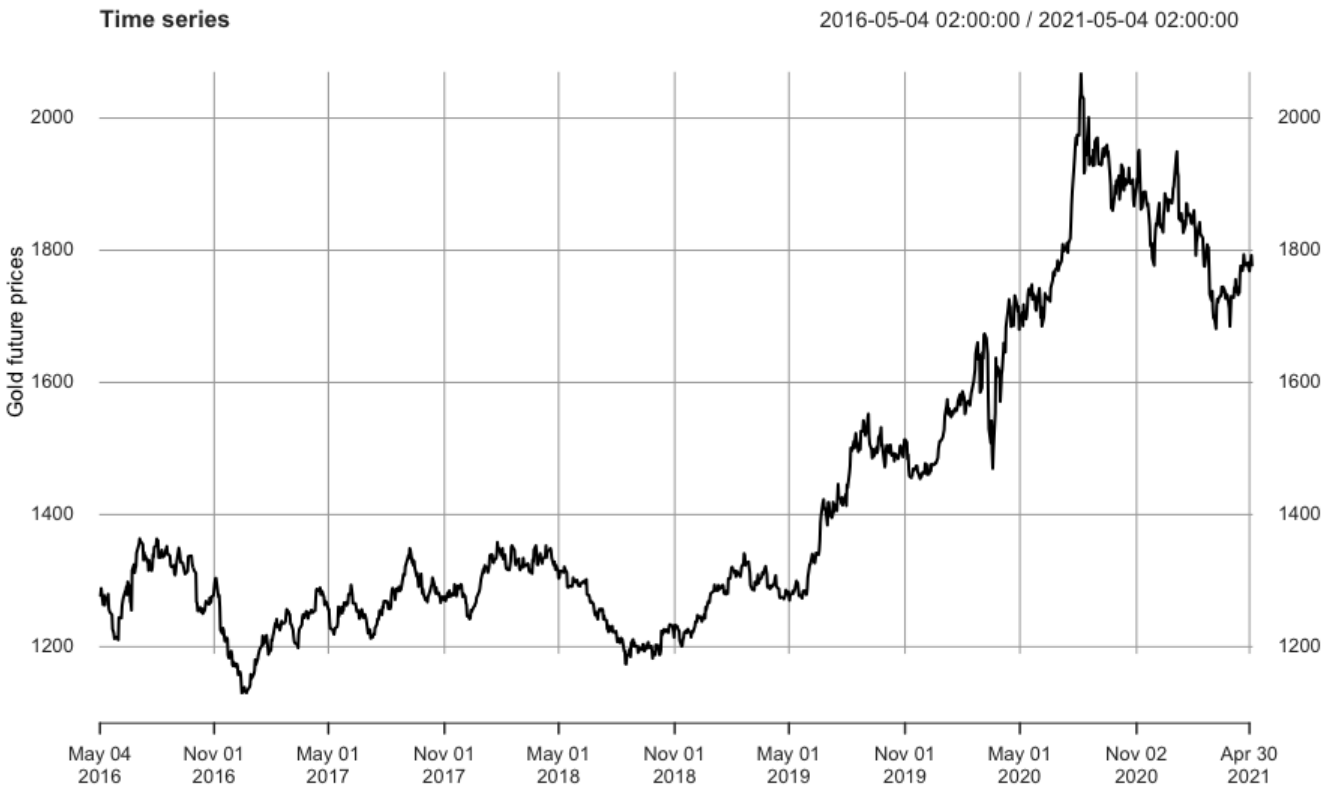


Figure 4.3: Futures prices series of gold (2016-2021)

4.3 Testing for Nonlinearity in the Commodity Markets prices

Investigations have revealed that commodity market price data is nonlinear, and this has dire impact on the type of method to be considered when modeling its price volatility. As a result, it is vital to test for the existence of nonlinearities in the market price series. A couple of practices have been found in the literature in testing for the existence of nonlinearity in a given data, however, we adopted the Keenan test and Tsay test due to their flexibility in their application. The two techniques have abilities to discover the memory effects to avoid multicollinearity in the series.

4.4 Keenan Test

Financial times data, such commodity market price data, exhibit non-linearity characteristics. Keenan (1985) suggested a non-linearity test in detecting for the presence of nonlinearity in the financial time

series under the hypothesis:

$$\begin{cases} H_0 : linearity \\ H_1 : nonlinearity \end{cases}$$

We adopted the model of Cryer and Chan (2008) in introducing the Keenan test. Akaike information criterion (AIC) was applied in selecting d, the autoregressive order, on the assumption that the H_0 process is linear. The Keenan test is originated from an expansion of second-order Volterra process. The application of Volterra expansion and Taylor processes are similar in operation, but the Volterra expansion is good for testing nonlinear time series data.

The Keenan test process can be estimated as:

$$x_t = \mu + \sum_{i=-\infty}^{\infty} \theta_i \delta_{t-i} + \sum_{i.j=-\infty}^{\infty} \theta_{i,j} \delta_{t-i} \delta_{t-j} + \sum_{i,j,k=-\infty}^{\infty} \theta_{ijk} \delta_{t-i} \delta_{t-j} \delta_{t-k} + ..., \tag{4.4.0.1}$$

in which δ_t, and $-\infty < t < \infty$

follows i.i.d. variables with zero mean. $x_1, ..., x_n$

represents the observed data points.

The procedure x_t is linear if $\sum_{i=-\infty}^{\infty} \theta_i \delta_{t-i} + \sum_{i.j=-\infty}^{\infty} \theta_{i,j} \delta_{t-i} \delta_{t-j} = 0$

On the other hand, Keenan (1985) suggested another approach for estimating the test as:

$$x_t = \theta + \psi_1 x_{t-1} + ... + \psi_l x_{t-l} + exp[\eta(\sum_{j=1}^{l} \psi_j x_{t-j})^2] + \delta_t, \tag{4.4.0.2}$$

in which the mean of δ_t is zero with a constant variance. If the regression coefficient,

$\eta = 0$, then the index term turns to 1 and it is incorporated in the intercept and the previous model changes to autoregressive model AR_m.

If the regression coefficient is $\eta \neq 0$, then the previous model is classified as a nonlinear. By simplifying $exp(c) \approx 1 + c$ term in the above expression, which is also valid for trivial value of c, it can be noticed that for small increase of η, x_t turns to a quadratic AR model, can be noticed that for small increase of η, x_t turns to

$$x_t = \theta + 1 + \psi_1 x_{t-1} + ... + \psi_l Y_{t-l} + \eta\{(\sum_{j=1}^{l} \psi_j x_{t-j})^2\} + \delta_t \tag{4.4.0.3}$$

Tsay Tsay (1986) identified flaws in applying Keenan's test in testing for nonlinearity presence in nonlinear data such as financial time series. Its application is less effective for nonlinear time series data, yet it is useful in detecting nonlinear features in the time series.

The Keenan test statistic is given:

$$T = \eta^2(\frac{d-2l-2}{rss-\eta^2}), \tag{4.4.0.4}$$

where T follows F-distribution with degrees of freedom 5 and $(d-2l-2)$.

4.5 Tsay's Test for Nonlinearity

Due to the shortcomings of the Keenan test, Tsay (1986) proposed an enhanced nonlinearity test, 'Tsay test' as an advanced method of testing for the existence of nonlinearity in nonlinear time series; once again, we follow Cryer and Chan (2008) in presenting the Tsay process.

The Tsay test enhanced the Keenan's technique by introducing the expression:

$\eta\{(\sum_{j-1}^{l}\phi_j x_{t-j})^2\}$

to

$exp(\varsigma_{1,1}x_{t-1}^2 + \varsigma_{1,2}Y_{t-1}x_{t-2} + ... + \varsigma_{1,l}Y_{t-l} + \varsigma_{2,2}x_{t-2}^2$

$+q_{2,3}x_{t-2}Y_{t-3} + ... + \varsigma_{2,l}x_{t-2}x_{t-l} + ... + \varsigma_{l-1,l-1}x_{t-l+1}$

$+\varsigma_{l-1,l}x_{t-l+1}q_{t-l+1}x_{t-l} + \varsigma_{l,l}x_{t-l}^2 + \delta_t$

By estimation, the nonlinear model turns to a quadratic AR model, along with unrestricted coefficients of the quadratic expressions.

The Tsay test process follows the quadratic regression:

$$x_t = \theta_0 + \phi_1 x_{t-1} + ... + \phi_l x_{t-l} \tag{4.5.0.1}$$

$+\varsigma_{1,2}x_{t-1}^2 + \varsigma_{1,2}q_{t-1}x_{t-2} + ... + \varsigma_{1,l}x_{t-1}x_{t-l}$

$+\varsigma_{2,2}x_{t-2}^2 + \varsigma_{2,3}x_{t-2}x_{t-3} + ...\varsigma_{2.l}x_{t-2}x_{t-l} + ...$

$+\varsigma_{l-1,l-1}x_{t-l+1}^2 + \varsigma_{l-1,l}x_{t-l+1}x_{t-l+1} + \varsigma_{l,l}x_{t-l}^2 + \delta_t$

and test if all

$l(\frac{l+1}{2})$ coefficients $\zeta_{ij} = 0$.

4.6 The results of the nonlinearity test

The two nonlinearity tests, Keenan's and Tsay's, revealed that day-to-day market price series of corn, crude oil, and gold are nonlinear, and this might be caused by economic factors. Table 4.3 presents the nonlinearity test results of the market prices of the three commodities. To choose a forecasting method that can accurately account for the nonlinearity patterns in the market price series of these commodities, this current study proposed a decomposition-based methods to study market prices fluctuation of the aforementioned commodities markets.

	Keen test	
	Test statistics	p-value
corn	23.3396	1.523e-06
crude oil	2.7635	0.0967
gold	0.9615	0.3270
	Tsay test	
corn	28.9700	8.733e-08
crude oil	3.0370	3.114e-42
gold	4.2330	0.0398

Table 4.3: Nonlinearity test results of the three commodities price series

4.7 The results of stationarity test

Stationarity tests were conducted on the corn, crude oil, and gold markets prices series, as illustrated in Table 4.4. The unit root test of the three markets prices by both Augmented Dicky-Fuller (ADF) and the Kwiatkowski-Phillips Schmidt-Shin (KPSS) tests, indicated that the three markets prices series are nonstationary. The evidence of nonstationary in the price series of the three markets suggested that forecasting methods that can cater for nonstationary are necessitated for accurate estimation, explanations and decision-taking.

4.8 EMD-Granger causality investigation

We introduced the EMD-Causality analysis to study the causal interdependence among the three commodities markets prices. Table 4.5 presents the EMD-causality test results of corn, crude oil, and gold

	ADF test	
	Test statistics	p-value
corn	1.8372	0.9900
crude oil	-2.2951	0.4534
gold	-1.7923	0.6662
	KPSS test	
corn	0.9412	0.0100
crude oil	1.6631	0.0001
gold	2.9431	0.0100

Table 4.4: Stationary test results of the three commodities price series

futures markets prices and their corresponding ρ values. The relationship between commodity market futures prices is determined by the value of ρ; for example, an ρ of 0.0140 suggests that there is EMD-causal relationship between crude oil and corn at the 5% significance level. This value implies that the crude oil market futures prices' movement could be used to describe the movement of corn market futures. It further suggests that from the knowledge of the crude oil futures market, one could understand the features of corn futures prices' movement. This relationship is the reason why agriculture market prices have been consistently affected by the crude oil market prices. Figures 4.1 and 4.2 indicate that crude oil prices go up continuously, thereby increasing corn price or if there was unprecedented corn production; this signifies an EMD-causal relationship between the two commodity futures prices. A ρ value of 0.0690 indicates that corn futures prices can also determine the futures of gold prices, however, the combined futures of crude oil and gold at the 5% level have an insignificant impact on corn futures.

In the case of crude oil futures, the gold futures is the EMD-causal of crude oil at the significance level of 0.05. It can be observed from Table 3 that the combination of market futures prices of gold and corn do not suggest any information about the futures prices movement of crude oil market futures because the ρ-value of 0.3220 is more than the threshold value of 0.05 at the significance level of 0.05. Finally, the gold price is relatively stable in EMD-causal, since gold is a commodity usually used to hedge against inflation.

	Corn	Crude oil	Gold
Corn		0.0690*	0.718 0
Crude oil	0.0140**		0.7300
Gold	0.0710**	0.0050**	
Corn+crude oil			1.0000
Corn+gold		0.3220	
Crude oil+gold	0.7310		

Table 4.5: EMD-causality test Results: Causality test is significant at 0.01 '*' 0.05 '**.' 0.1 ' ***' level

4.9 Conclusion

We investigated descriptive analysis and the statistical properties of the three commodities' markets in this chapter. The market price series of corn and gold price skewed to the right, while the price series of crude oil was skewed to the left, suggesting that the three commodities' markets price series were not normally distributed. It was also emerged that the daily spot prices of the three commodities were leptokurtic. The Jarque-Bera test revealed that the three commodities markets prices' series deviated from normal distribution. We also tested for the non-linearity and non-stationarity of the three commodities price series, utilizing the Keenan, Tsay, and ADF tests. The ADF examination results disclosed that the corn, crude oil, and gold price series were not stationary. It also emerged from the results of the Keenan and Tsay tests that the three commodities markets prices were nonlinear. Finally, the EMD-Granger Causality examination was introduced to analyze the causal relationships among three commodities markets prices. The results showed strong mutual relationships among the three commodities markets. This was the reason crude oil market prices have persistently affected the agricultural and industrial-based metal commodities' markets.

Chapter 5

Determinants of three commodities futures markets' prices, through EMD and VMD

Chapter summary

We applied the Hierarchical Clustering and Euclidean Distance Techniques based on Empirical mode Decomposition (EMD) and Variational mode Decomposition (VMD) to group- decomposed IMFs and modes to analyze the drivers of the three commodities futures markets prices series. We first utitilzed EMD and VMD to break down all the market price series into IMFs and modes and used the Hierarchical Clustering and Euclidean Distance Methods to classify the series into three frequency components: high, low, and trend. Subsequently, we applied statistical estimates, specifically, the Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation coefficient to calculate the correlation between the observed series and each IMF and the mode. This chapter combines the decompositions. It was found that the trend and low-frequency components are the major drivers of the commodity future market price variations. The low-frequency is caused by special events, while the trend occurs as a results of global economic market price movements. Overall, worldwide economic activities are the main influencers of commodity market futures prices' fluctuation, as compared to usual supply-demand disequilibrium caused by short-lived market price variations.

5.1 Introduction

The United States of America, China, and the European Union are regarded as the world's leading economies. These countries are also considered as the major consumers of corn, crude oil, and gold globally (Zou et al., 2007). This suggests that the aforementioned commodities play a very essential role in the day-to-day organization of these countries' economies. The USA is the primary consumers of corn worldwide. In 2019 / 2020 farm season, the US alone consumed 12.30 billion tons of corn, followed by China and European Union with annual consumption of 10.98 billion and 55,500 thousand tons respectively. The International Energy Agency (IEA) 2019 report shows that the US and China are the principal consumers of crude oil in the world; they purchased roughly 19.4 million and 14 million barrels per day respectively. Gold is considered as the most significant metal among the precious metal industry. It is used to estimate wealth worldwide and can retain its value over a long periods of time. It is a type of commodity used to hedge against inflation in the global market. The high-cost of gold is due to its expensive production cost associated with its extraction, how much people are ready to buy, it is exceptionally popular and can retain its value for centuries. From the futures prices' discovery mechanism perspective, these commodities are influenced by macroeconomic schemes, hence, the market price of these commodities are very crucial in supplying information to physical markets. Studying the components that drive the price fluctuations of corn, crude oil, and gold, therefore, is anticipated not to keep down the instability and reduce the risk in commodity markets, rather, to assist in serving as foundations for prudent and well-grounded programs for states.

Crude oil, corn, and gold commodities prices were selected over the commodity price market in carrying out this study; this was intended to discover the fundamental factors that cause the futures prices variations in energy, agriculture, and industrial metal markets, since these commodities futures prices are volatile, interdependent and complex in the commodity price market and responsive to macroeconomic schemes (Zou et al., 2007). In addition, the above-mentioned commodities play crucial parts in the worldwide economy, therefore, investigating the futures prices of these commodities market price series is vital for countries' future development plan since the price, supply and demand are mutually related.

Motivated by the above stated reasons, our investigation suggested an advanced signal identifier, EMD and VMD strategies, to break down the market price series of crude oil, corn, and gold to discover the elements that trigger commodity futures markets' prices. This thesis, therefore, employs EMD and VMD methods to study the constituents of futures prices of corn, crude oil, and gold from the agricultural, energies, and the industrial metal markets, respectively.

The EMD and VMD belong to data-driven techniques and the IMFs generated are localized (Huang et al.,

1998, Dragomiretskiy and Zosso, 2013). These methods, therefore, are suitable for studying non-linear and non-stationary time series, such as commodity market price data, which can reveal the underlying factors that cause commodity market price fluctuations. In addition, these methods are efficient, empirical, and local in analyzing commodity market price series. As mentioned in Chapter Two, these promising techniques have been applied in many study areas, however, few researchers have utilized novel approaches in commodity market, particularly corn, crude oil, and gold.

By analyzing the constituents that cause commodity futures markets prices of these exclusive commodities markets, this thesis extends the available studies in these ways:

1. Affirm that the EMD and VMD techniques are capable of revealing the cardinal components that lead to corn, crude oil, and gold markets price variations.

2. Categorized the IMFs and modes into three frequencies - high, low, and trend - using the combination of Hierarchical Clustering and Euclidean Distance Approaches.

3. Consequently, utilized distinct statistical methods to estimate the contribution of individual frequency to the overall variance of the selected commodities markets prices' volatility.

4. The economic explanations of the classified frequencies were given as short-live variations caused by ordinary disequilibrium of supply-demand, unforeseeable significant events, and a long-term trend. It can, thus, be inferred that economic growth is the prime factor of commodity futures prices' fluctuations, since low-frequency and trend are the primary factors of commodity markets prices series. By these establishments, efficient and reliable predicting methods are determined.

The rest of this chapter is divided into the following: Section 2 describes the EMD and VMD processes; Section 3 discusses the decomposition results of each commodity using EMD and VMD respectively and Section 4 is dedicated to an analysis of the IMFs and modes generated by the EMD and VMD techniques.

5.2 EMD and VMD Process

One major contribution from this thesis is the utilization of the EMD and VMD methods to discover the elements that cause commodity market futures prices' variations. We first used the EMD and VMD strategies to pre-process each commodity price series to find commodity futures price components. The actual price series of each commodity market futures' prices was disintegrated into several IMFs and a residue, RES, using the EMD approach. The extracted IMFs produced by EMD is constant for a given data series, that is, the IMFs decomposed by the EMD approach cannot be altered after the sifting process. This

is in contrast to the VMD approach where the decomposed modes can be changed to suit the requirements of the study. The EMD and VMD processes are given in the Figures 5.1 and 5.2 respectively.

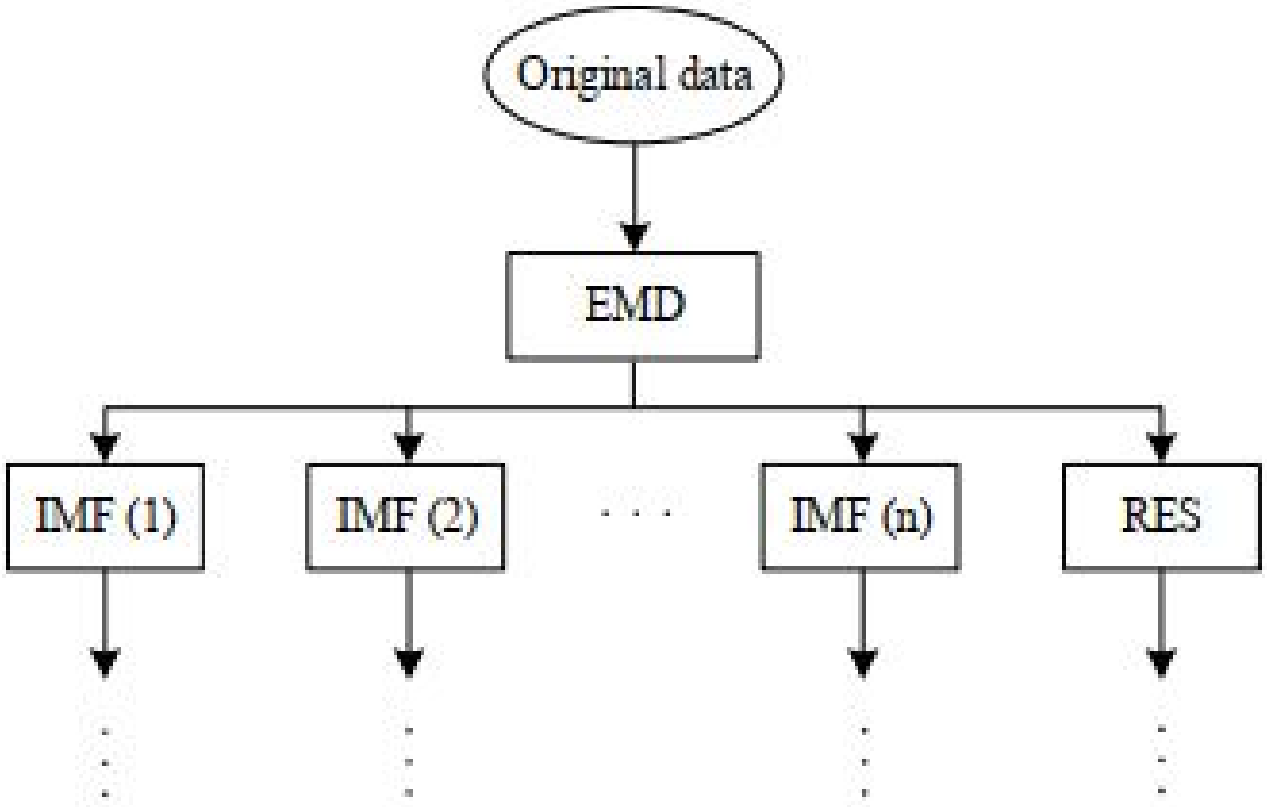


Figure 5.1: EMD process

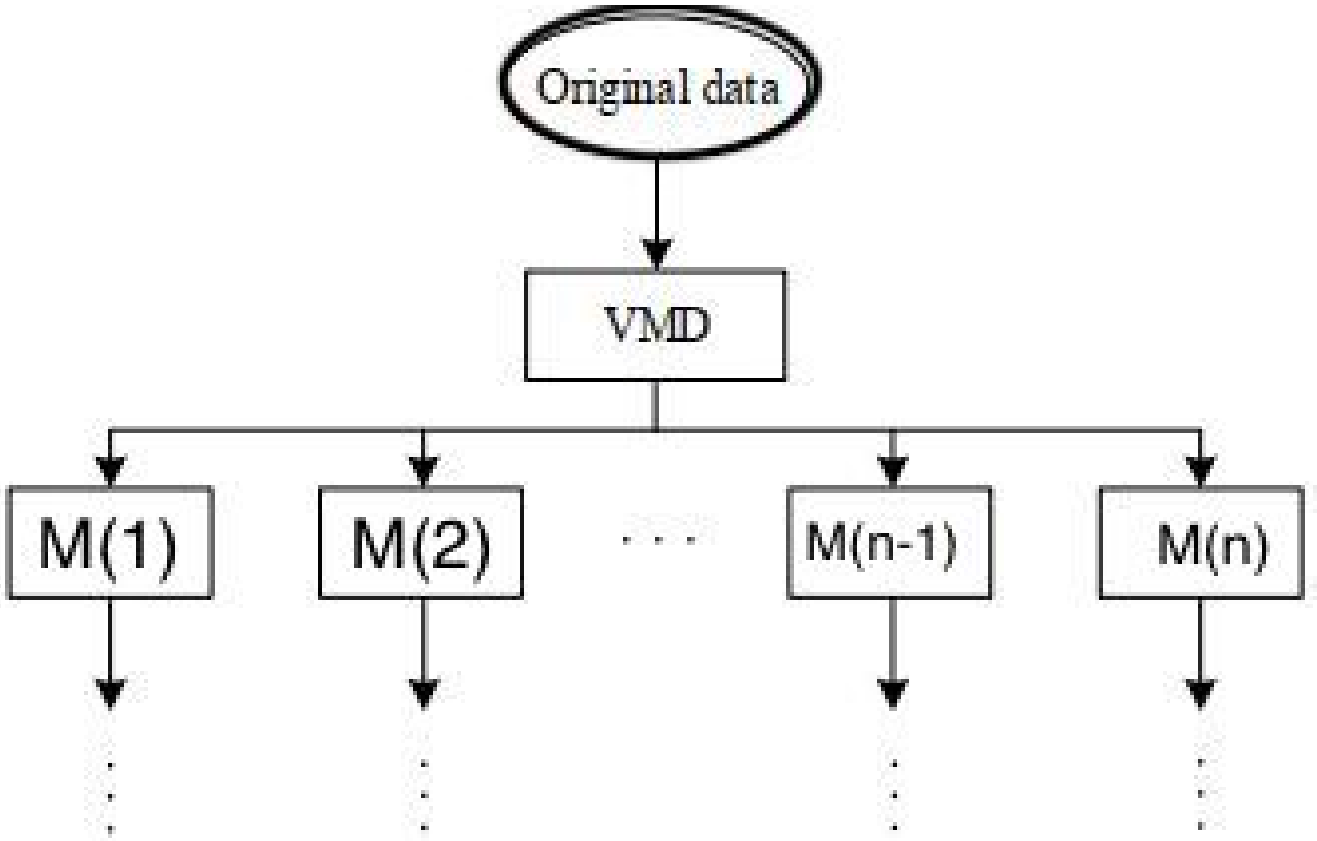


Figure 5.2: VMD process

5.3 Decomposition results of corn through EMD and VMD

The sifting process by EMD method automatically decomposed the corn price series into seven IMFs and one residue, expressed as IMF1, IMF2, ..., IMF7, as given in Figure 5.3. The IMFs are arranged in the order of: high-frequency, low-frequency, and the residue; that is, the IMFs are organized in order of intensity. These extracted IMFs and a residue denote the price fluctuations describe as a “trend” and market price instabilities, resulting from demand-supply, and unforeseeable events.

Likewise, the VMD strategy was applied to break down the corn price series into eight nodes. Figure 5.4 presents the decomposition results produced by the VMD method. Unlike EMD, the modes arranged themselves in descending order, thus, from low-frequency mode to high-frequency mode. The analysis compared eight distinct decomposition modes, describe as, M1, M2 to M8, separately.

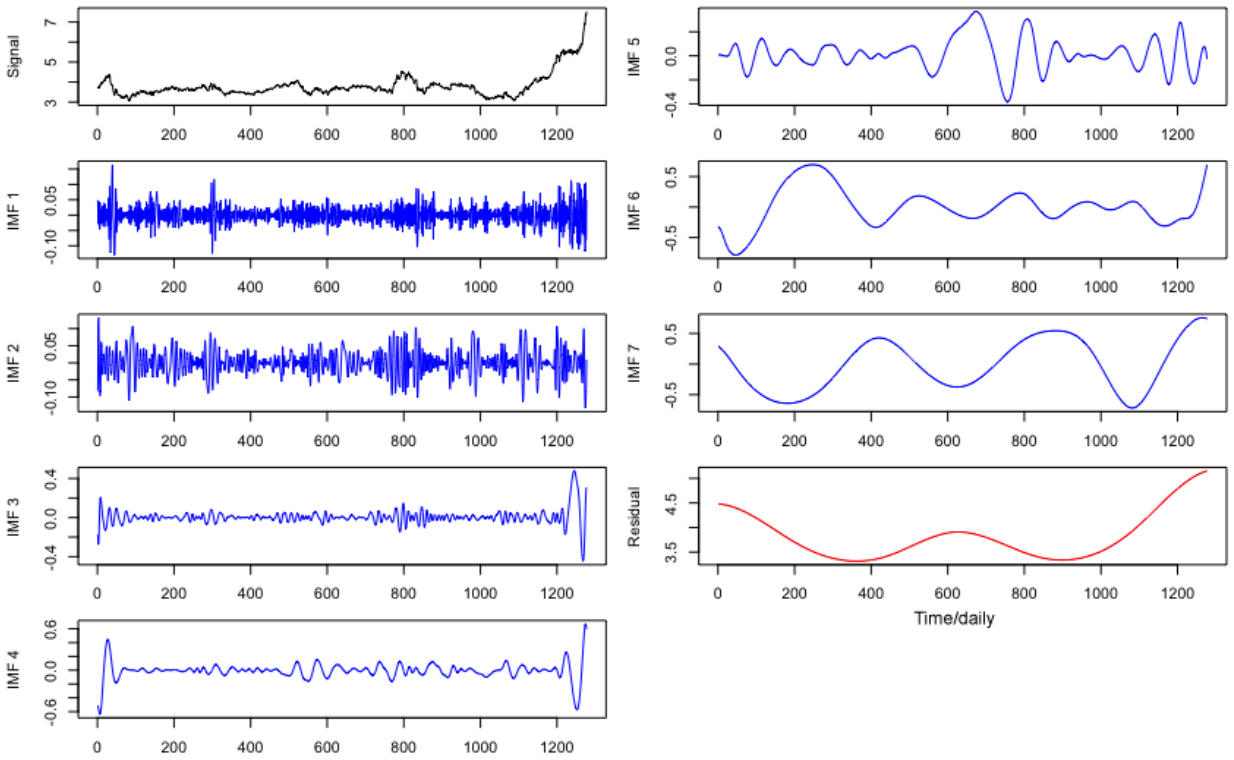


Figure 5.3: Empirical mode decomposition (EMD) curves for daily corn price series (May 2016-April 2021)

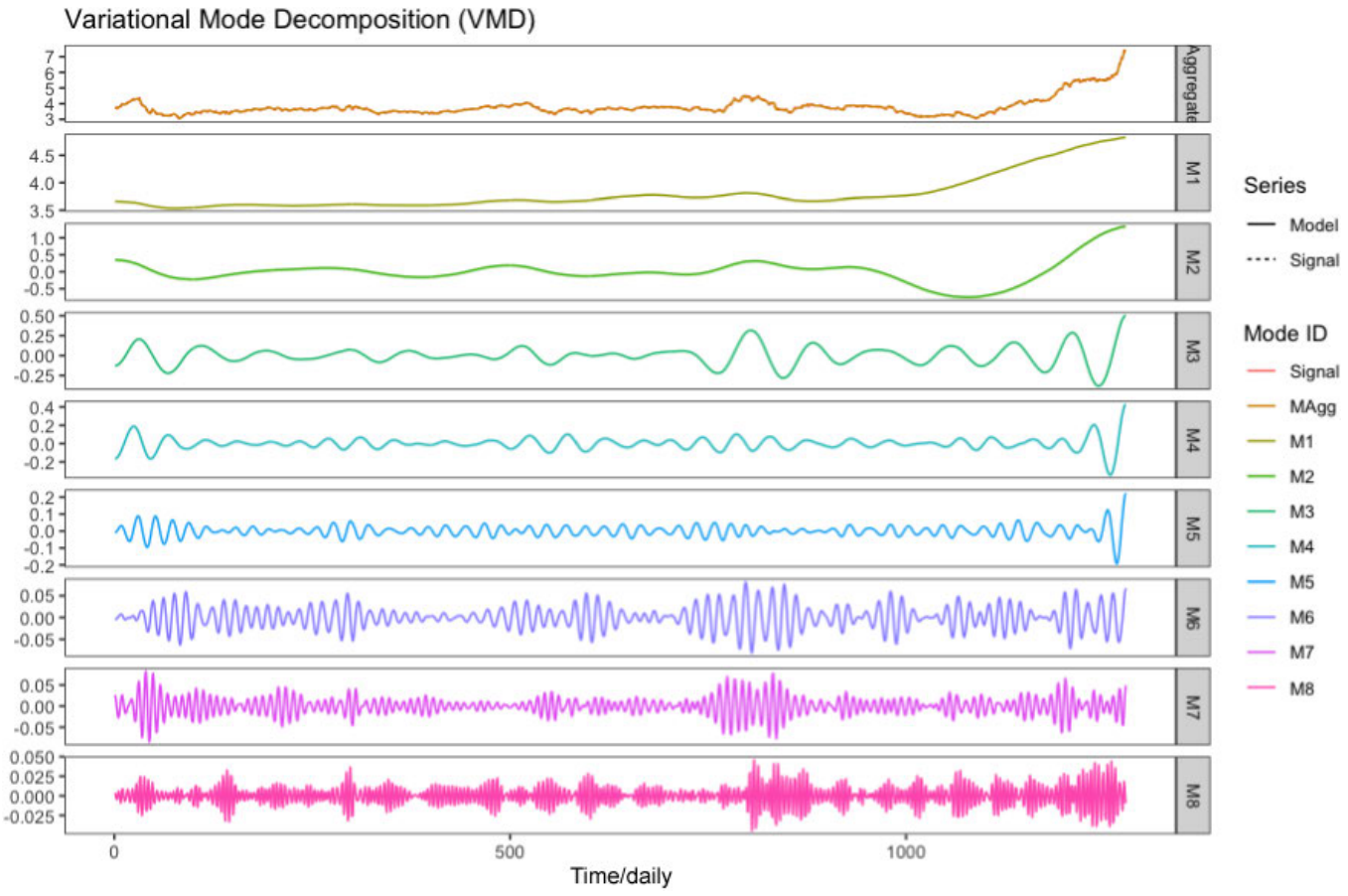


Figure 5.4: Variational mode decomposition (VMD) curves for daily corn price series (May 2016-April 2021)

5.4 IMF analysis of corn-obtained through EMD technique

The following produces are adopted in analyzing the IMFs: mean term of each IMF, the correlation between each IMF and the observed price, the variance, and the variance percentage of each IMF. The estimated values of the IMFs and modes generated by EMD and VMD methods for corn price series are shown in Table 5.1 and Table 5.3, separately.

The mean term is referred to as 'value' estimated by dividing the entire price series by the extrema for each IMF since the periods are not constant and frequent, and the amplitude of an IMF changes with time. We utilized the three correlation coefficients - the Pearson product-moment correlation coefficient, Kendall rank correlation coefficient, and the Spearman correlation coefficient - to evaluate the correlation between the IMFs and the original price series. The contribution of each IMF to the overall volatility of the original price series can be interpreted by adding up the variances and estimating the percentage of the variance

since the IMFs are free from each other. The sum of the variances of IMFs and the residue may differ from observed variance due to approximation in the calculations, an uneven in the observed price series, and the initiation of variance (Pegram et al., 2008). From Table 5.1, it can be seen that there is a positive 57.2450% difference and a negative 28.1570% difference from Table 5.3, respectively.

From the EMD technique, it can be seen that the dominant mode of corn futures prices' series was the residue with the mean period of 425.667, and this at the same time had the biggest correlation coefficient with the actual price; thus, the correlation coefficient between the residue and the original data attained a high point of 0.6148, 0.1078, and 0.1679, respectively. The residue accounted for over 41% of the overall variance of corn futures market price. The residue is responsible for long-term determination of trend of commodity futures prices (Huang et al., 1998). The IMF7 was the next remarkable mode, with the mean period of 212.8330, and the correlation between the actual price series of 0.5499, 0.3624, and 0.5118 in terms of Pearson, Kendall, and Spearman correlation coefficients; this donated more than 32% of the entire volatility of corn futures market prices.

The residue and IMF7 contributed over 73% of the total volatility of corn futures price. The IMF6 was the third important mode of corn futures price, which accounted for more than 18 % of the total variance of corn market price series. The IMF5 contributed over 3.823% of the whole variance of corn future price. The IMF4 and IMF3 donated 2.23% and 1.0760% of the variance of corn futures price series, respectively. The combined IMF1 and IMF2 contributed less than 1% of the variance of the corn futures price series.

5.5 Mode analysis of corn-obtained through VMD Technique

In the same manner, the VMD was utilized to decompose corn futures price series data into eight modes, as presented in Figure 5.4. The aggregate is the same as the observed data. It can be seen from Table 5.3 that the principal mode of the observed series was M2, which accounted for 50.56% of the overall volatility of corn futures price. The second eminent mode was M1, with an average period of 13.1650, which contributed over 40% of the net variance of the observed price series. Apart from M1and M2, M3, with an average period of 38.6970, provided over 5% of the whole variance of corn futures price. The contributions of M4, M5, M6, M7, and M8 to the entire variance of corn futures series, were insignificant, this can be observed from their low correlation coefficients between the original series. In descending order, M4-M8 contributed 1.7650%, 0.4360%, 0.2860%, 0.1870%, 0.0660% of the overall variance of the observed data, respectively.

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed series	2.0238				0.3353		
IMF1	1.6542	.0303	0.0580	0.0816	0.00103	0.3072%	0.1954%
IMF2	3.8580	0.0158	0.0555	0.0824	0.0013	0.3937%	0.2504%
IMF3	8.0315	0.1882	0.0842	0.1159	0.0057	1.6912%	1.0755%
IMF4	17.9859	0.1313	0.1608	0.2258	0.0149	4.4384%	2.8226%
IMF5	38.6970	0.1020	0.1992	0.2898	0.0182	5.4316%	3.4542%
IMF6	116.0909	0.0637	0.0232	0.0352	0.0984	29.3414%	18.6597%
IMF7	212.8333	0.5499	0.3624	0.5118	0.1698	50.6562%	32.2148%
Residue	425.6667	0.6148	0.1078	0.1679	0.2179	64.9854%	41.3274%
Sum						157.2450%	100.0000%

Table 5.1: Measures of IMFs and residue obtained through EMD of daily price series of corn (May 2016-April 2021): Correlation is significant at the level of 0.05 (2-tailed)

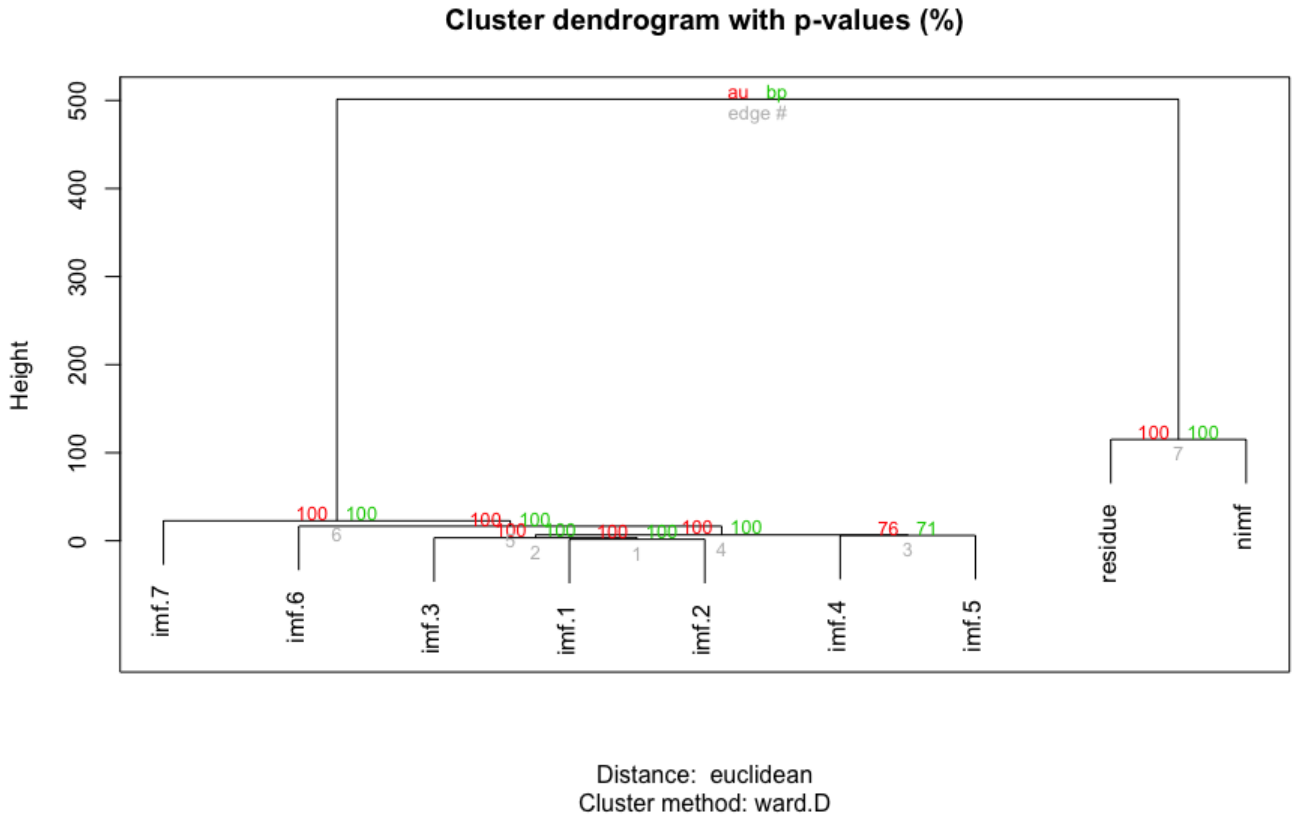


Figure 5.5: Hierarchical Clustering diagram obtained for the IMFs and residue through EMD

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed series	2.0238				0.3353		
High frequency	55.5217	0.5674	0.3988	0.5445	0.2137	63.7386%	45.5866%
Low frequency	32.7436	0.1535	0.2693	0.3720	0.03763	11.2241 %	8.0276%
Trend	425.6667	0.6148	0.1078	0.1679	0.2179	64.9854 %	46.4783%
Sum						139.9481%	100.0000%

Table 5.2: Correlation and variance of components obtained from the IMFs of EMD for daily price series of corn (May 2016-April 2021): Correlation is significant at the level of 0.05 (2-tailed)

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ modes
Observed series	2.0238				0.3353		
M1	13.1650	0.7698	0.3416	0.4408	0.0987	29.4309%	40.9657 %
M2	51.0800	0.8151	0.5652	0.7217	0.1218	36.3211 %	50.5563%
M3	38.6970	0.2742	0.2567	0.3341	0.0138	4.1222 %	5.7378%
M4	22.4035	0.1708	0.1499	0.2086	0.0043	1.2677%	1.7645%
M5	10.9145	0.0935	0.0812	0.1163	0.0011	0.3132%	0.4359 %
M6	7.3815	0.0721	0.0791	0.1140	0.0007	0.2058%	0.2865%
M7	4.7296	0.0567	0.0629	0.0895	0.0005	0.1342%	0.1868%
M8	2.6330	0.0392	0.0478	0.0687	0.0002	0.0477%	0.0664%
Sum						71.8430%	100.0000%

Table 5.3: Correlation and variance of components obtained from modes of VMD for daily price series of corn (May 2016-April 2021): Correlation is significant at the level of 0.05 (2-tailed)

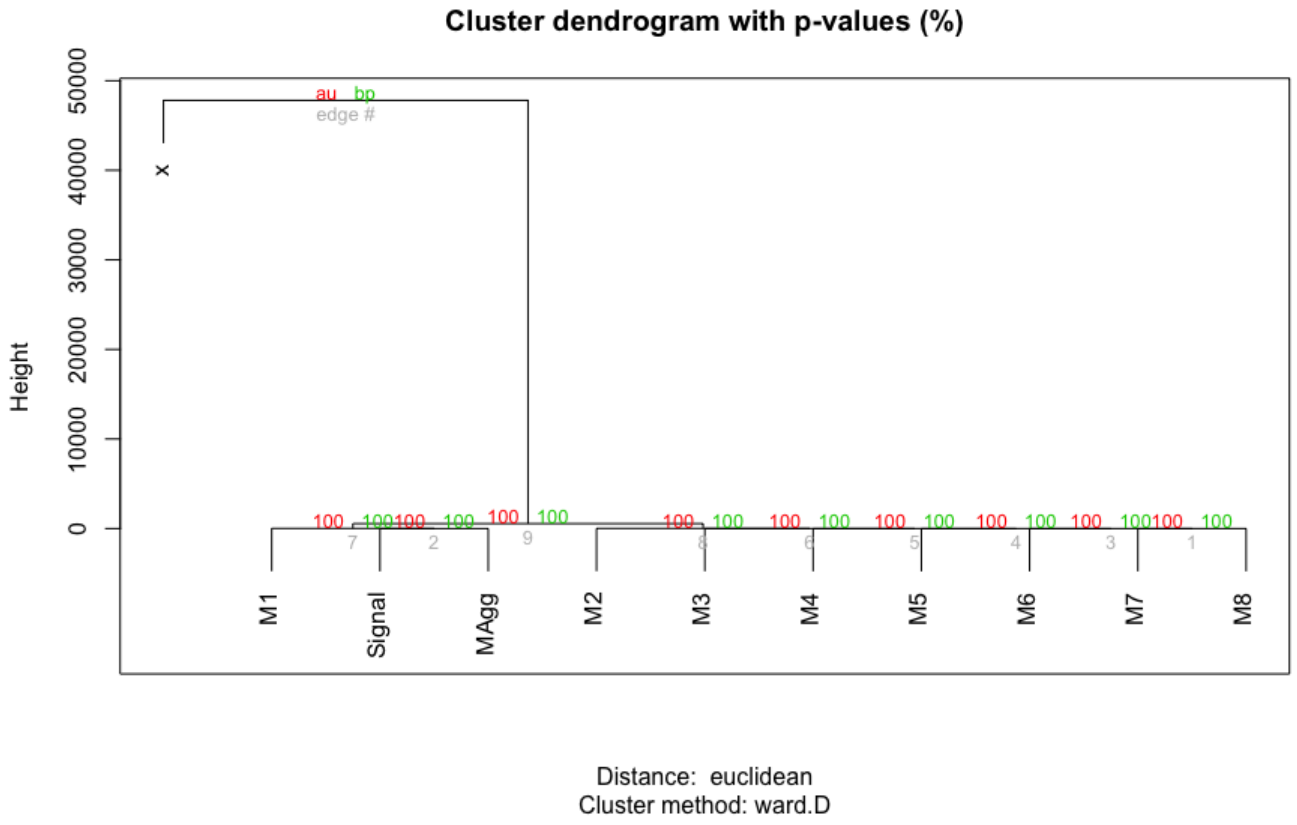


Figure 5.6: Hierarchical Clustering obtained for the modes and residue through VMD

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed series	2.0238				0.3353		
High frequency	4.7649	0.0943	0.1047	0.1504	0.0015	0.4325%	0.5809%
Low frequency	63.8500	0.8556	0.6676	0.7877	0.1495	44.5952%	59.8926%
Trend	13.1650	0.7698	0.3416	0.4408	0.0987	29.4309%	39.5265%
Sum						74.4586%	100.0000%

Table 5.4: Correlation and variance of components obtained from modes of VMD for daily price series of corn (May 2016-April 2021): Correlation is significant at the level of 0.05 (2-tailed)

5.6 Composition of corn price series

We adopted the Euclidean Distance and the Hierarchical Clustering Techniques in grouping the IMFs, modes, and residue into their respective components. In addition, we followed Zhu et al., (2015) in interpreting the IMFs, modes, and residue. The IMFs, modes, and residue were grouped according to the Euclidean Distance between the set of two IMFs, modes or a residue. If the Euclidean Distance between a set of two IMFs is less than 10, then the IMF is classified as high-frequency component, while an Euclidean Distance between the interval of 10-20 of two pairs of IMFs or residue is considered as low-frequency component. The Euclidean Distance between a set of two IMFs, modes or residue is described as trend component when the distance is more than 20. Table 5.2 presents the estimated contribution of each component to the overall variance of corn futures price series. Figures 5.5 and 5.7 represent the Euclidean Distance and the IMFs clustering and the three components obtained from the EMD approach, respectively.

By the above definition, IMF1-IMF3 and IMF6-IMF7 constitute the high-frequency components of corn future price series, while IMF4-IMF5 represent the low-frequency unit of the observed series of corn futures' price. These components have distinctive qualities and economic explanations - the high-frequency component represents the random events, while the low-frequency unit corresponds to short-term market price fluctuations. The residue changes gradually around the long-term mean, as stated earlier, and stand for long-term trend of corn futures prices.

For the case of VMD method, we discussed eight distinctive modes of corn futures market price series. M6-M8, with low amplitude, denotes the high-frequency part of the observed price series of corn and

constitute short-lived market variation. M2-M5 composes the low-frequency component of the original price series of corn and forms the random events, while M1 constitutes the long-term trend of the actual series of corn futures price. The grouping of the modes utilizing the Euclidean Distance approach is illustrated in Figure 5.5. A representation of the three components and their estimated values obtained from the VMD are given in Table 5.4 and Figure 5.6, separately.

5.7 Trend of corn futures prices

Trend contributed over 46.478% and 39.527% of the entire variance of actual price series of corn, in respect of EMD and VMD, and had the largest correlation coefficient with the observed price in terms of Pearson, Kendall, and Spearman, respectively. This suggests that trend is a very important component in determining corn futures prices, as given in Table 5.2 and Table 5.4, respectively. The continuous price variation of corn is a result of global economic growth, hence, it can be inferred that, although, the price of corn intermittently fluctuates due to random events like bad weather conditions, crop diseases, pest attack, and war (as happening now in Russia-Ukraine conflict), however, it returns to the trend as soon as the situation is over. This can be seen by comparing the variance of the trend with the observed prices' series of corn.

5.8 Effects of special events on corn price

Besides the trend, corn futures prices are affected by special events. The IMF6 and M2 form the low-frequency components of corn futures prices from the two decomposition techniques, respectively, and constitutes special events. The low-frequency arises from significant events such as unfavourable weather, pest attack, diseases, and conflicts, which have serious influence on futures prices of corn. The low-frequency accounted for more than 8% and 59% of the net variance of corn-price variation from EMD and VMD processes, separately. The three components of corn derived from EMD and VMD methods and their variance contributions are given in Tables 5.2 and 5.4, respectively. Based on an average period of the IMF and the mode, the lowest effect period lasted for more than 30 days and 60 days in terms of EMD and VMD, respectively. This suggest that the time span of these random events may last longer, and it takes some period for the market to adjust itself, considering the average period of these significant events. The graph of the three components of corn prices is illustrated in Figure 5.7 and Figure 5.8, separately.

In addition, some of the data points reached a high amplitude of 5.0USD, which indicates that a number of special events have dire impacts on the futures prices of corn. The high price variations of corn futures'

prices are due to special events, and occur at the medium term. The ordinary market price variation takes place at a high frequency, and the trend varies slowly. It is feasible to evaluate the effectiveness of each special event and split it from the actual price, and the results can be utilized to predict any similar events.

5.9 Effects of ordinary market disequilibrium on corn price

In addition to the trend and the significant events, futures market prices of corn is influence by normal market speculations. High-frequency occurs as a result of speculations within the market, financial schemes, trade wars, and unforeseen events such as an outbreak of pandemic. The high-frequency contributed over 45.5866% and 0.5809% of the entire volatility of corn price variation in respect of EMD and VMD. The ordinary market speculations are also referred to as “market disequilibrium” and responsible for short-term corn futures prices’ fluctuations. We referred to these events as short term, since the price series used in this study are day-to-day data. The disequilibrium of supply-demand has a significant effects on corn price. If these events occur continuously, they become one of the main drivers for pushing corn prices up. In the long-term trend prediction of corn futures’ price, the ordinary market variations can be neglected, although they are very important for short-term prediction. The corn price of 386,75USD per ton in the month of April 2021, thus, can be broken down as 106.81USD trend price, 242.56USD special event price, and 37.38USD ordinary market fluctuation derived from the VMD process.

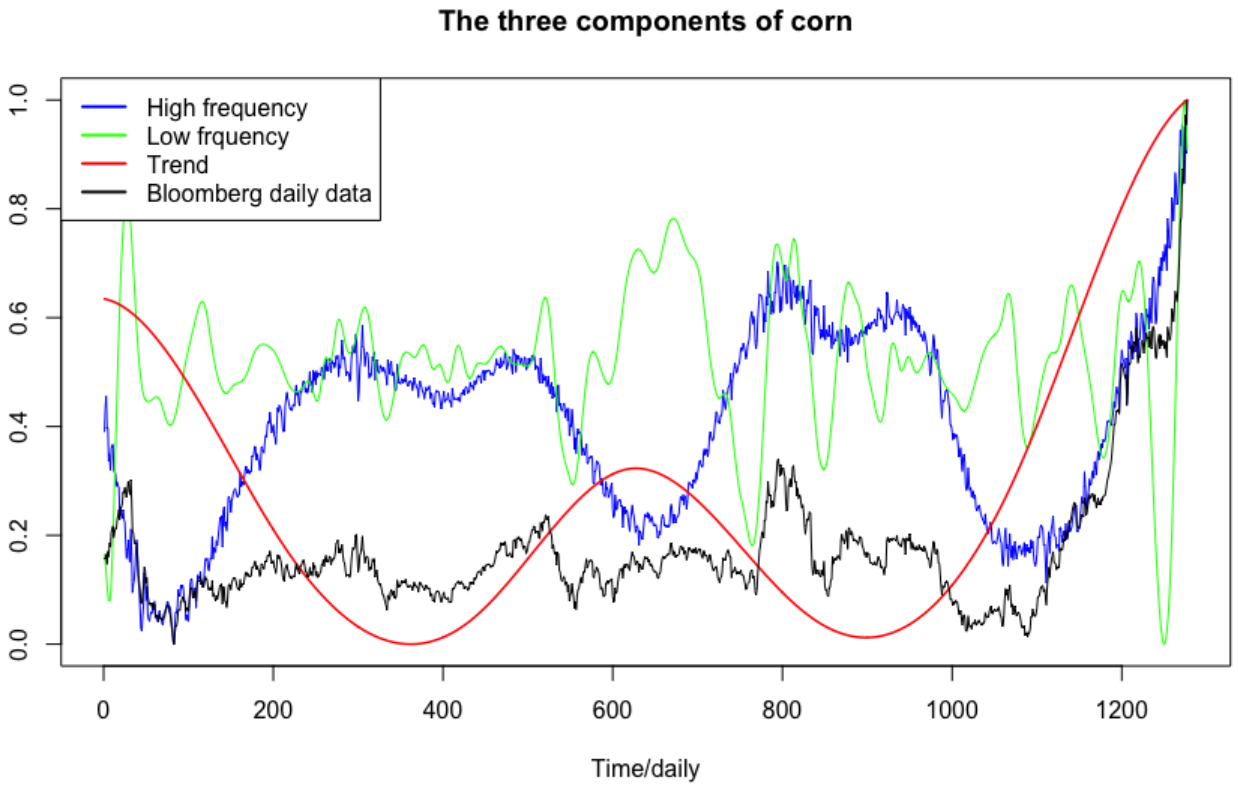


Figure 5.7: The three components of the Bloomberg daily data of corn (May 2016- April 2021) through EMD

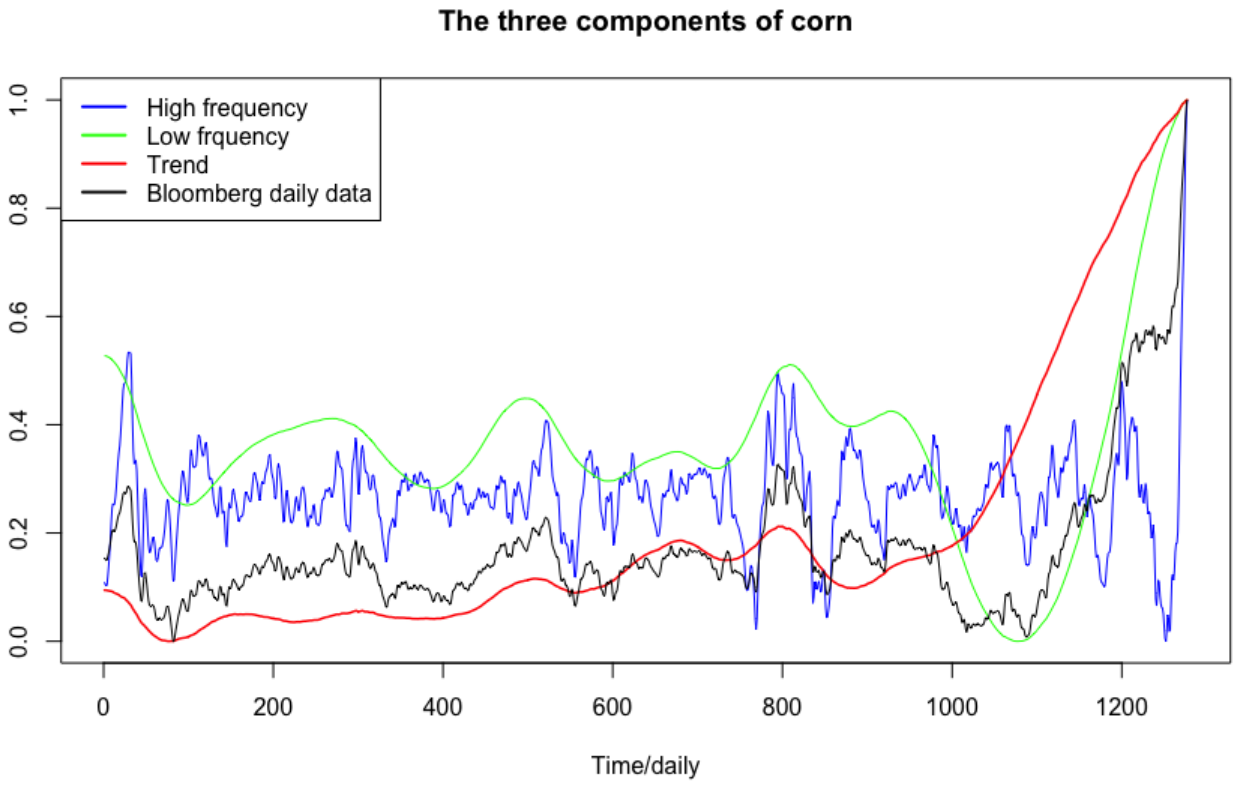


Figure 5.8: The three components of the Bloomberg daily data of corn (May 2016- April 2021) through VMD

5.10 Crude oil decomposition results

Again, the EMD and the VMD techniques were used to break down crude oil price data in the same way as that of corn price series. The EMD approach generated eight distinctive IMFs, described as IMF1-IMF8, and one residue, called RES, as presented in Figure 5.9. Interestingly, the VMD technique also decomposed the crude oil price series into eight modes, as illustrated in Figure 5.10.

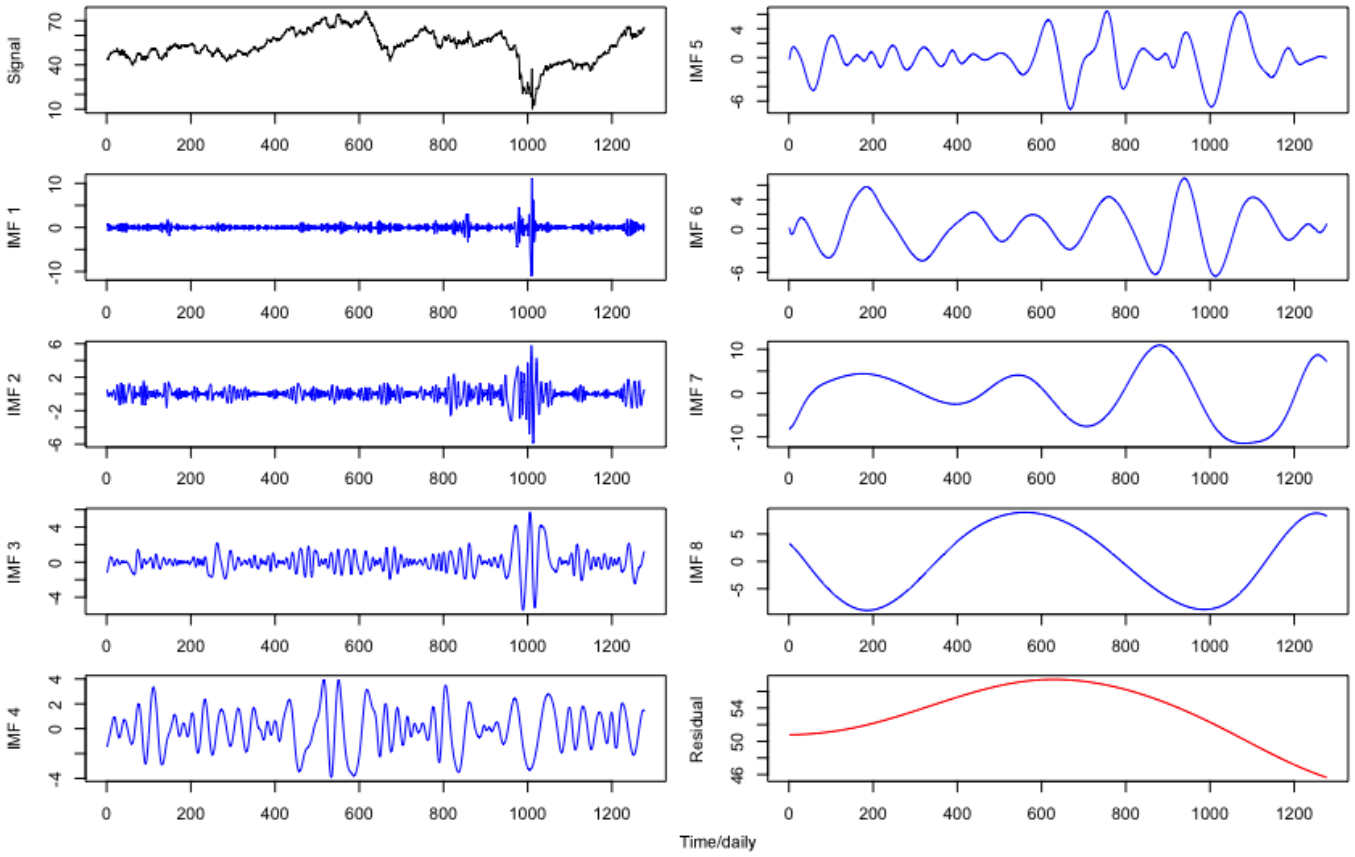


Figure 5.9: Empirical mode decomposition curves of daily crude oil price series (May 2016-April 2021

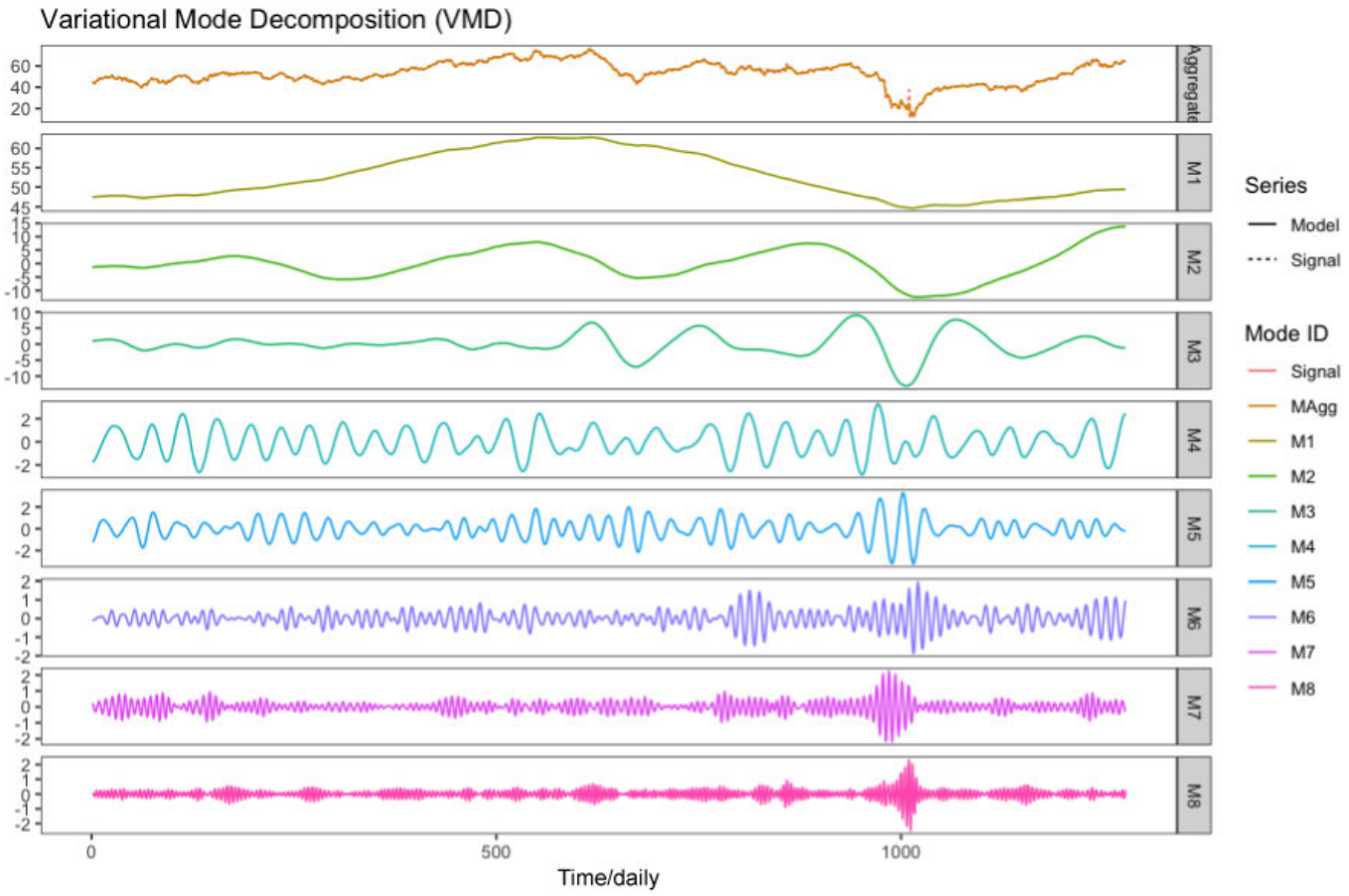


Figure 5.10: Variational mode decomposition curves of daily crude oil price (May 2016-April 2021)

5.11 IMF analysis of crude oil - obtained through EMD approach

We followed the same procedure and definitions of corn in analyzing the IMFs of crude oil price series decomposed by the EMD method. Again, we followed the interpretation of zhang et al., (2008) in estimating each IMF contribution to the entire variance of the crude oil price series. The IMF8 appeared as the leading mode of futures price series of crude oil, and accounted for over 37% of the whole variance of the actual price series, with an average period of 319.2500, and has the biggest correlation coefficients between the original series of 0.6150, 0.4260, and 0.6040 with respect to Pearson, Kendall, and Spearman, as given in Table 5.5. The IMF7 was the next significant mode of crude oil futures price series, which donated over 32% of the actual data's variability. The IMF8 and IMF7 were considered as the two most important modes in the crude oil price series, and contributed over 69% of the net variance of crude oil price volatility. The contribution of IMF5 and IMF6 to the total variance were 6,0792% and 8.6472%, respectively. The contributions of the rest of the IMFs to the entire variance of the observed price series were insignificant, suggesting that their combined effects on the crude oil price is negligible.

5.12 Mode analysis of crude oil-obtained through VMD method

The principal mode of crude oil price through VMD method was M1, which provided over 42% of the whole variance of the observed data, and has the highest correlation coefficients of 0.7310, 0.6070, and 0.7780 between the mode and the actual price series with respect to Pearson, Kendall, and Spearman correlation coefficients. The next influential mode of the crude oil price was M2, which supplied over 39% of the gross variance of crude oil price fluctuations. The collective effects of these two most important modes, accounted for 81.4400% of the grand variability of the original price of crude oil. The contribution of M3 and M4 to the gross variance of crude oil price series were 15.4200% and 1.7200% respectively. On the other hand, the M5, M6, M7, and M8 showed little correlation coefficients with the observed data and contributed less than 2% of the overall unevenness of crude oil price, indicating that their combined effects on crude oil market price is trivial.

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed seris	2.0079				113.1117		
IMF1	1.6003	0.1230	0.0392	0.0571	0.9429	0.8336%	0.8979%
IMF2	3.5082	0.0440	0.0274	0.0394	0.7835	0.6927%	0.7461%
IMF3	7.7394	0.0533	0.0452	0.0631	1.4582	1.2891%	1.3885%
IMF4	17.9859	0.1640	0.0726	0.1023	2.62998	2.32512%	2.50442%
IMF5	37.5588	0.3441	0.1576	0.2265	6.3840	5.6440%	6.0792%
IMF6	70.9444	0.31680	0.1756	0.2641	9.0807	8.0281%	8.6472%
IMF7	182.4286	0.4983	0.3155	0.4712	33.6216	29.7242%	32.0164%
IMF8	319.2500	0.6150	0.4260	0.6039	39.5897	35.0005%	37.6996 %
Residue		0.4821	0.4613	0.6134	10.5232	9.3033%	10.0208%
Sum						92.8407%	100.0000%

Table 5.5: Measures of IMFs and residue derived through EMD for Bloomberg daily price series of crude oil (May, 2016-April, 2021): Correlation is significant at the level of 0.05 (2-tailed)

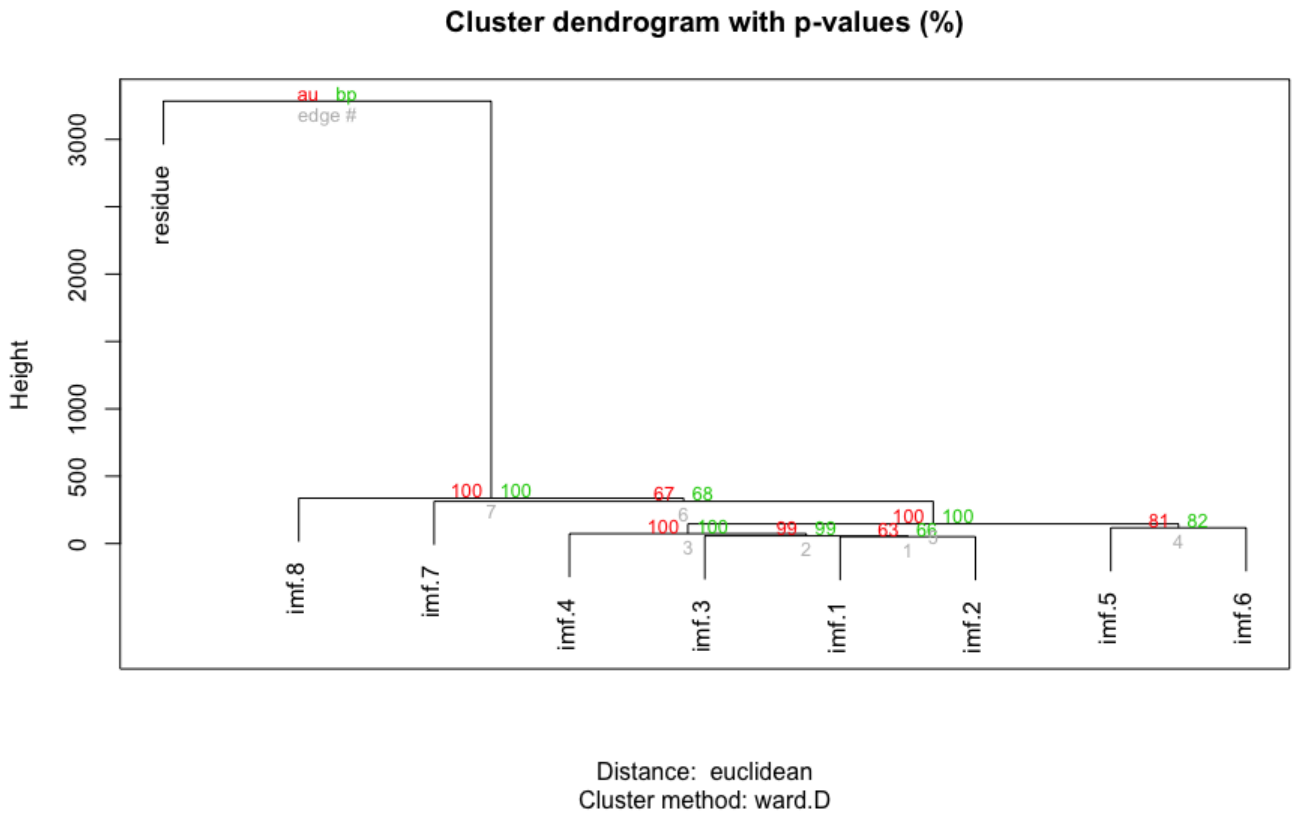


Figure 5.11: Hierarchical Clustering diagram obtained for the IMFs and residue through EMD decomposition

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed series	2.0079				113.1117		
High frequency	6.3532	0.2029	0.0860	0.1172	5.8019	5.1293%	5.7028%
Low frequency	55.5217	0.9287	0.7655	0.9279	85.4127	75.5118%	83.9538%
Trend	1277.0000	0.4821	0.4613	0.6134	10.5232	9.3033%	10.3434%
Sum						89.9444%	100.0000%

Table 5.6: Correlation and variance of components obtained from IMFs of EMD for daily price series of crude oil: Correlation is significant at 0.05 level (2-tailed)

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ modes
Observed series	2.0079				113.1117		
M1	21.2830	0.7310	0.6070	0.7780	34.7040	30.6800%	42.0300 %
M2	49.1150	0.7460	0.5470	0.7450	32.5420	28.7700%	39.4100%
M3	55.5230	0.4470	0.2260	0.2990	12.7280	11.2500%	15.4200%
M4	21.6440	0.1840	0.1180	0.1740	1.4170	1.2500%	1.7200%
M5	13.3020	0.1290	0.0660	0.0970	0.7330	0.6500%	0.8900%
M6	5.9400	0.0690	0.0420	0.0610	0.1900	0.1700%	0.2300%
M7	3.8350	0.0570	0.0350	0.0510	0.1570	0.1400%	0.1900 %
M8	2.4840	0.0490	0.0290	0.0410	0.0960	0.0800%	0.1200%
Sum						72.9900%	100.0000%

Table 5.7: Measures of modes and residue obtained through VMD for Bloomberg daily price series of crude oil (May 2016-April 2021): Correlation is significant at the level of 0.05 (2-tailed)

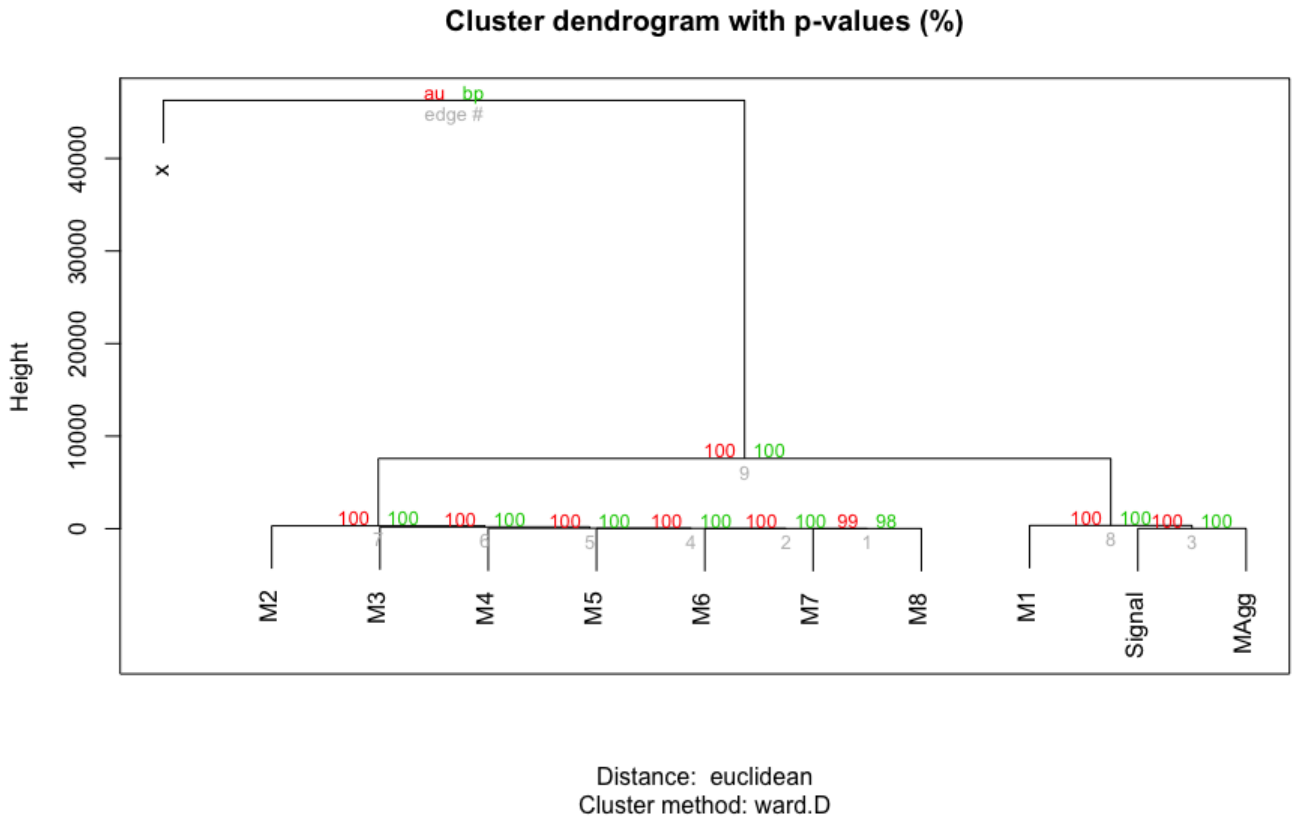


Figure 5.12: Hierarchical Clustering diagram obtained from the modes through VMD decomposition

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ modes
Observed series	2.0079				113.1117		
High frequency	6.3530	0.2030	0.0860	0.1170	5.8020	5.1200%	4.6080%
Low frequency	55.5220	0.9290	0.7650	0.9280	85.4130	75.5120%	67.8320%
Trend	21.2830	0.7310	0.6070	0.7780	34.7040	30.6810%	27.5610%
Sum						111.3220%	100.0000%

Table 5.8: Correlation and variance of components for Bloomberg daily data of crude oil (May 2016-April 2021 derived through VMD: Correlation is significant at 0.05 level (2-tailed)

5.13 Composition of crude oil price series

Likewise, we employed Hierarchical Clustering and the Euclidean Distance techniques in grouping the IMFs obtained through the EMD approach, into their specific components. The IMF1-IMF4 were grouped as high-frequency unit, while IMF5-IMF8 were categorized as the low-frequency part of the crude oil price. The residue classified as the trend component can be uniquely identify by its long-term average. Figure 5.11 shows all the partitions of the IMFs and the residue derived from EMD method. The residue with a long-term average constitutes the trend component of crude oil futures prices, while the low-frequency component corresponds to the impact of random events. The high-frequency component is characterized by short-term mean and amplitudes; and represents the shocks originated from short-term market uncertainties.

From the VMD technique, M4-M8 were categorized as high-frequency component, and constituted short-lived market instabilities. The M2-M3 formed the low-frequency unit of the crude oil price series, and corresponding to the significant events of the crude oil market price series. The M1 was categorized as the trend component of the observed series, as given in Table 5.8 and Figure 5.12.

5.14 The trend of crude oil futures prices

The trend is second influential component in crude oil futures prices determination; it contributed over 10% of the grand variance of crude oil futures prices volatility through EMD process, as presented in Table 5.6. This was affirmed by the VMD process; the trend again emerged as the second dominant component of the crude oil price unpredictability and donated over 27% of the entire variance of crude oil futures prices' variations, with correlation coefficient between the observed data as 0.7310, 0.6070, and 0.7780 in terms of Pearson, Kendall, and Spearman, respectively, thus, indicating that the trend is an important component in crude oil price determination in the long run, as illustrated in Table 5.8. Undoubtedly, the gradual increase in the crude oil prices is in accordance with global economic development by comparing the trend with the actual price series. It can be concluded that crude oil prices could vary owing to historic events, however, the price returns to the trend after the impact of the historic event is over.

5.15 Effects of special events on crude oil price

In addition to the trend, special events are main components of the crude oil price inconstancies. The IMF5-IMF8 and M2-M3 correspond to the low-frequency components of crude oil futures' prices from

EMD and VMD, respectively. From the EMD process, the special events constituted over 83% of the overall variance of crude oil price fluctuations, and maintained the biggest correlation coefficient of 0.9287, 0.7655 and 0.9279 between the observed price series in terms of Pearson, Kendall, and Spearman, respectively. The low-frequency contributed over 67% of the total variability of crude oil price through VMD, suggesting that special events are the primary components of crude oil prices' changeability. Intriguingly, the mean period of low-frequency from both EMD and VMD techniques were the same, as given in Tables 5.6 and 5.8, separately, suggesting that the time for special events may prevail for a longer time. Looking at the average period of the IMFs and the modes, it takes nearly two months for the market to regulate itself according to these special events.

Some of the price series could reach a high-level amplitude of 15 USD, therefore, suggesting that there is a remarkable impact of certain special events on crude oil price determination. The medium-term variation of oil price arises from special events by the virtue of the slow changes in trend and normal market fluctuation at a high frequency. In predicting, every special event can be estimated, and the outcomes can be utilized to forecast incoming special events of sameness by segregating special events from the entire data series.

5.16 Effects of ordinary market disequilibrium on crude oil price

Coupled with trends and special events, crude oil prices are influenced, as well, by the political setting, climatic conditions, inventory depletion, and industrial actions. The combination of these agents have no serious impacts on oil price, and constitute the high-frequency part of the observed data of crude oil futures prices. The influence of the normal market imbalance on the crude oil price is not significant, since it happens at a high frequency, thus, taking into consideration the amplitude levels of the IMFs and modes, its period is very short, indicating that their combined effects is minimal on crude oil price discovery. The short-term or near-term is adopted to define these impacts since the series used in this investigation was a daily data. The imbalance of supply-demand from normal market variation has no serious influence on the crude oil price, but if these events are not checked and occurred more frequently, they often become one of the primary drivers that cause crude oil prices instabilities. The normal market variation can be removed from the entire series in a long-term trend prediction, but it is pivotal for short-term forecasting; for example, the price of 64.8USD per barrel in April 2021 obtained through VMD method can be explicitly interpreted as a trend price of 17.86USD, a special event price of 43.96USD, and an ordinary market fluctuation of 2.99USD. The three components obtained through EMD and VMD techniques are given in Figure 5.13 and Figure 5.14, separately.

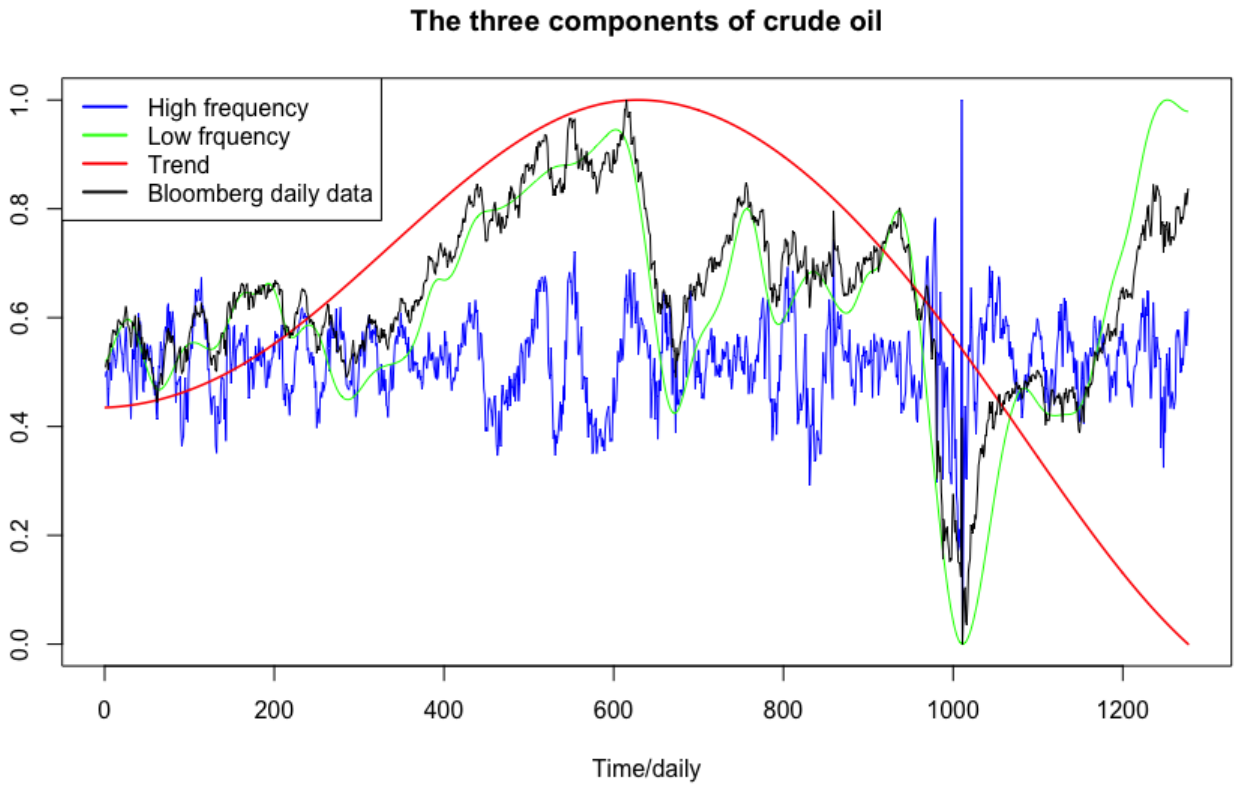


Figure 5.13: The three components derive from the Bloomberg daily data (May 2016-April 2021) through EMD

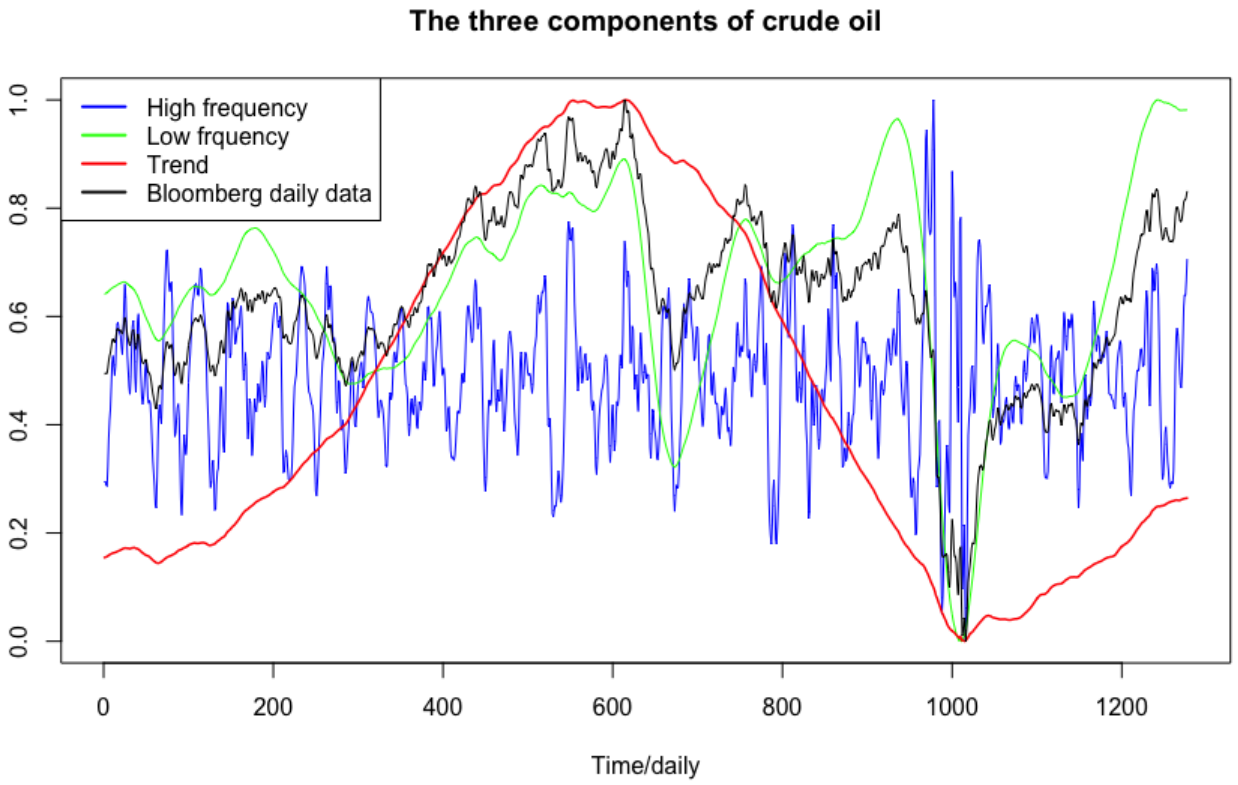


Figure 5.14: The three components derive from the Bloomberg daily data (May 2016-April 2021) through VMD

5.17 Decomposition results of gold through EMD and VMD methods

In the same vein, the EMD and VMD strategies were employed to decompose gold price series into their distinctive IMFs and modes. The decomposition results derived through both methods are presented in Figure 5.15 and Figure 5.16, respectively. The EMD technique produced eight distinctive IMFs of gold futures price series, described as IMF1-IMF8, and a residue, as given in Figure 5.15, while VMD process generated eight modes of the actual data of gold price series, as presented in Figure 5.16.

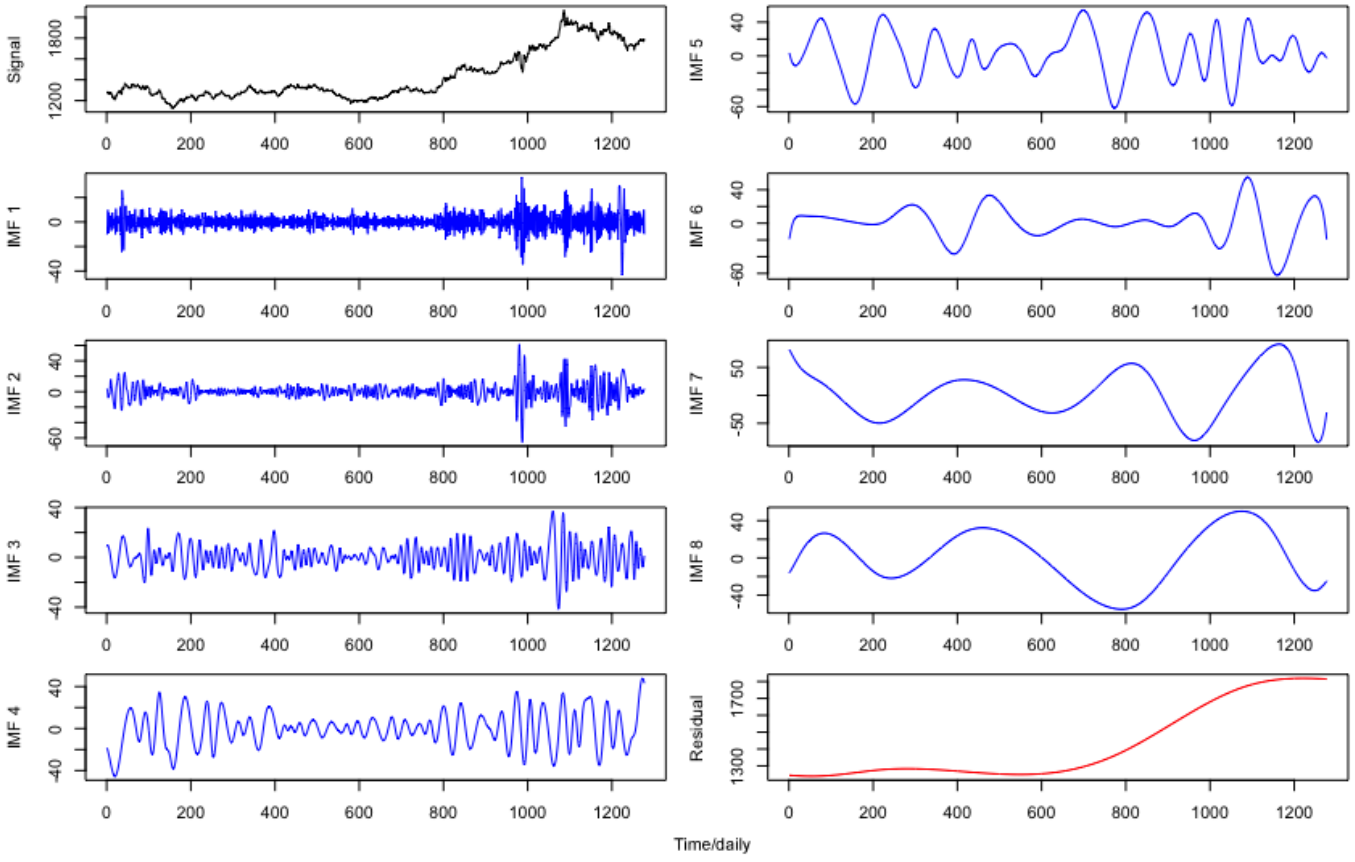


Figure 5.15: Empirical mode decomposition (EMD) curves for the daily gold price series (2016-2021)

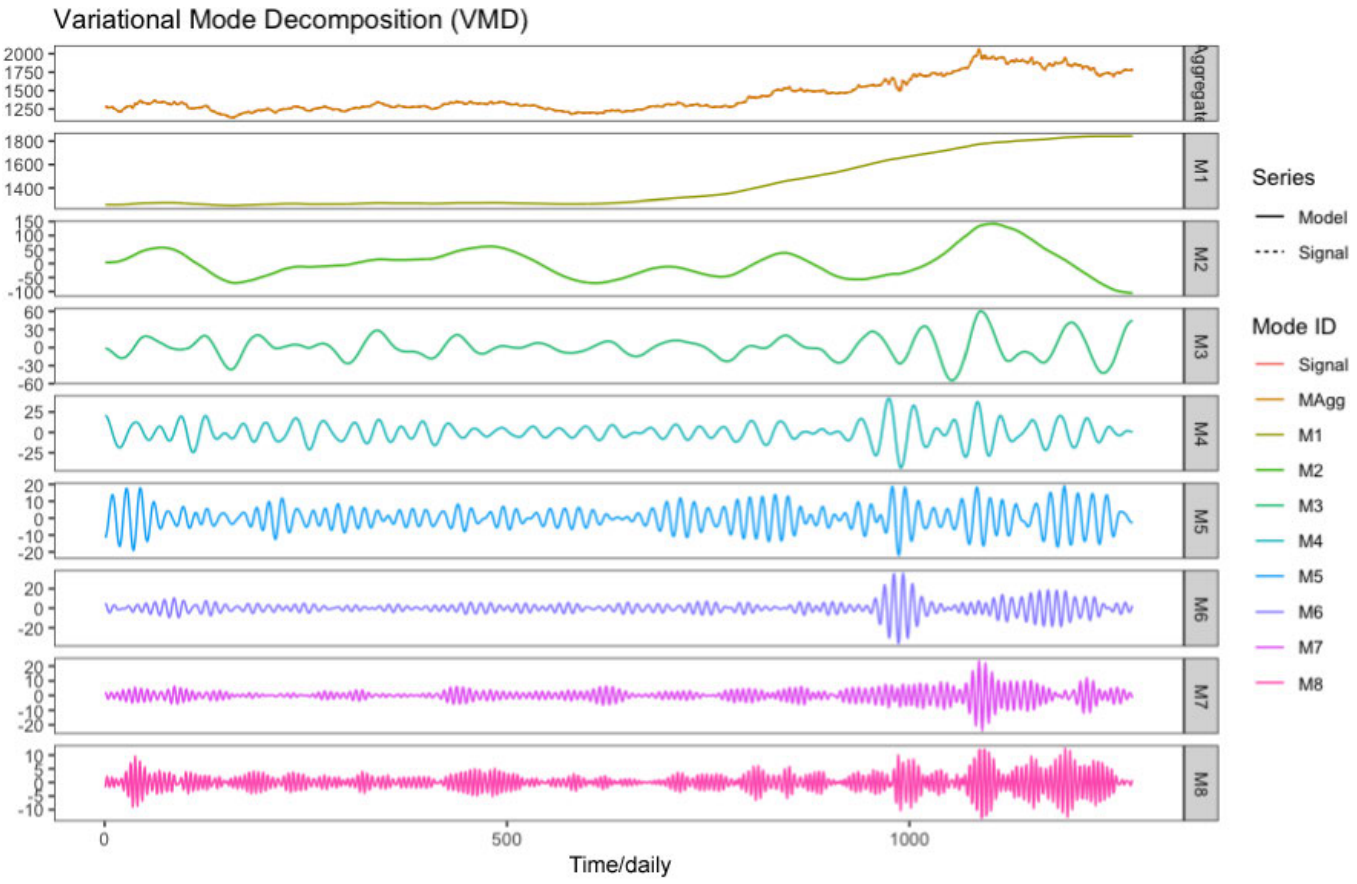


Figure 5.16: Variational mode decomposition (VMD) curves for the daily gold price series (May 2016-April 2021)

5.18 IMF analysis of gold price series through EMD

The residue emerged as the primary contributor of the gross variability of the gold price series, and constituted more than 34% of the entire variance. It also has the highest correlation coefficients between the actual price of gold with 0.9620, 0.5590, and 0.7430 with regard to Pearson, Kendall, and Spearman, separately, as given in Table 5.9 and Figure 5.15 . The second notable mode was IMF7, which contributed more than 27% of the actual series variance, followed byIMF8, which donated over 13% of the grand variance of the original series with correlation coefficients of 0.2850, 0.1860, and 0.2590 in respect of Pearson, Kendall, and Spearman, in that order. The IMF4-IMF6 contributed over 4%, 11% and 5%, respectively, of the net variance of gold price series. The Combined effects IMF4, IMF5, IMF6, IMF7, IMF8 and residue accounted for over 96% of the net variability of the gold price series. The IMF1-IMF3 contributions to the entire variance of gold price were insignificant, since they accounted for less than 3% of the grand variance of the gold price series.

5.19 Mode analysis of gold price series through VMD

In the same manner, the VMD technique disintegrated the actual price series of gold into eight individual modes, M1-M8, as given in Figure 5.16. The leading mode was M1, which donated over 93% of the gross variance of the gold price fluctuation, with correlation coefficients between the observed series of 0.9660, 0.7310, and 0.9020 in terms of Pearson, Kendall, and Spearman, respectively, as shown in Table 5.11. The next important mode was M2, which contributed 5.4200% of the total variance of the gold price series, with correlation coefficients between the actual price of 0.4170, 0.3270, and 0.4270 in terms of Pearson, Kendall, and Spearman, respectively. These two modes were the most important modes of gold futures price series, and accounted for over 98% of the overall variation of the volatility of the gold futures' series. The M3-M8 contributions to the net variance of the gold price series were insignificant, since they contributed less than 2% of the actual price series of gold. In addition, M3-M8 showed less correlations coefficients with the observed data, signifying that their combined effects on gold price determination is not important.

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed seris	1.9120				54236.2200		
IMF1	1.5610	0.0320	0.0350	0.0480	52.5560	0.0970%	0.7920 %
IMF2	3.9660	0.0700	0.0410	0.0600	108.516	0.200%	1.6360%
IMF3	8.5130	0.0570	0.0520	0.0720	101.7690	0.1880%	1.5340%
IMF4	19.0600	0.158	0.125	0.173	267.592	0.493%	4.034%
IMF5	47.2960	0.0920	0.1270	0.1820	736.3430	1.3580%	11.1010%
IMF6	79.8130	-0.0300	0.0870	0.1130	387.3370	0.7140%	5.8340%
IMF7	182.4290	0.2490	0.2390	0.3100	1794.4550	3.3090%	27.0530%
IMF8	212.8330	0.2850	0.1860	0.2590	876.0000	1.6150%	13.2060%
Residue	319.2500	0.9620	0.5590	0.7430	2308.6280	4.2570%	34.8040%
Sum						12.2310%	100.0000%

Table 5.9: Measures of IMFs and residue derived through EMD for Bloomberg daily price series of gold (May 2016-April 2021): Correlation is significant at the level of 0.05 (2-tailed)

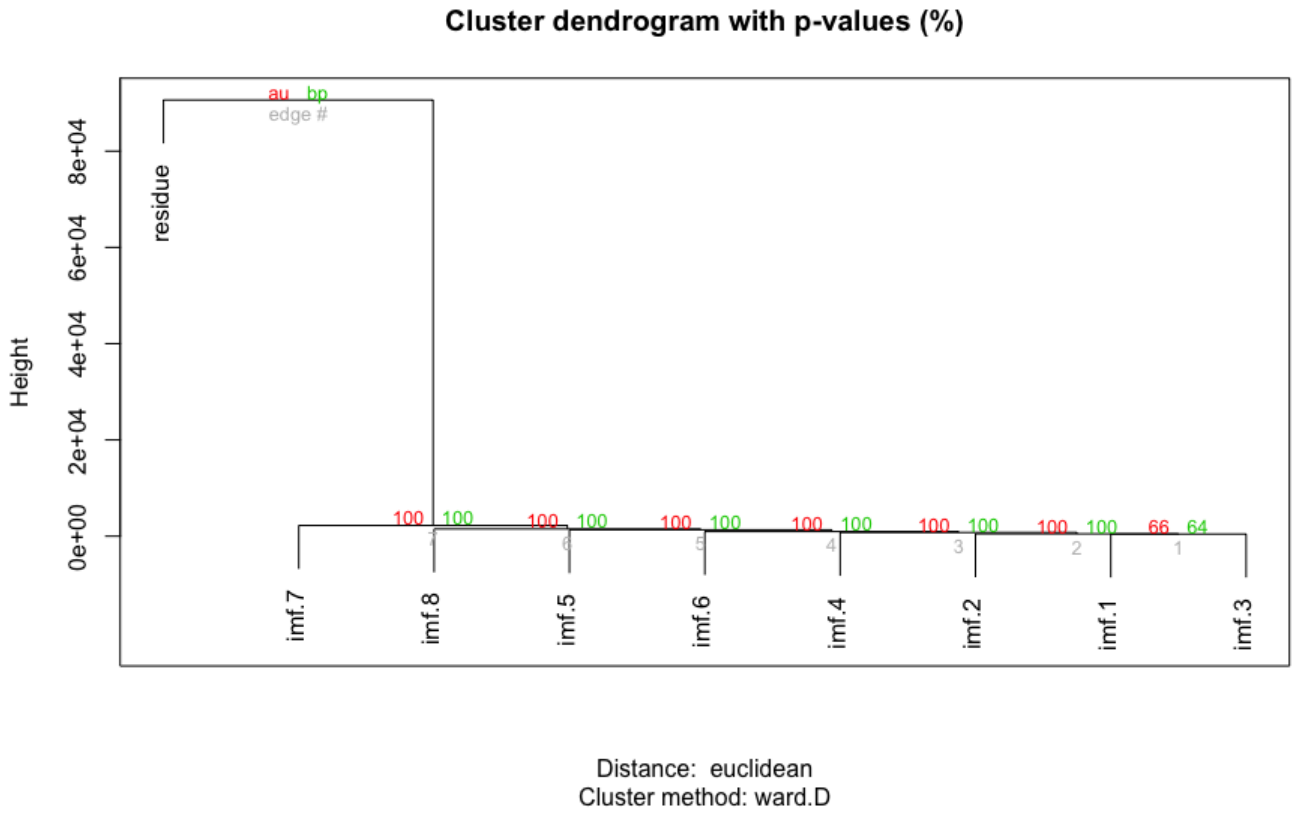


Figure 5.17: Hierarchical Clustering diagram obtained for the IMFs and residue of gold through EM

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ +residue
Observed series	1.9120				54236.2200		
High frequency	4.0800	0.0950	0.0640	0.0890	264.3760	0.4870%	0.5200%
Low frequency	67.2110	0.3750	0.3340	0.4230	3913.3720	7.2150%	7.6900%
Trend	319.2500	0.9620	0.5590	0.7430	46709.4500	86.1220%	91.7900%
Sum						93.8240%	100.0000%

Table 5.10: Correlation and variance of components for Bloomberg daily data of gold from May 2016-April 2021 derived through EMD: Correlation is significant at 0.05 level (2-tailed)

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ modes
Observed series	1.9120				54236.2200		
M1	18.7790	0.9660	0.7310	0.9020	45806.2700	84.4570%	93.4580 %
M2	44.0340	0.4170	0.3270	0.4270	2656.2980	4.8980%	5.4200%
M3	29.6980	0.1420	0.1230	0.1520	322.5260	0.5950 %	0.6580%
M4	15.2020	0.0750	0.0800	0.1130	122.1720	0.2250%	0.2490%
M5	8.4570	0.0590	0.0380	0.0540	43.4130	0.0800%	0.0890%
M6	5.6260	0.0470	0.0400	0.0510	37.1340	0.0680%	0.0760%
M7	3.9170	0.0290	0.0250	0.0330	15.2490	0.0280%	0.0310%
M8	2.9297	0.0240	0.0210	0.0290	9.6020	0.0180%	0.0200%
Sum						92.1960%	100.0000%

Table 5.11: Measures of modes and residue derived through VMD for Bloomberg daily price series of gold (May 2016-April 2019): Correlation is significant at the level of 0.05 (2-tailed)

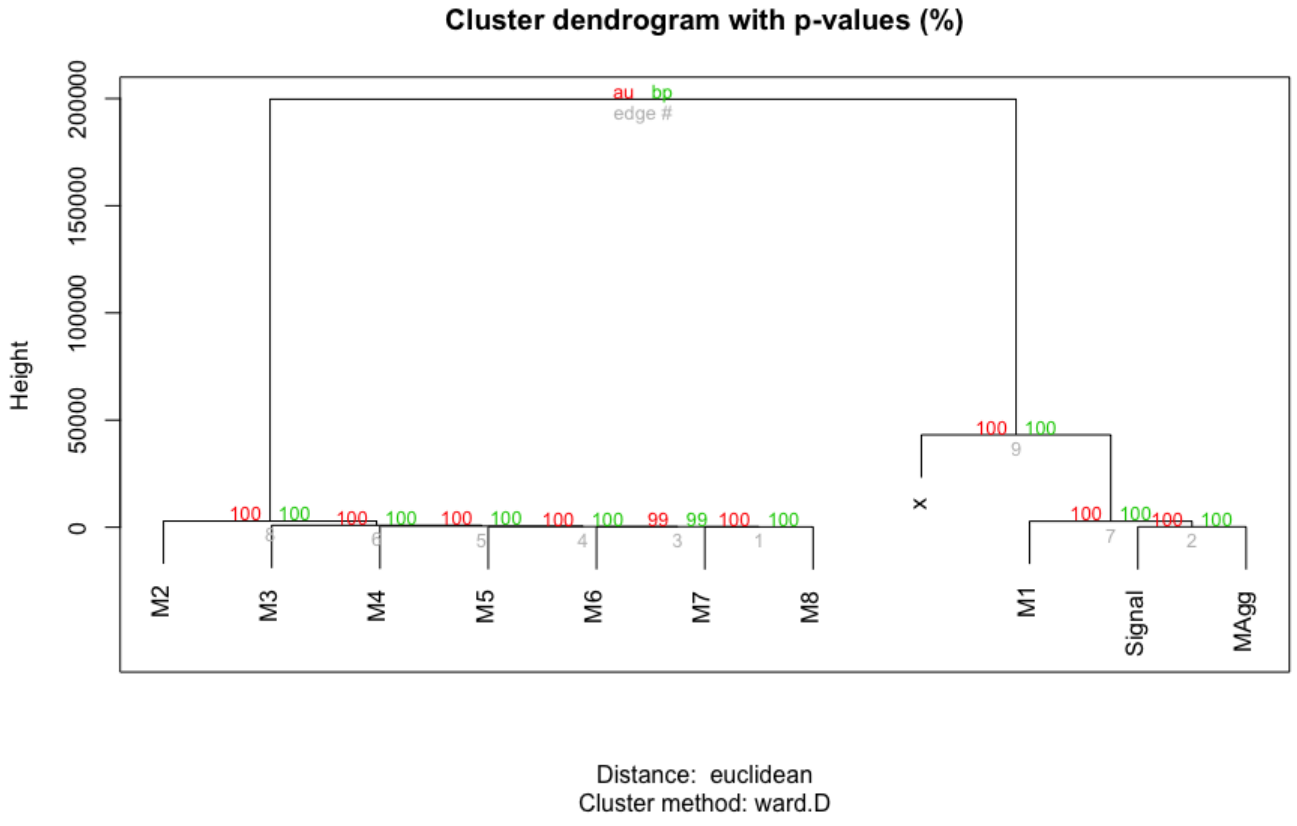


Figure 5.18: Hierarchical Clustering diagram obtained for the modes and residue through VMD

	Mean period	Pearson correlation	Kendall correlation	Spearman correlation	Variance	Variance as % of observed	Variance as % of Σ modes
Observed series	1.9120				54236.2200		
High frequency	5.1490	0.1610	0.1650	0.2170	649.3780	1.1970 %	1.3220%
Low frequency	44.0340	0.4170	0.3270	0.4270	2656.2980	4.8980%	5.4090%
Trend	18.7790	0.9660	0.7310	0.9020	45806.2700	84.4570%	93.2690%
Sum						90.5520%	100.0000%

Table 5.12: Correlation and variance of constituents for daily series of gold (May 2016-April 2021) derived through VMD: Correlation is significant at 0.05 level (2-tailed)

5.20 Composition of gold price series

In this section,we discuss the constitution of the gold price with respect to the IMFs and modes contributions, respectively. From the Hierarchical Clustering and the Euclidean Distance Techniques, the IMF1-IMF6 constituted the high-frequency components of EMD, while IMF7-IMF8 were classified as the low-frequency components of the actual price series, as given in Figure 5.17 and Table 5.10. These categorized IMFs and the residue can be economically explained and they give important information about gold market futures prices. The moderate change of the mean of the residue forms the long-term trend of the actual price series of gold. The low-frequency component constituted the special event of the observed price series. The impacts of short-term market variations are donated by the high-frequency components, and they are characterized by their shorter amplitudes, as given in Table 5.10 and Figure 5.17.

For VMD, M3-M8 constitute the high-frequency components of the observed price series of gold and accounted for the short-term impacts of normal market instabilities. M2 corresponds to the low-frequency component of the original price series of gold for the years under review, and indicates the significant events of gold futures price series. The long-term trend of the gold futures' prices represents by M1, are illustrate in Table 5.12 and Figure 5.18, separately.

5.21 The trend of gold price series

Trend accounted for over 91% and 93% of the overall variance of the actual price series of gold in terms of EMD and VMD, respectively. It possesses the biggest correlation coefficient with the original price series, suggesting that the trend is a dominant component responsible for the long-term variations of the

gold price, as shown in Table 5.10 and Table 5.12, respectively. Comparing the graph of trend with the Bloomberg day-to-day price series of gold, it can be seen that the trend mimics that of the observed series, showing that although, gold prices can fluctuate due to special events, they come back to the trend after they are over. Figure 5.19 and Figure 5.20 represents the three components of gold futures prices through EMD and VMD, respectively.

5.22 Effects of special events

The IMF7-IMF8 and M2 constitute the low-frequency units of futures prices of gold through EMD and VMD, respectively. Low-frequency accounted for over 7% and 5% of the net variance of the gold price volatility through EMD and VMD techniques, while the low-frequency corresponds to the shocks originating from special events of gold price. Based on the average period of the IMFs that constituted the low-frequencies, the time span for the shocks is a minimum of two months for EMD and a minimum of one and a half months for VMD; this signifies that it is difficult for the market to adjust itself to the effects brought about as results of special events. The period of the impacts of special events was a bit longer, which implies that many special events have a consequential effects on the price, therefore, intermittent prices' variations in gold, originate from special events.

5.23 Effects of ordinary market disequilibrium

In addition to trends and special events, the market price of precious metal, gold, is affected by political settings, adversed weather conditions, industrial unrest, and inventory depletion. The combination of these effects are described as 'high- frequency' units of gold price series, and constitute the normal market disequilibrium of observed price series of gold. The influence of normal market imbalance has a short-term effect on gold price because the periods for some of these constraints are short, thereby, indicating that the market disequilibrium has insignificant influence on the gold market price discovery. On the other hand, if these constraints are neglected for a long time, they become one of the primary factors contributing to gold price variation in the short-term. In long-term trend prediction, the ordinary market variations can be neglected, however, they are very important in short-term forecasting.

Overall, gold price comprises the trend, the primary contributor of gold price variations in the long-term, followed by the special events, the main causes of periodical price fluctuations of gold in the medium term, and the ordinary market unsteadiness, which is responsible for short-lived market price variabilities of gold. Considering the price of 1254.23USD per ounce in April 2021, through VMD process, this can

be broken down as 1169.81USD trend price, 67.84USD special event price, and 16.58USD normal market fluctuation. Figure 5.20 illustrates the three components derived through VMD process.

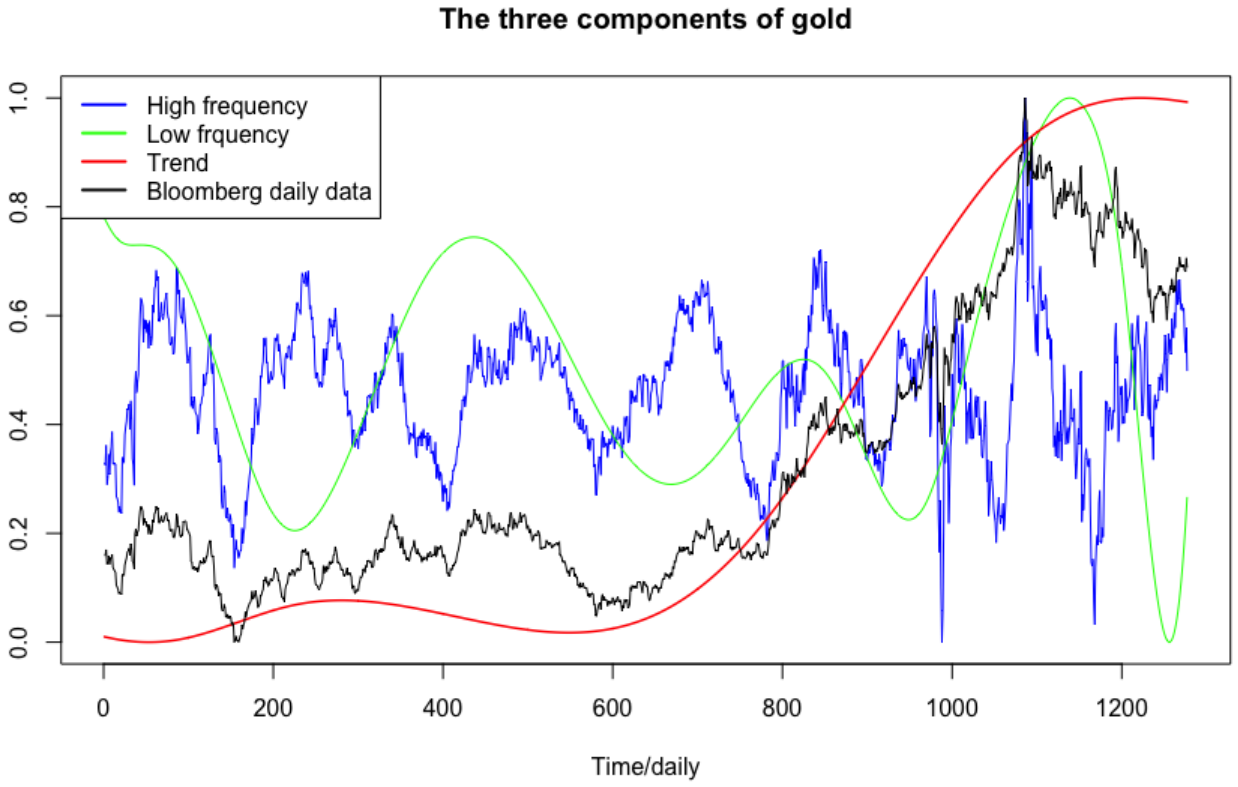


Figure 5.19: The three components of the Bloomberg daily data (May 2016-April 2021) through EMD

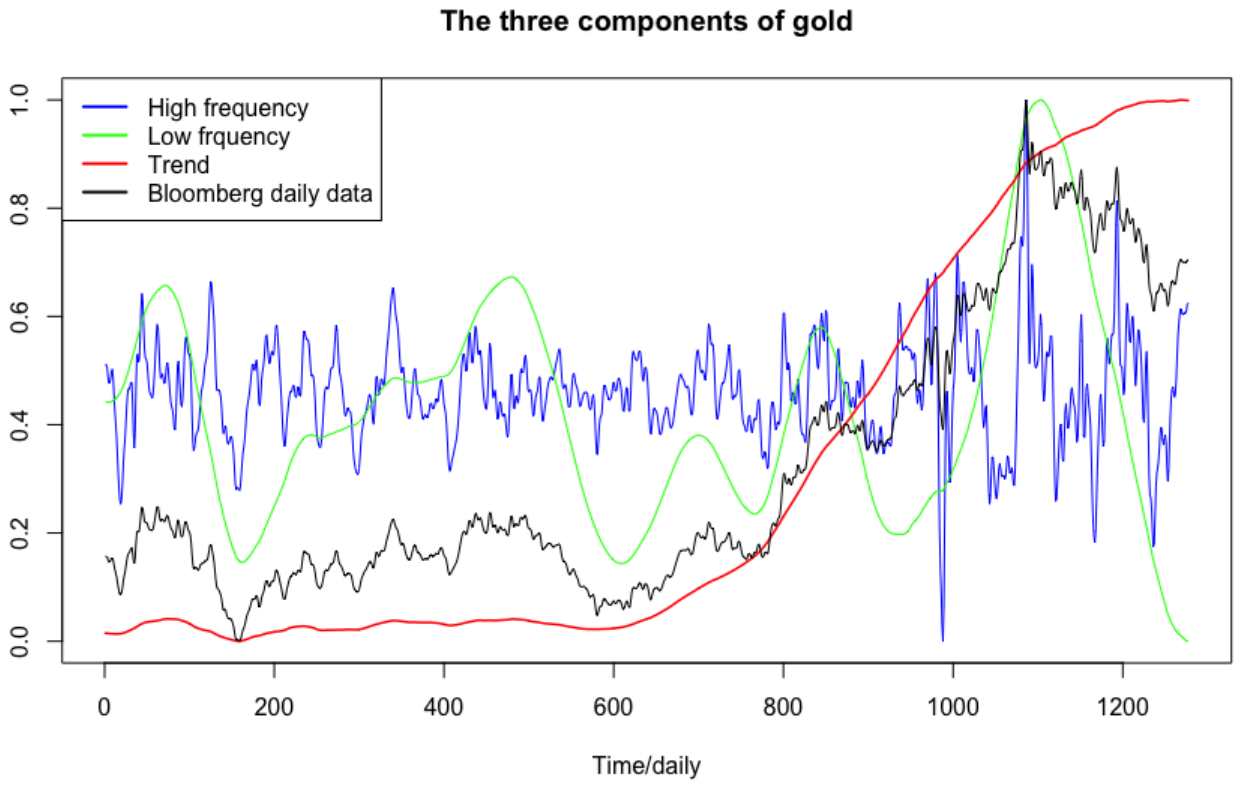


Figure 5.20: The composition Bloomberg's day-to-day price series of gold (May 2016-April 2021) derived from VMD process

5.24 Conclusion

This chapter analyzed the underlying factors that cause fluctuation in the three commodities markets - crude oil, corn, and gold markets prices - utilized the EMD and VMD approaches. These two innovative techniques were employed to breakdown day-to-day closing prices of corn, crude oil, and gold secured from the Bloomberg Commodity Index ranging from May 2016-April 2021, into their distinctive IMFs or modes and a residue to discover the constituents that trigger commodity futures' market prices variations.

Applied statistical measures, such as the Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation were utilized to estimate the contribution of each IMF's, mode, and residue to the overall variance on the commodity futures markets prices' variations. We continued by employing the Hierarchical Clustering and the Euclidean Distance techniques to categorize the IMFs,

residue, and modes into their distinctive frequencies: high-frequency, low-frequency, and trend components. It emerged that the trend and low-frequency components were the primary contributors of commodity futures prices fluctuations. The three commodities markets prices' fluctuations could be explained as a combination of a long-term trend, special events, and near-term fluctuations resulting from usual market activities, such as supply-demand imbalance. Trend and special events generally drive the three commodities prices. The effects from the trend deviate gradually and changes around the long-term mean. The unpredictable special events are responsible for irregular commodity price movements, and the effect could last for several years; this shows that normal market activities mainly trigger the short-term commodity price changes, and the effects have a very short duration, hence, their impact on prices is not serious. Interestingly, the three commodities' markets have similar behavior, since their prices series are mainly driven by the same components.

Significantly, utilizing the decomposition technique in evaluating compositions of commodity prices data series separately, divergent predicting strategies could be explored: first of all, based on the features of decomposed IMFs or modes, a suitable predicting technique can be considered to forecast each IMF or mode, for example, the residue can be estimated by utilizing a polynomial function, while Fourier transform can be considered in predicting low-frequency IMFs modes, and applying a non-linear technique to evaluate high-frequency IMFs or modes, and finally, in summing up the individual predicts to secure overall predicting outcomes. In the second place, the IMFs or modes can be classified as linear and non-linear series, and can be predicted differently, and then combining the individual estimated values as overall forecasted results, thus, curve fitting can be utilized to predict the trend component of commodity futures markets prices series, while non-linear predicting method, such as artificial neural network or backpropagation neural network, to predict the low-frequency component of commodity market futures prices series. Special events are hard to predict because they are caused by a number of factors, including trade wars, pandemic, climate conditions, financial schemes, and various complicated factors.

Eventually, we propose that a new model or consolidated predicting model must be evolved to cater for special events' influences on commodity market futures prices due to the fact that no one can predict the time they can occur and the particular place.

Chapter 6

Forecasting Commodity Futures Market Prices Using EMD and VMD Based Models

Chapter summary

Price uncertainty is a prime concern since commodity prices are associated with the livelihood and the economy of a nation; this means that any extraordinary price fluctuation in the futures market is very momentous event. The difficulties in predicting commodity prices are due to the unpredictability of world's financial issues, fiscal dispensation, the speculative market's exacerbation, and several other factors. This thesis aimed at modeling and forecasting the market price of commodity futures. we utilized decomposition-based methods, empirical mode decomposition (EMD), and variational mode decomposition (VMD) on three commodity markets mentioned earlier over the commodity spot market prices. Three commodity price data with different periods were decomposed into several intrinsic modes. This was done using three forecasting performance evaluation criteria, statistical measures, such as mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage erroe (MAPE) to compare the capabilities of the suggested models. We also introduced Diebold Mariano (DM) test in choosing the ideal model for each commodity futures market, since MAE, RMSE and MAPE have some shortcomings. The decomposition-based methods outperformed the individual BPNN and ARIMA methods in forecasting the corn and crude oil futures' prices series, while BPNN method emerged as the optimal model in forecasting gold futures' prices series. Variational mode decomposition (VMD) emerged as the ideal data

pre-treatment method and contributed to enhancing the predicting capability of the BPNN and the ARIMA methods. The empirical results showed that the models combined with decomposition methods predict commodity market futures prices accurately and can easily capture the volatility in commodity futures' prices.

6.1 Introduction

Current developments in the commodity markets have remarkable impacts on the world economy. The commodity markets, such as crude oil corn, and gold play a consequential role in the global financial markets. It is obvious that these commodities market prices are highly volatile and directly affect global market activities (Zhang et al., 2018, Wang and Wei, 2021). The futures prices of these three commodities are influenced by demand and supply, markets speculations, economic development, national schemes and state of affairs, such as spillover effect, an outbreak of pandemic, war, and recent global debt crisis (Zhao et al., 2016). These complicated components are the primary factors responsible for price fluctuations of corn, crude oil, and gold commodities futures' prices. The price instability of the above commodities market is a major concern as venture capitalists expect profit from their investments, hence, investigation of the prices of crude oil, corn and gold can provide investors with the necessary details to strategize and minimize the cost of doing business and increase profit. The price series of aforementioned commodities, however, are non-linear and non-stationary, making it strenuous to develop models to evaluate the price pattern of these commodities. Pragmatic investment decision-making, therefore, can help venture capitalists and policymakers curb the risk caused by price fluctuation, which can only be attained by explicitly predicting the futures market.

Crude oil, corn, and gold markets are deemed the topmost unstable, correlative and conglomerate in the commodity market and responsive to macroeconomic schemes (Zou et al., 2007). The above-mentioned commodities were selected from the commodity spot market price to perform this experiment, which focused on forecasting the price dynamics in energy, agriculture and industrial metal. In the global economy, corn, crude oil, and gold markets are crucial, therefore, a price study of these commodities is essential for any sustainable future planning due to the correlative relationship among price, supply and demand. For instance, a small increase in oil price often leads to increasing inflation, thereby affecting importing countries' economies, although reduced oil prices also bring about an economic downturn and political uncertainty in nations that export oil for economic development. The volatility of oil prices causes economic instability as research has shown that a relatively small rise in crude oil prices significantly affects the global economy (Raza et al., 2017). An increase of 10% of the price is equivalent to 0.6% to 2.5% of

Gross Domestic Product (GDP) growth for the USA (Raza et al., 2017, Plakandaras et al., 2015). High oil prices increase the cost of doing business, and these costs are ultimately passed on to consumers and businesses.

Motivated by the above-mentioned reasons, this study proposed a robust signal detector, EMD and VMD approach, to decompose crude oil, corn, and gold prices' series. This research adopted EMD and VMD techniques to study and predict the futures markets' prices of above stated markets. By investigating and predicting the data series of these selected commodities markets prices fluctuations, this research would beef up the existing literature in the following ways:

- it will provide a comprehensive causal relationship analysis among the three commodities markets can be studied.
- a critical comparative analysis of the EMD and the VMD data preprocess effect can be examined horizontally.
- the predicting performance ability of BPNN and ARIMA methods can be enhanced.
- the economic explanations of the above-mentioned commodities' markets prices variations can be understood, and
- it may be concluded that the decomposition-based methods have better forecasting performance than separate models in commodity market price prediction.

Researchers have lately used the decomposition framework to study times series in different fields, including exchange rates, climates, energy consumption, passenger travel movements, among others. For instance, in forecasting commodity futures market prices of corn, crude oil, and gold, Antwi et al., (2021) utilized both EMD and VMD to decompose price series of each commodity into several IMFs, modes, and residue. Decomposition-based models were employed to forecast each commodity futures prices series and compared with standard single models - BPNN and ARIMA models. Antwi et al., (2021) reported that combined models improve the forecasting capability of BPNN method and ARIMA method.

Novianty et al., (2022) utilized EMD to study the oil palm content in palm fruits. The study proposed the EMD method to break down signals and forecast the oil content in palm fruit. EMD was integrated with the ANN to form a decomposed model to analyze near-infrared signals and oil content in palm fruits. The RMSE and the coefficient of determination (R^2) were utilized as evaluation performance criteria to compare the robustness of the suggested method. The results demonstrated that the EMD-ANN method has high predictive ability and efficiency in extracting oil content in the palm fruits.

Similarly, Teng et al., (2022) employed a novel hybrid method forecasting model using the EMD technique and combined with sample entropy (SE) index and bidirectional long and short-term memory neural network (BiLSTM) to forecast fine particulate matters $(PM_{2.5})$ concentrations. The empirical results indicate that the combined model, BiLSTM has a high performance prediction on $(PM_{2.5})$ concentrate. In addition, the model could extract the trend in the concentration of $(PM_{2.5})$ compared to single-machine learning models.

To understand the US GDP data prediction deeper, Lin (2022) constructed a combined model using EMD and LSTM to form EMD-LSTM hybrid model to forecast the US GDP trend. First the US GDP was EMD-broken-down to generate IMFs then followed by LSTM model prediction. The experimental results showed that the integrated model, EMD-LSTM predicts US GDP better than the single model, LSTM in terms of revealing the embedded characteristics in the data.

In investigating the volatility spillover effect of Bitcoin and other financial markets, such as crude oil, gold, foreign exchange rate, stock, and natural gas, Jiang et al., (2022) proposed a VMD-based time varying parameter vector autoregressive (TVP-VAR) model to study the volatility spillovers of different financial markets and its effects on the worldwide economy. The VMD was utilized to decompose each market price. They reported that Bitcoin is considered a hedge in the financial market system. Furthermore, it was revealed that the entire volatility spillovers among the financial market system is caused by changes in external market consideration amidst the different markets.

To enhance the forecasting precision of the stock market price index, Niu et al., (2020) suggested hybrid models, VMD-LSTM and EMD-LSTM based on decomposition and ensemble basis. The VMD and EMD were used as a data pre-treatment method through which the actual data was disintegrated into several sub-units and the forecasting capability of VMD-LSTM and were verified using the complexity-invariant distance (CID), the VMD-LSTM method emerged as the best performing method in predicting stock market price index. The hybrid methods achieve remarkably good performance as compared to the separate methods, and the forecasting precision of the VMD-based methods is mostly higher than the EMD-based methods.

Agricultural commodity futures' market price is affected by several components, such as economic growth, financial schemes, climate conditions, and oil price instability, hence, the price series of this market is extremely complicated and nonlinear. To address the nonlinear features and accurate estimation of agricultural commodity futures market prices, Wang et al., (2022) suggested a forecast combination method based on an artificial bee colony algorithm (ABC) for predicting soybeans and corn markets. Three decomposition techniques, SSA, EMD, and VMD were utilized as data pre-processing to break down the

actual data of soybean and corn. Five combined predicting models: ARIMA, SVR, recurrent neural network (RNN), gated recurrent neural network (GRU), LSTM. Empirical results show that joint forecast models predict soybean and corn prices better than the individual models.

It is clear that extensive preceding study has led to several potential predicting methods being suggested, however, choosing the ideal method for a set of forecasts is a demanding (Wang and Li, 2018, Zhang and Zhang, 2018, Kourentzes et al., 2019). As a result, decomposition-based model forecasting has evolved as one of the efficient techniques in predicting nonlinear time series, as it helps increasing forecast precision as compared to separate model forecasts (Bates and Granger, 1969, Ismael, 2008, Nowotarski et al., 2016).

In view of this, we suggest decomposition-based framework to forecast three commodities markets' prices:

- corn, the most versatile food product in the agricultural industry,
- crude oil, the main global source of fuel, and
- gold, a fascinating and precious metal in the metal industry over the commodity market.

This work presents a neural network predicting method based on EMD and VMD techniques. The EMD and VMD techniques were utilized to generate a series of combined models: the EMD-BPNN, EMD-ARIMA, VMD-ARIMA, and VMD-BPNN. The performance of the EMD and VMD methods, therefore, are compared horizontally utilizing the three classical forecasting criteria performance measures: root mean square error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). We also employed the novel DM test to compare the forecasting abilities of the proposed methods. From this approach, it was emerged that the decomposition techniques integrated with the BPNN and the ARIMA models were superior to separate models.

6.2 Forecasting Performance Evaluation Criteria for Commodity prices

Generally, many statistical techniques have been developed in literature in estimating commodity futures prices' forecasting models in the literature. The most common evaluation criteria used are MAE, RMSE and MAPE. This study utilized these three error measure indices to evaluate the viability of the proposed methods, in addition to Diebold Mariano (DM) test to differentiate the predicting ability of models suggested in this study, since traditional evaluation criteria have drawbacks, including stochastic process, in choosing the ideal forecasting method.

Mathematically, the three error methods can be given as:

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n}|Y(i) - \hat{Y}(i)| \tag{6.2.0.1}$$

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(Y(i)) - \hat{Y}(i))^2} \tag{6.2.0.2}$$

$$MAPE = \frac{1}{n}|\frac{Y(i) - \hat{Y}(i)}{Y(i)}| \tag{6.2.0.3}$$

where the sample size is denoted by n, the actual dataset is given by $Y(i)$, and $\hat{Y}(i)$ corresponds to the predicted value of the series.

6.3 Diebold Mariano Test (DM)

Generally, utilizing traditional predicting evaluation of MAE, RMSE and MAPE as forecasting performance criteria have some drawbacks in selecting the best predicting method. For example, after evaluating the MAE of the forecasting model, if the difference between the two models is negligible, it is difficult to arrive at a conclusion at this stage. A decision cannot be taken based on slight difference of MAE result. Furthermore, basing on slight MAE difference between the models to accept one model may mean that the best method would be rejected, since the slight differentiation in the estimation may be due to stochastic process. In solving the drawbacks related to the classical approach, this research proposed the DM test suggested by Diebold and Mariano (1995), which can quantifiably estimate the forecasting precision of commodity market futures prices methods in choosing the best method for forecasting each and every commodity market price series.

We followed Chen et al., (2014) in introducing DM process. The DM process is given by:

Let q_i define the observed data, and $\hat{q}_{i,t}^{p}$ represent the i^{th} competing p-step forecasting series.

supposing the forecasting errors from the i^{th} competing methods are $e_{i,t}^{p}(i = 1, 2, 3, ..., n)$, where n is the number of forecasting methods. The p-step forecasting errors $e_{i,t}^{p}$, is expressed as:

$$e_{i,t}^{p} = y_t^p - \hat{q}_{i,t}^{p}(i = 1, 2, 3, ..., k) \tag{6.3.0.1}$$

The estimation of each forecast is assessed utilizing the loss function:

$$S(q_t^p, \hat{q}_{i,t}^{p}) = S(e_{i,t}^{p}) \tag{6.3.0.2}$$

This research set p value to 1, and discarded the superscripted p in the conditions below. In reality, numerous loss functions have been utilized in commodity markets and the usual one include the SE and the AE losses functions.

The squared-error function can be expressed as:

$$S_2(q_t, \hat{q}^p_{i,t}) = S_2(e_{i,t}) = \Sigma^T_{t=1}(e_{i,t})^2 \tag{6.3.0.3}$$

Absolute-error loss function is given by:

$$S_1(q_t, \hat{q}^p_{i,t}) = S_1(e_{i,t}) = \Sigma^T_{t=1}|e_{i,t}| \tag{6.3.0.4}$$

The two errors, SE and AE, are symmetrical about the origin. Additionally, the SE loss function can severely deal with larger errors.

To assess if one forecasting method, for example, the method A prediction is superior to the method B, then we test for equality accuracy of the two methods in the null hypothesis:

$$H_0 : E[S(e_{1,t})] = E[S(e_{2,t})] \tag{6.3.0.5}$$

The alternative hypothesis that one method has better predicting precision than the other method is given by:

$$H_1 : E[S(e_{1,t})] \neq E[S(e_{2,t})] \tag{6.3.0.6}$$

The DM process is derived by differentiating the differential loss function, d_t:

$$d_t = S(e_{1,t}) - S(e_{2,t}) \tag{6.3.0.7}$$

The equal predictive accuracy of the null hypothesis is expressed as: $H_0 : E[d_t] = 0$. Then, let $\bar{d}$ denote sample mean loss, $\bar{d}$ is expressed as:

$$\bar{d} = \frac{1}{T}\Sigma^T_{t=1}[S(e_{1,t}) - S(e_{2,t})] \tag{6.3.0.8}$$

The DM test statistic is expressed as:

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi \hat{f}_d(0)}{T}}} \tag{6.3.0.9}$$

where $2\pi \hat{f}_d(0)$ represents a constant evaluator of the asymptotic variance of $\sqrt{Td}$ and d is normally distrbuted. The variance is used on the statistics because d_t, the sample loss differentials, are succeeding correlated $p > 1$. We reject the H_0 at 5% level if $|DM|$ greater than 1.96, or else, if $|DM|$ less or equal to1.96, we fail to reject H_0, since the DM statistics converge to normal distribution. The DM test is good in revealing all the impedements of the sample stochastic difference, in such a way that the best predicting method can be established statistically.

6.4 Empirical results and analysis

In time series forecasting using the BPNN model, many prior points are selected as input to forecast the subsequent one since the predicting accuracy is affected by input's limit. The ideal limit of the BPNN's input series was ten after a provisional analysis of the different forecasting results with distinct input limit. With reference to the time series $T_i, i = 1, 2, \ldots, n$, this research uses $T_1, T_2, T_3, T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$ to predict $T_1 1$; and so forth. Hence, T_{i+10} could be forecasted by $T_i, T_{i+1}, T_{i+2}, T_{i+3}, T_{i+4}, T_{i+5}, T_{i+6}, T_{i+7}, T_{i+9}$. The main threshold variables of the BPNN were set as follows: smoothing threshold 0.01%, the nodes 10, hidden layers 2, number of outputs 1, and number of iteration 100. It is important to note that these variables stated were established through a series of tests, and maintained these settings throughout our discussions. We consider BPNN method as the benchmark model, and the ARIMA as comparative model and compare their forecasting precision with the four proposed decomposition-based methods.

6.5 Experimental procedure

From Figure 6.1, it can be seen that there are four steps involves in this experiment:

1. EDM and VMD techniques were utilized to break down each price series of the three commodities into sub-series separately;

2. normalized each sub-series by means of distinctive linear transformation prior to the forecasting phase;

3. normalized sub-series were entered into the EMD-BPNN, EMD-ARIMA, VMD-BPNN, and VMD-ARIMA models, producing an array of predictions which were considered and reversed using the normalization process; and

4. obtain the final predict by adding all forecasted points.

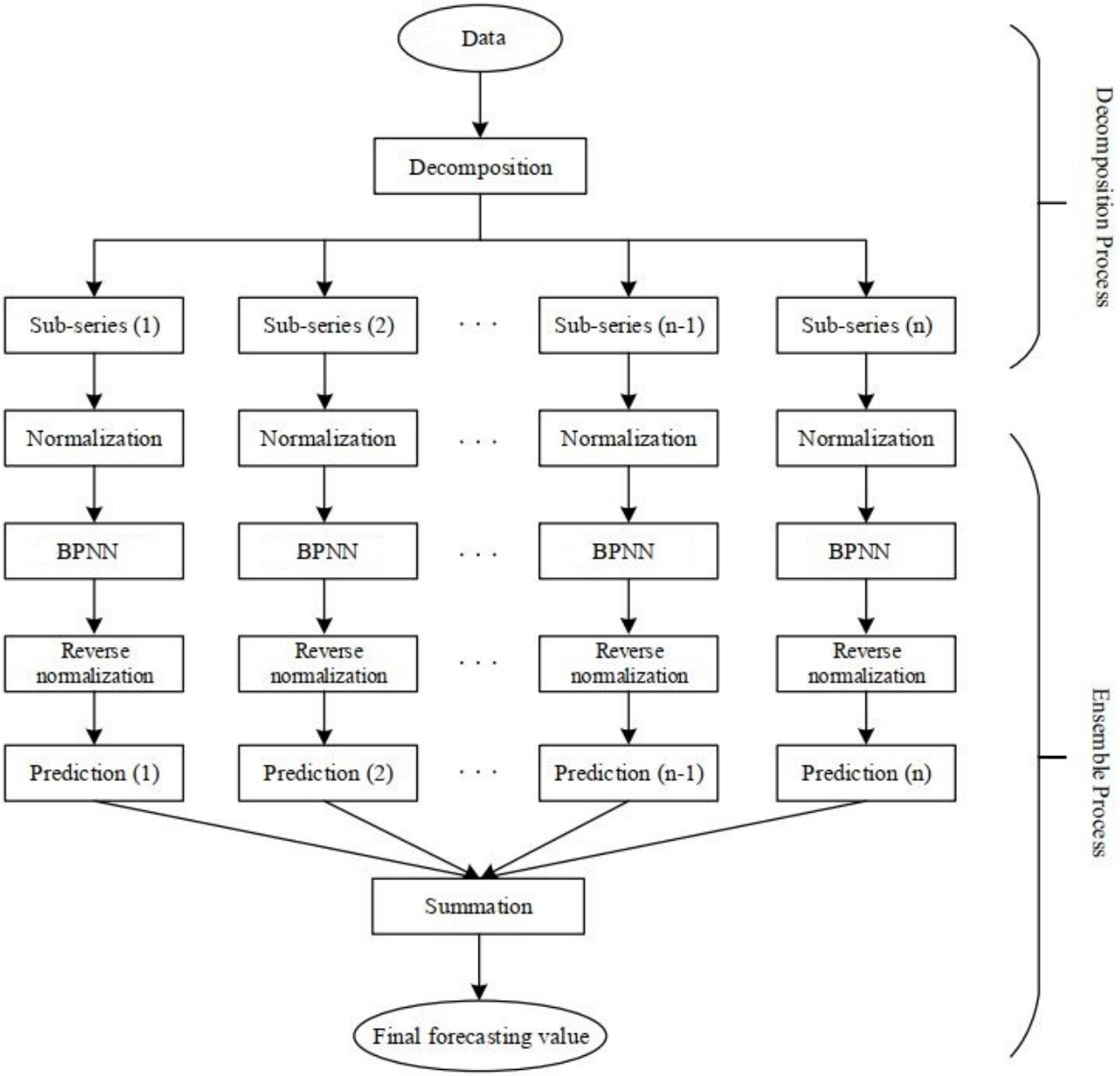


Figure 6.1: Structure of proposed Decomposition-based model.

6.6 Decomposition results of day-to-day price series of corn through EMD and VMD

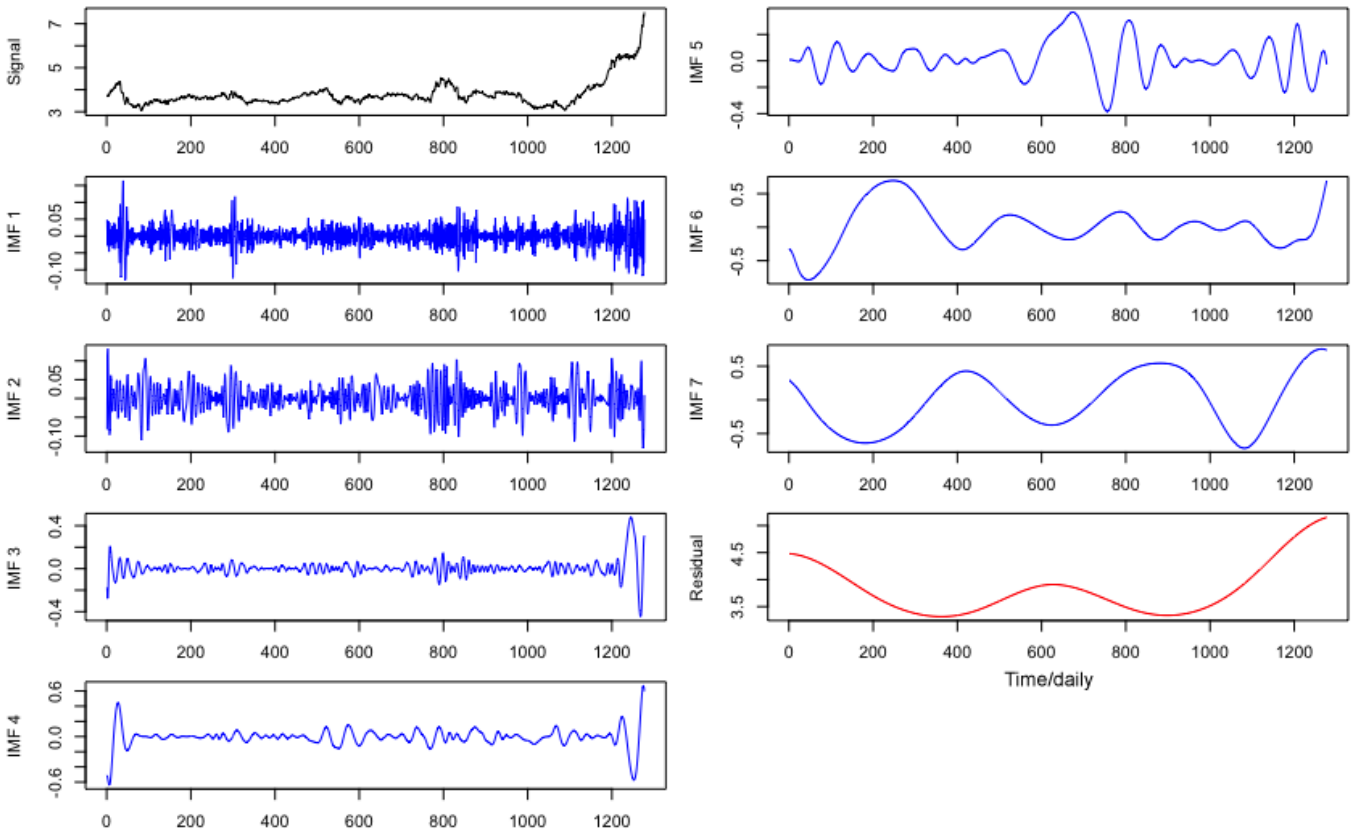


Figure 6.2: Empirical mode decomposition (EMD) curves for daily corn price series (May 2016-April 2021)

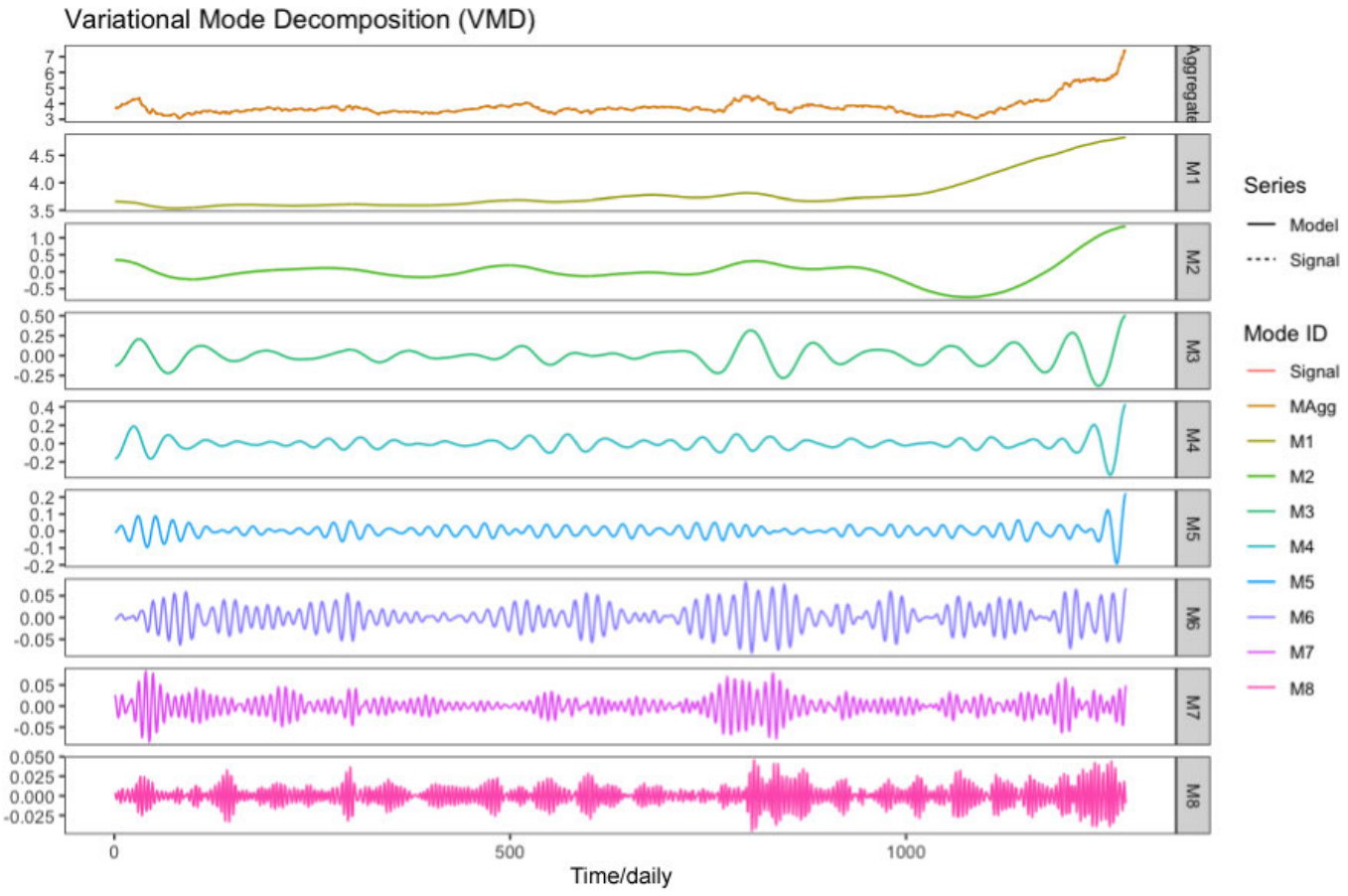


Figure 6.3: Variational mode decomposition (VMD) curves for daily corn price series (May 2016-April 2021)

6.7 Comparative analysis of corn futures Series

As it can be observed from the experimental procedure, after the decomposition of corn price series into different sub-series using the EMD and VMD methods, we used the BPNN method to estimate the last 20% of the decomposed corn price series, precisely, two hundred and fifty-five (255) predicting points of the overall sub-series and added up to obtain two hundred and fifty-five (255) predicting points of corn futures' prices. The corn price series was fitted into BPNN method devoid of decomposing the price series and serves as a benchmark model to assess the decompositions' ability of EMD and VMD. Four decomposition-based models, namely, EMD-BPNN, VMD-BPNN, EMD-ARIMA, and VMD-ARIMA were formed.

Thereafter, we fitted ARIMA model, which serves as the relative model to assess the effectiveness of the forecasting strength of the BPNN method. Comparative study of one-day ahead out-sample forecasting

outcomes of six (6) methods were discussed as give in Figure 6.5. The models' predictive capability is evaluated utilizing three statistical measures - MAE, RMSE, and MAPE as given in Table 6.1.

It can be observed from Table 6.1 and Figure 6.4 that the VMD-ARIMA forecasting performance surpassed all the suggested predicting methods, that is, EMD-BPNN, EMD-ARIMA, VMD-BPNN, BPNN, and ARIMA, in terms of MAE, RMSE, and MAPE, with error measures 0.3566, 0.5886 and 0.0954, respectively. This means that the decomposition methods proposed in this research could raise the expectedness of the ARIMA method in corn futures market prices series prediction. The VMD-based model forecasting ability was superior to all models integrated with EMD approaches, indicating that the VMD technique has high data pre-treatment ability over the EMD method. The VMD-ARIMA method has the lowest forecasting error value in comparison with VMD-BPNN, EMD-BPNN, and EMD-ARIMA with reference to the three rating measures. Moreover, the VMD-ARIMA model has the decreased MAE, RMSE, and MAPE values in the real data point of view. In comparing the BPNN and the ARIMA methods, the BPNN model has high predictive ability than the ARIMA model, suggesting that the BPNN method has demonstrated to be adaptable, respect to MAE, RMSE, and MAPE, as presented in Table 6.1 and 6.4, separately. In addition, it can be seen from Table 6.1and Figure 6.4 that the EMD techniques could not increase the forecasting precision of BPNN method in terms of the three predicting criteria measures.

6.8 Pairwise Comparison of suggested models of Corn by DM Test: Two-sided

This part of the study compares the forecasting strength of the proposed methods by utilizing the DM test in choosing the most suitable method in estimating corn futures market price series. As stated earlier, using MAE, RMSE, and MAPE to rate predicting model abilities have peculiar drawbacks, including stochastic differences in choosing the best predicting method. The traditional indices approach sometimes give bias outcomes if the effect of stochastic difference is high, hence, the present study, adopted the DM method as a novel approach in selecting the most suitable method in predicting commodities futures market prices series based on a statistical hypothesis test.

We followed Chen et al., (2014) in studying the predicting abilities of the proposed methods. The null hypothesis is that the two predicting methods have equal forecasting accuracy. An alternative "two-sided", hypothesis is that the two methods have distinctive qualities of forecasting accuracy. A DM-value above 1.9600 between two methods suggests that method 1 and method 2 have distinctive levels of forecasting accuracy. A DM-value below 1.9600 indicates that the two methods have equal degrees of forecasting accuracy. A positive DM test statistic shows that the method 1 is more efficient than method 2, while a

negative DM value implies that the method 2 is superior to method 1. Table 6.2 presents the pairwise comparison of the suggested forecasting methods. From Table 6.2, the comparison between VMD-ARIMA and EMD-ARIMA, DM test value of 9.2190 is more than the threshold value of 1.9600 with reference to absolute-error loss, we rejected the H_0 at the 5% level of significance, which is to say, there is significant differences between the two forecasting methods, VMD-ARIMA model has a higher predicting accuracy over the EMD-ARIMA model. In addition, according to square-error loss, the absolute DM value of 4.5420 is above the benchmark value of 1.96, and alternative hypothesis is accepted at the 5% level of significance, indicating a remarkable predicting differences between the two methods, and concluded that the VMD-ARIMA model has a quality forecasting precision than the EMD-ARIMA model. Likewise, the DM test value of 11.6520 with respect to percentage-error loss, is greater than the standard value of 1.9600, the null hypothesis is rejected at the 5% significance level, hence, the VMD-ARIMA method and EMD-ARIMA method have distinctive forecasting abilities, and the VMD-ARIMA method has higher predicting quality as against the EMD-ARIMA method.

Similarly, the predictive analysis of the BPNN method and the ARIMA method from Table 6.2, the three DM tests, namely, the absolute-error loss, the square-error loss, and the percentage-error loss evaluate the BPNN method as the most efficient forecasted method, which suggests that the BPNN method forecasting precision is ahead of the ARIMA method. The forecasting analysis of the EMD-BPNN and the ARIMA methods, the DM test in terms of absolute-error loss, percentage-error loss, and percentage-error loss, shows that there is a remarkable forecasting differences between the two methods, since the DM values of 3.7040, 5.3130 and 4.3230 are more than 1.9600, suggesting that EMD-BPNN method has best forecasting performance as against ARIMA method. By comparison, there exists no significant forecasting performance between the VMD-BPNN method and the BPNN method, and this might be caused by stochastic differences. Furthermore, it can be seen from Table 19 that the predicting capability of the VMD-ARIMA method and the BPNN method were the same, and the differences might be as a results of stochastic differences in the series.

Finally, the predicting comparisons of the remaining methods are summarized and presented in Table 6.2. Essentially, the VMD-ARIMA method is the most suitable method in forecasting corn futures market prices data, since it raises forecasting precision, including the MAPE, in contrast to the EMD-BPNN, EMD-ARIMA, and ARIMA methods. It can observed that models integrated with the VMD techniques performance were satisfied due to number of reasons:

1. VMD has ability to calculate all the associated signal by means of searching for a single frequency spectrum using Hilbert transformation;

2. VMD technique is suitable for random sampling and non-linear data, including agricultural commodity futures market price series, and

3. VMD utilizes the bandwidth evaluator strategy by means of Gaussian smoothness to extract signals before decomposing the data.

As a whole, VMD is fully joined to the Wiener filter, making the VMD method acceptable for decomposing noisy data (Dragomiretskiy and Zosso, 2013). In addition, it is capable in capturing short and long memories fluctuations in time series as against other decomposition methods. One-day ahead forecasting of corn's futures market price using the suggested methods is presented in Figure 6.5 .

	Criteria		
Model	MAE	RMSE	MAPE
BPNN	0.3875	0.6605	0.1017
EMD-BPNN	3.7688	4.0085	12.5925
EMD-ARIMA	1.0331	1.1897	0.2073
VMD-BPNN	0.3902	0.6052	0.1017
VMD-ARIMA	0.3566	0.5886	0.0954
ARIMA	3.7217	3.7742	49.1343

Table 6.1: Predicting performance evaluation of suggested models of corn

	MAE		RMSE		MAPE	
Model	DM value	p-value	DM value	p-value	DM value	p-value
EMD-BPNN*VMD-BPNN	23.4660	2.2e-16	10.4050	2.2e-16	26.244	2.2e-16
EMD-BPNN*EMD-ARIMA	20.402	2.2e-16	10.251	2.2e-16	23.5600	2.2e-16
EMD-BPNN*VMD-ARIMA	23.514	2.2e-16	10.4100	2.2e-16	26.2800	2.2e-16
EMD-BPNN*BPNN	23.8470	2.2e-16	10.4840	2.2e-16	26.2430	2.2e-16
EMD-BPNN*ARIMA	3.7040	0.0003	5.3130	2.357e-07	4.3230	2.206e-05
VMD-BPNN*EMD-ARIMA	-9.1130	2.2e-16	-4.8550	2.11e-06	-11.4480	2.2e-16
VMD-BPNN*VMD-ARIMA	1.4277	0.155	-0.162	0.8710	2.3370	0.0200
VMD-BPNN*BPNN	-1.4690	0.1430	-1.9301	0.0546	0.1420	0.8870
VMD-BPNN*ARIMA	-39.632	2.2e-16	-11.8860	2.2e-16	-549.06	2.2e-16
VMD-ARIMA*EMD-ARIMA	9.2190	2.2e-16	4.5422	8.608e-06	11.6520	2.2e-16
EMD-ARIMA*ARIMA	-29.4190	2.2e-16	-11.5090	2.2e-16	-93.9310	2.2e-16
EMD-ARIMA*BPNN	6.9710	2.706e-11	1.1440	0.2540	11.1490	2.2e-16
VMD-ARIMA*ARIMA	-39.928	2.2e-16	-11.9030	2.2e-16	-636.4900	2.2e-16
VMD-ARIMA*BPNN	-1.9320	0.0545	-1.9540	0.0520	-1.7110	0.0880
ARIMA*BPNN	42.8310	2.2e-16	12.1930	2.2e-16	508.7700	2.2e-16

Table 6.2: DM Test of Suggested models of Corn: Two sided

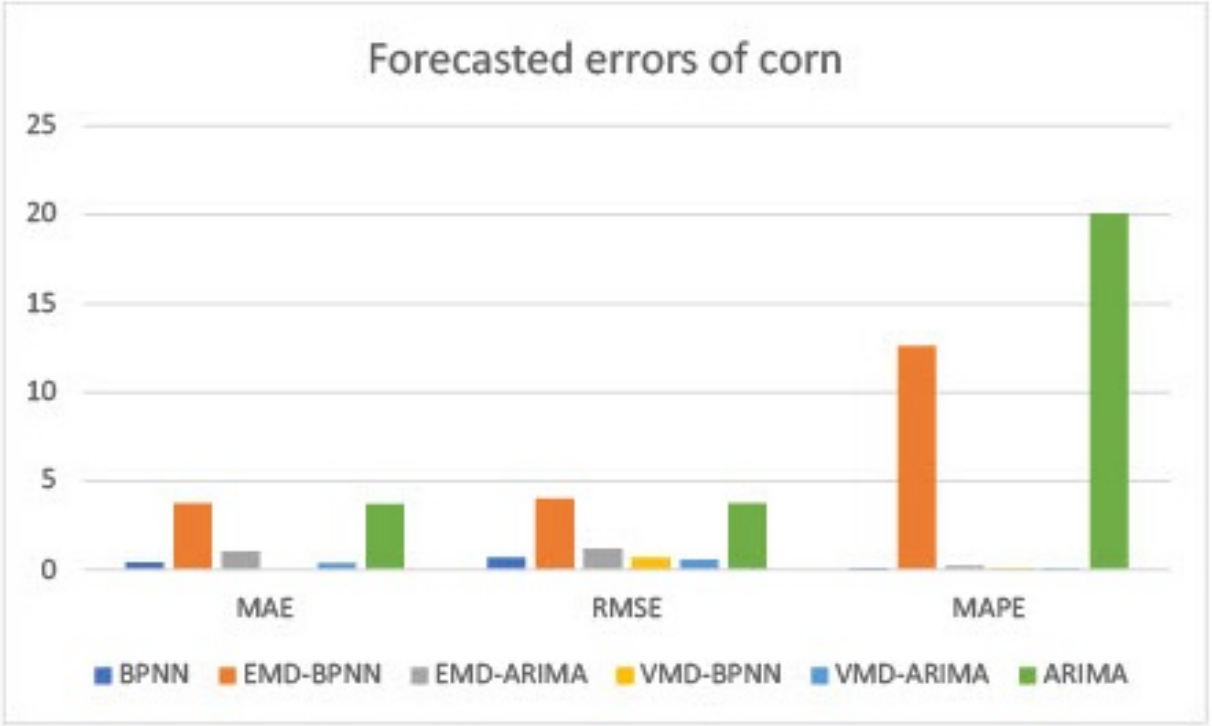


Figure 6.4: Graphical representation of errors of suggested models of corn (2016-2021)

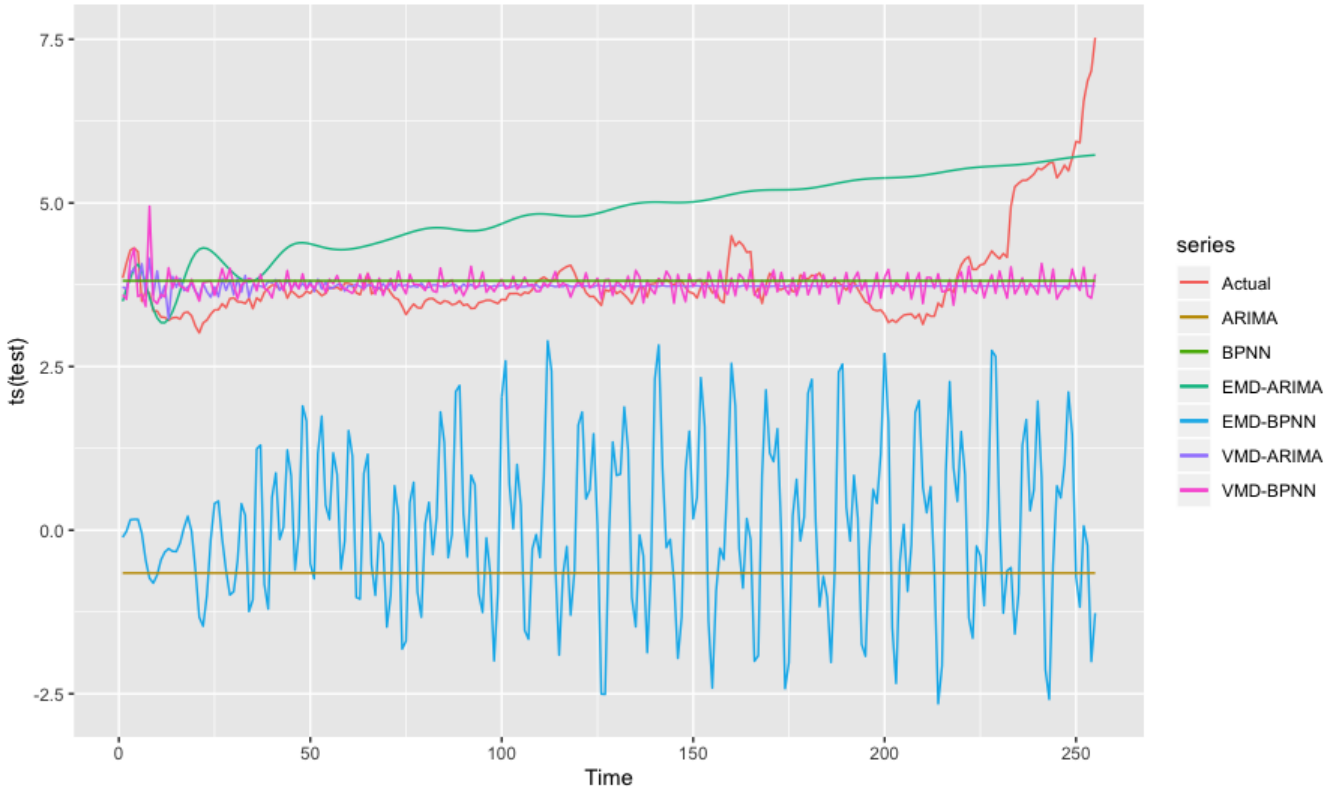


Figure 6.5: A day-ahead out of sample predicting results of suggested models of corn (2016-2021)

6.9 Decomposition results of daily crude oil price series through EMD and VMD

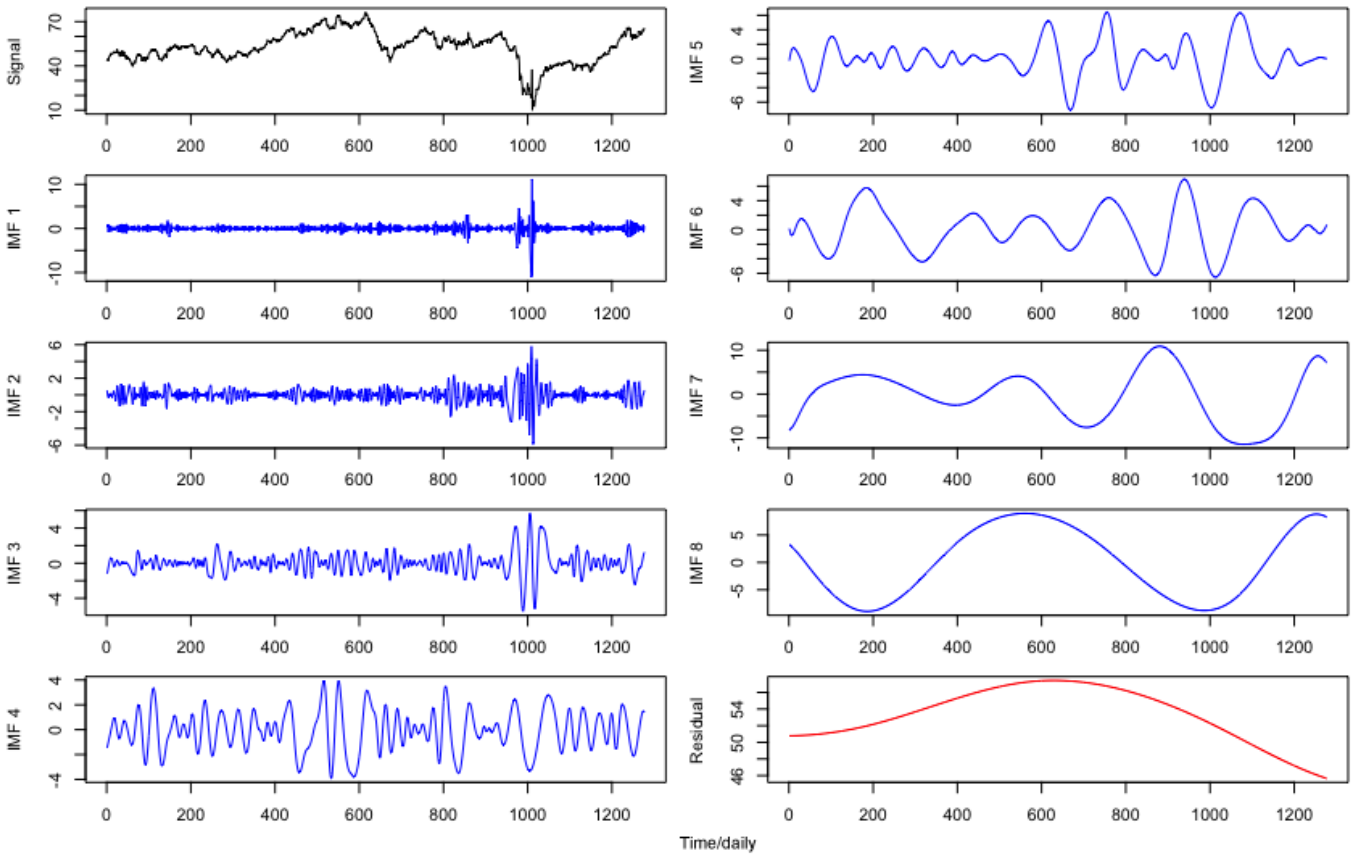


Figure 6.6: Empirical mode decomposition curves of daily crude oil price series (May 2016-April 2021)

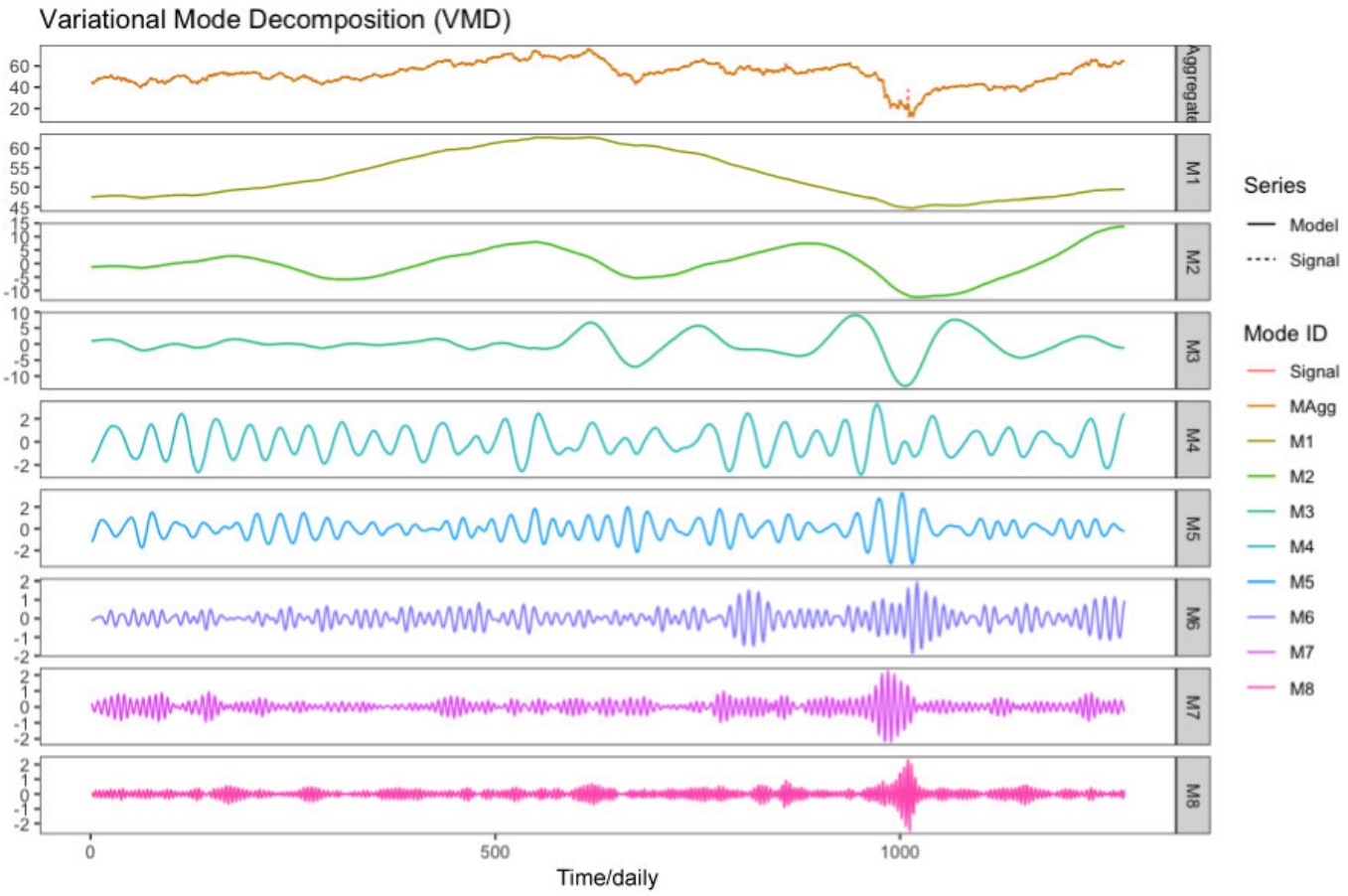


Figure 6.7: Variational mode decomposition curves of daily crude oil price (May 2016-April 2021)

6.10 Comparative analysis of crude Oil

The comparative and forecasting analysis of day-to-day crude oil price series using the proposed decomposition-based methods is presented in this section of the study. With reference to the decomposition results in Figures 6.6 and 6.7, we utilized the decomposition-based methods to predict the crude oil futures' prices, which constitute 1277 data points. We employed the BPNN method to predict the latest 20% of the crude oil price series devoid of decomposition methods, securing 255 forecasting points overall of each sub-series. The entire 255 predicting points of oil futures' prices series are shown in Figure 6.9.

The BPNN method appeared as the optimal method as against the EMD-BPNN model, VMD-BPNN model, EMD-ARIMA model, VMD-ARIMA model, and ARIMA model, with reference to MAE, as indicated in Table 6.3. According to MAE, the performance of the six methods, in ascending order, is BPNN (8.1535), VMD-ARIMA (8.3016), EMD-ARIMA (8.3621), VMD-BPNN (9.2334), EMD-BPNN (51.6060) and ARIMA (23.7060). The VMD-ARIMA method came out as the best model in estimating crude oil futures price series with reference to RMSE estimation criterion. The VMD-ARIMA method has quality predicting accuracy as compared to other decomposed-based methods. The performance of the models in order of importance based on RMSE criterion, is VMD-ARIMA (10.6912), BPNN (10.7716), EMD-ARIMA (10.8019), VMD-BPNN (11.6129), ARIMA (25.4995) and EMD-ARIMA (53.9339). Comparably, the EMD-ARIMA method appeared to be the most appropriate method in forecasting oil price series based on MAPE. According to MAPE, the performance abilities of the suggested methods of oil price series are: VMD-ARIMA (0.1532%), BPNN (0.1546%), VMD-ARIMA (0.15587%), VMD-BPNN (0.1796%), ARIMA (0.7913%) and EMD-PBNN (16.4053%). It can be observed that each evaluating criterion selected different model as the best method in predicting crude oil price series, this brought a contradiction as which method is suitable for forecasting oil price series, hence, DM method was introduced to assess the forecasting performance precision of the proposed methods among the three statistical measures- MAE, RMSE, and MAPE - to select the most efficient method to estimate the crude oil price series. The errors of the proposed methods are given in Figure 6.8.

6.11 Pairwise Comparison of suggested models of Crude oil Based on DM Test: Two sided

According to AE loss from Table 6.4, the DM-AE value of 26.61 exceeded the threshold point of 1.96, hence, the H_0 is discarded at 5% level of significance, and say that, there is a notable predicting distinction involving the two models, and that the forecasting performance of VMD-BPNN model is superior to the

EMD-BPNN model as illustrated in Table 6.4. Likewise, as reported by SE loss and PE loss, both DM-SE and DM-PE of 14.8000 and 16.7980 are greater than 1.9600, suggesting that the VMD-BPNN method has high forecasting precision than the EMD-BPNN method in estimating oil price series. Equivalently, the forecasting comparison relating to EMD-BPNN method and EMD-ARIMA method, the three loss functions- AE loss, SE loss, and PE loss- evaluations pointed out that there is a remarkable forecasting difference involving the EMD-BPNN and EMD-ARIMA methods, therefore, the EMD-ARIMA method has better predicting precision than EMD-BPNN method, since the DM test results of 26.3160, 14.8000 and 17.1880, with respect to DM-AE, DM-SE, and DM-PE are above the standard value as presented in Table 6.4.

According to DM test report, all the three loss functions, namely, AE loss, SE loss and PE loss affirm that there exist a noticeable predicting performance involving the EMD-BPNN method and the VMD-ARIMA method, hence, the predictive performance of the VMD-ARIMA method is more satisfactory than the EMD-BPNN method. Based on DM-AE, DM-SE and DM-PE loss functions, the DM test report showed that there is no predicting differences involving the VMD-BPNN and BPNN predictability methods, since the DM test statistic is lower than the benchmark value and the difference in them might be originated from stochastic procedure. In comparing BPNN and EMD-BPNN methods, the three DM tests report, DM-AE loss, DM-SE loss, and DM-PE loss confirm that the forecasting ability of the BPNN method is superior to the EMD-BPNN. Comparably, in assessing predictability of the EMD-BPNN method and ARIMA method, according to the errors - AE loss, SE loss and PE loss- evaluated that the forecasting strength of the EMD-BPNN method is more acceptable than ARIMA method. The BPNN method has better forecasting ability than the ARIMA method according to DM-AE and DM-SE test respectively. In conclusion, the EMD-ARIMA method appeared as the most preferred method for forecasting crude oil futures market price series.

	Criteria		
Model	MAE	RMSE	MAPE
BPNN	8.1535	10.7716	0.1546
EMD-BPNN	51.6060	53.9339	16.4053
EMD-ARIMA	8.3621	10.8019	0.1532
VMD-BPNN	9.2334	11.6129	0.1796
VMD-ARIMA	8.3016	10.6912	0.1559
ARIMA	23.7060	25.4995	0.7913

Table 6.3: Forecasting performance evaluation of suggested models of crude oil

	MAE		RMSE		MAPE	
Model	DM value	p-value	DM value	p-value	DM value	p-value
EMD-BPNN*VMD-BPNN	26.61	2.2e-16	14.8000	2.2e-16	16.7980	2.2e-16
EMD-BPNN*EMD-ARIMA	26.3160	2.2e-16	14.7900	2.2e-16	17.1880	2.2e-16
EMD-BPNN*VMD-ARIMA	5.8766	1.31e-08	6.9303	3.447e-11	4.2954	2.485e-05
EMD-BPNN*BPNN	26.4420	2.2e-16	14.7970	2.2e-16	16.917	2.2e-16
EMD-BPNN*ARIMA	24.341	2.2e-16	14.452	2.2e-16	17.8160	2.2e-16
VMD-BPNN*EMD-ARIMA	2.5517	0.01131	1.0881	0.2776	-0.2578	0.7968
VMD-BPNN*VMD-ARIMA	-36.447	2.2e-16	-20.7590	2.2e-16	-8.2758	7.28e-15
VMD-BPNN*BPNN	1.1991	0.2316	0.3931	0.6946	0.4611	0.6451
VMD-BPNN*ARIMA	-16.6990	2.2e-16	-10.3450	2.2e-16	-0.4431	0.6581
EMD-ARIMA*VMD-ARIMA	-35.3020	2.2e-16	-20.6810	2.2e-16	-8.4249	2.702e-15
EMD-ARIMA*ARIMA	-16.5070	2.2e-16	-10.3710	2.2e-16	-0.4310	0.6669
EMD-ARIMA*BPNN	0.0440	0.9649	0.0784	0.9375	0.4923	0.6229
VMD-ARIMA*ARIMA	39.5570	2.2e-16	21.2030	2.2e-16	135.1600	2.2e-16
VMD-ARIMA*BPNN	35.7050	2.2e-16	20.6990	2.2e-16	26.193	2.2e-16
ARIMA*BPNN	16.297	2.2e-16	10.5610	2.2e-16	2.6326	0.0090

Table 6.4: Diebold Mariano Test: Two-sided

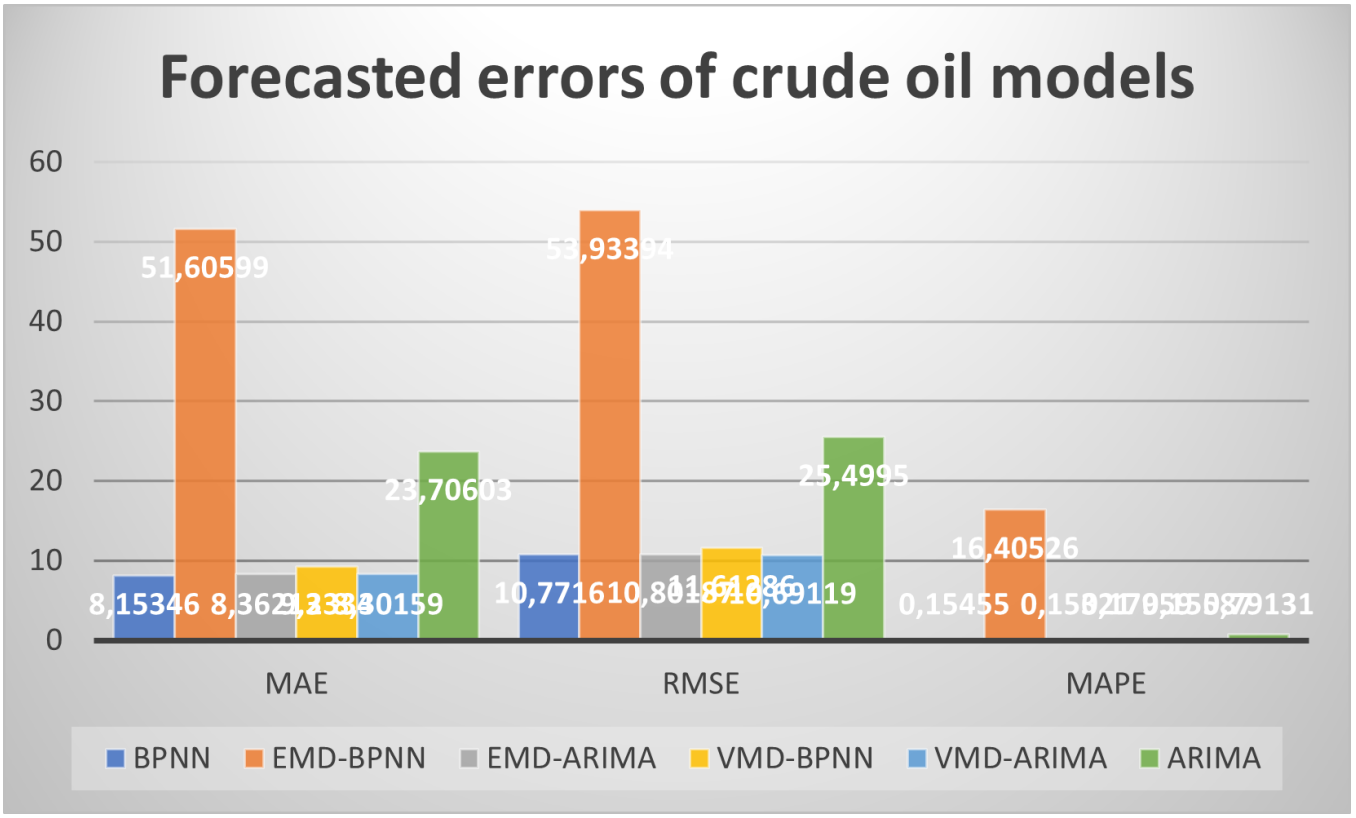


Figure 6.8: Graphical representation of errors of suggested models of crude oil

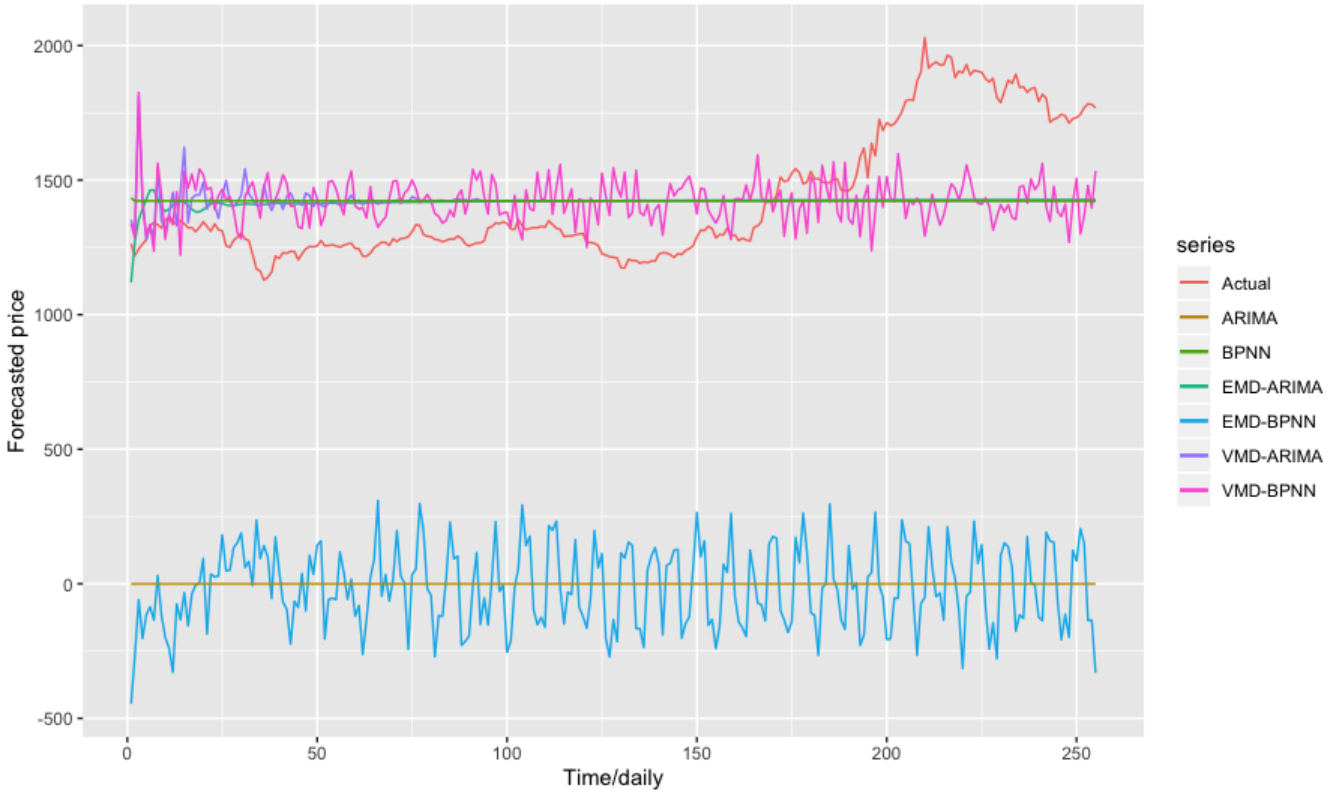


Figure 6.9: A day ahead out of sample predicting results of different models of gold (2016-2021)

6.12 Decomposition results of day-to-day price series of gold through EMD and VMD

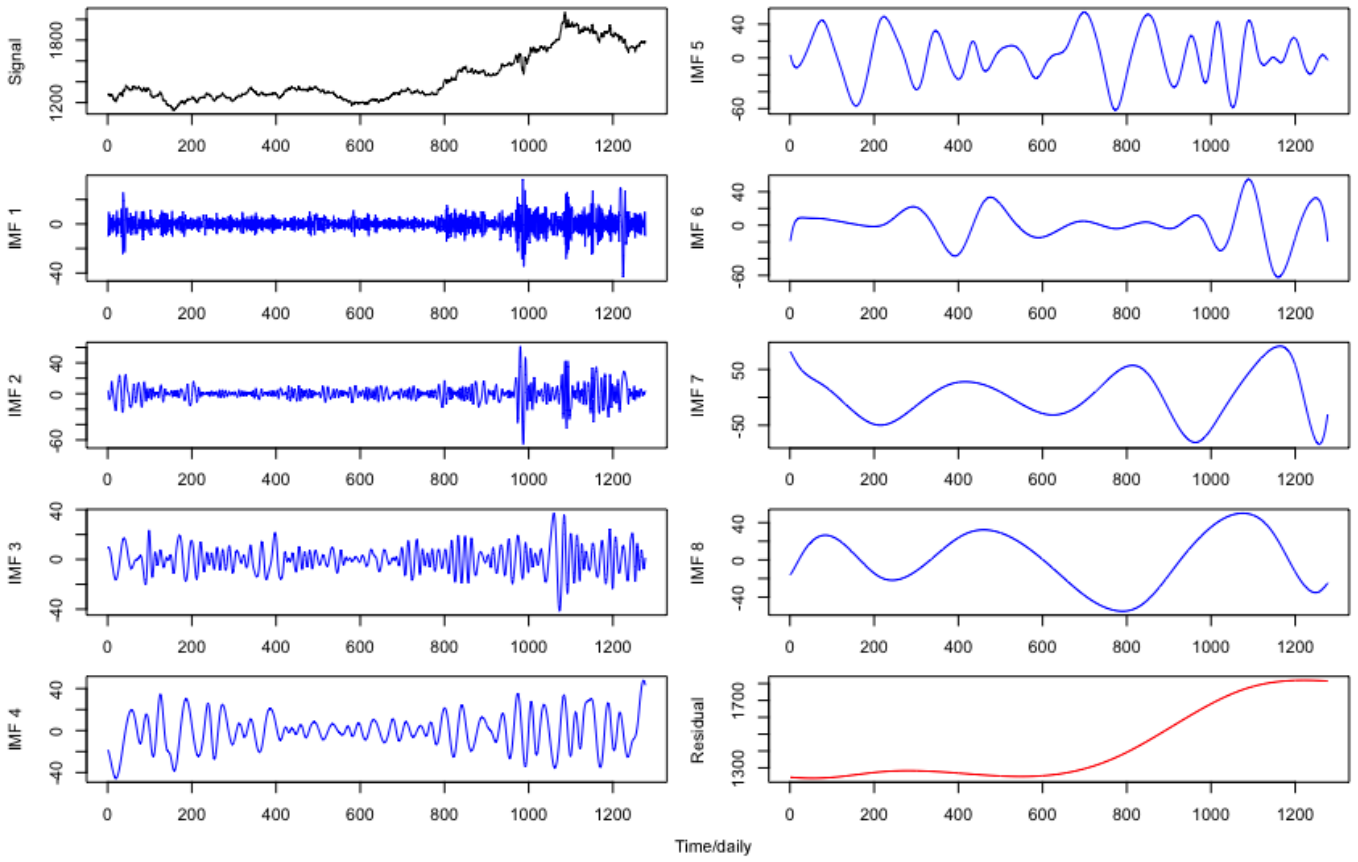


Figure 6.10: Empirical mode decomposition (EMD) curves for the daily gold price series (May 2016-April 2021)

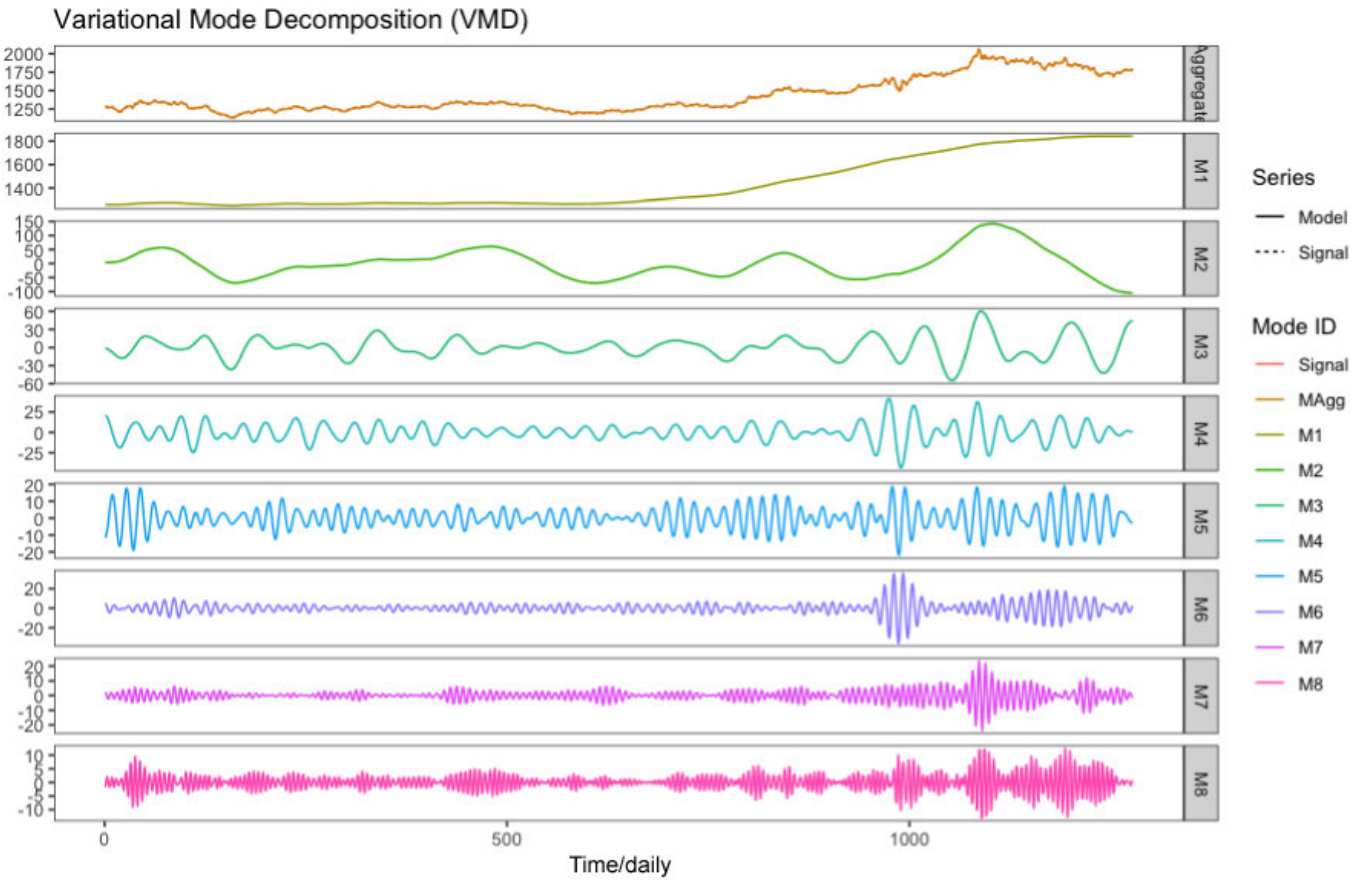


Figure 6.11: Variational mode decomposition (VMD) curves for the daily gold price series (May 2016-April 2021)

6.13 Comparative Analysis of gold

This part presents the comaparative study of gold prices' series using EMD and VMD techniques. Figures 6.10 and 6.11 present the decomposition results of the two suggested methods, separately. The proposed methods were utilized to predict latest 20% price series of gold as shown in Table 6.5 and Figure 6.13. Figure 6.12 represents the errors analysis of the proposed methods.

Four decomposition-based models were applied to predict the gold futures prices, which comprise 1277 data points. The BPNN method was employed to predict the latest 20% price series of gold without decomposing the price series of gold and serves as benchmark model, securing 255 predicting points overall of every sub-series. A total of two hundred and fifty-five (255) predicting points of prices of gold, as presented in Figure 6.13.

Correspondingly, the forecasting outcome revealed BPNN method as the best predicting model as compared to the four proposed decomposition-based methods, EMD-BPNN, EMD-ARIMA, VMD-ARIMA,

VMD-BPNN, and the ARIMA method, according to the MAE, RMSE, and MAPE,respectively, as given in Table 6.5 and Figure 6.12. Based on the three estimating criteria, the VMD-ARIMA method and EMD-ARIMA method predictabilities are superior to the ARIMA method, in other words, the decomposition used in this study increases forecasting precision of BPNN and ARIMA models, as given in Table 6.5. In comparing the EMD-ARIMA and VMD-ARIMA methods, the report from the three criteria revealed that the EMD-ARIMA method has a superior forecasting ability to the VMD-ARIMA method. Similarly, the comparison involving the EMD-BPNN method and VMD-BPNN method suggested that the VMD- BPNN method possess quality predicting capability than EMD-BPNN method in terms of three criteria, MAE, RMSE and MAPE. With reference to MAPE, the predicting abilities of the six methods could be arranged in order of magnitude as: BPNN (0.1386%), EMD-ARIMA (0.1402%), VMD-ARIMA (0.1437%), VMD-BPNN (0.1466%), ARIMA (0.8058%), and EMD-BPNN (55.9444%). The decomposition techniques turn to increase ARIMA method but fail to improve the predicting capabilty of BPNN method.

6.14 Pairwise Comparison of suggested models of gold Based on DM Test: Two-sided

Equivalently, the proposed methods of gold predictability performance were tested using DM test to assess their forecasting precisions, as illustrated in Table 6.6. The BPNN method came out as the most suitable model for esimating gold prices' series, in terms of all three forecasting loss functions, DM-AE, DM-SE, and DM-PE, respectively. According to DM test report, the VMD-ARIMA method is considered as the next important model in forecasting gold price series. The EMD-ARIMA method appeared as the third most suitable method in forecasting from DM test analysis of the three loss functions. The remaining pairwise analysis performance ability of the models is outlined in Table 6.6. It is suggested that the BPNN method is deemed appropriate in predicting market prices of gold.

	Criteria		
Model	MAE	RMSE	MAPE
BPNN	197.2114	236.5327	0.1386
EMD-BPNN	1467.8270	1495.7420	55.9444
EMD-ARIMA	199.3610	241.8806	0.1402
VMD-BPNN	208.9359	259.2293	0.1466
VMD-ARIMA	205.3709	249.3515	0.1437
ARIMA	636.5843	679.0113	0.8058

Table 6.5: predicting performance evaluation of gold

	MAE		RSME		MAPE	
Model	DM value	p-value	DM value	p-value	DM value	p-value
EMD-BPNN*VMD-BPNN	41.4280,	2.2e-16	19.7540	2.2e-16	73.4550	2.2e-16
EMD-BPNN*EMD-ARIMA	41.66 00	2.2e-16	19.7503	2.2e-16	74.3120	2.2e-16
EMD-BPNN*VMD-ARIMA	2.8361	0.0049	3.7064	0.0003	2.7998	0.0055
EMD-BPNN*BPNN	41.412	2.2e-16	19.7520	2.2e-16	74.8950	2.2e-16
EMD-BPNN*ARIMA	45.4820	2.2e-16	20.3230	2.2e-16	58.5510	2.2e-16
VMD-BPNN*EMD-ARIMA	3.531	0.0005	3.1258	0.0020	3.7150	0.0003
VMD-BPNN*VMD-ARIMA	-48.0290	2.2e-16	-21.6760	2.2e-16	-559.4200	2.2e-16
VMD-BPNN*BPNN	2.7844	0.0058	2.07110	0.0394	2.6956	0.0075
VMD-BPNN*ARIMA	-21.4260	2.2e-16	-10.9050	2.2e-16	-40.6510	2.2e-16
EMD-ARIMA*VMD-ARIMA	-48.2470	2.2e-16	-21.6700	2.2e-16	-784.7500	2.2e-16
EMD-ARIMA*ARIMA	-22.0390	2.2e-16	-10.9260	2.2e-16	-44.4810	2.2e-16
EMD-ARIMA*BPNN	0.9622	0.3369	0.5068	0.6127	-0.2385	0.8117
VMD-ARIMA*ARIMA	63.2740	2.2e-16	23.0400	2.2e-16	169.3200	2.2e-16
VMD-ARIMA*BPNN	47.5500	2.2e-16	21.6560	2.2e-16	778.0100	2.2e-16
ARIMA*BPNN	21.2830	2.2e-16	10.8770	2.2e-16	41.4630	2.2e-16

Table 6.6: Diebold Mariano Test; Two-sided

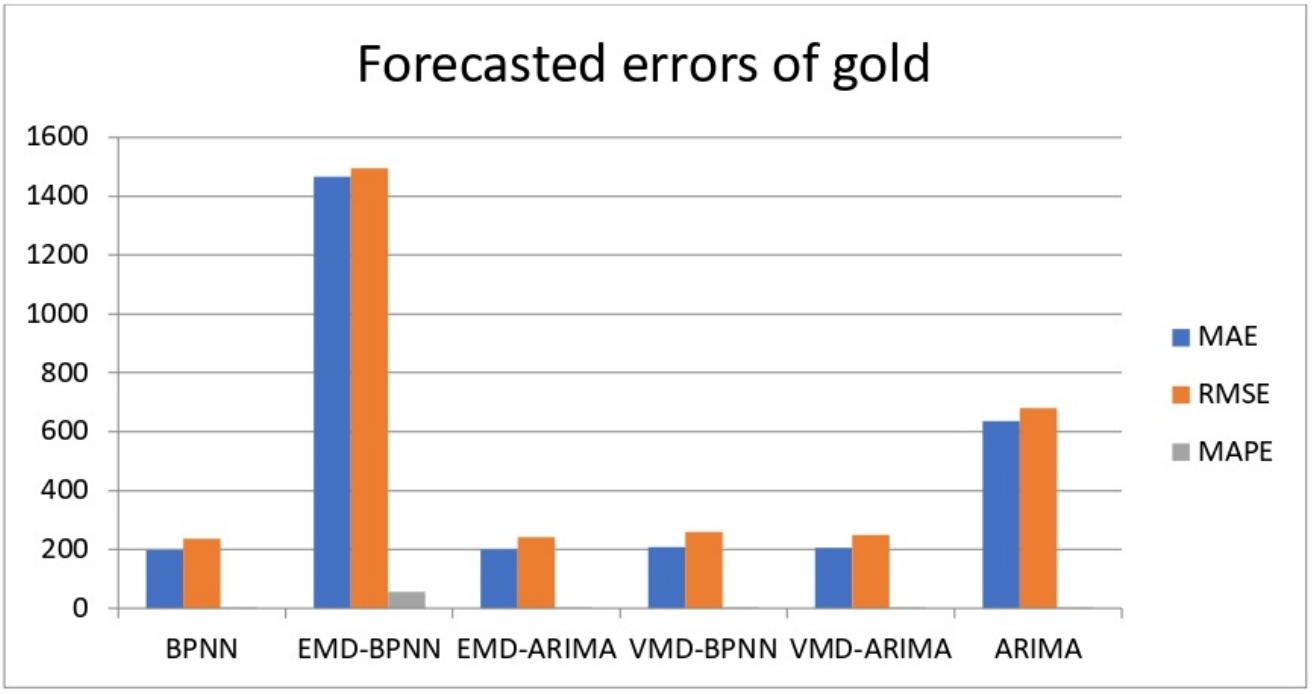


Figure 6.12: Graphical representation of errors of suggested models of gold

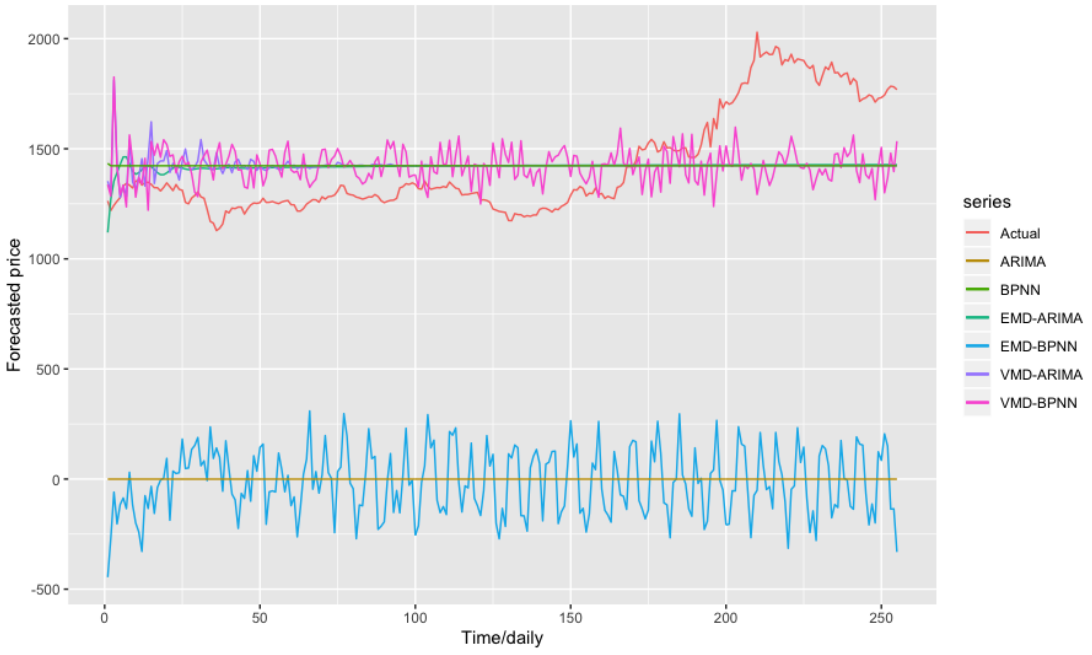


Figure 6.13: A day ahead out of sample predicting results of different models of gold (2016-2021)

6.15 Conclusion

The effectiveness of two decomposition methods, EMD and VMD have been presented in this chapter.The decomposition techniques integrated with BPNN method and ARIMA method were utilized to predict market price series of three commodities - crude oil, corn, and gold - chosen across commodity markets. We utilized the EMD technique and VMD technique to decompose each commodity series into distinctive IMFs, modes, and residue to forecast the futures market price of above stated commodities markets. We used the decompsition methods to enhance the forecasting precision of BPNN method and ARIMA method by comparing the predicting ability of proposed method using the forecasting performance criteria, MAE, RMSE, and MAPE in rating the six methods. Subsequently, we introduced Diebold Mariano test to choose the best predicting method for each commodity market price series, since it is not reliable to use the traditional methods- MAE, RMSE and MAPE- in selecting the ideal method due to some peculiar drawbacks stated earlier.

It was revealed that, the decomposition-based method suggested, outperformed the benchmark methods, BPNN and ARIMA, in predicting corn market prices' series and crude oil market prices' series, however, it failed to predict the gold price series.

By these findings, we arrive at the following inferences:

1. the VMD-ARIMA method appeared as the most suitable model in predicting corn futures market price series as against the EMD-BPNN, EMD-ARIMA, VMD-BPNN, BPNN, and ARIMA methods in respect of evaluating criteria, MAE, RMSE, and MAPE respectively, while EMD-ARIMA method came out as the ideal model for forecasting crude oil futures market price series, and BPNN method is the most efficient model in forecasting gold futures market prices series;

2. in all instances, decomposition-based models predicting performance were more satisfactory than the ARIMA method throughout the study;

3. the predicting ability of the VMD-BPNN method was better than the EMD-BPNN method, EMD-ARIMA method, and VMD-ARIMA method, indicating that the VMD technique is robust in pretreating data since it enhances the predicting accuracy, and

4. applying a decomposition-based techniques in predicting commodity futures market prices, lawmakers, regimes, and firms can judiciously take resolutions to reduce loss and increase profit. It is suggested that analysts interested in predicting commodity futures market prices should think about the decomposition-based methods since they raise forecasting precision.

Chapter 7

General Conclusions, Contributions, Recommendations, Limitations and Further Research

7.1 General Conclusions

Commodity market price instability is critical to economic growth and political steadiness. Commodity futures' prices play a vital part in hedging risk and discovering prices, accurate prediction of commodity futures market prices plays an important role in production operations and investment decisions and provides some productive suggestions on how the government might adjust its policies. Nevertheless, the commodity futures market is affected by many factors, such as economic development, oil price instability, invasion, pandemics, financial speculation, weather condition, hence, it is extremely complex, non-stationary, and non-linear (White, 2000, Nazlioglu and Soytas, 2012, Huchet-Bourdon, 2011, Rausser and De G, 2014). These complex features in commodity futures prices series make accurate prediction of commodity futures market prices very challenging.

Substantial previous studies have suggested potential predicting models, however, choosing the ideal model for a given competing forecast models is a demanding task (Wang and Li, 2018, Kourentzes et al.,

2019, Zhang et al., 2019). Decomposition-based forecasting has become one of the most robust methods for predicting complex data series, and it has proven to give more definite and precise results, than using individual forecasts (Bates and Granger, 1969, Ismael, 2008, Nowotarski et al., 2016).

The decomposition-based models stemmed from conventional times series models and machine learning models. Conventional times series models, such as the ARIMA models, have been used extensively in explaining and predicting commodity price series (Ball, 1985, Maghyereh, 2006, Mohammadi and Su, 2010, Sun et al., 2010). The intrinsic features of commodity price data, however, hinder the use of conventional models as they are less effective in analyzing commodity data making these models less reliable and accurate as they could produce false results and conclusions (Tsay, 1987).

Commodity data series are usually non-stationary and inevitably noisy, as a result, decomposition methods are often used to pre-treat such time series to extract the noise impediment (Abu-Mostafa and Atiya, 1996, Adam et al., 2022). This process allows only signals with relevant information in the time series. Previous studies have proposed different decomposition techniques based on different model assumptions, to cater for the irregularities in the commodity data series. These include singular spectral analysis (SSA) Vautard et al., (1992), empirical mode decomposition (EMD) Huang et al., (1998) and variational mode decomposition (VMD) Dragomiretskiy and Zosso (2013). In response to the setbacks of the traditional models, this thesis suggested a novel technique hinged on empirical mode decomposition and variational mode decomposition to study the determinants, mutual relationships, and the forecasting of commodity futures market prices using commodity data of corn, crude oil, and gold from Bloomberg Commodity Index. EMD and VMD are instinctive, adaptive, and robust. EMD and VMD can decompose complex data into simple units called 'IMFs' or 'modes', thus, they enhance times series analysis of individual models and increase forecasting precision of BPNN and ARIMA models. In the subsequent paragraphs, we present chapter-by-chapter details of the thesis.

In the first chapter, we presented a general introduction of the study concentrating on the background of the study, rationale, problem statement, research objectives, scope and the expected contributions of the study to literature.

In Chapter Two, we reviewed the relevant literature on empirical mode decomposition and variational mode decomposition and their competing models, such as singular spectral analysis, wavelet transforms, and discrete wavelet transform. It was discovered that EMD and VMD outperformed the SSA, DWT, Fourier transforms, and Wavelet transforms, in terms of their uniqueness in extracting the embedded intrinsic factors in the data - flexibility, empirical, and ability to decompose non-linear and nonstationary times series data. In addition to this, due to the existence of non-linear features in the commodity market price series data generation, the EMD and VMD as a data pre-treatment techniques increased the

forecasting accuracy and analysis of commodity futures' markets.

In the third chapter, we presented the theoretical background of the suggested methods used in this study - the EMD and the VMD techniques. Then the back-propagation neural network and autoregressive integrated moving average were introduced as a benchmark and comparative models, respectively. It was revealed that BPNN could estimate any non-linear continuous function to any desirable level compared to the traditional methods.

In Chapter Four, we examined the descriptive analysis and the statistical properties of the three commodities markets. It was observed that corn and gold price series were positively skewed, while crude oil price series was negatively skewed, hence, these three commodities market prices did not follow normally distribution curve. It was also revealed that the daily spot prices of the three commodities were leptokurtic. The Jarque-Bera test revealed that the three-commodities price series were not normally distributed. We also tested for the non-linearity and non-stationarity of the three-commodities price series using Keenan, Tsay, and ADF tests. The ADF test results revealed that the corn, crude oil, and gold price series were nonstationary. It was emerged from the results of the Keenan and Tsay's tests that corn, crude oil, and gold markets prices were nonlinear. Finally, the EMD-Causality analysis was introduced to examine the causal relationships among three commodities markets' prices. The results showed strong mutual relationships among the three commodities markets, indicating the reason why crude oil market prices have persistently affected the agricultural commodity market prices.

In the fifth chapter, EMD and VMD strategies were utilized to breakdown corn, crude oil, and gold price series into IMFs or modes and residue to analyze the factors that influence commodity market price fluctuations. We used statistical measures such as Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation to evaluate the effect of each IMF or mode and residue on the commodity futures' prices. Subsequently, we employed Hierarchical Clustering and the Euclidean Distance to group the IMFs, modes, and residue into high-frequency, low-frequency, and trend components. The empirical results showed that the trend and the low-frequency components were the main drivers of commodity market price fluctuations, hence, corn, crude oil, and gold prices could be analyzed as a combination of a long-term trend, special events, and short-term fluctuations. The three commodities markets prices' fluctuations, therefore, could be explained as a combination of a long-term trend, special events, and near-term fluctuations resulting from usual market activities, such as supply-demand imbalance. Trends and special events generally drive the three commodities prices. The effects from the trend deviate gradually and change around the long-term mean. The unpredictable special events are responsible for irregular commodity price movements, and their effect could last for several years, indicating the normal market activities mainly trigger the short-term commodity price changes, and the effects have a

very short duration, hence, their impact on commodity price is not serious. Interestigly, the three commodities' markets were similar, since the three markets prices' series fluctuations were affected by the same components.

In Chapter Six, combined models proposed in this thesis were used to predict corn, crude oil, and gold futures market prices. EMD and VMD methods were used to decompose the three-commodities price series into their respective IMFs, modes, and residue and combined with BPNN and ARIMA to predict one-day-ahead out-of- sample of the aforementioned commodities market prices. We compared the forecasting capability of this novel approach using forecasting performance criteria MAE, RMSE, and MAPE. The Diebold Mariano test was also performed to select the ideal model for predicting each commodity market futures price. It was found that the VMD-ARIMA model emerged as the optimal model for predicting corn futures price series, while the EMD-ARIMA model was the best predicting model for crude oil futures price series, however, the combined models failed to improve the forecasting ability of gold futures price series. The BPNN model outperformed all model combination suggested for forecasting gold futures price series. Using a hybrid model method, market participants, government, policymakers, and businesses can make sustainable decisions to reduce loss and increase profit.

7.2 Main Contributions of the Study

The main contribution of this thesis to the existing literature are as follows:

- The research contributes to the existing literature by comparing the data pre-treatment effect of EMD and VMD horizontally, in decomposing commodity market price series and studying the underlying components that cause corn, crude oil, and gold markets price fluctuations is a novel approach in studying commodity market prices.
- Utilizing Hierarchical Clustering and Euclidean Distance Approaches, the IMFs, residue and modes were classified into their distinctive frequencies, namely, high-frequency, low-frequency, and trend units. The effect of these frequencies and trends on commodity market price fluctuation is the first of its kind in the literature.
- Using statistical measures such as Pearson product-moment correlation coefficient, Kendall rank correlation, and Spearman rank correlation coefficient to evaluate the contribution of the IMFs, residue, and modes to the net variance of the volatility of crude oil, corn, and gold markets price fluctuations, is an innovative approach to studying financial time series.

- The EMD-Causality technique proposed to study the causal relationship of corn, crude oil, and gold futures prices movement, is novel in the financial market. This new approach to study price movement of commodity markets, will provide a vital information about one commodity market to explain the other commodity market price fluctuations in various markets.

- Decomposing of financial data before forecasting has high forecasting precision accuracy in commodity futures price prediction. In addition, using decomposition techniques in agricultural, energy, and industrial metal commodities futures markets, effectively, minimizes the prediction complexity. Furthermore, using econometric and machine learning models incorporated with decomposition methods can capture the price series information up to acceptable degrees.

- Decomposition-based predicting techniques can effectively raise the predicting performance capability of BPNN and ARIMA models and reduce errors, thus, the proposed novel combination method can statistically improve forecast accuracy. This study, therefore, may assist in arresting the agriculture, energy, and industrial commodities markets trends and estimate volatility risk factors accurately, consequently serving as a guide for investors, governments policymakers and related sectors such as agricultural, energy, and industrial to take prudent and sustainable planning and investment decisions.

7.3 Recommendations

The proposed decomposition approach, especially VMD-based, is robust in analyzing the determinants, modeling, and forecasting commodity futures markets prices fluctuations, thereby, improving forecasting precision accuracy. It can reveal the embedded characteristics, the driving factors in the commodity market futures price and can handle also non-linearity and non-stationarity in the commodity market data generation. Decomposition-based model predictions have better forecasting performance than individual model forecasts. Significantly, in utilizing the decomposition technique in evaluating compositions of commodity prices data series individually, divergent forecasting strategies can be explored. First of all, based on the features of decomposed IMFs or modes, a suitable predicting technique can be considered to forecast each IMF or mode; for example, the residue can be estimated by utilizing a polynomial function, while Fourier transform can be considered in predicting low-frequency IMFs or modes. The approach can be applied in a non-linear technique to evaluate high-frequency IMFs or modes, and finally to sum up the individual predicts to secure the overall predicting results. In the second place, the IMFs or modes can be classified as linear and non-linear series , and can be predicted differently, and then combining the individual estimated values as overall forecasted results. This would mean that curve fitting can be utilized

to predict the trend component of commodity futures markets prices series, while non-linear predicting method such as artificial neural network or backpropagation neural network can be used to predict the low-frequency component of commodity market futures prices' series. Special events are hard to predict because they are caused by a number of factors, including trade wars, pandemic, climatic conditions, financial schemes, and various complicated factors, hence, it is recommended that researchers, institutions, investors, and policymakers interested in studying commodity price movements should consider using this novel technique to achieve better results. It is further suggested that the decomposition approach could be utilized in other areas of financial data modeling and analysis instead of conventional methods.

7.4 Limitation of the Study

The EMD and VMD are typical adaptive decomposition process, therefore, setting frequency resolution is crucial parameter in determining the range of application. Frequency determination deal with the number of sifting iterations, stopping criterion threshold setting and the amplitude ratio between different elements (Rilling and Flandrin, 2007, Roy and Doherty, 2008, Sun et al., 2010). Presently, there is no unique approach for stopping criterion threshold setting, which may render EMD and VMD lacking in conceptual basis. This suggests that when using EMD and VMD in processing signal with a high sampling frequency yields a good decomposition result, otherwise, an inadequate sampling frequency could add false components forming part of the IMFs.

7.5 Suggestion for Future Studies

Further study can extend the proposed methodology, the EMD and VMD and combine with the data-driven methods such as - Autoregressive Moving Average (ARMA), Autoregressive Conditional Hetroscedasticity (ARCH), Radial Basis Function Neural Network (RBFNN) models, Markov regime-switching (MRS), support vector machine (SVM), and genetic algorithm (GA) in studying financial markets. Eventually, we propose that a new model or consolidated predicting technique should be investigated to cater for special events' influences on commodity market prices since no one can predict the time and the place they will occur.

Bibliography

D Zhang, G Zang, J Li, K Ma, and H Liu. Prediction of soybean price in china using qr-rbf neural network model. *Computers and Electronics in Agriculture*, 154:10–17, 2018.

W Wang and L Wei. Impacts of agricultural price support policy on price variability and welfare: Evidence from china's soybean market. *Agricultural Economics*, 52(1):3–17, 2021.

L Zhao, X Zhang, S Wang, and S Xu. The effects of oil price shocks on output and inflation in china. *Energy Economics*, 53:101–110, 2016.

HF Zou, GP Xia, FT Yang, and HY Wang. An investigation and comparison of artificial neural network and time series models for chinese food grain price forecasting. *Neurocomputing*, 70(16-18):2913–2923, 2007.

G Kaur and D N Rao. Efficiency of indian commodities market: A study of agricultural commodity derivatives traded on ncdex. *Available at SSRN 1600687*, 2010.

P Arendas. *The Halloween Effect on Financial Markets*. Wolters Kluwer, 2017.

J Ali and K B Gupta. Efficiency in agricultural commodity futures markets in india: Evidence from cointegration and causality tests. *Agricultural Finance Review*, 2011.

E F Fama and K R French. Comparing cross-section and time-series factor models. *The Review of Financial Studies*, 33(5):1891–1926, 2020.

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

E F Fama. Efficient capital markets: A review of theory and empirical work." journal of finance 25: 385-417.. 1980. *Agency Problems and the Theory of the Firm." Journal of Political Economy*, 88:288–307, 1970.

H Arora, B Singh, S Gupta, A Singh, V S Tomar, K K De, J Kaur, J Kaur, S Karthick, S Kanojia, et al. Testing weak-form efficiency of indian stock market using high frequency data 07. 2017.

C Aktan, E E Sahin, and I Kucukkaplan. Testing the information efficiency in emerging markets. *Financial management from an emerging market perspective*, pages 49–66, 2018.

D Chhabra and M Gupta. Market efficiency and calendar anomalies in commodity futures markets: a review. *Agricultural Economics Research Review*, 33(347-2021-788), 2020.

B G Malkiel. The efficient market hypothesis and its critics. *Journal of economic perspectives*, 17(1): 59–82, 2003.

J Wang and X Li. A combined neural network model for commodity price forecasting with ssa. *Soft Computing*, 22(16):5323–5333, 2018.

H Boubaker and S A Raza. A wavelet analysis of mean and volatility spillovers between oil and brics stock markets. *Energy Economics*, 64:105–117, 2017.

A Maghyereh. Oil price shocks and emerging stock markets: A generalized var approach. In *Global stock markets and portfolio management*, pages 55–68. Springer, 2006.

A A Salisu and T F Oloko. Modeling oil price–us stock nexus: A varma–bekk–agarch approach. *Energy Economics*, 50:1–12, 2015.

J L Zhang, Y J Zhang, and L Zhang. A novel hybrid method for crude oil price forecasting. *Energy Economics*, 49:649–659, 2015.

W Kristjanpoller and M C Minutolo. Gold price volatility: A forecasting approach using the artificial neural network–garch model. *Expert systems with applications*, 42(20):7245–7251, 2015.

S Motlaghi, F Jalali, and M N Ahmadabadi. An expert system design for a crude oil distillation column with the neural networks model and the process optimization using genetic algorithm framework. *Expert systems with applications*, 35(4):1540–1545, 2008.

R. Bacon. Modelling the price of oil. *Oxford review of economic policy*, 7(2):17–34, 1991.

N E Huang, Z Shen, S R Long, M C Wu, H H Shih, Q Zheng, N C Yen, C C Tung, and H H Liu. The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences*, 454(1971):903–995, 1998.

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

A D Addison, B M Battista, and C C Knapp. Improved hydrogeophysical parameter estimation from empirical mode decomposition processed ground penetrating radar data. *Journal of Environmental & Engineering Geophysics*, 14(4):171–178, 2009.

K Dragomiretskiy and D Zosso. Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544, 2013.

Q Zhu, F Zhang, S Liu, Y Wu, and L Wang. A hybrid vmd–bigru model for rubber futures time series forecasting. *Applied Soft Computing*, 84:105739, 2019.

T Jin, H Ding, B Li, H Xia, and C Xue. Valuation of interest rate ceiling and floor based on the uncertain fractional differential equation in caputo sense. *Journal of Intelligent & Fuzzy Systems*, (Preprint):1–10, 2021.

S Dees, P Karadeloglou, R K Kaufmann, and M Sanchez. Modelling the world oil market: Assessment of a quarterly econometric model. *Energy policy*, 35(1):178–191, 2007.

C Morana. A semiparametric approach to short-term oil price forecasting. *Energy Economics*, 23(3): 325–338, 2001.

P Sadorsky. Time-varying risk premiums in petroleum futures prices. *Energy Economics*, 24(6):539–556, 2002.

S Mirmirani and H C Li. A comparison of var and neural networks with genetic algorithm in forecasting price of oil. In *Applications of artificial intelligence in finance and economics*. Emerald Group Publishing Limited, 2004.

B Zhu, P Wang, J Chevallier, and Y Wei. Carbon price analysis using empirical mode decomposition. *Computational Economics*, 45(2):195–206, 2015.

D Wang, C Yue, S Wei, and J Lv. Performance analysis of four decomposition-ensemble models for one-day-ahead agricultural commodity futures price forecasting. *Algorithms*, 10(3):108, 2017.

H Miao, S Ramchander, T Wang, and D Yang. Influential factors in crude oil price forecasting. *Energy Economics*, 68:77–88, 2017.

S Wu and R Chen. Threshold variable determination and threshold variable driven switching autoregressive models. *Statistica Sinica*, 17(1):241–S38, 2007.

X Zhang, K K Lai, and S Y Wang. A new approach for crude oil price analysis based on empirical mode decomposition. *Energy economics*, 30(3):905–918, 2008.

Y Jian-Hui and D Wei. Prediction of gold price based on wt-svr and emd-svr model. In *2012 Eighth International Conference on Computational Intelligence and Security*, pages 415–419. IEEE, 2012.

C S Lin, S H Chiu, and T Y Lin. Empirical mode decomposition–based least squares support vector regression for foreign exchange rate forecasting. *Economic Modelling*, 29(6):2583–2590, 2012.

B Premanode and C Toumazou. Improving prediction of exchange rates using differential emd. *Expert systems with applications*, 40(1):377–384, 2013.

N An, W Zhao, J Wang, D Shang, and E Zhao. Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting. *Energy*, 49: 279–288, 2013.

S Abadan and A Shabri. Hybrid empirical mode decomposition-arima for forecasting price of rice. *Applied Mathematical Sciences*, 8(63):3133–3143, 2014.

V Plakandaras, T Papadimitriou, and P Gogas. Forecasting daily and monthly exchange rates with machine learning techniques. *Journal of Forecasting*, 34(7):560–573, 2015.

Q Hua and T Jiang. The prediction for london gold price: improved empirical mode decomposition. *Applied Economics Letters*, 22(17):1404–1408, 2015.

T Xiong, C Li, Y Bao, Z Hu, and L Zhang. A combination method for interval forecasting of agricultural commodity futures prices. *Knowledge-Based Systems*, 77:92–102, 2015.

H Chen, H Liao, B J Tang, and Y M Wei. Impacts of opec's political risk on the international crude oil prices: An empirical analysis based on the svar models. *Energy Economics*, 57:42–49, 2016.

L Crosato, L Grossi, and F Nan. Forecasting the volatility of electricity prices by robust estimation: An application to the italian market. In *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, pages 279–283. Springer, 2018.

J Meng, H Nie, B Mo, and Y Jiang. Risk spillover effects from global crude oil market to china's commodity sectors. *Energy*, 202:117208, 2020.

V Korotin, M Dolgonosov, V Popov, O Korotina, and I Korolkova. The ukrainian crisis, economic sanctions, oil shock and commodity currency: Analysis based on emd approach. *Research in International Business and Finance*, 48:156–168, 2019.

Y Zhu, D Xu, J Cheng, and S H Ali. Estimating the impact of china's export policy on tin prices: a mode decomposition counterfactual analysis method. *Resources policy*, 59:250–264, 2018.

B Zhu, L Huang, L Yuan, S Ye, and P Wang. Exploring the risk spillover effects between carbon market and electricity market: A bidimensional empirical mode decomposition based conditional value at risk approach. *International Review of Economics & Finance*, 67:163–175, 2020.

L Y Wei and C H Cheng. A hybrid recurrent neural networks model based on synthesis features to forecast the taiwan stock market. *Int. J. Innov. Comput. Inf. Control*, 8(8):5559–5571, 2012.

L B Ngo. Application of the empirical mode decomposition on the characterization and forecasting of the arrival data of an enterprise cluster. 2011.

E Antwi, E N Gyamfi, K Kyei, R Gill, and A M Adam. Determinants of commodity futures prices: Decomposition approach. *Mathematical Problems in Engineering*, 2021, 2021.

S Lahmiri. Intraday stock price forecasting based on variational mode decomposition. *Journal of Computational Science*, 12:23–27, 2016.

C Aneesh, S Kumar, P M Hisham, and K P Soman. Performance comparison of variational mode decomposition over empirical wavelet transform for the classification of power quality disturbances using support vector machine. *Procedia Computer Science*, 46:372–380, 2015.

Y Seo, S Kim, and V P Singh. Machine learning models coupled with variational mode decomposition: a new approach for modeling daily rainfall-runoff. *Atmosphere*, 9(7):251, 2018.

H Niu, K Xu, and W Wang. A hybrid stock price index forecasting model based on variational mode decomposition and lstm network. *Applied Intelligence*, 50(12):4296–4309, 2020.

E Jianwei, J Ye, and H Jin. A novel hybrid model on the prediction of time series and its application for the gold price analysis and forecasting. *Physica A: Statistical Mechanics and its Applications*, 527: 121454, 2019.

K He, G K F Tso, Y Zou, and J Liu. Crude oil risk forecasting: New evidence from multiscale analysis approach. *Energy economics*, 76:574–583, 2018.

Y Li, X Chen, J Yu, and X Yang. A fusion frequency feature extraction method for underwater acoustic signal based on variational mode decomposition, duffing chaotic oscillator and a kind of permutation entropy. *Electronics*, 8(1):61, 2019.

E S Fianu. Analyzing and forecasting multi-commodity prices using variants of mode decomposition-based extreme learning machine hybridization approach. *Available at SSRN 4046171*, 2022.

F Daoui, S Mefteh-Wali, and S B Jabeur. Multiresolutional statistical machine learning for testing interdependence of power markets: A variational mode decomposition-based approach. *Expert Systems with Applications*, page 118161, 2022.

W Sun and Z Xu. Carbon price prediction model based on adaptive variational mode decomposition and optimized extreme learning machine. *International Journal of Environmental Science and Technology*, pages 1–21, 2022.

R Li, Y Hu, J Heng, and X Chen. A novel multiscale forecasting model for crude oil price time series. *Technological Forecasting and Social Change*, 173:121181, 2021.

W Liu, C Wang, Y Li, Y Liu, and K Huang. Ensemble forecasting for product futures prices using variational mode decomposition and artificial neural networks. *Chaos, Solitons & Fractals*, 146:110822, 2021.

Q Zhang, J Wu, Y Ma, G Li, J Ma, and C Wang. Short-term load forecasting method with variational mode decomposition and stacking model fusion. *Sustainable Energy, Grids and Networks*, 30:100622, 2022.

Y Lin, Q Lu, B Tan, and Y Yu. Forecasting energy prices using a novel hybrid model with variational mode decomposition. *Energy*, 246:123366, 2022.

P. Levy. Propriétés asymptotiques des sommes de variables indépendantes ou enchainées. *Journal des mathématiques pures et appliquées. Series*, 9(14):4, 1935.

J. Ville. Etude critique de la notion de collectif. *Bull. Amer. Math. Soc*, 45(11):824, 1939.

Joseph L Doob. What is a martingale? *The American Mathematical Monthly*, 78(5):451–463, 1971.

JY Campbell. Lo, aw and mackinlay, ac (1997) the econometrics of financial markets. *Princeton: Princeton University Press. Number of ISO*, 14001:242–253, 2004.

S L Jones and J M Netter. Efficient capital markets. *The Concise Encyclopedia of Economic*, 15, 2008.

E S Gardner Jr. Exponential smoothing: The state of the art—part ii. *International journal of forecasting*, 22(4):637–666, 2006.

A C Harvey. Forecasting, structural time series models and the kalman filter. 1990.

R Hyndman, A B Koehler, J K Ord, and R D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

G E P Box, M Jenkins Gwilym, C Reinsel Gregory, and M Ljung Greta. Time series analysis: forecasting and control. *San Francisco: Holden Bay*, 1976.

P H Franses. *Expert adjustments of model forecasts: Theory, practice and strategies for improvement*. Cambridge University Press, 2014.

HM Markowitz. Portfolio selection (1952) 7 j.

W F Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *The journal of finance*, 19(3):425–442, 1964.

J W Taylor, P E McSharry, and R Buizza. Wind power density forecasting using ensemble predictions and time series models. *IEEE Transactions on Energy Conversion*, 24(3):775–782, 2009.

T Gneiting. Making and evaluating point forecasts. *Journal of the American Statistical Association*, 106 (494):746–762, 2011.

R Engle. Risk and volatility: Econometric models and financial practice. *American economic review*, 94 (3):405–420, 2004.

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

M P Wand. Error analysis for general multtvariate kernel estimators. *Journal of Nonparametric Statistics*, 2(1):1–15, 1992.

A Timmermann. Moments of markov switching models. *Journal of econometrics*, 96(1):75–111, 2000.

G Zhang, B E Patuwo, and M Y Hu. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, 14(1):35–62, 1998.

K Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

J Park and I W Sandberg. Universal approximation using radial-basis-function networks. *Neural computation*, 3(2):246–257, 1991.

Anton M and H G Zimmermann. Recurrent neural networks are universal approximators. In *International Conference on Artificial Neural Networks*, pages 632–640. Springer, 2006.

L R Glosten, R Jagannathan, and D E Runkle. On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, 48(5):1779–1801, 1993.

H Tong. *Non-linear time series: a dynamical system approach*. Oxford university press, 1990.

C WS Chen, T C Chiang, and M KP So. Asymmetrical reaction to us stock-return news: evidence from major stock markets based on a double-threshold model. *Journal of economics and business*, 55(5-6): 487–502, 2003.

C W S Chen and M K P So. On a threshold heteroscedastic model. *International Journal of Forecasting*, 22(1):73–89, 2006.

R Gerlach, C W S Chen, D S Y Lin, and M H Huang. Asymmetric responses of international stock markets to trading volume. *Physica A: Statistical Mechanics and its applications*, 360(2):422–444, 2006.

R Chen. Threshold variable selection in open-loop threshold autoregressive models. *Journal of time series analysis*, 16(5):461–481, 1995.

J G De Gooijer. Testing non-linearities in world stock market prices. *Economics Letters*, 31(1):31–35, 1989.

S Ling. On the probabilistic properties of a double threshold arma conditional heteroskedastic model. *Journal of Applied probability*, 36(3):688–705, 1999.

A Amendola, M Niglio, and C Vitale. The moments of setarma models. *Statistics & probability letters*, 76(6):625–633, 2006.

K S Chan and H Tong. On estimating thresholds in autoregressive models. *Journal of time series analysis*, 7(3):179–190, 1986.

B Efron and R Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75, 1986.

E Carlstein. Resampling techniques for stationary time-series: some recent developments. Technical report, North Carolina State University. Dept. of Statistics, 1990.

H. R Kunsch. The jackknife and the bootstrap for general stationary observations. *The annals of Statistics*, pages 1217–1241, 1989.

D N Politis and J P Romano. The stationary bootstrap. *Journal of the American Statistical association*, 89(428):1303–1313, 1994.

P Hlmann. Sieve bootstrap for time series. *Bernoulli*, pages 123–148, 1997.

DA Friedman. Bootstrap regression models. *Ann Stat*, 9:1218–1228, 1981.

A Zagda. Prediction intervals for stationary time series using the sieve bootstrap method. *Demonstratio Mathematica*, 34(2):257–270, 2001.

M A Andre'es, D Pena, and J Romo. Forecasting time series with sieve bootstrap. *Journal of Statistical Planning and Inference*, 100(1):1–11, 2002.

C Cordeiro and M Neves. The bootstrap methodology in time series forecasting. *Proceedings of Comp-Stat2006"(J. Black and A. White, Eds.), Springer Verlag*, pages 1067–1073, 2006.

C Cordeiro and M M Neves. Forecasting time series with boot. expos procedure. *REVSTAT-Statistical Journal*, 7(2):135–149, 2009.

C Cordeiro and M M Neves. Boot. expos in nngc competition. In *The 2010 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2010.

P J Bickel and K A Doksum. An analysis of transformations revisited. *Journal of the american statistical association*, 76(374):296–311, 1981.

J E Angus. Asymptotic theory for bootstrapping the extremes. *Communications in Statistics-Theory and Methods*, 22(1):15–30, 1992.

D Zelterman. A semiparametric bootstrap technique for simulating extreme order statistics. *Journal of the American Statistical Association*, 88(422):477–485, 1993.

F J Fabozzi, K C Ma, and B J Oliphant. Sin stock returns. *The Journal of Portfolio Management*, 35(1): 82–94, 2008.

G E P Box and D R Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

V M Guerrero. Time-series analysis supported by power transformations. *Journal of forecasting*, 12(1): 37–48, 1993.

C W J Granger and P Newbold. Forecasting transformed series. *Journal of the Royal Statistical Society: Series B (Methodological)*, 38(2):189–203, 1976.

J M G Taylor. The retransformed mean after a fitted power transformation. *Journal of the American Statistical Association*, 81(393):114–118, 1986.

A Pankratz and U Dudley. Forecasts of power-transformed series. *Journal of Forecasting*, 6(4):239–248, 1987.

C H D Buys-Ballot. Leo claemert periodiques de temperature. *Kemint et Fills, Utrecht*, 1847.

J H Poynting. A comparison of the fluctuations in the price of wheat and in the cotton and silk imports into great britain. *Journal of the Statistical Society of London*, 47(1):34–74, 1884.

V O Anderson. Nochmals über" the elimination of spurious correlation due to position in time or space. *Biometrika*, pages 269–279, 1914.

R H Hooker. The suspension of the berlin produce exchange and its effect upon corn prices. *Journal of the Royal Statistical Society*, 64(4):574–613, 1901.

J Spencer. On the graduation of the rates of sickness and mortality presented by the experience of the manchester unity of oddfellows during the period 1893–97. *Journal of the Institute of Actuaries*, 38(4): 334–343, 1904.

M T Copeland. Statistical indices of business conditions. *The Quarterly Journal of Economics*, pages 522–562, 1915.

F R Macaulay. Appendices to" the smoothing of time series". In *The Smoothing of Time Series*, pages 118–169. NBER, 1931.

G G S Pegram, M C Peel, and T A McMahon. Empirical mode decomposition using rational splines: an application to rainfall time series. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 464(2094):1483–1501, 2008.

Z Guo, W Zhao, H Lu, and J Wang. Multi-step forecasting for wind speed using a modified emd-based artificial neural network model. *Renewable energy*, 37(1), 2012.

Co K Hamza E. Improving artificial neural networks' performance in seasonal time series forecasting. *Information Sciences*, 178(23):4550–4559, 2008.

I Khandelwal, R Adhikari, and G Verma. Time series forecasting using hybrid arima and ann models based on dwt decomposition. *Procedia Computer Science*, 48:173–179, 2015.

G P Zhang and M Qi. Neural network forecasting for seasonal and trend time series. *European journal of operational research*, 160(2):501–514, 2005.

S S Reddy, C M Jung, and K J Seog. Day-ahead electricity price forecasting using back propagation neural networks and weighted least square technique. *Frontiers in Energy*, 10(1):105–113, 2016.

F S Lasheras, Ana S De C J, Francisco J and, A Krzemie, and P R Fern. Forecasting the comex copper spot price by means of neural networks and arima models. *Resources Policy*, 45:37–43, 2015.

J Z Wang, J J Wang, Z G Zhang, and S P Guo. Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, 38(11):14346–14355, 2011.

T Jo. Vtg schemes for using back propagation for multivariate time series prediction. *Applied Soft Computing*, 13(5):2692–2702, 2013.

G E P Box, G M Jenkins, G C Reinsel, and G M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

G P Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

D M Keenan. A tukey nonadditivity-type test for time series nonlinearity. *Biometrika*, 72(1):39–44, 1985.

J D Cryer and K k Chan. *Time series analysis: with applications in R*, volume 2. Springer, 2008.

R S Tsay. Nonlinearity tests for time series. *Biometrika*, 73(2):461–466, 1986.

M Q Raza, M Nadarajah, D Q Hung, and Z Baharudin. An intelligent hybrid short-term load forecasting model for smart power grids. *Sustainable Cities and Society*, 31:264–275, 2017.

I Novianty, R G Baskoro, M I Nurulhaq, and M A Nanda. Empirical mode decomposition of near-infrared spectroscopy signals for predicting oil content in palm fruits. *Information Processing in Agriculture*, 2022.

M Teng, S Li, J Xing, G Song, J Yang, J Dong, X Zeng, and Y Qin. 24-hour prediction of pm2. 5 concentrations by combining empirical mode decomposition and bidirectional long short-term memory neural network. *Science of The Total Environment*, page 153276, 2022.

S L Lin. Application of empirical mode decomposition to improve deep learning for us gdp data forecasting. *Heliyon*, page e08748, 2022.

S Jiang, Y Li, Q Lu, S Wang, and Y Wei. Volatility communicator or receiver? investigating volatility spillover mechanisms among bitcoin and other financial markets. *Research in International Business and Finance*, 59:101543, 2022.

J Wang, Z Wang, X Li, and H Zhou. Artificial bee colony-based combination approach to forecasting agricultural commodity prices. *International Journal of Forecasting*, 38(1):21–34, 2022.

Y J Zhang and J L Zhang. Volatility forecasting of crude oil market: A new hybrid method. *Journal of Forecasting*, 37(8):781–789, 2018.

N Kourentzes, D Barrow, and F Petropoulos. Another look at forecast selection and combination: Evidence from forecast pooling. *International Journal of Production Economics*, 209:226–235, 2019.

J M Bates and C W J Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969.

S Ismael. Adaptive combination of forecasts with application to wind energy. *International Journal of Forecasting*, 24(4):679–693, 2008.

J Nowotarski, B Liu, R Weron, and T Hong. Improving short term load forecast accuracy via combining sister forecasts. *Energy*, 98:40–49, 2016.

F X Diebold and R S Mariano. Com paring predictive accu racy. *Journal of Business and Economic Statistics*, 13(3):253–263, 1995.

H Chen, Q Wan, and Y Wang. Refined diebold-mariano test methods for the evaluation of wind power forecasting models. *Energies*, 7(7):4185–4198, 2014.

B White. The importance of climate variability and seasonal forecasting to the australian economy. In *Applications of seasonal climate forecasting in agricultural and natural ecosystems*, pages 1–22. Springer, 2000.

S Nazlioglu and U Soytas. Oil price, agricultural commodity prices, and the dollar: A panel cointegration and causality analysis. *Energy Economics*, 34(4):1098–1104, 2012.

M Huchet-Bourdon. Agricultural commodity price volatility: An overview. 2011.

G C Rausser and H De G. Us policy contributions to agricultural commodity price fluctuations, 2006–12. *Food Price Policy in an Era of Market Instability*, page 433, 2014.

Y J Zhang, T Yao, L Y He, and R Ripple. Volatility forecasting of crude oil market: Can the regime switching garch model beat the single-regime garch models? *International Review of Economics & Finance*, 59:302–317, 2019.

V E Ball. Output, input, and productivity measurement in us agriculture 1948–79. *American Journal of Agricultural Economics*, 67(3):475–486, 1985.

H Mohammadi and L Su. International evidence on crude oil price dynamics: Applications of arima-garch models. *Energy Economics*, 32(5):1001–1008, 2010.

W F Sun, Y H Peng, Y Yang, et al. Method for improving frequency resolution of empirical mode decomposition. *Comput. Eng. Appl*, 46:129–133, 2010.

R S Tsay. Conditional heteroscedastic time series models. *Journal of the American Statistical association*, 82(398):590–604, 1987.

Y S Abu-Mostafa and A F Atiya. Introduction to financial forecasting. *Applied intelligence*, 6(3):205–213, 1996.

A M Adam, K Kyei, S Moyo, R Gill, and E N Gyamfi. Multifrequency network for sadc exchange rate markets using eemd-based dcca. *Journal of Economics and Finance*, 46(1):145–166, 2022.

R Vautard, P Yiou, and M Ghil. Singular-spectrum analysis: A toolkit for short, noisy chaotic signals. *Physica D: Nonlinear Phenomena*, 58(1-4):95–126, 1992.

G Rilling and P Flandrin. One or two frequencies? the empirical mode decomposition answers. *IEEE transactions on signal processing*, 56(1):85–95, 2007.

A Roy and J F Doherty. Empirical mode decomposition frequency resolution improvement using the pre-emphasis and de-emphasis method. In *2008 42nd Annual Conference on Information Sciences and Systems*, pages 453–457. IEEE, 2008.

Appendix

```
R-Codes for Data Validation
setwd("/Users/emanuel/Desktop/VMDs/")

################## PLOTTING TIME SERIES ###############
library(xts)
ts.corn <- xts(CP$Value, order.by = as.POSIXct(CP$Date, "UTC"))
plot(xts(COPVMD$Value, order.by = as.POSIXct(COPVMD$Date, "UCT")), main = "Time
series", ylab = "Corn future prices", xlab = "Time/Daily")
plot(xts(COPVMD$Value, order.by = as.POSIXct(COPVMD$Date, "UCT")), main = "Time
series",  ylab = "Crude oil future prices", xlab = "Time/Daily")
plot(xts(COPVMD$Value, order.by = as.POSIXct(COPVMD$Date, "UCT")), main = "Time
series", ylab = "gold future prices", xlab = "Time/Daily")

######### Test for existance of nonlinearity ###################

#Keenan's test: (test for nonlinearity against the null hypothesis
#that the time series follows some AR process)
library(TSA)
Keenan.test(CP$Value)
#Time series does not follow an AR process

#Tsay test for quadratic nonlinearity in a time series(
#The null hypothesis is that the true model is an AR process)
Tsay.test(CP$Value)
#The true model is not an AR

library(nonlinearTseries)
nonlinearityTest(CP$Value, verbose = TRUE)
nonlinearityTest(CP$Value)
############################### Stationarity test#############
library(tseries)
adf.test(CP$Value)
adf.test(CP$Value)
## KPSS test
kpss.test(COPVMD$Value, null = "Trend")

############ load library ###############
library(tidyverse)
library(magrittr)
library(readxl)
summary(COPVMD)
library(Hmisc)
library(vmd)
library(EMD)
library(hht)
library(pvclust)
library(Rlibeemd)
```

Figure 7.1:

```
###############Decomposition of corn with EMD. ######################
Price<- emd(COPVMD$Value, tt=NULL,
tol=sd(COPVMD$Value)*0.1^2,boundary="periodic")
write.table(Price,file="emdCorn.v3.csv")
#################  plot subplot. ################
par(mfcol=c(5, 2), mar=c(2,4,1,1), oma=c(2,2,2,2))
#mtext("Decomp", side=3,line=0)
#######Signal is the plot of Observation index against observed price ###############
plot(ts(COPVMD$Value), xlab = "Time/daily",ylab="Signal", type="l", main ="",)
#############################################################################
############Observation index against IMFs ########################
plot(Price$imf[,1], type="l", xlab=" ",ylab="IMF 1",col="blue")
plot(Price$imf[,2], type="l", xlab="",ylab="IMF 2",col="blue")
plot(Price$imf[,3], type="l", xlab="",ylab="IMF 3",col="blue")
plot(Price$imf[,4], type="l", xlab="",ylab="IMF 4",col="blue")
plot(Price$imf[,5], type="l", xlab="TIME/Daily",ylab="IMF 5",col="blue")
plot(Price$imf[,6], type="l", xlab="",ylab="IMF 6",col="blue")
plot(Price$imf[,7], type="l", xlab="",ylab="IMF 7",col="blue")
plot(Price$imf[,8], type="l", xlab="",ylab="IMF 8",col="blue")
plot(Price$residue, type="l", xlab="TIME/Daily",ylab="Residual",col="red")

#################.  FITTING HCLUSTER. ####################
fit <- pvclust(Corn1[,1:9], method.hclust="ward", method.dist="euclidean")
fit
plot(fit) # dendogram with p values
box()
ask.bak <- par()$ask
# add rectangles around groups highly supported by the data
pvrect(fit, alpha = 0.95)
##############CORRELATION COEFFICIENTS ######################
cor.test(COPVMD$Value, trend, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, trend, method = "kendall", alternative = "greater")
cor.test(COPVMD$Value, trend, method = "spearman", alternative = "greater",exact = F)
################# Variance of IMFs and residue ############
var(Price$imf[,1])
var(Price$imf[,2])
var(Price$imf[,3])
var(Price$imf[,4])
var(Price$imf[,5])
var(Price$imf[,6])
var(Price$imf[,7])
var(Price$residue)
var(COPVMD$Value)
#################### Extrema ##############################
pp<-extrema(Price$residue)
pp
```

Figure 7.2:

```
lowfrequency<-(Price$imf5+Price$imf[,6])
highfrequency <-
(Price$imf[,1]+Price$imf[,2]+Price$imf[,3]+Price$imf[,4])+Price$imf[,5]+Price$imf[,7]+Pric
e$imf[,8]
plot.ts(highfrequency, xlab="Time/monthly",ylab=" ",col='blue')
lines(lowfrequency,col='red')
line(lowfrequency,add=TRUE)
lines(Corn_Price)
pp<-extrema(highfrequency)
pp
var(highfrequency)
var(highfrequency)
trend<-(Price$residue)
var(Price$imf[,6])
################  Decomposition of crude oil with EMD ##########
Price<-emd(COPVMD$Value, tt=NULL,
tol=sd(COPVMD$Value)*0.1^2,boundary="periodic")
names(Price)
emd(COPVMD$Value)
write.table(Price,file="emdCorn.v3.csv")
#plot(Price,type="l", xlab="Time/Daily")
################### plot subplot ##################
par(mfcol=c(5, 2), mar=c(2,4,1,1), oma=c(2,2,2,2))
#mtext("Decomp", side=3,line=0)
###########. Signal is the plot of Observation index against observed price #####
plot(ts(COPVMD$Value), xlab = "Time/daily",ylab="Signal", type="l", main ="",)
##############Observation index against IMFs ########################
plot(Price$imf[,1], type="l", xlab=" ",ylab="IMF 1",col="blue")
plot(Price$imf[,2], type="l", xlab="",ylab="IMF 2",col="blue")
plot(Price$imf[,3], type="l", xlab="",ylab="IMF 3",col="blue")
plot(Price$imf[,4], type="l", xlab="",ylab="IMF 4",col="blue")
plot(Price$imf[,5], type="l", xlab="TIME/Daily",ylab="IMF 5",col="blue")
plot(Price$imf[,6], type="l", xlab="",ylab="IMF 6",col="blue")
plot(Price$imf[,7], type="l", xlab="",ylab="IMF 7",col="blue")
plot(Price$imf[,8], type="l", xlab="",ylab="IMF 8",col="blue")
plot(Price$residue, type="l", xlab="TIME/Daily",ylab="Residual",col="red")

########## Extrema ####################
pp<-extrema(Price$imf[,1])
pp

bb<-extrema(Price$residue)
bb
######################### FITTING HCLUSTER ################
fit <- pvclust(Corn1[,1:9], method.hclust="ward", method.dist="euclidean")
fit
plot(fit)
```

Figure 7.3:

```
# dendogram with p values
box()
ask.bak <- par()$ask
# add rectangles around groups highly supported by the data
pvrect(fit, alpha = 0.95)
 ###############CORRELATION COEFFICIENTS ######################
cor.test(COPVMD$Value, trend, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, trend, method = "kendall", alternative = "greater")
cor.test(COPVMD$Value, trend, method = "spearman", alternative = "greater",exact = F)
################## Variance of IMFs and residue #############
var(Price$imf[,1])
var(Price$imf[,2])
var(Price$imf[,3])
var(Price$imf[,4])
var(Price$imf[,5])
var(Price$imf[,6])
var(Price$imf[,7])
var(Price$imf[,8)
var(Price$residue)
var(COPVMD$Value)
############### Decomposition of gold with EMD ####################
Price<- emd(COPVMD$Value, tt=NULL,
tol=sd(COPVMD$Value)*0.1^2,boundary="periodic")
write.table(Price,file="emdCorn.v3.csv")
##################plot subplot#################
par(mfcol=c(5, 2), mar=c(2,4,1,1), oma=c(2,2,2,2))
#mtext("Decomp", side=3,line=0)
#############################################################################
#Signal is the plot of Observation index against observed price
#############################################################################
plot(ts(COPVMD$Value), xlab = "Time/daily",ylab="Signal", type="l", main ="",)
##############Observation index against IMFs ############################
plot(Price$imf[,1], type="l", xlab=" ",ylab="IMF 1",col="blue")
plot(Price$imf[,2], type="l", xlab="",ylab="IMF 2",col="blue")
plot(Price$imf[,3], type="l", xlab="",ylab="IMF 3",col="blue")
plot(Price$imf[,4], type="l", xlab="",ylab="IMF 4",col="blue")
plot(Price$imf[,5], type="l", xlab="TIME/Daily",ylab="IMF 5",col="blue")
plot(Price$imf[,6], type="l", xlab="",ylab="IMF 6",col="blue")
plot(Price$imf[,7], type="l", xlab="",ylab="IMF 7",col="blue")
plot(Price$imf[,8], type="l", xlab="",ylab="IMF 8",col="blue")
plot(Price$residue, type="l", xlab="TIME/Daily",ylab="Residual",col="red")
################## Variance of IMFs and residue #############
var(Price$imf[,1])
var(Price$imf[,2])
var(Price$imf[,3])
var(Price$imf[,4])
var(Price$imf[,5])
```

Figure 7.4:

```
var(Price$imf[,6])
var(Price$imf[,7])
var(Price$imf[,8)
var(Price$residue)
var(COPVMD$Value)
############## CORRELATION COEFFICIENTS ######################
cor.test(COPVMD$Value, Price$residue, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, Price$residue,method =  "kendall", alternative = "greater")
cor.test(COPVMD$Value, Price$residue, method = "spearman", alternative =
"greater",exact = F)

cor.test(COPVMD$Value, trend, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, trend, method = "kendall", alternative = "greater")
cor.test(COPVMD$Value, trend, method = "spearman", alternative = "greater",exact = F)
########## Extrema ####################
pp<-extrema(Price$imf[,1])
pp
bb<-extrema(Price$residue)
bb
######################### FITTING HCLUSTER #################
fit <- pvclust(Corn1[,1:9], method.hclust="ward", method.dist="euclidean")
fit
plot(fit)
# dendogram with p values
box()
ask.bak <- par()$ask
# add rectangles around groups highly supported by the data
pvrect(fit, alpha = 0.95)
########################### EMD THREE COMPONENTS ############
GV <- vmd(COPVMD$Value, alpha = 2000,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
GV <- as.data.frame(GV)
HGV <- ts(GV$M8+GV$M7+GV$M6+GV$M3+GV$M4+GV$M5)
LGV <- ts(GV$M2)
ResGV <- ts(GV$M1)
OGV <- ts(GV$MAgg)
HGV_N <- ts(normalizeData(HGV,type="0_1"))
LGV_N <-  ts(normalizeData(LGV,type="0_1"))
ResGV_N <-  ts(normalizeData(ResGV,type="0_1"))
OGV_N <-  ts(normalizeData(OGV,type="0_1"))
win.graph()
plot(HGV_N,col="blue",main="The three components of gold",xlab="Time/daily",ylab="
")
lines(LGV_N,col="green")
```

Figure 7.5:

```
################## VMD Decomposition ##################
library(vmd)
library(RSNNS)
library(ggplot2)
################### Decomposition of corn with VMD ####################
COPVMD$Value <- as.numeric(as.character(COPVMD$Value))
COPVMD$Date <- as.Date(as.character(COPVMD$Date))
vmdCOP <- vmd(COPVMD$Value, alpha = 2000,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
#To Data Frame
df <- as.data.frame(vmdCOP)
#Plot Results
plot(vmdCOP, facet="bymode",scales="free")
plot(vmdCOP, facet="bymode", scale="free")
fit <- pvclust(df[,1:11], method.hclust="ward", method.dist="euclidean")fit
############ dendogram with p values ##################
plot(fit)
box()
ask.bak <- par()$ask
# add rectangles around groups highly supported by the data
pvrect(fit, alpha = 0.95)
plot.ts(COPVMD [,2], xlab="Time/daily", ylab="corn futures prices", main="Time series")
high<-(df$M3+df$M4+df$M5+df$M6+df$M7+df$M8)
low<-(df$M2)
trend<-(df$M1)
plot.ts(low, xlab="Time/monthly",ylab=" ",col='blue')
lines(high,col="red", lty=2)
lines(trend,col="yellow")
legend("topright",col=c("red","blue", "yellow"), lty=1:2,lwd=2,legend=c("High frequency",
"Low frequency","Trend"))
######### variance of the mode ##########################
var(df$M1)
var(df$M2)
var(df$M3)
var(df$M4)
var(df$M5)
var(df$M6)
var(df$M7)
var(df$M8)
var(df$MAgg)
h<-extrema(df$M1)
h
ht<-extrema(high)
ht
plot(xts(COPVMD$Value, order.by = as.POSIXct(COPVMD$Date, "UCT")), main = "Time
series",
    ylab = "Corn future prices", xlab = "Time/Daily")
```

Figure 7.6:

```
OG<-COPVMD$Value
cor.test(OG,high , method = "pearson", alternative = "greater")
cor.test(OG, high, method = "kendall", alternative = "greater")
cor.test(OG, high, method = "spearman", alternative = "greater",exact = F)

cor.test(COPVMD$Value, df$M1, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, df$M1, method = "kendall", alternative = "greater")
cor.test(COPVMD$Value, df$M1, method = "spearman", alternative = "greater",exact = F)

COPVMD$Value <- as.numeric(as.character(COPVMD$Value))
COPVMD$Date <- as.Date(as.character(COPVMD$Date))

################## Decomposition of crude oil with VMD #############
vmdCOP <- vmd(COPVMD$Value, alpha = 2000,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
#To Data Frame
df <- as.data.frame(vmdCOP)
#Plot Results
plot(vmdCOP, facet="bymode",scales="free")
plot(vmdCOP, facet="bymode", scale="free")
fit <- pvclust(df[,1:11], method.hclust="ward", method.dist="euclidean")fit
########### dendogram with p values ##################
plot(fit)
box()
ask.bak <- par()$ask
# add rectangles around groups highly supported by the data
pvrect(fit, alpha = 0.95)
high<-(df$M3+df$M4+df$M5+df$M6+df$M7+df$M8)
low<-(df$M2)
trend<-(df$M1)
######################## Plotting of the three components ##############
high<-(df$M4+df$M5+df$M6+df$M7+df$M8)
low<-(df$M2+df$M3)
trend<-(df$M1)
plot.ts(low, xlab="Time/monthly",ylab=" ",col='blue')
lines(high,col="red", lty=2)
lines(trend,col="yellow")
legend("topright",col=c("red","blue", "yellow"), lty=1:2,lwd=2,legend=c("High frequency",
"Low frequency","Trend"))
######### variance of the mode ########################
var(df$M1)
var(df$M2)
var(df$M3)
var(df$M4)
var(df$M5)
var(df$M6)
```

Figure 7.7:

```
var(df$M7)
var(df$M8)
var(df$MAgg)
######################################## Extrema ##########################
h<-extrema(df$M1)
h
ht<-extrema(high)
ht
################ Correlation ##################
OG<-COPVMD$Value
cor.test(OG,high , method = "pearson", alternative = "greater")
cor.test(OG, high, method = "kendall", alternative = "greater")
cor.test(OG, high, method = "spearman", alternative = "greater",exact = F)

cor.test(COPVMD$Value, df$M1, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, df$M1, method = "kendall", alternative = "greater")
cor.test(COPVMD$Value, df$M1, method = "spearman", alternative = "greater",exact = F)
################## Decomposition of gold with VMD #############
vmdCOP <- vmd(COPVMD$Value, alpha = 2000,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
#To Data Frame
df <- as.data.frame(vmdCOP)
#Plot Results
plot(vmdCOP, facet="bymode",scales="free")
plot(vmdCOP, facet="bymode", scale="free")
fit <- pvclust(df[,1:11], method.hclust="ward", method.dist="euclidean")fit
########### dendogram with p values #################
plot(fit)
box()
ask.bak <- par()$ask
# add rectangles around groups highly supported by the data
pvrect(fit, alpha = 0.95)
high<-(df$M3+df$M4+df$M5+df$M6+df$M7+df$M8)
low<-(df$M2)
trend<-(df$M1)
######################## Plotting of the three components ##############
high<-(df$M4+df$M5+df$M6+df$M7+df$M8)
low<-(df$M2+df$M3)
trend<-(df$M1)
plot.ts(low, xlab="Time/monthly",ylab=" ",col='blue')
lines(high,col="red", lty=2)
lines(trend,col="yellow")
legend("topright",col=c("red","blue", "yellow"), lty=1:2,lwd=2,legend=c("High frequency",
"Low frequency","Trend"))
```

Figure 7.8:

```
######### variance of the mode ########################
var(df$M1)
var(df$M2)
var(df$M3)
var(df$M4)
var(df$M5)
var(df$M6)
var(df$M7)
var(df$M8)
var(df$MAgg)
######################################## Extrema ##########################
h<-extrema(df$M1)
h
ht<-extrema(high)
ht
################ Correlation ###################
OG<-COPVMD$Value
cor.test(OG,high , method = "pearson", alternative = "greater")
cor.test(OG, high, method = "kendall", alternative = "greater")
cor.test(OG, high, method = "spearman", alternative = "greater",exact = F)

cor.test(COPVMD$Value, df$M1, method = "pearson", alternative = "greater")
cor.test(COPVMD$Value, df$M1, method = "kendall", alternative = "greater")
cor.test(COPVMD$Value, df$M1, method = "spearman", alternative = "greater",exact = F)

################### Forecasting corn futures prices  #################
library(DescTools)
library(forecast)
library(EMD)
library(vmd)
library(neuralnet)
library(MASS)
library(tseries)
library(TSA)
library(readxl)
set.seed(2015)
#########################################################################
##1_Corn_EMD_BPNN
CP <- read_excel("/Users/emanuel/Desktop/NEWDATA_COMM/COMMODITY
DATA.xlsx",  sheet = "CORN", col_types = c("date", "numeric"))

#CP <- (CP$Value)
Corn_Price<- emd(CP$Value, tt=NULL, tol=sd(CP$Value)*0.1^2,boundary="periodic")
write.table(Corn_Price,file="emdCP.csv")
#plot(Corn_Price,type="l", xlab="Time/Daily")
plot.ts(Corn_Price$imf[,1])
```

Figure 7.9:

```
###################### Datapartition ######################################
index <- sample(nrow(CP),round(0.8*nrow(CP)))
train <- CP[index,]
nrow(train)
test <- CP[-index,]
nrow(test)
train_D <- train$Value
test_D <- test$Value

normalizetest <- function(test){return((test-min(test))/(max(test)-min(test)))}
test_norm <- normalize(test)
####################Decomposition#############################################
df_IMFs<- emd(train_D, tt=NULL, tol=sd(train_D)*0.1^2,boundary="periodic")
adf.test(df_IMFs$imf[,1]) #Stratinonary
adf.test(df_IMFs$imf[,2]) #Stratinonary
adf.test(df_IMFs$imf[,3]) #Stratinonary
adf.test(df_IMFs$imf[,4]) #Stratinonary
adf.test(df_IMFs$imf[,5]) #Stratinonary
adf.test(df_IMFs$imf[,6]) #Stratinonary
adf.test(df_IMFs$imf[,7]) #Stratinonary
adf.test(df_IMFs$imf[,8]) #Not stratinonary
adf.test(df_IMFs$imf[,9]) # Not stratinonary
########### Normalisation of trained data ##########
IMF1 <- as.data.frame(df_IMFs$imf[,1]);Res <- as.data.frame(df_IMFs$residue)
ts.plot(IMF1)
IMF1_Norm <- ts((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))
################### Time series, Fitting and Forecasting ###############
fit_1 <- nnetar(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
fcastedvaluesExtracted
minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- ts(fcastedvaluesExtracted*(maxval-minval)+minval)
ts.plot(test_D,col="red")
plot(F_1)
#############################################################################
IMF2 <- as.data.frame(df_IMFs$imf[,2])
normalize2 <- function(IMF2){return((IMF2-min(IMF2))/(max(IMF2)-min(IMF2)))}
IMF2_Norm <- ts(normalize2(IMF2))
###################Time series, Fitting and Forecasting ##############
fit_2 <- nnetar(IMF2_Norm)
nfcast_2 <- forecast(fit_2,h=255) #newdata =test1
fcastedvaluesExtracted2 <- as.numeric(nfcast_2$mean)
minval2 <- min(IMF2);maxval2 <- max(IMF2)
F_2 <- fcastedvaluesExtracted2*(maxval2-minval2)+minval2
#####################################################
IMF3 <- as.data.frame(df_IMFs$imf[,3])
```

Figure 7.10:

```
normalize3 <- function(IMF3){return((IMF3-min(IMF3))/(max(IMF3)-min(IMF3)))}
IMF3_Norm <- ts(normalize3(IMF3))
##################Time series, Fitting and Forecasting ################
fit_3 <- nnetar(IMF3_Norm)
nfcast_3 <- forecast(fit_3,h=255) #newdata =test1
fcastedvaluesExtracted3 <- as.numeric(nfcast_3$mean)
minval3 <- min(IMF3);maxval3 <- max(IMF3)
F_3 <- fcastedvaluesExtracted3*(maxval3-minval3)+minval3
#########################################################################
#####################
IMF4 <- as.data.frame(df_IMFs$imf[,4])
normalize4 <- function(IMF4){return((IMF4-min(IMF4))/(max(IMF4)-min(IMF4)))}
IMF4_Norm <- ts(normalize4(IMF4))
###################Time series, Fitting and Forecasting ####################
fit_4 <- nnetar(IMF4_Norm)
nfcast_4 <- forecast(fit_4,h=255) #newdata =test1
fcastedvaluesExtracted4 <- as.numeric(nfcast_4$mean)
minval4 <- min(IMF4);maxval4 <- max(IMF4)
F_4 <- fcastedvaluesExtracted4*(maxval4-minval4)+minval4
#########################################################################
IMF5 <- as.data.frame(df_IMFs$imf[,5])
normalize5 <- function(IMF5){return((IMF5-min(IMF5))/(max(IMF5)-min(IMF5)))}
IMF5_Norm <- ts(normalize5(IMF5))
###################Time series, Fitting and Forecasting ####################
fit_5 <- nnetar(IMF5_Norm)
nfcast_5 <- forecast(fit_5,h=255) #newdata =test1
fcastedvaluesExtracted5 <- as.numeric(nfcast_5$mean)
minval5 <- min(IMF5);maxval5 <- max(IMF5)
F_5 <- fcastedvaluesExtracted5*(maxval5-minval5)+minval5
#########################################################################
IMF6 <- as.data.frame(df_IMFs$imf[,6])
normalize6 <- function(IMF6){return((IMF6-min(IMF6))/(max(IMF6)-min(IMF6)))}
IMF6_Norm <- ts(normalize6(IMF6))
###################Time series, Fitting and Forecasting ###########################
fit_6 <- nnetar(IMF6_Norm)
nfcast_6 <- forecast(fit_6,h=255) #newdata =test1
fcastedvaluesExtracted6 <- as.numeric(nfcast_6$mean)
minval6 <- min(IMF6);maxval6 <- max(IMF6)
F_6 <- fcastedvaluesExtracted6*(maxval6-minval6)+minval6
#########################################################################
IMF7 <- as.data.frame(df_IMFs$imf[,7])
normalize7 <- function(IMF7){return((IMF7-min(IMF7))/(max(IMF7)-min(IMF7)))}
IMF7_Norm <- ts(normalize7(IMF7))
###################Time series, Fitting and Forecasting ####################
fit_7 <- nnetar(IMF7_Norm)
nfcast_7 <- forecast(fit_7,h=255) #newdata =test1
fcastedvaluesExtracted7 <- as.numeric(nfcast_7$mean)
```

Figure 7.11:

```
minval7 <- min(IMF7);maxval7 <- max(IMF7)
F_7 <- fcastedvaluesExtracted7*(maxval7-minval7)+minval7
#########################################################################
IMF8 <- as.data.frame(diff(df_IMFs$imf[,8]))
normalize8 <- function(IMF8){return((IMF8-min(IMF8))/(max(IMF8)-min(IMF8)))}
IMF8_Norm <- ts(normalize8(IMF8))
###################Time series, Fitting and Forecasting ######################
fit_8 <- nnetar(IMF8_Norm)
nfcast_8 <- forecast(fit_8,h=255) #newdata =test1
fcastedvaluesExtracted8 <- as.numeric(nfcast_8$mean)
minval8 <- min(IMF8);maxval8 <- max(IMF8)
F_8 <- fcastedvaluesExtracted8*(maxval8-minval8)+minval8
#####################################################################
Res <- as.data.frame(diff(df_IMFs$residue))
normalizeRes <- function(Res){return((Res-min(Res))/(max(Res)-min(Res)))}
Res_Norm <- ts(normalizeRes(Res))
###################Time series, Fitting and Forecasting#####################
fit_Res <- nnetar(Res_Norm)
nfcast_Res <- forecast(fit_Res,h=255) #newdata =test1
fcastedvaluesExtracted_Res <- as.numeric(nfcast_Res$mean)
minvalRes <- min(Res);maxvalRes <- max(Res)
F_Res <- fcastedvaluesExtracted_Res*(maxvalRes-minvalRes)+minvalRes
#############################################################################
#Sum up all prediction = Final forecasting value
FinalForecast_emd_BPNN_Res <- ts(F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8+F_Res)
FinalForecast_emd_BPNN_Res
FinalForecast_emd_BPNN <- ts(F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8)
FinalForecast_emd_BPNN
dd <- data.frame(test_D ,FinalForecast_emd_BPNN_Res)
col_headings <- c("Actual Price","Forecast Price")
names(dd) <- col_headings;attach(dd)
percentage_err=((dd$`Actual Price`-dd$`Forecast Price`)/(dd$`Actual Price`))
percentage_err
mean(percentage_err)
MAE(dd$`Actual Price`,dd$`Forecast Price`)
RMSE(dd$`Actual Price`,dd$`Forecast Price`)
MAPE(dd$`Actual Price`,dd$`Forecast Price`)
A<-MAE
plot.ts(dd$`Actual Price`)
lines(dd$`Forecast Price`,col="red")
lines(dd$`Forecast Price`, col="red")
FinalForecast_emd_BPNN_Res <- ts(FinalForecast_emd_BPNN_Res)
test_D <- ts(test_D)
plot.ts(FinalForecast_emd_BPNN_Res,col="blue", main="The three
components",xlab="Time/monthly",ylab=" " )
lines(test_D,col="red")
legend("bottom",col=c("blue", "red"),lwd=2,
```

Figure 7.12:

```
    legend=c("Forecasted", "Test"))
plot.ts(train_D)
plot.ts(`Actual Price`, col="red")
lines(train_D)
############################################################
#2_CRUDE_VMD_PBNN
library(DescTools)
library(vmd)
dd_vmd <- vmd(train_D, alpha =100,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
dd_vmd <- as.data.frame(dd_vmd)

###M1 (Residue)
M1 <- dd_vmd$M1
normalize <- function(M1){return((M1-min(M1))/(max(M1)-min(M1)))}
M1_Norm <- normalize(M1)
fit.M1 <- nnetar(M1_Norm)
fcast.M1 <- forecast(fit.M1,h=255) #newdata =test1
fcastExtracted <- as.numeric(fcast.M1$mean)
fcastExtracted <- as.data.frame(fcastExtracted)
minval <- min(M1);maxval <- max(M1)
finalfcast.M1 <- fcastExtracted*(maxval-minval)+minval
#######M2#########################################################################
M2 <- dd_vmd$M2
normalize2 <- function(M2){return((M2-min(M2))/(max(M2)-min(M2)))}
M2_Norm <- normalize(M2)
fit.M2 <- nnetar(M2_Norm)
fcast.M2 <- forecast(fit.M2,h=255) #newdata =test1
fcastExtracted2 <- as.numeric(fcast.M2$mean)
fcastExtracted2 <- as.data.frame(fcastExtracted2)
minval2 <- min(M2);maxval2 <- max(M2)
finalfcast.M2 <- fcastExtracted2*(maxval2-minval2)+minval2
#######M2#########################################################################
M3 <- dd_vmd$M3
normalize3 <- function(M3){return((M3-min(M3))/(max(M3)-min(M3)))};M3_Norm <-
normalize(M3)
fit.M3 <- nnetar(M3_Norm)
fcast.M3 <- forecast(fit.M3,h=255) #newdata =test1
fcastExtracted3 <- as.numeric(fcast.M3$mean);fcastExtracted3 <-
as.data.frame(fcastExtracted3)
minval3 <- min(M3);maxval3 <- max(M3)
finalfcast.M3 <- fcastExtracted3*(maxval3-minval3)+minval3
#######M2#########################################################################
M4 <- dd_vmd$M4
normalize4 <- function(M4){return((M4-min(M4))/(max(M4)-min(M4)))};M4_Norm <-
normalize(M4)
fit.M4 <- nnetar(M4_Norm)
```

Figure 7.13:

```
fcast.M4 <- forecast(fit.M4,h=255) #newdata =test1
fcastExtracted4 <- as.numeric(fcast.M4$mean);fcastExtracted4 <-
as.data.frame(fcastExtracted4)
minval4 <- min(M4);maxval4 <- max(M4)
finalfcast.M4 <- fcastExtracted4*(maxval4-minval4)+minval4
#######M5#############################################################################
M5 <- dd_vmd$M5
normalize5 <- function(M5){return((M5-min(M5))/(max(M5)-min(M5)))};M5_Norm <-
normalize(M5)
fit.M5 <- nnetar(M5_Norm)
fcast.M5 <- forecast(fit.M5,h=255) #newdata =test1
fcastExtracted5 <- as.numeric(fcast.M5$mean);fcastExtracted5 <-
as.data.frame(fcastExtracted5)
minval5 <- min(M5);maxval5 <- max(M5)
finalfcast.M5 <- fcastExtracted5*(maxval5-minval5)+minval5
#######M6#############################################################################
M6 <- dd_vmd$M6
normalize6 <- function(M6){return((M6-min(M6))/(max(M6)-min(M6)))};M6_Norm <-
normalize(M6)
fit.M6 <- nnetar(M6_Norm)
fcast.M6 <- forecast(fit.M6,h=255) #newdata =test1
fcastExtracted6 <- as.numeric(fcast.M6$mean);fcastExtracted6 <-
as.data.frame(fcastExtracted6)
minval6 <- min(M6);maxval6 <- max(M6)
finalfcast.M6 <- fcastExtracted6*(maxval6-minval6)+minval6
#######M7#############################################################################
############
M7 <- dd_vmd$M7
normalize7 <- function(M7){return((M7-min(M7))/(max(M7)-min(M7)))};M7_Norm <-
normalize(M7)
fit.M7 <- nnetar(M7_Norm)
fcast.M7 <- forecast(fit.M7,h=255) #newdata =test1
fcastExtracted7 <- as.numeric(fcast.M7$mean);fcastExtracted7 <-
as.data.frame(fcastExtracted7)
minval7 <- min(M7);maxval7 <- max(M7)
finalfcast.M7 <- fcastExtracted7*(maxval7-minval7)+minval7
#######M8#############################################################################
############
M8 <- dd_vmd$M8
normalize8 <- function(M8){return((M8-min(M8))/(max(M8)-min(M8)))};M8_Norm <-
normalize(M8)
fit.M8 <- nnetar(M8_Norm)
fcast.M8 <- forecast(fit.M8,h=255) #newdata =test1
fcastExtracted8 <- as.numeric(fcast.M8$mean);fcastExtracted8 <-
as.data.frame(fcastExtracted8)
minval8 <- min(M8);maxval8 <- max(M8)
finalfcast.M8 <- fcastExtracted8*(maxval8-minval8)+minval8
```

Figure 7.14:

```
#############################################################################
FinalForecast_vmd_BPNN <- finalfcast.M1
+finalfcast.M2+finalfcast.M3+finalfcast.M4+finalfcast.M5+finalfcast.M6+finalfcast.M7+fin
alfcast.M8
vd <- data.frame(test_D ,FinalForecast_vmd_BPNN)
col_headings <- c("Actual Price","Forecast Price")
names(vd) <- col_headings;attach(vd)
MAE(vd$`Actual Price`,vd$`Forecast Price`)
RMSE(vd$`Actual Price`,vd$`Forecast Price`)
MAPE(vd$`Actual Price`,vd$`Forecast Price`)
B<-MAE
B
plot.ts(test_D,col="red", main="The three components",xlab="Time/monthly",ylab="Corn
Future Price ")
lines(FinalForecast_vmd_BPNN, col="blue")
legend("topright",col=c("blue", "red"),lwd=2,
       legend=c("Forecasted", "Test"))
A<-MAE
############################ EMD_ARIMA FORECASTING ########################

###3_EMD_ARIMA
library(MASS)
library(tseries)
library(forecast)

df_IMFs<- emd(train_D, tt=NULL, tol=sd(train_D)*0.1^2,boundary="periodic")
################### Normalisation of trained data ##################
IMF1 <- as.data.frame(df_IMFs$imf[,1]);Res <- as.data.frame(df_IMFs$residue)
ts.plot(IMF1)
IMF1_Norm <- ts((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))

####################Time series, Fitting and Forecasting ##########################
fit_1 <- nnetar(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
fcastedvaluesExtracted

minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- ts(fcastedvaluesExtracted*(maxval-minval)+minval)
ts.plot(test_D,col="red")
plot(F_1)

IMF1 <- as.data.frame(df_IMFs$imf[,1])
normalize <- function(IMF1){return((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))}
IMF1_Norm <- ts(normalize(IMF1))
####################Time series, Fitting and Forecasting ##########################
fit_1 <- auto.arima(IMF1_Norm)
```

Figure 7.15:

```
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- fcastedvaluesExtracted*(maxval-minval)+minval
################################################################################
IMF2 <- as.data.frame(df_IMFs$imf[,2])
normalize2 <- function(IMF2){return((IMF2-min(IMF2))/(max(IMF2)-min(IMF2)))}
IMF2_Norm <- ts(normalize2(IMF2))
###################Time series, Fitting and Forecasting ####################
fit_2 <- auto.arima(IMF2_Norm)
nfcast_2 <- forecast(fit_2,h=255) #newdata =test1
fcastedvaluesExtracted2 <- as.numeric(nfcast_2$mean)
minval2 <- min(IMF2);maxval2 <- max(IMF2)
F_2 <- fcastedvaluesExtracted2*(maxval2-minval2)+minval2
################################################################################
IMF3 <- as.data.frame(df_IMFs$imf[,3])
normalize3 <- function(IMF3){return((IMF3-min(IMF3))/(max(IMF3)-min(IMF3)))}
IMF3_Norm <- ts(normalize3(IMF3))
###################Time series, Fitting and Forecasting ################
fit_3 <- auto.arima(IMF3_Norm)
nfcast_3 <- forecast(fit_3,h=255) #newdata =test1
fcastedvaluesExtracted3 <- as.numeric(nfcast_3$mean)
minval3 <- min(IMF3);maxval3 <- max(IMF3)
F_3 <- fcastedvaluesExtracted3*(maxval3-minval3)+minval3
################################################################################
IMF4 <- as.data.frame(df_IMFs$imf[,4])
normalize4 <- function(IMF4){return((IMF4-min(IMF4))/(max(IMF4)-min(IMF4)))}
IMF4_Norm <- ts(normalize4(IMF4))
###################Time series, Fitting and Forecasting ####################
fit_4 <- auto.arima(IMF4_Norm)
nfcast_4 <- forecast(fit_4,h=255) #newdata =test1
fcastedvaluesExtracted4 <- as.numeric(nfcast_4$mean)
minval4 <- min(IMF4);maxval4 <- max(IMF4)
F_4 <- fcastedvaluesExtracted4*(maxval4-minval4)+minval4
################################################################################
IMF5 <- as.data.frame(df_IMFs$imf[,5])
normalize5 <- function(IMF5){return((IMF5-min(IMF5))/(max(IMF5)-min(IMF5)))}
IMF5_Norm <- ts(normalize5(IMF5))
###################Time series, Fitting and Forecasting ########################
fit_5 <- auto.arima(IMF5_Norm)
nfcast_5 <- forecast(fit_5,h=255) #newdata =test1
fcastedvaluesExtracted5 <- as.numeric(nfcast_5$mean)
minval5 <- min(IMF5);maxval5 <- max(IMF5)
F_5 <- fcastedvaluesExtracted5*(maxval5-minval5)+minval5
################################################################################
IMF6 <- as.data.frame(df_IMFs$imf[,6])
normalize6 <- function(IMF6){return((IMF6-min(IMF6))/(max(IMF6)-min(IMF6)))}
```

Figure 7.16:

```
IMF6_Norm <- ts(normalize6(IMF6))
####################Time series, Fitting and Forecasting#####################
fit_6 <- auto.arima(IMF6_Norm)
nfcast_6 <- forecast(fit_6,h=255) #newdata =test1
fcastedvaluesExtracted6 <- as.numeric(nfcast_6$mean)
minval6 <- min(IMF6);maxval6 <- max(IMF6)
F_6 <- fcastedvaluesExtracted6*(maxval6-minval6)+minval6
##############################################################################
IMF7 <- as.data.frame(df_IMFs$imf[,7])
normalize7 <- function(IMF7){return((IMF7-min(IMF7))/(max(IMF7)-min(IMF7)))}
IMF7_Norm <- ts(normalize7(IMF7))
####################Time series, Fitting and Forecasting ######################
fit_7 <- auto.arima(IMF7_Norm)
nfcast_7 <- forecast(fit_7,h=255) #newdata =test1
fcastedvaluesExtracted7 <- as.numeric(nfcast_7$mean)
minval7 <- min(IMF7);maxval7 <- max(IMF7)
F_7 <- fcastedvaluesExtracted7*(maxval7-minval7)+minval7
##############################################################################
IMF8 <- as.data.frame(df_IMFs$imf[,8])
normalize8 <- function(IMF8){return((IMF8-min(IMF8))/(max(IMF8)-min(IMF8)))}
IMF8_Norm <- ts(normalize8(IMF8))
####################Time series, Fitting and Forecasting ######################
fit_8 <- auto.arima(IMF8_Norm)
nfcast_8 <- forecast(fit_8,h=255) #newdata =test1
fcastedvaluesExtracted8 <- as.numeric(nfcast_8$mean)
minval8 <- min(IMF8);maxval8 <- max(IMF8)
F_8 <- fcastedvaluesExtracted8*(maxval8-minval8)+minval8

####################Time series, Fitting and Forecasting ####################
Res <- as.data.frame(df_IMFs$residue)
normalizeRes <- function(Res){return((Res-min(Res))/(max(Res)-min(Res)))}
Res_Norm <- ts(normalize(Res))
fit_Res <- auto.arima(Res_Norm)
nfcast_Res <- forecast(fit_Res,h=255) #newdata =test1
fcastedvaluesExtractedRes <- as.numeric(nfcast_Res$mean)
minvalRes <- min(Res);maxvalRes <- max(Res)
F_Res <- fcastedvaluesExtractedRes*(maxvalRes-minvalRes)+minvalRes

################### FinalForecast.emd.ARIMA ####################
FinalForecast.emd.ARIMA <- F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8+F_Res
FinalForecast.emd.ARIMA<- ts(FinalForecast.emd.ARIMA)
ead <- data.frame(test_D ,FinalForecast.emd.ARIMA)
col_headings <- c("Actual Price","Forecast Price")
names(ead) <- col_headings;attach(ead)
MAE(ead$`Actual Price`,ead$`Forecast Price`)
RMSE(ead$`Actual Price`,ead$`Forecast Price`)
MAPE(ead$`Actual Price`,ead$`Forecast Price`)
```

Figure 7.17:

```
plot(FinalForecast.emd.ARIMA)
FinalForecast_vmd_BPNN
plot.ts(`Actual Price`)
#################### VMD_ARIMA ##############################
###4_VMD_ARIMA
###M1 (Residue)
M1 <- dd_vmd$M1
normalize <- function(M1){return((M1-min(M1))/(max(M1)-min(M1)))}
M1_Norm <- normalize(M1)
fit.M1 <- auto.arima(M1_Norm)
fcast.M1 <- forecast(fit.M1,h=255) #newdata =test1
fcastExtracted <- as.numeric(fcast.M1$mean);fcastExtracted <-
as.data.frame(fcastExtracted)
minval <- min(M1);maxval <- max(M1)
finalfcast.M1 <- as.numeric(fcast.M1$mean)
#finalfcast.M1 <- fcastExtracted*(maxval-minval)+minval
#### M2
M2 <- dd_vmd$M2
normalize2 <- function(M2){return((M2-min(M2))/(max(M2)-min(M2)))};M2_Norm <-
normalize(M2)
fit.M2 <- auto.arima(M2_Norm)
fcast.M2 <- forecast(fit.M2,h=255) #newdata =test1
fcastExtracted2 <- as.numeric(fcast.M2$mean)
fcastExtracted2 <- as.data.frame(fcastExtracted2)
minval2 <- min(M2);maxval2 <- max(M2)
finalfcast.M2 <- as.numeric(fcast.M2$mean)
#finalfcast.M2 <- fcastExtracted2*(maxval2-minval2)+minval2
#######M2#########################################################################
M3 <- dd_vmd$M3
normalize3 <- function(M3){return((M3-min(M3))/(max(M3)-min(M3)))};M3_Norm <-
normalize(M3)
fit.M3 <- auto.arima(M3_Norm)
fcast.M3 <- forecast(fit.M3,h=255) #newdata =test1
fcastExtracted3 <- as.numeric(fcast.M3$mean);fcastExtracted3 <-
as.data.frame(fcastExtracted3)
minval3 <- min(M3);maxval3 <- max(M3)
finalfcast.M3 <- as.numeric(fcast.M3$mean)
#finalfcast.M3 <- fcastExtracted3*(maxval3-minval3)+minval3
#######M2#########################################################################
M4 <- dd_vmd$M4
normalize4 <- function(M4){return((M4-min(M4))/(max(M4)-min(M4)))};M4_Norm <-
normalize(M4)
fit.M4 <- auto.arima(M4_Norm)
fcast.M4 <- forecast(fit.M4,h=255) #newdata =test1
fcastExtracted4 <- as.numeric(fcast.M4$mean);fcastExtracted4 <-
as.data.frame(fcastExtracted4)
minval4 <- min(M4);maxval4 <- max(M4)
```

Figure 7.18:

```
finalfcast.M4 <- as.numeric(fcast.M4$mean)
#finalfcast.M4 <- fcastExtracted4*(maxval4-minval4)+minval4
#######M5#########################################################################
M5 <- dd_vmd$M5
normalize5 <- function(M5){return((M5-min(M5))/(max(M5)-min(M5)))};M5_Norm <-
normalize(M5)
fit.M5 <- auto.arima(M5_Norm)
fcast.M5 <- forecast(fit.M5,h=255) #newdata =test1
fcastExtracted5 <- as.numeric(fcast.M5$mean);fcastExtracted5 <-
as.data.frame(fcastExtracted5)
minval5 <- min(M5);maxval5 <- max(M5)
finalfcast.M5 <- as.numeric(fcast.M5$mean)
#finalfcast.M5 <- fcastExtracted5*(maxval5-minval5)+minval5
#######M6#########################################################################
M6 <- dd_vmd$M6
normalize6 <- function(M6){return((M6-min(M6))/(max(M6)-min(M6)))};M6_Norm <-
normalize(M6)
fit.M6 <- auto.arima(M6_Norm)
fcast.M6 <- forecast(fit.M6,h=255) #newdata =test1
fcastExtracted6 <- as.numeric(fcast.M6$mean);fcastExtracted6 <-
as.data.frame(fcastExtracted6)
minval6 <- min(M6);maxval6 <- max(M6)
finalfcast.M6 <- as.numeric(fcast.M6$mean)
#finalfcast.M6 <- fcastExtracted6*(maxval6-minval6)+minval6
#######M7#########################################################################
M7 <- dd_vmd$M7
normalize7 <- function(M7){return((M7-min(M7))/(max(M7)-min(M7)))};M7_Norm <-
normalize(M7)
fit.M7 <- auto.arima(M7_Norm)
fcast.M7 <- forecast(fit.M7,h=255) #newdata =test1
fcastExtracted7 <- as.numeric(fcast.M7$mean);fcastExtracted7 <-
as.data.frame(fcastExtracted7)
minval7 <- min(M7);maxval7 <- max(M7)
finalfcast.M7 <- as.numeric(fcast.M7$mean)
#finalfcast.M7 <- fcastExtracted7*(maxval7-minval7)+minval7
#######M8#########################################################################
M8 <- dd_vmd$M8
normalize8 <- function(M8){return((M8-min(M8))/(max(M8)-min(M8)))};M8_Norm <-
normalize(M8)
fit.M8 <- auto.arima(M8_Norm)
fcast.M8 <- forecast(fit.M8,h=255) #newdata =test1
fcastExtracted8 <- as.numeric(fcast.M8$mean);fcastExtracted8 <-
as.data.frame(fcastExtracted8)
minval8 <- min(M8);maxval8 <- max(M8)
finalfcast.M8 <- as.numeric(fcast.M8$mean)
#finalfcast.M8 <- fcastExtracted8*(maxval8-minval8)+minval8
```

Figure 7.19:

```
#############################################################################
#FinalForecast_vmd_arima
FinalForecast_vmd_arima <-
finalfcast.M1+finalfcast.M2+finalfcast.M3+finalfcast.M4+finalfcast.M5+finalfcast.M6+fina
lfcast.M7+finalfcast.M8
vad <- data.frame(test_D ,FinalForecast_vmd_arima)
col_headings <- c("Actual Price","Forecast Price")
names(vad) <- col_headings;attach(vad)
MAE(vad$`Actual Price`,vad$`Forecast Price`)
RMSE(vad$`Actual Price`,vad$`Forecast Price`)
MAPE(vad$`Actual Price`,vad$`Forecast Price`)
FinalForecast_vmd_arima <- ts(FinalForecast_vmd_arima)
plot(FinalForecast_vmd_arima)
(FinalForecast_vmd_arima)
lines(test, col="red")
dm.test(ead$'Actual Price'-ead$'Forecast Price',vad$'Actual Price'-vad$'Forecast Price')
#two models have different accuracy
CP <- data.frame(CP$Value)
index <- sample(nrow(CP),round(0.8*nrow(CP)))
train <- CP[index,]
test <- CP[-index,]
CP_Norm <- ts((train-min(train))/(max(train)-min(train)))
normalizeCP<- function(train){return((train-min(train))/(max(train)-min(train)))}
CP_Norm <- normalize(train)
fit_og <- nnetar(CP_Norm)
nfcast_og <- forecast(fit_og,h=255) #newdata =test1
FVE <- as.numeric(nfcast_og$mean)
F1<-FVE
minval <- min(train);maxval <- max(train)
F1 <- FVE*(maxval-minval)+minval
ts.plot(test,col="red")
plot(F1)
ff <- data.frame(test ,F1)
col_headings <- c("Actual Price","Forecast Price")
names(ff) <- col_headings;attach(ff)
MAE(ff$`Actual Price`,ff$`Forecast Price`)
RMSE(ff$`Actual Price`,ff$`Forecast Price`)
MAPE(ff$`Actual Price`,ff$`Forecast Price`)
CP2<-CP
CP2
index <- sample(nrow(CP2),round(0.8*nrow(CP2)))
train <- CP2[index,]
test <- CP2[-index,]
test
CP2_Norm <- ts(2*(train-min(train))/(max(train)-min(train))-1)
max(CP2_Norm)
min(CP2_Norm)
```

Figure 7.20:

```
fit_O <- auto.arima(CP2_Norm)
nfcast_O <- forecast(fit_O,h=255) #newdata =test1
plot(nfcast_O)
as.data.frame(nfcast_O)
FVEO <- ts(as.numeric(nfcast_O$mean))
plot(FVEO)
minval <- min(train);maxval <- max(train)
FO <- ts(FVEO*(maxval-minval)+minval)
FO
ddO <- data.frame(test ,FO)
col_headings <- c("Actual Price","Forecast Price")
names(ddO) <- col_headings;attach(ddO)
MAE(ddO$`Actual Price`,ddO$`Forecast Price`)
RMSE(ddO$`Actual Price`,ddO$`Forecast Price`)
MAPE(ddO$`Actual Price`,ddO$`Forecast Price`)
#####################################################################
#autoplot(FinalForecast_emd_BPNN)
#plot(FinalForecast_emd_BPNN)
#lines(FinalForecast_emd_BPNN_Res, col="violet")
#lines(FinalForecast_vmd_arima, col="red")
#lines(FinalForecast.emd.ARIMA, col="green")
#lines(FinalForecast_vmd_BPNN, col="blue")
#lines(F1,col="yellow")
#lines(FVEO)
#lines(test)
test_norm <- (test-min(test))/(max(test)-min(test))
ts.test <- xts(test$Value, order.by = as.POSIXct(test$Date, "UTC"))
plot(ts.test)
lines(FinalForecast_vmd_arima)

autoplot(ts(test), series = "Actual")+
autolayer(FinalForecast.emd.ARIMA, series = "EMD-ARIMA")+
 autolayer(FinalForecast_emd_BPNN, series = "EMD-BPNN")+
  autolayer(ts(FinalForecast_vmd_arima), series = "VMD-ARIMA")+
   autolayer(ts(FinalForecast_vmd_BPNN), series = "VMD-BPNN")+
    autolayer(ts(F1), series = "BPNN")+
     autolayer(ts(FVEO), series = "ARIMA")
plot(FinalForecast.emd.ARIMA)
lines(test_norm)
plot()
plot(FinalForecast_emd_BPNN_Res, col="black",xlab="Time/daily",ylab="Crude oil Future
Price " )
lines(FinalForecast_vmd_arima, col="red")
lines(FinalForecast.emd.ARIMA, col="green")
lines(FinalForecast_vmd_BPNN, col="blue")
lines(F1,col="yellow")
lines(FVEO, col="brown")
```

Figure 7.21:

```
lines(`Actual Price`, col="purple")
legend("topright",col=c("black", "red", "green", "blue", "yellow", "brown",
"purple"),lwd=2,
      legend=c("EMD-BPNN", "VMD-ARIMA", "EMD-ARIMA", "EMD-BPNN", "BPNN",
"ARIMA", "Actual"))
###################### Dm Test ############################
library(forecast)
#null hypothesis is that the two methods have the same forecast accuracy

######### Comparing forecasts accuracy MAE ##########
dm.test(mae_emd_bpnn,mae_vmd_bpnn)
dm.test(mae_emd_bpnn,mae_emd_arima)
dm.test(mae_emd_bpnn,mae_vmd_arima)
dm.test(mae_emd_bpnn,mae_bpnn)
dm.test(mae_emd_bpnn,mae_arima)

dm.test(mae_vmd_bpnn,mae_emd_arima)
dm.test(mae_vmd_bpnn,mae_vmd_arima)
dm.test(mae_vmd_bpnn,mae_bpnn)
dm.test(mae_vmd_bpnn,mae_arima)

dm.test(mae_emd_arima,mae_vmd_arima)
dm.test(mae_emd_arima,mae_arima)
dm.test(mae_emd_arima,mae_bpnn)

dm.test(mae_vmd_arima,mae_arima)
dm.test(mae_vmd_arima,mae_bpnn)

dm.test(mae_arima,mae_bpnn)

#Comparing forecasts accuracy RMSE
dm.test(mse_emd_bpnn,mse_vmd_bpnn)
dm.test(mse_emd_bpnn,mse_emd_arima)
dm.test(mse_emd_bpnn,mse_vmd_arima)
dm.test(mse_emd_bpnn,mse_bpnn)
dm.test(mse_emd_bpnn,mse_arima)

dm.test(mse_vmd_bpnn,mse_emd_arima)
dm.test(mse_vmd_bpnn,mse_vmd_arima)
dm.test(mse_vmd_bpnn,mse_bpnn)
dm.test(mse_vmd_bpnn,mse_arima)

dm.test(mse_emd_arima,mse_vmd_arima)
dm.test(mse_emd_arima,mse_arima)
dm.test(mse_emd_arima,mse_bpnn)

dm.test(mse_vmd_arima,mse_arima)
```

Figure 7.22:

```
dm.test(mse_vmd_arima,mse_bpnn)
dm.test(mse_arima,mse_bpnn)
##############################################################################

#Comparing forecasts accuracy MAPE
dm.test(mape_emd_bpnn,mape_vmd_bpnn)
dm.test(mape_emd_bpnn,mape_emd_arima)
dm.test(mape_emd_bpnn,mape_vmd_arima)
dm.test(mape_emd_bpnn,mape_bpnn)
dm.test(mape_emd_bpnn,mape_arima)

dm.test(mape_vmd_bpnn,mape_emd_arima)
dm.test(mape_vmd_bpnn,mape_vmd_arima)
dm.test(mape_vmd_bpnn,mape_bpnn)
dm.test(mape_vmd_bpnn,mape_arima)

dm.test(mape_emd_arima,mape_vmd_arima)
dm.test(mape_emd_arima,mape_arima)
dm.test(mape_emd_arima,mape_bpnn)

dm.test(mape_vmd_arima,mape_arima)
dm.test(mape_vmd_arima,mape_bpnn)

dm.test(mape_arima,mape_bpnn)

########################## FORECASTING OF CRUDE OIL #################
###################### Data partition# ################################
index <- sample(nrow(CP),round(0.8*nrow(CP)))
train <- CP[index,]
nrow(train)
test <- CP[-index,]
nrow(test)
train_D <- train$Value
test_D <- test$Value

normalizetest <- function(test){return((test-min(test))/(max(test)-min(test)))}
test_norm <- normalize(test)

###################Decomposition#############################################
df_IMFs<- emd(train_D, tt=NULL, tol=sd(train_D)*0.1^2,boundary="periodic")

###################Normalisation of trained data#################################
IMF1 <- as.data.frame(df_IMFs$imf[,1]);Res <- as.data.frame(df_IMFs$residue)
ts.plot(IMF1)
IMF1_Norm <- ts((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))
###################Time series, Fitting and Forecasting ##################
fit_1 <- nnetar(IMF1_Norm)
```

Figure 7.23:

```
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
fcastedvaluesExtracted

minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- ts(fcastedvaluesExtracted*(maxval-minval)+minval)
ts.plot(test_D,col="red")
plot(F_1)
#############################################################################
IMF2 <- as.data.frame(df_IMFs$imf[,2])
normalize2 <- function(IMF2){return((IMF2-min(IMF2))/(max(IMF2)-min(IMF2)))}
IMF2_Norm <- ts(normalize2(IMF2))
###################Time series, Fitting and Forecasting ######################
fit_2 <- nnetar(IMF2_Norm)
nfcast_2 <- forecast(fit_2,h=255) #newdata =test1

fcastedvaluesExtracted2 <- as.numeric(nfcast_2$mean)
minval2 <- min(IMF2);maxval2 <- max(IMF2)
F_2 <- fcastedvaluesExtracted2*(maxval2-minval2)+minval2
#############################################################################
IMF3 <- as.data.frame(df_IMFs$imf[,3])
normalize3 <- function(IMF3){return((IMF3-min(IMF3))/(max(IMF3)-min(IMF3)))}
IMF3_Norm <- ts(normalize3(IMF3))
###################Time series, Fitting and Forecasting ######################
fit_3 <- nnetar(IMF3_Norm)
nfcast_3 <- forecast(fit_3,h=255) #newdata =test1

fcastedvaluesExtracted3 <- as.numeric(nfcast_3$mean)
minval3 <- min(IMF3);maxval3 <- max(IMF3)
F_3 <- fcastedvaluesExtracted3*(maxval3-minval3)+minval3
#############################################################################
IMF4 <- as.data.frame(df_IMFs$imf[,4])
normalize4 <- function(IMF4){return((IMF4-min(IMF4))/(max(IMF4)-min(IMF4)))}
IMF4_Norm <- ts(normalize4(IMF4))
###################Time series, Fitting and Forecasting ######################
fit_4 <- nnetar(IMF4_Norm)
nfcast_4 <- forecast(fit_4,h=255) #newdata =test1
fcastedvaluesExtracted4 <- as.numeric(nfcast_4$mean)
minval4 <- min(IMF4);maxval4 <- max(IMF4)
F_4 <- fcastedvaluesExtracted4*(maxval4-minval4)+minval4
#############################################################################
IMF5 <- as.data.frame(df_IMFs$imf[,5])
normalize5 <- function(IMF5){return((IMF5-min(IMF5))/(max(IMF5)-min(IMF5)))}
IMF5_Norm <- ts(normalize5(IMF5))
###################Time series, Fitting and Forecasting ######################
fit_5 <- nnetar(IMF5_Norm)
nfcast_5 <- forecast(fit_5,h=255) #newdata =test1
```

Figure 7.24:

```
fcastedvaluesExtracted5 <- as.numeric(nfcast_5$mean)
minval5 <- min(IMF5);maxval5 <- max(IMF5)
F_5 <- fcastedvaluesExtracted5*(maxval5-minval5)+minval5
#############################################################################
IMF6 <- as.data.frame(df_IMFs$imf[,6])
normalize6 <- function(IMF6){return((IMF6-min(IMF6))/(max(IMF6)-min(IMF6)))}
IMF6_Norm <- ts(normalize6(IMF6))
####################Time series, Fitting and Forecasting ########################
fit_6 <- nnetar(IMF6_Norm)
nfcast_6 <- forecast(fit_6,h=255) #newdata =test1
fcastedvaluesExtracted6 <- as.numeric(nfcast_6$mean)
minval6 <- min(IMF6);maxval6 <- max(IMF6)
F_6 <- fcastedvaluesExtracted6*(maxval6-minval6)+minval6
#############################################################################
IMF7 <- as.data.frame(df_IMFs$imf[,7])
normalize7 <- function(IMF7){return((IMF7-min(IMF7))/(max(IMF7)-min(IMF7)))}
IMF7_Norm <- ts(normalize7(IMF7))
####################Time series, Fitting and Forecasting ########################
fit_7 <- nnetar(IMF7_Norm)
nfcast_7 <- forecast(fit_7,h=255) #newdata =test1
fcastedvaluesExtracted7 <- as.numeric(nfcast_7$mean)
minval7 <- min(IMF7);maxval7 <- max(IMF7)
F_7 <- fcastedvaluesExtracted7*(maxval7-minval7)+minval7
#############################################################################
IMF8 <- as.data.frame(diff(df_IMFs$imf[,8]))
normalize8 <- function(IMF8){return((IMF8-min(IMF8))/(max(IMF8)-min(IMF8)))}
IMF8_Norm <- ts(normalize8(IMF8))
####################Time series, Fitting and Forecasting ######################
fit_8 <- nnetar(IMF8_Norm)
nfcast_8 <- forecast(fit_8,h=255) #newdata =test1
fcastedvaluesExtracted8 <- as.numeric(nfcast_8$mean)
minval8 <- min(IMF8);maxval8 <- max(IMF8)
F_8 <- fcastedvaluesExtracted8*(maxval8-minval8)+minval8
#############################################################################
Res <- as.data.frame(diff(df_IMFs$residue))
normalizeRes <- function(Res){return((Res-min(Res))/(max(Res)-min(Res)))}
Res_Norm <- ts(normalizeRes(Res))
####################Time series, Fitting and Forecasting ######################
fit_Res <- nnetar(Res_Norm)
nfcast_Res <- forecast(fit_Res,h=255) #newdata =test1
fcastedvaluesExtracted_Res <- as.numeric(nfcast_Res$mean)
minvalRes <- min(Res);maxvalRes <- max(Res)
F_Res <- fcastedvaluesExtracted_Res*(maxvalRes-minvalRes)+minvalRes
#############################################################################
#Sum up all prediction = Final forecasting value
FinalForecast_emd_BPNN_Res <- ts(F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8+F_Res)
FinalForecast_emd_BPNN_Res
```

Figure 7.25:

```
FinalForecast_emd_BPNN <- ts(F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8)
FinalForecast_emd_BPNN
dd <- data.frame(test_D ,FinalForecast_emd_BPNN_Res)
col_headings <- c("Actual Price","Forecast Price")
names(dd) <- col_headings;attach(dd)
percentage_err=((dd$`Actual Price`-dd$`Forecast Price`)/(dd$`Actual Price`))
percentage_err
mean(percentage_err)
MAE(dd$`Actual Price`,dd$`Forecast Price`)
MSE(dd$`Actual Price`,dd$`Forecast Price`)
MAPE(dd$`Actual Price`,dd$`Forecast Price`)

mae_emd_bpnn  <- abs(dd$`Actual Price`-dd$`Forecast Price`)
mse_emd_bpnn<- (dd$`Actual Price`-dd$`Forecast Price`)^2
mape_emd_bpnn <- (abs((dd$`Actual Price`-dd$`Forecast Price`)/dd$`Actual Price`))*100
A<-MAE
plot.ts(dd$`Actual Price`)
lines(dd$`Forecast Price`,col="red")
lines(dd$`Forecast Price`, col="red")
FinalForecast_emd_BPNN_Res <- ts(FinalForecast_emd_BPNN_Res)
test_D <- ts(test_D)
plot.ts(FinalForecast_emd_BPNN_Res,col="blue", main="The three
components",xlab="Time/monthly",ylab=" " )
lines(test_D,col="red")
legend("bottom",col=c("blue", "red"),lwd=2,
     legend=c("Forecasted", "Test"))
plot.ts(train_D)
plot.ts(`Actual Price`, col="red")
lines(train_D)
######################################################################
#2_CRUDE_VMD_PBNN
library(DescTools)
library(vmd)
dd_vmd <- vmd(train_D, alpha =100,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
dd_vmd <- as.data.frame(dd_vmd)
###M1 (Residue)
M1 <- dd_vmd$M1
normalize <- function(M1){return((M1-min(M1))/(max(M1)-min(M1)))}
M1_Norm <- normalize(M1)
fit.M1 <- nnetar(M1_Norm)
fcast.M1 <- forecast(fit.M1,h=255) #newdata =test1
fcastExtracted <- as.numeric(fcast.M1$mean)
fcastExtracted <- as.data.frame(fcastExtracted)
minval <- min(M1);maxval <- max(M1)
finalfcast.M1 <- fcastExtracted*(maxval-minval)+minval
```

Figure 7.26:

```
#######M2#########################################################################
M2 <- dd_vmd$M2
normalize2 <- function(M2){return((M2-min(M2))/(max(M2)-min(M2)))}
M2_Norm <- normalize(M2)
fit.M2 <- nnetar(M2_Norm)
fcast.M2 <- forecast(fit.M2,h=255) #newdata =test1
fcastExtracted2 <- as.numeric(fcast.M2$mean)
fcastExtracted2 <- as.data.frame(fcastExtracted2)
minval2 <- min(M2);maxval2 <- max(M2)
finalfcast.M2 <- fcastExtracted2*(maxval2-minval2)+minval2
#######M2#########################################################################
M3 <- dd_vmd$M3
normalize3 <- function(M3){return((M3-min(M3))/(max(M3)-min(M3)))};M3_Norm <-
normalize(M3)
fit.M3 <- nnetar(M3_Norm)
fcast.M3 <- forecast(fit.M3,h=255) #newdata =test1
fcastExtracted3 <- as.numeric(fcast.M3$mean);fcastExtracted3 <-
as.data.frame(fcastExtracted3)
minval3 <- min(M3);maxval3 <- max(M3)
finalfcast.M3 <- fcastExtracted3*(maxval3-minval3)+minval3
#######M2#########################################################################
M4 <- dd_vmd$M4
normalize4 <- function(M4){return((M4-min(M4))/(max(M4)-min(M4)))};M4_Norm <-
normalize(M4)
fit.M4 <- nnetar(M4_Norm)
fcast.M4 <- forecast(fit.M4,h=255) #newdata =test1
fcastExtracted4 <- as.numeric(fcast.M4$mean);fcastExtracted4 <-
as.data.frame(fcastExtracted4)
minval4 <- min(M4);maxval4 <- max(M4)
finalfcast.M4 <- fcastExtracted4*(maxval4-minval4)+minval4
#######M5#########################################################################
M5 <- dd_vmd$M5
normalize5 <- function(M5){return((M5-min(M5))/(max(M5)-min(M5)))};M5_Norm <-
normalize(M5)
fit.M5 <- nnetar(M5_Norm)
fcast.M5 <- forecast(fit.M5,h=255) #newdata =test1
fcastExtracted5 <- as.numeric(fcast.M5$mean);fcastExtracted5 <-
as.data.frame(fcastExtracted5)
minval5 <- min(M5);maxval5 <- max(M5)
finalfcast.M5 <- fcastExtracted5*(maxval5-minval5)+minval5
#######M6#########################################################################
M6 <- dd_vmd$M6
normalize6 <- function(M6){return((M6-min(M6))/(max(M6)-min(M6)))};M6_Norm <-
normalize(M6)
fit.M6 <- nnetar(M6_Norm)
fcast.M6 <- forecast(fit.M6,h=255) #newdata =test1
```

Figure 7.27:

```
fcastExtracted6 <- as.numeric(fcast.M6$mean);fcastExtracted6 <-
as.data.frame(fcastExtracted6)
minval6 <- min(M6);maxval6 <- max(M6)
finalfcast.M6 <- fcastExtracted6*(maxval6-minval6)+minval6
#######M7#############################################################
M7 <- dd_vmd$M7
normalize7 <- function(M7){return((M7-min(M7))/(max(M7)-min(M7)))};M7_Norm <-
normalize(M7)
fit.M7 <- nnetar(M7_Norm)
fcast.M7 <- forecast(fit.M7,h=255) #newdata =test1
fcastExtracted7 <- as.numeric(fcast.M7$mean);fcastExtracted7 <-
as.data.frame(fcastExtracted7)
minval7 <- min(M7);maxval7 <- max(M7)
finalfcast.M7 <- fcastExtracted7*(maxval7-minval7)+minval7
#######M8#############################################################
M8 <- dd_vmd$M8
normalize8 <- function(M8){return((M8-min(M8))/(max(M8)-min(M8)))};M8_Norm <-
normalize(M8)
fit.M8 <- nnetar(M8_Norm)
fcast.M8 <- forecast(fit.M8,h=255) #newdata =test1
fcastExtracted8 <- as.numeric(fcast.M8$mean);fcastExtracted8 <-
as.data.frame(fcastExtracted8)
minval8 <- min(M8);maxval8 <- max(M8)
finalfcast.M8 <- fcastExtracted8*(maxval8-minval8)+minval8
#####################################################################################
FinalForecast_vmd_BPNN <- finalfcast.M1
+finalfcast.M2+finalfcast.M3+finalfcast.M4+finalfcast.M5+finalfcast.M6+finalfcast.M7+finalf
cast.M8
vd <- data.frame(test_D ,FinalForecast_vmd_BPNN)
col_headings <- c("Actual Price","Forecast Price")
names(vd) <- col_headings;attach(vd)
MAE(vd$`Actual Price`,vd$`Forecast Price`)
RMSE(vd$`Actual Price`,vd$`Forecast Price`)
MAPE(vd$`Actual Price`,vd$`Forecast Price`)

mae_vmd_bpnn  <- abs(vd$`Actual Price`-vd$`Forecast Price`)
mse_vmd_bpnn<- (vd$`Actual Price`-vd$`Forecast Price`)^2
mape_vmd_bpnn <- (abs((vd$`Actual Price`-vd$`Forecast Price`)/vd$`Actual Price`))*100

plot.ts(test_D,col="red", main="The three components",xlab="Time/monthly",ylab="Corn
Future Price ")
lines(FinalForecast_vmd_BPNN, col="blue")
legend("topright",col=c("blue", "red"),lwd=2,
    legend=c("Forecasted", "Test"))
A<-MAE
```

Figure 7.28:

```
dm.test(dd$'Actual Price'-dd$'Forecast Price',vd$'Actual Price'-vd$'Forecast Price') #two
models have different accuracy


###Comparing forecast
#null hypothesis is that the two methods have the same forecast accuracy
library(forecast)
dm.test(dd$'Actual Price'-dd$'Forecast Price',vd$'Actual Price'-vd$'Forecast Price') #two
models have different accuracy


dm.test(dd$'Actual Price'-dd$'Forecast Price',vd$'Actual Price'-vd$'Forecast Price',
alternative = "greater")
#vd has greater accuracy

dm.test(dd$'Actual Price'-dd$'Forecast Price',vd$'Actual Price'-vd$'Forecast Price',
alternative = "less")
#########################################################
###3_EMD_ARIMA
library(MASS)
library(tseries)
library(forecast)

df_IMFs<- emd(train_D, tt=NULL, tol=sd(train_D)*0.1^2,boundary="periodic")
###################Normalisation of trained data############################
IMF1 <- as.data.frame(df_IMFs$imf[,1]);Res <- as.data.frame(df_IMFs$residue)
ts.plot(IMF1)
IMF1_Norm <- ts((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))

###################Time series, Fitting and Forecasting ########################
fit_1 <- nnetar(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
fcastedvaluesExtracted
minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- ts(fcastedvaluesExtracted*(maxval-minval)+minval)
ts.plot(test_D,col="red")
plot(F_1)
IMF1 <- as.data.frame(df_IMFs$imf[,1])
normalize <- function(IMF1){return((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))}
IMF1_Norm <- ts(normalize(IMF1))
###################Time series, Fitting and Forecasting ########################
fit_1 <- auto.arima(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- fcastedvaluesExtracted*(maxval-minval)+minval
```

Figure 7.29:

```
#########################################################################
IMF2 <- as.data.frame(df_IMFs$imf[,2])
normalize2 <- function(IMF2){return((IMF2-min(IMF2))/(max(IMF2)-min(IMF2)))}
IMF2_Norm <- ts(normalize2(IMF2))
###################Time series, Fitting and Forecasting ##########################
fit_2 <- auto.arima(IMF2_Norm)
nfcast_2 <- forecast(fit_2,h=255) #newdata =test1
fcastedvaluesExtracted2 <- as.numeric(nfcast_2$mean)
minval2 <- min(IMF2);maxval2 <- max(IMF2)
F_2 <- fcastedvaluesExtracted2*(maxval2-minval2)+minval2
#########################################################################
IMF3 <- as.data.frame(df_IMFs$imf[,3])
normalize3 <- function(IMF3){return((IMF3-min(IMF3))/(max(IMF3)-min(IMF3)))}
IMF3_Norm <- ts(normalize3(IMF3))
###################Time series, Fitting and Forecasting ########################
fit_3 <- auto.arima(IMF3_Norm)
nfcast_3 <- forecast(fit_3,h=255) #newdata =test1
fcastedvaluesExtracted3 <- as.numeric(nfcast_3$mean)
minval3 <- min(IMF3);maxval3 <- max(IMF3)
F_3 <- fcastedvaluesExtracted3*(maxval3-minval3)+minval3
#########################################################################
IMF4 <- as.data.frame(df_IMFs$imf[,4])
normalize4 <- function(IMF4){return((IMF4-min(IMF4))/(max(IMF4)-min(IMF4)))}
IMF4_Norm <- ts(normalize4(IMF4))
###################Time series, Fitting and Forecasting ######################
fit_4 <- auto.arima(IMF4_Norm)
nfcast_4 <- forecast(fit_4,h=255) #newdata =test1
fcastedvaluesExtracted4 <- as.numeric(nfcast_4$mean)
minval4 <- min(IMF4);maxval4 <- max(IMF4)
F_4 <- fcastedvaluesExtracted4*(maxval4-minval4)+minval4
#########################################################################
IMF5 <- as.data.frame(df_IMFs$imf[,5])
normalize5 <- function(IMF5){return((IMF5-min(IMF5))/(max(IMF5)-min(IMF5)))}
IMF5_Norm <- ts(normalize5(IMF5))
###################Time series, Fitting and Forecasting ######################
fit_5 <- auto.arima(IMF5_Norm)
nfcast_5 <- forecast(fit_5,h=255) #newdata =test1
fcastedvaluesExtracted5 <- as.numeric(nfcast_5$mean)
minval5 <- min(IMF5);maxval5 <- max(IMF5)
F_5 <- fcastedvaluesExtracted5*(maxval5-minval5)+minval5
#########################################################################
IMF6 <- as.data.frame(df_IMFs$imf[,6])
normalize6 <- function(IMF6){return((IMF6-min(IMF6))/(max(IMF6)-min(IMF6)))}
IMF6_Norm <- ts(normalize6(IMF6))
###################Time series, Fitting and Forecasting ######################
fit_6 <- auto.arima(IMF6_Norm)
nfcast_6 <- forecast(fit_6,h=255) #newdata =test1
```

Figure 7.30:

```
fcastedvaluesExtracted6 <- as.numeric(nfcast_6$mean)
minval6 <- min(IMF6);maxval6 <- max(IMF6)
F_6 <- fcastedvaluesExtracted6*(maxval6-minval6)+minval6
################################################################################
IMF7 <- as.data.frame(df_IMFs$imf[,7])
normalize7 <- function(IMF7){return((IMF7-min(IMF7))/(max(IMF7)-min(IMF7)))}
IMF7_Norm <- ts(normalize7(IMF7))
####################Time series, Fitting and Forecasting ############################
fit_7 <- auto.arima(IMF7_Norm)
nfcast_7 <- forecast(fit_7,h=255) #newdata =test1
fcastedvaluesExtracted7 <- as.numeric(nfcast_7$mean)
minval7 <- min(IMF7);maxval7 <- max(IMF7)
F_7 <- fcastedvaluesExtracted7*(maxval7-minval7)+minval7
################################################################################
IMF8 <- as.data.frame(df_IMFs$imf[,8])
normalize8 <- function(IMF8){return((IMF8-min(IMF8))/(max(IMF8)-min(IMF8)))}
IMF8_Norm <- ts(normalize8(IMF8))
####################Time series, Fitting and Forecasting ############################
fit_8 <- auto.arima(IMF8_Norm)
nfcast_8 <- forecast(fit_8,h=255) #newdata =test1
fcastedvaluesExtracted8 <- as.numeric(nfcast_8$mean)
minval8 <- min(IMF8);maxval8 <- max(IMF8)
F_8 <- fcastedvaluesExtracted8*(maxval8-minval8)+minval8
####################Time series, Fitting and Forecasting#######################
Res <- as.data.frame(df_IMFs$residue)
normalizeRes <- function(Res){return((Res-min(Res))/(max(Res)-min(Res)))}
Res_Norm <- ts(normalize(Res))
fit_Res <- auto.arima(Res_Norm)
nfcast_Res <- forecast(fit_Res,h=255) #newdata =test1
fcastedvaluesExtractedRes <- as.numeric(nfcast_Res$mean)
minvalRes <- min(Res);maxvalRes <- max(Res)
F_Res <- fcastedvaluesExtractedRes*(maxvalRes-minvalRes)+minvalRes
################################################################################
FinalForecast.emd.ARIMA <- F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8+F_Res
FinalForecast.emd.ARIMA<- ts(FinalForecast.emd.ARIMA)
ead <- data.frame(test_D ,FinalForecast.emd.ARIMA)
col_headings <- c("Actual Price","Forecast Price")
names(ead) <- col_headings;attach(ead)
MAE(ead$`Actual Price`,ead$`Forecast Price`)
RMSE(ead$`Actual Price`,ead$`Forecast Price`)
MAPE(ead$`Actual Price`,ead$`Forecast Price`)

mae_emd_arima <- abs(ead$`Actual Price`-ead$`Forecast Price`)
mse_emd_arima<- (ead$`Actual Price`-ead$`Forecast Price`)^2
mape_emd_arima <- (abs((ead$`Actual Price`-ead$`Forecast Price`)/ead$`Actual
Price`))*100
```

Figure 7.31:

```
plot(FinalForecast.emd.ARIMA)
FinalForecast_vmd_BPNN
plot.ts(`Actual Price`)

###4_VMD_ARIMA
###M1 (Residue)
M1 <- dd_vmd$M1
normalize <- function(M1){return((M1-min(M1))/(max(M1)-min(M1)))}
M1_Norm <- normalize(M1)
fit.M1 <- auto.arima(M1_Norm)
fcast.M1 <- forecast(fit.M1,h=255) #newdata =test1
fcastExtracted <- as.numeric(fcast.M1$mean);fcastExtracted <-
as.data.frame(fcastExtracted)
minval <- min(M1);maxval <- max(M1)
finalfcast.M1 <- as.numeric(fcast.M1$mean)
#finalfcast.M1 <- fcastExtracted*(maxval-minval)+minval
#######M2#############################################################################
M2 <- dd_vmd$M2
normalize2 <- function(M2){return((M2-min(M2))/(max(M2)-min(M2)))};M2_Norm <-
normalize(M2)
fit.M2 <- auto.arima(M2_Norm)
fcast.M2 <- forecast(fit.M2,h=255) #newdata =test1
fcastExtracted2 <- as.numeric(fcast.M2$mean)
fcastExtracted2 <- as.data.frame(fcastExtracted2)
minval2 <- min(M2);maxval2 <- max(M2)
finalfcast.M2 <- as.numeric(fcast.M2$mean)
#finalfcast.M2 <- fcastExtracted2*(maxval2-minval2)+minval2
#######M2#############################################################################
M3 <- dd_vmd$M3
normalize3 <- function(M3){return((M3-min(M3))/(max(M3)-min(M3)))};M3_Norm <-
normalize(M3)
fit.M3 <- auto.arima(M3_Norm)
fcast.M3 <- forecast(fit.M3,h=255) #newdata =test1
fcastExtracted3 <- as.numeric(fcast.M3$mean);fcastExtracted3 <-
as.data.frame(fcastExtracted3)
minval3 <- min(M3);maxval3 <- max(M3)
finalfcast.M3 <- as.numeric(fcast.M3$mean)
#finalfcast.M3 <- fcastExtracted3*(maxval3-minval3)+minval3
#######M2#############################################################################
M4 <- dd_vmd$M4
normalize4 <- function(M4){return((M4-min(M4))/(max(M4)-min(M4)))};M4_Norm <-
normalize(M4)
fit.M4 <- auto.arima(M4_Norm)
fcast.M4 <- forecast(fit.M4,h=255) #newdata =test1
fcastExtracted4 <- as.numeric(fcast.M4$mean);fcastExtracted4 <-
as.data.frame(fcastExtracted4)
minval4 <- min(M4);maxval4 <- max(M4)
```

Figure 7.32:

```
finalfcast.M4 <- as.numeric(fcast.M4$mean)
#finalfcast.M4 <- fcastExtracted4*(maxval4-minval4)+minval4
#######M5############################################################
M5 <- dd_vmd$M5
normalize5 <- function(M5){return((M5-min(M5))/(max(M5)-min(M5)))};M5_Norm <-
normalize(M5)
fit.M5 <- auto.arima(M5_Norm)
fcast.M5 <- forecast(fit.M5,h=255) #newdata =test1
fcastExtracted5 <- as.numeric(fcast.M5$mean);fcastExtracted5 <-
as.data.frame(fcastExtracted5)
minval5 <- min(M5);maxval5 <- max(M5)
finalfcast.M5 <- as.numeric(fcast.M5$mean)
#finalfcast.M5 <- fcastExtracted5*(maxval5-minval5)+minval5
#######M6#########################################################################
M6 <- dd_vmd$M6
normalize6 <- function(M6){return((M6-min(M6))/(max(M6)-min(M6)))};M6_Norm <-
normalize(M6)
fit.M6 <- auto.arima(M6_Norm)
fcast.M6 <- forecast(fit.M6,h=255) #newdata =test1
fcastExtracted6 <- as.numeric(fcast.M6$mean);fcastExtracted6 <-
as.data.frame(fcastExtracted6)
minval6 <- min(M6);maxval6 <- max(M6)
finalfcast.M6 <- as.numeric(fcast.M6$mean)
#finalfcast.M6 <- fcastExtracted6*(maxval6-minval6)+minval6
#######M7#########################################################################
M7 <- dd_vmd$M7
normalize7 <- function(M7){return((M7-min(M7))/(max(M7)-min(M7)))};M7_Norm <-
normalize(M7)
fit.M7 <- auto.arima(M7_Norm)
fcast.M7 <- forecast(fit.M7,h=255) #newdata =test1
fcastExtracted7 <- as.numeric(fcast.M7$mean);fcastExtracted7 <-
as.data.frame(fcastExtracted7)
minval7 <- min(M7);maxval7 <- max(M7)
finalfcast.M7 <- as.numeric(fcast.M7$mean)
#finalfcast.M7 <- fcastExtracted7*(maxval7-minval7)+minval7
#######M8#########################################################################
M8 <- dd_vmd$M8
normalize8 <- function(M8){return((M8-min(M8))/(max(M8)-min(M8)))};M8_Norm <-
normalize(M8)
fit.M8 <- auto.arima(M8_Norm)
fcast.M8 <- forecast(fit.M8,h=255) #newdata =test1
fcastExtracted8 <- as.numeric(fcast.M8$mean);fcastExtracted8 <-
as.data.frame(fcastExtracted8)
minval8 <- min(M8);maxval8 <- max(M8)
finalfcast.M8 <- as.numeric(fcast.M8$mean)
#finalfcast.M8 <- fcastExtracted8*(maxval8-minval8)+minval8
```

Figure 7.33:

```
#############################################################################
#FinalForecast_vmd_arima
FinalForecast_vmd_arima <-
finalfcast.M1+finalfcast.M2+finalfcast.M3+finalfcast.M4+finalfcast.M5+finalfcast.M6+finalfc
ast.M7+finalfcast.M8
vad <- data.frame(test_D ,FinalForecast_vmd_arima)
col_headings <- c("Actual Price","Forecast Price")
names(vad) <- col_headings;attach(vad)
MAE(vad$`Actual Price`,vad$`Forecast Price`)
RMSE(vad$`Actual Price`,vad$`Forecast Price`)
MAPE(vad$`Actual Price`,vad$`Forecast Price`)

mae_vmd_arima <- abs(vad$`Actual Price`-vad$`Forecast Price`)
mse_vmd_arima<- (vad$`Actual Price`-vad$`Forecast Price`)^2
mape_vmd_arima <- (abs((vad$`Actual Price`-vad$`Forecast Price`)/vad$`Actual Price`))*100

FinalForecast_vmd_arima <- ts(FinalForecast_vmd_arima)
plot(FinalForecast_vmd_arima)
(FinalForecast_vmd_arima)
lines(test, col="red")
dm.test(ead$'Actual Price'-ead$'Forecast Price',vad$'Actual Price'-vad$'Forecast Price')
#two models have different accuracy

CP <- data.frame(CP$Value)
index <- sample(nrow(CP),round(0.8*nrow(CP)))
train <- CP[index,]
test <- CP[-index,]
CP_Norm <- ts((train-min(train))/(max(train)-min(train)))
normalizeCP<- function(train){return((train-min(train))/(max(train)-min(train)))}
CP_Norm <- normalize(train)
fit_og <- nnetar(CP_Norm)
nfcast_og <- forecast(fit_og,h=255) #newdata =test1
FVE <- as.numeric(nfcast_og$mean)
F1<-FVE
minval <- min(train);maxval <- max(train)
F1 <- FVE*(maxval-minval)+minval
ts.plot(test,col="red")
plot(F1)

ff <- data.frame(test ,F1)
col_headings <- c("Actual Price","Forecast Price")
names(ff) <- col_headings;attach(ff)
MAE(ff$`Actual Price`,ff$`Forecast Price`)
RMSE(ff$`Actual Price`,ff$`Forecast Price`)
MAPE(ff$`Actual Price`,ff$`Forecast Price`)

mae_bpnn <- abs(ff$`Actual Price`-ff$`Forecast Price`)
```

Figure 7.34:

```
mse_bpnn<- (ff$`Actual Price`-ff$`Forecast Price`)^2
mape_bpnn <- (abs((ff$`Actual Price`-ff$`Forecast Price`)/ff$`Actual Price`))*100

CP2<-CP
CP2
index <- sample(nrow(CP2),round(0.8*nrow(CP2)))
train <- CP2[index,]
test <- CP2[-index,]
test
CP2_Norm <- ts(2*(train-min(train))/(max(train)-min(train))-1)
max(CP2_Norm)
min(CP2_Norm)
fit_O <- auto.arima(CP2_Norm)
nfcast_O <- forecast(fit_O,h=255) #newdata =test1
plot(nfcast_O)
as.data.frame(nfcast_O)
FVEO <- ts(as.numeric(nfcast_O$mean))
plot(FVEO)
minval <- min(train);maxval <- max(train)
FO <- ts(FVEO*(maxval-minval)+minval)
FO
ddO <- data.frame(test ,FO)
col_headings <- c("Actual Price","Forecast Price")
names(ddO) <- col_headings;attach(ddO)
MAE(ddO$`Actual Price`,ddO$`Forecast Price`)
RMSE(ddO$`Actual Price`,ddO$`Forecast Price`)
MAPE(ddO$`Actual Price`,ddO$`Forecast Price`)

mae_arima <- abs(ddO$`Actual Price`-ddO$`Forecast Price`)
mse_arima<- (ddO$`Actual Price`-ddO$`Forecast Price`)^2
mape_arima <- (abs((ddO$`Actual Price`-ddO$`Forecast Price`)/ddO$`Actual Price`))*100
#####################################################################
#autoplot(FinalForecast_emd_BPNN)
#plot(FinalForecast_emd_BPNN)
#lines(FinalForecast_emd_BPNN_Res, col="violet")
#lines(FinalForecast_vmd_arima, col="red")
#lines(FinalForecast.emd.ARIMA, col="green")
#lines(FinalForecast_vmd_BPNN, col="blue")
#lines(F1,col="yellow")
#lines(FVEO)
#lines(test)
test_norm <- (test-min(test))/(max(test)-min(test))
ts.test <- xts(test$Value, order.by = as.POSIXct(test$Date, "UTC"))
plot(ts.test)
lines(FinalForecast_vmd_arima)
```

Figure 7.35:

```
autoplot(ts(test), series = "Actual")+
autolayer(FinalForecast.emd.ARIMA, series = "EMD-ARIMA")+
 autolayer(FinalForecast_emd_BPNN, series = "EMD-BPNN")+
  autolayer(ts(FinalForecast_vmd_arima), series = "VMD-ARIMA")+
   autolayer(ts(FinalForecast_vmd_BPNN), series = "VMD-BPNN")+
    autolayer(ts(F1), series = "BPNN")+
     autolayer(ts(FVEO), series = "ARIMA")

plot(FinalForecast.emd.ARIMA)
lines(test_norm)
plot()

plot(FinalForecast_emd_BPNN_Res, col="black",xlab="Time/daily",ylab="Crude oil Future
Price " )
lines(FinalForecast_vmd_arima, col="red")
lines(FinalForecast.emd.ARIMA, col="green")
lines(FinalForecast_vmd_BPNN, col="blue")
lines(F1,col="yellow")
lines(FVEO, col="brown")
lines(`Actual Price`, col="purple")
legend("topright",col=c("black", "red", "green", "blue", "yellow", "brown", "purple"),lwd=2,
       legend=c("EMD-BPNN", "VMD-ARIMA", "EMD-ARIMA", "EMD-BPNN", "BPNN",
"ARIMA", "Actual"))
###################### Dm Test ##############
#null hypothesis is that the two methods have the same forecast accuracy
library(forecast)
#################################################################################
#Comparing forecasts accuracy MAE
dm.test(mae_emd_bpnn,mae_vmd_bpnn)
dm.test(mae_emd_bpnn,mae_emd_arima)
dm.test(mae_emd_bpnn,mae_vmd_arima)
dm.test(mae_emd_bpnn,mae_bpnn)
dm.test(mae_emd_bpnn,mae_arima)

dm.test(mae_vmd_bpnn,mae_emd_arima)
dm.test(mae_vmd_bpnn,mae_vmd_arima)
dm.test(mae_vmd_bpnn,mae_bpnn)
dm.test(mae_vmd_bpnn,mae_arima)

dm.test(mae_emd_arima,mae_vmd_arima)
dm.test(mae_emd_arima,mae_arima)
dm.test(mae_emd_arima,mae_bpnn)

dm.test(mae_vmd_arima,mae_arima)
dm.test(mae_vmd_arima,mae_bpnn)

dm.test(mae_arima,mae_bpnn)
```

Figure 7.36:

```
##############################################################################
#Comparing forecasts accuracy RMSE
dm.test(mse_emd_bpnn,mse_vmd_bpnn)
dm.test(mse_emd_bpnn,mse_emd_arima)
dm.test(mse_emd_bpnn,mse_vmd_arima)
dm.test(mse_emd_bpnn,mse_bpnn)
dm.test(mse_emd_bpnn,mse_arima)

dm.test(mse_vmd_bpnn,mse_emd_arima)
dm.test(mse_vmd_bpnn,mse_vmd_arima)
dm.test(mse_vmd_bpnn,mse_bpnn)
dm.test(mse_vmd_bpnn,mse_arima)

dm.test(mse_emd_arima,mse_vmd_arima)
dm.test(mse_emd_arima,mse_arima)
dm.test(mse_emd_arima,mse_bpnn)

dm.test(mse_vmd_arima,mse_arima)
dm.test(mse_vmd_arima,mse_bpnn)

dm.test(mse_arima,mse_bpnn)
##############################################################################
#Comparing forecasts accuracy MAPE
dm.test(mape_emd_bpnn,mape_vmd_bpnn)
dm.test(mape_emd_bpnn,mape_emd_arima)
dm.test(mape_emd_bpnn,mape_vmd_arima)
dm.test(mape_emd_bpnn,mape_bpnn)
dm.test(mape_emd_bpnn,mape_arima)

dm.test(mape_vmd_bpnn,mape_emd_arima)
dm.test(mape_vmd_bpnn,mape_vmd_arima)
dm.test(mape_vmd_bpnn,mape_bpnn)
dm.test(mape_vmd_bpnn,mape_arima)

dm.test(mape_emd_arima,mape_vmd_arima)
dm.test(mape_emd_arima,mape_arima)
dm.test(mape_emd_arima,mape_bpnn)

dm.test(mape_vmd_arima,mape_arima)
dm.test(mape_vmd_arima,mape_bpnn)

dm.test(mape_arima,mape_bpnn)
```

Figure 7.37:

```
########################## FORECASTING OF GOLD PRICE SERIES ################
###################### Data partition# ##############################
index <- sample(nrow(CP),round(0.8*nrow(CP)))
train <- CP[index,]
nrow(train)
test <- CP[-index,]
nrow(test)
train_D <- train$Value
test_D <- test$Value

normalizetest <- function(test){return((test-min(test))/(max(test)-min(test)))}
test_norm <- normalize(test)
###################Decomposition#########################################
df_IMFs<- emd(train_D, tt=NULL, tol=sd(train_D)*0.1^2,boundary="periodic")
###################Normalisation of trained data#################################
IMF1 <- as.data.frame(df_IMFs$imf[,1]);Res <- as.data.frame(df_IMFs$residue)
ts.plot(IMF1)
IMF1_Norm <- ts((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))
###################Time series, Fitting and Forecasting ##################
fit_1 <- nnetar(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
fcastedvaluesExtracted

minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- ts(fcastedvaluesExtracted*(maxval-minval)+minval)
ts.plot(test_D,col="red")
plot(F_1)
#############################################################################
IMF2 <- as.data.frame(df_IMFs$imf[,2])
normalize2 <- function(IMF2){return((IMF2-min(IMF2))/(max(IMF2)-min(IMF2)))}
IMF2_Norm <- ts(normalize2(IMF2))
###################Time series, Fitting and Forecasting ######################
fit_2 <- nnetar(IMF2_Norm)
nfcast_2 <- forecast(fit_2,h=255) #newdata =test1

fcastedvaluesExtracted2 <- as.numeric(nfcast_2$mean)
minval2 <- min(IMF2);maxval2 <- max(IMF2)
F_2 <- fcastedvaluesExtracted2*(maxval2-minval2)+minval2
#############################################################################
IMF3 <- as.data.frame(df_IMFs$imf[,3])
normalize3 <- function(IMF3){return((IMF3-min(IMF3))/(max(IMF3)-min(IMF3)))}
IMF3_Norm <- ts(normalize3(IMF3))
###################Time series, Fitting and Forecasting ######################
fit_3 <- nnetar(IMF3_Norm)
nfcast_3 <- forecast(fit_3,h=255) #newdata =test1
```

Figure 7.38:

```
fcastedvaluesExtracted3 <- as.numeric(nfcast_3$mean)
minval3 <- min(IMF3);maxval3 <- max(IMF3)
F_3 <- fcastedvaluesExtracted3*(maxval3-minval3)+minval3
################################################################################
IMF4 <- as.data.frame(df_IMFs$imf[,4])
normalize4 <- function(IMF4){return((IMF4-min(IMF4))/(max(IMF4)-min(IMF4)))}
IMF4_Norm <- ts(normalize4(IMF4))
###################Time series, Fitting and Forecasting ########################
fit_4 <- nnetar(IMF4_Norm)
nfcast_4 <- forecast(fit_4,h=255) #newdata =test1
fcastedvaluesExtracted4 <- as.numeric(nfcast_4$mean)
minval4 <- min(IMF4);maxval4 <- max(IMF4)
F_4 <- fcastedvaluesExtracted4*(maxval4-minval4)+minval4
################################################################################
IMF5 <- as.data.frame(df_IMFs$imf[,5])
normalize5 <- function(IMF5){return((IMF5-min(IMF5))/(max(IMF5)-min(IMF5)))}
IMF5_Norm <- ts(normalize5(IMF5))
###################Time series, Fitting and Forecasting ########################
fit_5 <- nnetar(IMF5_Norm)
nfcast_5 <- forecast(fit_5,h=255) #newdata =test1
fcastedvaluesExtracted5 <- as.numeric(nfcast_5$mean)
minval5 <- min(IMF5);maxval5 <- max(IMF5)
F_5 <- fcastedvaluesExtracted5*(maxval5-minval5)+minval5
################################################################################
IMF6 <- as.data.frame(df_IMFs$imf[,6])
normalize6 <- function(IMF6){return((IMF6-min(IMF6))/(max(IMF6)-min(IMF6)))}
IMF6_Norm <- ts(normalize6(IMF6))
###################Time series, Fitting and Forecasting ########################
fit_6 <- nnetar(IMF6_Norm)
nfcast_6 <- forecast(fit_6,h=255) #newdata =test1
fcastedvaluesExtracted6 <- as.numeric(nfcast_6$mean)
minval6 <- min(IMF6);maxval6 <- max(IMF6)
F_6 <- fcastedvaluesExtracted6*(maxval6-minval6)+minval6
################################################################################
IMF7 <- as.data.frame(df_IMFs$imf[,7])
normalize7 <- function(IMF7){return((IMF7-min(IMF7))/(max(IMF7)-min(IMF7)))}
IMF7_Norm <- ts(normalize7(IMF7))
###################Time series, Fitting and Forecasting ########################
fit_7 <- nnetar(IMF7_Norm)
nfcast_7 <- forecast(fit_7,h=255) #newdata =test1
fcastedvaluesExtracted7 <- as.numeric(nfcast_7$mean)
minval7 <- min(IMF7);maxval7 <- max(IMF7)
F_7 <- fcastedvaluesExtracted7*(maxval7-minval7)+minval7
################################################################################
IMF8 <- as.data.frame(diff(df_IMFs$imf[,8]))
normalize8 <- function(IMF8){return((IMF8-min(IMF8))/(max(IMF8)-min(IMF8)))}
IMF8_Norm <- ts(normalize8(IMF8))
```

Figure 7.39:

```
###################Time series, Fitting and Forecasting ######################
fit_8 <- nnetar(IMF8_Norm)
nfcast_8 <- forecast(fit_8,h=255) #newdata =test1
fcastedvaluesExtracted8 <- as.numeric(nfcast_8$mean)
minval8 <- min(IMF8);maxval8 <- max(IMF8)
F_8 <- fcastedvaluesExtracted8*(maxval8-minval8)+minval8
#############################################################################
Res <- as.data.frame(diff(df_IMFs$residue))
normalizeRes <- function(Res){return((Res-min(Res))/(max(Res)-min(Res)))}
Res_Norm <- ts(normalizeRes(Res))
###################Time series, Fitting and Forecasting ######################
fit_Res <- nnetar(Res_Norm)
nfcast_Res <- forecast(fit_Res,h=255) #newdata =test1
fcastedvaluesExtracted_Res <- as.numeric(nfcast_Res$mean)
minvalRes <- min(Res);maxvalRes <- max(Res)
F_Res <- fcastedvaluesExtracted_Res*(maxvalRes-minvalRes)+minvalRes
#############################################################################
#Sum up all prediction = Final forecasting value
FinalForecast_emd_BPNN_Res <- ts(F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8+F_Res)
FinalForecast_emd_BPNN_Res
FinalForecast_emd_BPNN <- ts(F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8)
FinalForecast_emd_BPNN
dd <- data.frame(test_D ,FinalForecast_emd_BPNN_Res)
col_headings <- c("Actual Price","Forecast Price")
names(dd) <- col_headings;attach(dd)
percentage_err=((dd$`Actual Price`-dd$`Forecast Price`)/(dd$`Actual Price`))
percentage_err
mean(percentage_err)
MAE(dd$`Actual Price`,dd$`Forecast Price`)
MSE(dd$`Actual Price`,dd$`Forecast Price`)
MAPE(dd$`Actual Price`,dd$`Forecast Price`)

mae_emd_bpnn  <- abs(dd$`Actual Price`-dd$`Forecast Price`)
mse_emd_bpnn<- (dd$`Actual Price`-dd$`Forecast Price`)^2
mape_emd_bpnn <- (abs((dd$`Actual Price`-dd$`Forecast Price`)/dd$`Actual Price`))*100

A<-MAE
plot.ts(dd$`Actual Price`)
lines(dd$`Forecast Price`,col="red")
lines(dd$`Forecast Price`, col="red")
FinalForecast_emd_BPNN_Res <- ts(FinalForecast_emd_BPNN_Res)
test_D <- ts(test_D)
plot.ts(FinalForecast_emd_BPNN_Res,col="blue", main="The three
components",xlab="Time/monthly",ylab=" " )
lines(test_D,col="red")
legend("bottom",col=c("blue", "red"),lwd=2,
    legend=c("Forecasted", "Test"))
```

Figure 7.40:

```
plot.ts(train_D)
plot.ts(`Actual Price`, col="red")
lines(train_D)
#####################################################################
#2_CRUDE_VMD_PBNN
library(DescTools)
library(vmd)
dd_vmd <- vmd(train_D, alpha =100,tau=0, DC=FALSE,init = 0, tol = 1e-
3,K=8,orderModes=TRUE)
dd_vmd <- as.data.frame(dd_vmd)

###M1 (Residue)
M1 <- dd_vmd$M1
normalize <- function(M1){return((M1-min(M1))/(max(M1)-min(M1)))}
M1_Norm <- normalize(M1)
fit.M1 <- nnetar(M1_Norm)
fcast.M1 <- forecast(fit.M1,h=255) #newdata =test1
fcastExtracted <- as.numeric(fcast.M1$mean)
fcastExtracted <- as.data.frame(fcastExtracted)
minval <- min(M1);maxval <- max(M1)
finalfcast.M1 <- fcastExtracted*(maxval-minval)+minval
#######M2#############################################################################
M2 <- dd_vmd$M2
normalize2 <- function(M2){return((M2-min(M2))/(max(M2)-min(M2)))}
M2_Norm <- normalize(M2)
fit.M2 <- nnetar(M2_Norm)
fcast.M2 <- forecast(fit.M2,h=255) #newdata =test1
fcastExtracted2 <- as.numeric(fcast.M2$mean)
fcastExtracted2 <- as.data.frame(fcastExtracted2)
minval2 <- min(M2);maxval2 <- max(M2)
finalfcast.M2 <- fcastExtracted2*(maxval2-minval2)+minval2
#######M2#############################################################
M3 <- dd_vmd$M3
normalize3 <- function(M3){return((M3-min(M3))/(max(M3)-min(M3)))};M3_Norm <-
normalize(M3)
fit.M3 <- nnetar(M3_Norm)
fcast.M3 <- forecast(fit.M3,h=255) #newdata =test1
fcastExtracted3 <- as.numeric(fcast.M3$mean);fcastExtracted3 <-
as.data.frame(fcastExtracted3)
minval3 <- min(M3);maxval3 <- max(M3)
finalfcast.M3 <- fcastExtracted3*(maxval3-minval3)+minval3
#######M2#############################################################################
M4 <- dd_vmd$M4
normalize4 <- function(M4){return((M4-min(M4))/(max(M4)-min(M4)))};M4_Norm <-
normalize(M4)
fit.M4 <- nnetar(M4_Norm)
fcast.M4 <- forecast(fit.M4,h=255) #newdata =test1
```

Figure 7.41:

```
fcastExtracted4 <- as.numeric(fcast.M4$mean);fcastExtracted4 <-
as.data.frame(fcastExtracted4)
minval4 <- min(M4);maxval4 <- max(M4)
finalfcast.M4 <- fcastExtracted4*(maxval4-minval4)+minval4
#######M5#############################################################
M5 <- dd_vmd$M5
normalize5 <- function(M5){return((M5-min(M5))/(max(M5)-min(M5)))};M5_Norm <-
normalize(M5)
fit.M5 <- nnetar(M5_Norm)
fcast.M5 <- forecast(fit.M5,h=255) #newdata =test1
fcastExtracted5 <- as.numeric(fcast.M5$mean);fcastExtracted5 <-
as.data.frame(fcastExtracted5)
minval5 <- min(M5);maxval5 <- max(M5)
finalfcast.M5 <- fcastExtracted5*(maxval5-minval5)+minval5
#######M6#############################################################
M6 <- dd_vmd$M6
normalize6 <- function(M6){return((M6-min(M6))/(max(M6)-min(M6)))};M6_Norm <-
normalize(M6)
fit.M6 <- nnetar(M6_Norm)
fcast.M6 <- forecast(fit.M6,h=255) #newdata =test1
fcastExtracted6 <- as.numeric(fcast.M6$mean);fcastExtracted6 <-
as.data.frame(fcastExtracted6)
minval6 <- min(M6);maxval6 <- max(M6)
finalfcast.M6 <- fcastExtracted6*(maxval6-minval6)+minval6
#######M7#############################################################
M7 <- dd_vmd$M7
normalize7 <- function(M7){return((M7-min(M7))/(max(M7)-min(M7)))};M7_Norm <-
normalize(M7)
fit.M7 <- nnetar(M7_Norm)
fcast.M7 <- forecast(fit.M7,h=255) #newdata =test1
fcastExtracted7 <- as.numeric(fcast.M7$mean);fcastExtracted7 <-
as.data.frame(fcastExtracted7)
minval7 <- min(M7);maxval7 <- max(M7)
finalfcast.M7 <- fcastExtracted7*(maxval7-minval7)+minval7
#######M8#############################################################
M8 <- dd_vmd$M8
normalize8 <- function(M8){return((M8-min(M8))/(max(M8)-min(M8)))};M8_Norm <-
normalize(M8)
fit.M8 <- nnetar(M8_Norm)
fcast.M8 <- forecast(fit.M8,h=255) #newdata =test1
fcastExtracted8 <- as.numeric(fcast.M8$mean);fcastExtracted8 <-
as.data.frame(fcastExtracted8)
minval8 <- min(M8);maxval8 <- max(M8)
finalfcast.M8 <- fcastExtracted8*(maxval8-minval8)+minval8
```

Figure 7.42:

```
###################Normalisation of trained data##########################
IMF1 <- as.data.frame(df_IMFs$imf[,1]);Res <- as.data.frame(df_IMFs$residue)
ts.plot(IMF1)
IMF1_Norm <- ts((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))

###################Time series, Fitting and Forecasting ######################
fit_1 <- nnetar(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
fcastedvaluesExtracted
minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- ts(fcastedvaluesExtracted*(maxval-minval)+minval)
ts.plot(test_D,col="red")
plot(F_1)
IMF1 <- as.data.frame(df_IMFs$imf[,1])
normalize <- function(IMF1){return((IMF1-min(IMF1))/(max(IMF1)-min(IMF1)))}
IMF1_Norm <- ts(normalize(IMF1))
###################Time series, Fitting and Forecasting ######################
fit_1 <- auto.arima(IMF1_Norm)
nfcast_1 <- forecast(fit_1,h=255) #newdata =test1
fcastedvaluesExtracted <- as.numeric(nfcast_1$mean)
minval <- min(IMF1);maxval <- max(IMF1)
F_1 <- fcastedvaluesExtracted*(maxval-minval)+minval
#############################################################################
IMF2 <- as.data.frame(df_IMFs$imf[,2])
normalize2 <- function(IMF2){return((IMF2-min(IMF2))/(max(IMF2)-min(IMF2)))}
IMF2_Norm <- ts(normalize2(IMF2))
###################Time series, Fitting and Forecasting ##########################
fit_2 <- auto.arima(IMF2_Norm)
nfcast_2 <- forecast(fit_2,h=255) #newdata =test1
fcastedvaluesExtracted2 <- as.numeric(nfcast_2$mean)
minval2 <- min(IMF2);maxval2 <- max(IMF2)
F_2 <- fcastedvaluesExtracted2*(maxval2-minval2)+minval2
#############################################################################
IMF3 <- as.data.frame(df_IMFs$imf[,3])
normalize3 <- function(IMF3){return((IMF3-min(IMF3))/(max(IMF3)-min(IMF3)))}
IMF3_Norm <- ts(normalize3(IMF3))
###################Time series, Fitting and Forecasting ########################
fit_3 <- auto.arima(IMF3_Norm)
nfcast_3 <- forecast(fit_3,h=255) #newdata =test1
fcastedvaluesExtracted3 <- as.numeric(nfcast_3$mean)
minval3 <- min(IMF3);maxval3 <- max(IMF3)
F_3 <- fcastedvaluesExtracted3*(maxval3-minval3)+minval3
#############################################################################
IMF4 <- as.data.frame(df_IMFs$imf[,4])
normalize4 <- function(IMF4){return((IMF4-min(IMF4))/(max(IMF4)-min(IMF4)))}
IMF4_Norm <- ts(normalize4(IMF4))
```

Figure 7.43:

```
###################Time series, Fitting and Forecasting ##########################
fit_4 <- auto.arima(IMF4_Norm)
nfcast_4 <- forecast(fit_4,h=255) #newdata =test1
fcastedvaluesExtracted4 <- as.numeric(nfcast_4$mean)
minval4 <- min(IMF4);maxval4 <- max(IMF4)
F_4 <- fcastedvaluesExtracted4*(maxval4-minval4)+minval4
#############################################################################
IMF5 <- as.data.frame(df_IMFs$imf[,5])
normalize5 <- function(IMF5){return((IMF5-min(IMF5))/(max(IMF5)-min(IMF5)))}
IMF5_Norm <- ts(normalize5(IMF5))
###################Time series, Fitting and Forecasting ##########################
fit_5 <- auto.arima(IMF5_Norm)
nfcast_5 <- forecast(fit_5,h=255) #newdata =test1
fcastedvaluesExtracted5 <- as.numeric(nfcast_5$mean)
minval5 <- min(IMF5);maxval5 <- max(IMF5)
F_5 <- fcastedvaluesExtracted5*(maxval5-minval5)+minval5
#############################################################################
IMF6 <- as.data.frame(df_IMFs$imf[,6])
normalize6 <- function(IMF6){return((IMF6-min(IMF6))/(max(IMF6)-min(IMF6)))}
IMF6_Norm <- ts(normalize6(IMF6))
###################Time series, Fitting and Forecasting ##########################
fit_6 <- auto.arima(IMF6_Norm)
nfcast_6 <- forecast(fit_6,h=255) #newdata =test1
fcastedvaluesExtracted6 <- as.numeric(nfcast_6$mean)
minval6 <- min(IMF6);maxval6 <- max(IMF6)
F_6 <- fcastedvaluesExtracted6*(maxval6-minval6)+minval6
#############################################################################
IMF7 <- as.data.frame(df_IMFs$imf[,7])
normalize7 <- function(IMF7){return((IMF7-min(IMF7))/(max(IMF7)-min(IMF7)))}
IMF7_Norm <- ts(normalize7(IMF7))
###################Time series, Fitting and Forecasting ##########################
fit_7 <- auto.arima(IMF7_Norm)
nfcast_7 <- forecast(fit_7,h=255) #newdata =test1
fcastedvaluesExtracted7 <- as.numeric(nfcast_7$mean)
minval7 <- min(IMF7);maxval7 <- max(IMF7)
F_7 <- fcastedvaluesExtracted7*(maxval7-minval7)+minval7
#############################################################################
IMF8 <- as.data.frame(df_IMFs$imf[,8])
normalize8 <- function(IMF8){return((IMF8-min(IMF8))/(max(IMF8)-min(IMF8)))}
IMF8_Norm <- ts(normalize8(IMF8))
###################Time series, Fitting and Forecasting ##########################
fit_8 <- auto.arima(IMF8_Norm)
nfcast_8 <- forecast(fit_8,h=255) #newdata =test1
fcastedvaluesExtracted8 <- as.numeric(nfcast_8$mean)
minval8 <- min(IMF8);maxval8 <- max(IMF8)
F_8 <- fcastedvaluesExtracted8*(maxval8-minval8)+minval8
```

Figure 7.44:

```
###################Time series, Fitting and Forecasting ##########################
Res <- as.data.frame(df_IMFs$residue)
normalizeRes <- function(Res){return((Res-min(Res))/(max(Res)-min(Res)))}
Res_Norm <- ts(normalize(Res))
fit_Res <- auto.arima(Res_Norm)
nfcast_Res <- forecast(fit_Res,h=255) #newdata =test1
fcastedvaluesExtractedRes <- as.numeric(nfcast_Res$mean)
minvalRes <- min(Res);maxvalRes <- max(Res)
F_Res <- fcastedvaluesExtractedRes*(maxvalRes-minvalRes)+minvalRes
#################################################################################
FinalForecast.emd.ARIMA <- F_1+F_2+F_3+F_4+F_5+F_6+F_7+F_8+F_Res
FinalForecast.emd.ARIMA<- ts(FinalForecast.emd.ARIMA)
ead <- data.frame(test_D ,FinalForecast.emd.ARIMA)
col_headings <- c("Actual Price","Forecast Price")
names(ead) <- col_headings;attach(ead)
MAE(ead$`Actual Price`,ead$`Forecast Price`)
RMSE(ead$`Actual Price`,ead$`Forecast Price`)
MAPE(ead$`Actual Price`,ead$`Forecast Price`)

mae_emd_arima <- abs(ead$`Actual Price`-ead$`Forecast Price`)
mse_emd_arima<- (ead$`Actual Price`-ead$`Forecast Price`)^2
mape_emd_arima <- (abs((ead$`Actual Price`-ead$`Forecast Price`)/ead$`Actual
Price`))*100

plot(FinalForecast.emd.ARIMA)
FinalForecast_vmd_BPNN
plot.ts(`Actual Price`)

###4_VMD_ARIMA
###M1 (Residue)
M1 <- dd_vmd$M1
normalize <- function(M1){return((M1-min(M1))/(max(M1)-min(M1)))}
M1_Norm <- normalize(M1)
fit.M1 <- auto.arima(M1_Norm)
fcast.M1 <- forecast(fit.M1,h=255) #newdata =test1
fcastExtracted <- as.numeric(fcast.M1$mean);fcastExtracted <-
as.data.frame(fcastExtracted)
minval <- min(M1);maxval <- max(M1)
finalfcast.M1 <- as.numeric(fcast.M1$mean)
#finalfcast.M1 <- fcastExtracted*(maxval-minval)+minval
#######M2#########################################################################
M2 <- dd_vmd$M2
normalize2 <- function(M2){return((M2-min(M2))/(max(M2)-min(M2)))};M2_Norm <-
normalize(M2)
fit.M2 <- auto.arima(M2_Norm)
fcast.M2 <- forecast(fit.M2,h=255) #newdata =test1
fcastExtracted2 <- as.numeric(fcast.M2$mean)
```

Figure 7.45:

```
fcastExtracted2 <- as.data.frame(fcastExtracted2)
minval2 <- min(M2);maxval2 <- max(M2)
finalfcast.M2 <- as.numeric(fcast.M2$mean)
#finalfcast.M2 <- fcastExtracted2*(maxval2-minval2)+minval2
#######M2############################################################
M3 <- dd_vmd$M3
normalize3 <- function(M3){return((M3-min(M3))/(max(M3)-min(M3)))};M3_Norm <-
normalize(M3)
fit.M3 <- auto.arima(M3_Norm)
fcast.M3 <- forecast(fit.M3,h=255) #newdata =test1
fcastExtracted3 <- as.numeric(fcast.M3$mean);fcastExtracted3 <-
as.data.frame(fcastExtracted3)
minval3 <- min(M3);maxval3 <- max(M3)
finalfcast.M3 <- as.numeric(fcast.M3$mean)
#finalfcast.M3 <- fcastExtracted3*(maxval3-minval3)+minval3
#######M2############################################################
M4 <- dd_vmd$M4
normalize4 <- function(M4){return((M4-min(M4))/(max(M4)-min(M4)))};M4_Norm <-
normalize(M4)
fit.M4 <- auto.arima(M4_Norm)
fcast.M4 <- forecast(fit.M4,h=255) #newdata =test1
fcastExtracted4 <- as.numeric(fcast.M4$mean);fcastExtracted4 <-
as.data.frame(fcastExtracted4)
minval4 <- min(M4);maxval4 <- max(M4)
finalfcast.M4 <- as.numeric(fcast.M4$mean)
#finalfcast.M4 <- fcastExtracted4*(maxval4-minval4)+minval4
#######M5############################################################
M5 <- dd_vmd$M5
normalize5 <- function(M5){return((M5-min(M5))/(max(M5)-min(M5)))};M5_Norm <-
normalize(M5)
fit.M5 <- auto.arima(M5_Norm)
fcast.M5 <- forecast(fit.M5,h=255) #newdata =test1
fcastExtracted5 <- as.numeric(fcast.M5$mean);fcastExtracted5 <-
as.data.frame(fcastExtracted5)
minval5 <- min(M5);maxval5 <- max(M5)
finalfcast.M5 <- as.numeric(fcast.M5$mean)
#finalfcast.M5 <- fcastExtracted5*(maxval5-minval5)+minval5
#######M6############################################################
M6 <- dd_vmd$M6
normalize6 <- function(M6){return((M6-min(M6))/(max(M6)-min(M6)))};M6_Norm <-
normalize(M6)
fit.M6 <- auto.arima(M6_Norm)
fcast.M6 <- forecast(fit.M6,h=255) #newdata =test1
fcastExtracted6 <- as.numeric(fcast.M6$mean);fcastExtracted6 <-
as.data.frame(fcastExtracted6)
minval6 <- min(M6);maxval6 <- max(M6)
finalfcast.M6 <- as.numeric(fcast.M6$mean)
```

Figure 7.46:

```
#finalfcast.M6 <- fcastExtracted6*(maxval6-minval6)+minval6
#######M7########################################################################
M7 <- dd_vmd$M7
normalize7 <- function(M7){return((M7-min(M7))/(max(M7)-min(M7)))};M7_Norm <-
normalize(M7)
fit.M7 <- auto.arima(M7_Norm)
fcast.M7 <- forecast(fit.M7,h=255) #newdata =test1
fcastExtracted7 <- as.numeric(fcast.M7$mean);fcastExtracted7 <-
as.data.frame(fcastExtracted7)
minval7 <- min(M7);maxval7 <- max(M7)
finalfcast.M7 <- as.numeric(fcast.M7$mean)
#finalfcast.M7 <- fcastExtracted7*(maxval7-minval7)+minval7
#######M8########################################################################
M8 <- dd_vmd$M8
normalize8 <- function(M8){return((M8-min(M8))/(max(M8)-min(M8)))};M8_Norm <-
normalize(M8)
fit.M8 <- auto.arima(M8_Norm)
fcast.M8 <- forecast(fit.M8,h=255) #newdata =test1
fcastExtracted8 <- as.numeric(fcast.M8$mean);fcastExtracted8 <-
as.data.frame(fcastExtracted8)
minval8 <- min(M8);maxval8 <- max(M8)
finalfcast.M8 <- as.numeric(fcast.M8$mean)
#finalfcast.M8 <- fcastExtracted8*(maxval8-minval8)+minval8
################################################################################
#FinalForecast_vmd_arima
FinalForecast_vmd_arima <-
finalfcast.M1+finalfcast.M2+finalfcast.M3+finalfcast.M4+finalfcast.M5+finalfcast.M6+finalfc
ast.M7+finalfcast.M8
vad <- data.frame(test_D ,FinalForecast_vmd_arima)
col_headings <- c("Actual Price","Forecast Price")
names(vad) <- col_headings;attach(vad)
MAE(vad$`Actual Price`,vad$`Forecast Price`)
RMSE(vad$`Actual Price`,vad$`Forecast Price`)
MAPE(vad$`Actual Price`,vad$`Forecast Price`)
mae_vmd_arima <- abs(vad$`Actual Price`-vad$`Forecast Price`)
mse_vmd_arima<- (vad$`Actual Price`-vad$`Forecast Price`)^2
mape_vmd_arima <- (abs((vad$`Actual Price`-vad$`Forecast Price`)/vad$`Actual Price`))*100

FinalForecast_vmd_arima <- ts(FinalForecast_vmd_arima)
plot(FinalForecast_vmd_arima)
(FinalForecast_vmd_arima)
lines(test, col="red")
dm.test(ead$'Actual Price'-ead$'Forecast Price',vad$'Actual Price'-vad$'Forecast Price')
#two models have different accuracy
CP <- data.frame(CP$Value)
index <- sample(nrow(CP),round(0.8*nrow(CP)))
```

Figure 7.47:

```
train <- CP[index,]
test <- CP[-index,]
CP_Norm <- ts((train-min(train))/(max(train)-min(train)))
normalizeCP<- function(train){return((train-min(train))/(max(train)-min(train)))}
CP_Norm <- normalize(train)
fit_og <- nnetar(CP_Norm)
nfcast_og <- forecast(fit_og,h=255) #newdata =test1
FVE <- as.numeric(nfcast_og$mean)
F1<-FVE
minval <- min(train);maxval <- max(train)
F1 <- FVE*(maxval-minval)+minval
ts.plot(test,col="red")
plot(F1)
ff <- data.frame(test ,F1)
col_headings <- c("Actual Price","Forecast Price")
names(ff) <- col_headings;attach(ff)
MAE(ff$`Actual Price`,ff$`Forecast Price`)
RMSE(ff$`Actual Price`,ff$`Forecast Price`)
MAPE(ff$`Actual Price`,ff$`Forecast Price`)

mae_bpnn <- abs(ff$`Actual Price`-ff$`Forecast Price`)
mse_bpnn<- (ff$`Actual Price`-ff$`Forecast Price`)^2
mape_bpnn <- (abs((ff$`Actual Price`-ff$`Forecast Price`)/ff$`Actual Price`))*100
CP2<-CP
CP2
index <- sample(nrow(CP2),round(0.8*nrow(CP2)))
train <- CP2[index,]
test <- CP2[-index,]
test
CP2_Norm <- ts(2*(train-min(train))/(max(train)-min(train))-1)
max(CP2_Norm)
min(CP2_Norm)
fit_O <- auto.arima(CP2_Norm)
nfcast_O <- forecast(fit_O,h=255) #newdata =test1
plot(nfcast_O)
as.data.frame(nfcast_O)
FVEO <- ts(as.numeric(nfcast_O$mean))
plot(FVEO)
minval <- min(train);maxval <- max(train)
FO <- ts(FVEO*(maxval-minval)+minval)
FO
ddO <- data.frame(test ,FO)
col_headings <- c("Actual Price","Forecast Price")
names(ddO) <- col_headings;attach(ddO)
MAE(ddO$`Actual Price`,ddO$`Forecast Price`)
RMSE(ddO$`Actual Price`,ddO$`Forecast Price`)
MAPE(ddO$`Actual Price`,ddO$`Forecast Price`)
```

Figure 7.48:

```
mae_arima <- abs(ddO$`Actual Price`-ddO$`Forecast Price`)
mse_arima<- (ddO$`Actual Price`-ddO$`Forecast Price`)^2
mape_arima <- (abs((ddO$`Actual Price`-ddO$`Forecast Price`)/ddO$`Actual Price`))*100
#####################################################################
#autoplot(FinalForecast_emd_BPNN)
#plot(FinalForecast_emd_BPNN)
#lines(FinalForecast_emd_BPNN_Res, col="violet")
#lines(FinalForecast_vmd_arima, col="red")
#lines(FinalForecast.emd.ARIMA, col="green")
#lines(FinalForecast_vmd_BPNN, col="blue")
#lines(F1,col="yellow")
#lines(FVEO)
#lines(test)

test_norm <- (test-min(test))/(max(test)-min(test))

ts.test <- xts(test$Value, order.by = as.POSIXct(test$Date, "UTC"))
plot(ts.test)
lines(FinalForecast_vmd_arima)
autoplot(ts(test), series = "Actual")+
autolayer(FinalForecast.emd.ARIMA, series = "EMD-ARIMA")+
 autolayer(FinalForecast_emd_BPNN, series = "EMD-BPNN")+
  autolayer(ts(FinalForecast_vmd_arima), series = "VMD-ARIMA")+
   autolayer(ts(FinalForecast_vmd_BPNN), series = "VMD-BPNN")+
    autolayer(ts(F1), series = "BPNN")+
     autolayer(ts(FVEO), series = "ARIMA")

plot(FinalForecast.emd.ARIMA)
lines(test_norm)
plot()
plot(FinalForecast_emd_BPNN_Res, col="black",xlab="Time/daily",ylab="Crude oil Future
Price " )
lines(FinalForecast_vmd_arima, col="red")
lines(FinalForecast.emd.ARIMA, col="green")
lines(FinalForecast_vmd_BPNN, col="blue")
lines(F1,col="yellow")
lines(FVEO, col="brown")
lines(`Actual Price`, col="purple")

legend("topright",col=c("black", "red", "green", "blue", "yellow", "brown", "purple"),lwd=2,
       legend=c("EMD-BPNN", "VMD-ARIMA", "EMD-ARIMA", "EMD-BPNN", "BPNN",
"ARIMA", "Actual"))
```

Figure 7.49:

```
###################### Dm Test ##############
#null hypothesis is that the two methods have the same forecast accuracy
library(forecast)
#################################################################################
#Comparing forecasts accuracy MAE
dm.test(mae_emd_bpnn,mae_vmd_bpnn)
dm.test(mae_emd_bpnn,mae_emd_arima)
dm.test(mae_emd_bpnn,mae_vmd_arima)
dm.test(mae_emd_bpnn,mae_bpnn)
dm.test(mae_emd_bpnn,mae_arima)

dm.test(mae_vmd_bpnn,mae_emd_arima)
dm.test(mae_vmd_bpnn,mae_vmd_arima)
dm.test(mae_vmd_bpnn,mae_bpnn)
dm.test(mae_vmd_bpnn,mae_arima)

dm.test(mae_emd_arima,mae_vmd_arima)
dm.test(mae_emd_arima,mae_arima)
dm.test(mae_emd_arima,mae_bpnn)

dm.test(mae_vmd_arima,mae_arima)
dm.test(mae_vmd_arima,mae_bpnn)

dm.test(mae_arima,mae_bpnn)
#################################################################################
#Comparing forecasts accuracy RMSE
dm.test(mse_emd_bpnn,mse_vmd_bpnn)
dm.test(mse_emd_bpnn,mse_emd_arima)
dm.test(mse_emd_bpnn,mse_vmd_arima)
dm.test(mse_emd_bpnn,mse_bpnn)
dm.test(mse_emd_bpnn,mse_arima)

dm.test(mse_vmd_bpnn,mse_emd_arima)
dm.test(mse_vmd_bpnn,mse_vmd_arima)
dm.test(mse_vmd_bpnn,mse_bpnn)
dm.test(mse_vmd_bpnn,mse_arima)

dm.test(mse_emd_arima,mse_vmd_arima)
dm.test(mse_emd_arima,mse_arima)
dm.test(mse_emd_arima,mse_bpnn)
dm.test(mse_vmd_arima,mse_arima)
dm.test(mse_vmd_arima,mse_bpnn)

dm.test(mse_arima,mse_bpnn)
```

Figure 7.50:

```
#############################################################################
#Comparing forecasts accuracy MAPE
dm.test(mape_emd_bpnn,mape_vmd_bpnn)
dm.test(mape_emd_bpnn,mape_emd_arima)
dm.test(mape_emd_bpnn,mape_vmd_arima)
dm.test(mape_emd_bpnn,mape_bpnn)
dm.test(mape_emd_bpnn,mape_arima)

dm.test(mape_vmd_bpnn,mape_emd_arima)
dm.test(mape_vmd_bpnn,mape_vmd_arima)
dm.test(mape_vmd_bpnn,mape_bpnn)
dm.test(mape_vmd_bpnn,mape_arima)

dm.test(mape_emd_arima,mape_vmd_arima)
dm.test(mape_emd_arima,mape_arima)
dm.test(mape_emd_arima,mape_bpnn)

dm.test(mape_vmd_arima,mape_arima)
dm.test(mape_vmd_arima,mape_bpnn)

dm.test(mape_arima,mape_bpnn)
```

Figure 7.51: