



UNIVERSIDAD NACIONAL DE COLOMBIA

# Predicción de series de tiempo no lineales usando un modelo híbrido basado en la descomposición wavelet

Michael Vásquez Londoño

Universidad Nacional de Colombia

Escuela de Estadística

Medellín, Colombia

2018



# Predicción de series de tiempo no lineales usando un modelo híbrido basado en la descomposición wavelet

Michael Vásquez Londoño

Tesis o trabajo de grado presentada(o) como requisito parcial para optar al título de:  
**Maestría en Ciencias-Estadística**

Director:  
Norman Diego Giraldo Gómez  
Profesor Asociado, MsC (Matemáticas)

Línea de Investigación:  
Series de Tiempo

Universidad Nacional de Colombia  
Escuela de Estadística  
Medellín, Colombia  
2018



«We are born at sunset and we die ere morn, and the whole darkness of the world we know...»

Fernando Pessoa



# Agradecimientos

Se agradece la colaboración del profesor Norman Giraldo, director de esta tesis. Asimismo se resaltan las incontables observaciones y aportes de las profesoras Gladys Elena Salcedo Echeverry y Luz Marina Gómez Gómez que como jurados evaluadores ayudaron en la clarificación de múltiples conceptos.



## Resumen

El pronóstico de series de tiempo no lineales ha recibido especial atención debido a que los patrones subyacentes en los datos son difíciles de capturar. Durante los últimos años, un gran número de modelos han sido propuestos para superar el pronóstico de las metodologías tradicionales. En este trabajo, se usa una metodología adecuada para pronosticar series de tiempo, con un alto nivel de ruido, que evidencien no linealidad y no estacionariedad. Específicamente, se combina la transformación wavelet discreta de máximo traslape (MODWT) con el modelo ARFIMA-HYGARCH y redes neuronales. Además, se implementa el modelo LSW propuesto para el pronóstico de series de tiempo no estacionarias. Ambas metodologías, se aplican para pronosticar la tasa de cambio USD/COP. Los resultados sugieren que la metodología híbrida, basada en wavelets y redes neuronales, provee pronósticos más precisos para pronosticar una apreciación/depreciación del tipo de cambio.

**Palabras clave:** Pronóstico, modelos híbridos, ARFIMA-HYGARCH, Descomposición Wavelet, modelo LSW, tipos de cambio.

## Abstract

The forecast of nonlinear time series has received special attention due to the difficulty to capture the underlying patterns in the data. In recent years, a number of models have been proposed to improve the performance of the traditional models. In this research, we used a methodology to forecast nonstationary and nonlinear time series contaminated with high noise levels. In particular, we combine the maximal overlap discrete wavelet transform (MODWT) with the model ARFIMA-HYGARCH and neural networks (NN). In addition, the LSW model is implemented to forecast non-stationary time series. In the present study, both methodologies are applied to forecast USD/COP exchange rate. The results suggest that the hybrid methodology, based on wavelets and neural networks, provided more accurate forecast of an appreciation or a depreciation of the exchange rates.

**Keywords:** Forecast, Hybrid models, ARFIMA-HYGARCH, wavelet decomposition, LSW model, exchange rates

# Contenido

<b>Agradecimientos</b>	<b>vii</b>
<b>Resumen</b>	<b>ix</b>
<b>1 Introducción</b>	<b>2</b>
<b>2 Modelación de series de tiempo</b>	<b>7</b>
2.1 Series de tiempo estacionarias . . . . .	7
2.2 Linealidad y no linealidad de series de tiempo . . . . .	8
2.3 Retornos logarítmicos . . . . .	9
2.4 Algunos modelos para series de tiempo . . . . .	10
2.4.1 Modelo ARIMA Y ARFIMA . . . . .	10
2.4.1.1 Estimador GPH del parámetro de diferenciación . . . . .	12
2.4.1.2 Estimador de Reisen . . . . .	12
2.4.1.3 Estimador de Castaño . . . . .	13
2.4.1.4 Identificación modelo ARFIMA . . . . .	13
2.5 Modelos GARCH . . . . .	14
2.6 Generalidades sobre redes neuronales . . . . .	16
<b>3 Modelación de series de tiempo usando el análisis wavelet</b>	<b>20</b>
3.1 Análisis multiescala . . . . .	20
3.2 Transformación Haar sobre funciones . . . . .	22
3.3 Transformación wavelet discreta . . . . .	23
3.3.1 Transformación Wavelet No-Diezmada . . . . .	29
3.4 Proceso Wavelet Localmente Estacionario (LSW) y pronóstico . . . . .	30
3.4.1 Detección de no estacionariedades . . . . .	35
3.4.2 Pronóstico de un proceso LSW . . . . .	36
3.5 Metodología híbrida basada en wavelets . . . . .	38
<b>4 Resultados empíricos</b>	<b>42</b>
4.1 Estadísticas preliminares . . . . .	42
4.2 Ajuste modelo ARFIMA-HYGARCH . . . . .	45
4.2.1 Ajuste del modelo ARFIMA . . . . .	46
4.2.2 Ajuste del modelo en varianza condicional . . . . .	46

---

4.3	Ajuste del proceso LSW . . . . .	48
4.4	Metodología Híbrida . . . . .	53
4.5	Evaluación de los pronósticos . . . . .	58
<b>5</b>	<b>Conclusiones</b>	<b>62</b>
<b>6</b>	<b>Anexo</b>	<b>63</b>
6.1	Modelos ARFIMA-HYGARCH para subseries . . . . .	63
6.2	Test BDS sobre residuales ARFIMA-IGARCH para subseries . . . . .	63
6.3	Validación cruzada redes neuronales . . . . .	63
6.4	Código: Análisis descriptivo de la serie . . . . .	63
6.5	Código: análisis de memoria larga . . . . .	77
6.6	Código: identificación modelo ARFIMA . . . . .	79
6.7	Código: especificación Garch . . . . .	80
6.8	Código: prueba de no estacionariedad y proceso LSW . . . . .	84
6.9	Código: metodología híbrida: Modelo ARFIMA-HYGARCH . . . . .	86
6.10	Código: metodología híbrida: Modelo de redes neuronales . . . . .	96
6.10.1	Selección de variables . . . . .	96
6.10.2	Código: validación cruzada redes neuronales . . . . .	97
6.10.3	Código: pesos iniciales redes neuronales . . . . .	101

# 1 Introducción

El análisis de series de tiempo que exhiben estructuras no lineales ha recibido especial atención debido a la dificultad intrínseca para capturar los patrones subyacentes en los datos. En general, en la literatura se han propuesto una gran cantidad de metodologías, paramétricas y no paramétricas, para modelar el proceso generador de datos. Sin embargo, un hecho común surge en la mayoría de trabajos. El modelo se ajusta a la serie temporal de los datos que está contaminada, usualmente, por un alto nivel de ruido, lo cual no permite grandes mejoras predictivas cuando se comparan con otras metodologías tradicionales de tipo no lineal. [Stock, 2001] muestra que dado que las especificaciones no lineales no son significativamente mejores para pronosticar que las lineales, la relación costo beneficio estaría en contra de las primeras.

Asimismo, una gran cantidad de series de tiempo en las ciencias aplicadas muestran una estructura de segundo orden que varía en función del tiempo (especialmente para retornos logarítmicos, ver [Fryzlewicz et al., 2003]). A menudo, la modelización es llevada a cabo mediante modelos no lineales pero que conservan el supuesto de estacionariedad, lo que genera un nivel de predicción pobre especialmente cuando existe una estructura poco homogénea al final de los datos.

Recientemente se ha propuesto la estimación a través de wavelets que en lugar de considerar la serie temporal de los datos tiene en cuenta el dominio tiempo escala. El análisis de wavelets, en el contexto de series de tiempo univariadas, se divide en dos corrientes: por un lado, los modelos de pronóstico basados en la estimación del periodograma mediante wavelets y, por otro lado, los modelos híbridos que combinan la metodología wavelet con otros modelos tradicionales de pronóstico.

En la primera corriente, se destaca el trabajo de [Fryzlewicz et al., 2003]. A partir del modelo Wavelet Localmente Estacionario (LSW), el cual se basa en una representación tiempo escala de una función de autocovarianza que cambia en el tiempo, [Fryzlewicz et al., 2003] proponen un marco teórico para estimar y pronosticar procesos no estacionarios. Además, en un trabajo posterior, [Fryzlewicz, 2005] muestra que la metodología LSW captura varios hechos estilizados de los retornos logarítmicos de las series financieras. Entre ellos podemos destacar que su distribución presenta colas pesadas y que la volatilidad es abrupta y tiende a agruparse. Sin embargo, contrario a la especificación GARCH que también captura estas propiedades, no requiere del supuesto de estacionariedad. Asimismo, en la misma línea, [Wong et al., 2003] propone un método de modelación y pronóstico basado en el análisis por multiresolución.

---

Por su parte, la corriente alternativa combina la metodología wavelet con modelos de pronóstico tradicionales de tipo paramétricos y semiparamétricos. En este sentido, las wavelets permiten descomponer la serie de tiempo en diferentes niveles de resolución. En cada nivel se obtiene una señal de aproximación y una señal de detalle. La señal de aproximación toma en cuenta las frecuencias bajas, mientras que la señal de detalle corresponde a los componentes de alta frecuencia. Por lo tanto, el análisis de wavelets híbrido busca descomponer una serie de tiempo, que puede ser no estacionaria o provenir de un proceso generador de datos no lineal, en un conjunto de componentes más simples que pueden ser modeladas independientemente.

A continuación, se profundiza en la revisión de literatura de las dos corrientes. El análisis de wavelets permite descomponer una serie de tiempo como un conjunto de vectores discretos en distintas escalas. [Nason et al., 2000] propone el uso de wavelets para modelar series con una estructura de segundo orden que varía en función del tiempo. Para ello, fundamentan teóricamente un modelo que denominan proceso wavelet localmente estacionario (LSW), proponen un método de estimación y derivan propiedades asintóticas para dicho proceso.

Posteriormente, [Fryzlewicz et al., 2003] establecen una metodología de pronóstico a partir de un proceso LSW. Para validar la metodología propuesta, modelan la velocidad del viento sobre la superficie del océano en una región del pacífico. La serie es medida en términos de desviaciones respecto a la media y es importante porque permite entender el efecto de El Niño sobre esta región. Existe una evidencia significativa de que la serie es no estacionaria, y cuando se compara con el modelo autorregresivo (AR), los resultados del pronóstico basados en un proceso LSW, presentan intervalos de predicción más estrechos.

[Fryzlewicz, 2005] propone el uso de un proceso LSW para modelar y pronosticar retornos logarítmicos financieros. Para ello muestra que la mayoría de los hechos estilizados reportados en la literatura para los retornos logarítmicos son compatibles con el modelo LSW. Además, muestra que el ruido blanco modulado (TMWN) es un caso especial de dicho proceso. El modelo es aplicado al índice bursátil de la Bolsa de Londres (FTSE 100) y al índice Dow Jones (DJIA). [Xie et al., 2009] muestran que la metodología de pronóstico propuesto por [Fryzlewicz et al., 2003] puede ser sensible a la existencia de valores atípicos y modifican el algoritmo imponiendo algunas restricciones. Evalúan la nueva metodología mediante el pronóstico por fuera de la muestra para la volatilidad de los retornos del índice Standard & Poors 500 (S&P 500).

Por otro lado, una corriente alternativa ha usado el análisis de wavelets en conjunto con otros modelos tradicionales de pronóstico para modelar series de tiempo que presentan características de no linealidad o no estacionariedad. En la literatura se encuentran múltiples metodologías, paramétricas y semiparamétricas. A continuación se mencionan algunos trabajos.

En cuanto a la combinación de wavelets con el modelo lineal ARMA se destaca el trabajo de [Conejo et al., 2005] que propone un modelo híbrido para el pronóstico de los precios de la electricidad en el mercado español, específicamente, emplean un modelo wavelet-ARMA. La

metodología híbrida es superior, en comparación con el modelo ARMA, para todas las estaciones del año. [Karthikeyan and Nagesh Kumar, 2013] emplean las wavelets en combinación con modelos ARMA para la predicción de series hidrológicas no estacionarias; particularmente, emplean series mensuales relativas a caudales. A pesar de la no estacionariedad, el modelo wavelets-ARMA funciona de gran manera para predecir eventos de sequía a largo plazo. [Zhu et al., 2014] proponen una metodología híbrida que combina una transformación wavelet no diezmada y el modelo ARMA. Específicamente, emplean la Transformada Wavelet Discreta de Máximo Traslape (MODWT). El modelo propuesto es evaluado para una serie de precipitaciones diarias y la serie MacKey-Glass. En términos del pronóstico, el modelo híbrido MODWT-ARMA resulta superior que su contraparte híbrida basada en la Transformada Wavelet Discreta (DWT).

Algunos trabajos han empleado las wavelets con el modelo GARCH. [Tan et al., 2010] proponen un modelo mixto que combina la metodología wavelet con el modelo ARMA-GARCH. El modelo se aplica para la serie de precios de la electricidad del mercado español. Los resultados sugieren una evidencia clara a favor de la superioridad de la metodología híbrida respecto al modelo tradicional simple. [Ismail et al., 2016] comparan el modelo GARCH y el MODWT-GARCH para pronosticar la volatilidad del índice de precios diarios para cuatro países africanos. La metodología híbrida arroja pronósticos más precisos.

En cuanto al uso de metodologías no paramétricas en combinación con la descomposición wavelet es común el uso de redes neuronales y Máquinas de soporte vectorial (SVM). [Soltani, 2002] propone el uso del análisis multiescala en combinación con redes neuronales para la predicción de la serie MacKey-Glass. La metodología resulta superior comparada con otras técnicas semiparamétricas como las SVM y Funciones de Base Radial. [Fernandez, 2007] usa distintas series de tiempo mensuales de la industria manufacturera estadounidense para comparar la capacidad predictiva de varios modelos tradicionales y una metodología híbrida basada en wavelets. Particularmente compara un modelo ARIMA estacional multiplicativo (SARIMA), SVM, un modelo de componentes no observables y un modelo híbrido SARIMA-wavelets. Además, realiza una combinación lineal de los pronósticos para capturar las distintas características de cada modelo. Los resultados sugieren, en general, la superioridad de la combinaciones basadas en wavelets. [Ben Mabrouk et al., 2008] proponen el uso de un modelo autorregresivo (AR) en combinación con wavelets y estimadores Kernel. Los datos analizados corresponden a la serie de manchas solares y a una serie generada por la ecuación MacKey-Glass. La metodología propuesta resulta superior, en términos del pronóstico, cuando se compara con modelos no paramétricos, el modelo autorregresivo de umbral y el modelo propuesto por [Soltani, 2002]. [Guo et al., 2008] proponen el uso de wavelets con SVM para pronosticar series de producción de diferentes industrias en Japón. Empleando distintas medidas de error de pronóstico, encuentran que el modelo híbrido es superior a las SVM tradicionales.

En cuanto al tipo de cambio, la mayoría de la investigación empírica ha puesto de manifiesto la incapacidad de los modelos estructurales macroeconómicos para explicar el comportamien-

to del tipo de cambio nominal en el corto, mediano y largo plazo, ver [Meese and K.S., 1983]. Lo anterior ha desplazado la investigación al análisis de las propiedades estadísticas de la serie temporal y a considerar el tipo de cambio como el precio de un activo financiero (véase [Groen, 1999], [Rowland, 2002] y [Cheung et al., 2005]). Bajo este enfoque, adquiere especial importancia el mecanismo generador de las expectativas de los agentes.

Los estudios iniciales conservaron el supuesto de expectativas racionales de los modelos macroeconómicos neoclásicos, que en las finanzas cuantitativas se traducen en la Hipótesis del Mercado Eficiente (HME) propuesta por [Bachelier, 1964] y generalizada por [Osborne, 1964]. La HME sostiene que los precios reflejan toda la información disponible y los agentes son homogéneos. Estadísticamente, esto se traduce en que el precio de un activo puede ser modelado como

$$r_{t+1} = r_t + \varepsilon_t, \quad \varepsilon_t \sim iid(0, \sigma^2), \quad (1-1)$$

donde  $r_t$  es el logaritmo del precio en el período  $t$  y  $\varepsilon_t$  es un proceso ruido blanco. Por lo tanto, el mercado no tiene memoria y los rendimientos pasados no sirven para predecir los rendimientos futuros. Sin embargo, la evidencia empírica rechaza la hipótesis de eficiencia, ver [Lo and C., 1988], [Chang and Ting, 2000], y [Singh and Dey, 2002]. En cambio, el comportamiento del tipo de cambio se caracteriza por:

- Una volatilidad abrupta y que tiende a agruparse. Por lo tanto, se tiene una alta persistencia e indicios de que el proceso tiene una varianza condicionada no estacionaria.
- Dependencia de las series a largo plazo, los hechos pasados son fundamentales en la evolución del sistema dinámico.
- Su distribución presenta colas pesadas, es leptocúrtica y asimétrica. Por lo tanto, se incrementa la probabilidad de ocurrencia de eventos extremos que tenían una baja probabilidad en una distribución normal.
- Fuerte dependencia no lineal, [Kilian and Taylor, 2003].

Para el tipo de cambio Peso/Dólar se han propuesto varias alternativas de modelación. [Patiño and Alonso, 2005] evalúan la capacidad predictiva de los modelos macroeconómicos contra un paseo aleatorio, [Castaño et al., 2008a] proponen modelar la media y la varianza condicional usando una estrategia de modelos anidados ARFIMA-HYGARCH. Los autores encuentran evidencia de memoria larga y de un proceso integrado en la varianza. En esta misma línea se encuentra el trabajo de [Montenegro, 2010] quien propone el uso de modelos de la familia GARCH. También se han propuesto modelos no paramétricos, [Jalil and Misas, 2007] comparan una red neuronal contra la especificación ARIMA y un paseo aleatorio. En el estudio se emplean funciones de costo simétricas y asimétricas. La red neuronal resulta claramente superior. [Parra, 2014] estima la volatilidad de la tasa de cambio a través de un modelo de volatilidad estocástica ( $SV$ ) y los resultados son comparados con el modelo GARCH. Ambos

modelos logran capturar la alta persistencia, exceso de curtosis y clusters de volatilidad. Sin embargo, el supuesto de normalidad parece no ser adecuado.

[Maya and Gómez, 2008] analizan el mercado cambiario para distintos países latinoamericanos encontrando evidencia de memoria larga en media para el caso colombiano. Para el Peso Chileno encuentran evidencia de un  $ARMA(1,1)$ , para el resto de retornos de las tasas de cambio no rechazan la hipótesis de media cero. En cuanto a la varianza condicional encuentran evidencia de procesos integrados para el Peso Chileno/Euro y peso Colombiano/Dólar. Además, encuentran evidencia de especificaciones hiperbólicas y fraccionalmente integradas. La distribución que mejor se ajusta a los datos es la distribución de error generalizada ( $GED$ ).

En este trabajo se comparará la capacidad predictiva del uso de wavelets, dentro del marco LSW, y su implementación híbrida, tanto paramétrica como semiparamétrica, con otros modelos tradicionales. Como lo señala [Fryzlewicz, 2005] la principal motivación para el uso de wavelets es el análisis de fenómenos multiescala. Muchos autores observan que varios factores operan en distintas escalas de tiempo. Esta característica es usual en retornos logarítmicos financieros. En este trabajo se emplean los retornos diarios del tipo de cambio para el Peso Colombiano (COP)/Dólar Estadounidense (USD).

Este trabajo consta de cuatro capítulos incluyendo el presente. En general, se comienza exponiendo los modelos tradicionales en el análisis de series de tiempo y la transformada wavelet, para posteriormente presentar el proceso LSW y la metodología híbrida.

Específicamente en el capítulo dos se presentan de forma general dos conceptos fundamentales en el análisis de series de tiempo como estacionariedad y no linealidad. Posteriormente, en el mismo capítulo se presentan los modelos lineales de memoria corta ARIMA y de memoria larga ARFIMA. Finalmente, se presentan dos especificaciones no lineales usadas comúnmente en la literatura, el modelo GARCH Hiperbólico (HYGARCH) y las redes neuronales (NN). En el capítulo tres se presenta el análisis wavelet en el contexto de series de tiempo. En este sentido, se introducen conceptos básicos provenientes del análisis multiescala y se generaliza hasta presentar la transformada wavelet discreta y su versión no-diezmada. A continuación se presenta el proceso LSW y su metodología de estimación y pronóstico. El capítulo finaliza con la metodología híbrida de pronóstico basada en wavelets, se muestra la transformación wavelet discreta de máximo traslape (MODWT) y se presentan las especificaciones híbridas wavelets-ARFIMA-HYGARCH y wavelets-red neuronal. Finalmente, en el capítulo cuatro se presentan los resultados para los retornos de la tasa de cambio USDCOP.

# 2 Modelación de series de tiempo

## 2.1. Series de tiempo estacionarias

Una serie de tiempo se define de forma general como la realización de un proceso estocástico. Este último, es una familia de variables aleatorias  $\{X_t\}_{t=-\infty}^{\infty}$ , donde  $t$  representa el tiempo, tal que para cada serie finita de elecciones de  $t$ ,  $(t_1, t_2, \dots, t_n)$ , se define una distribución de probabilidad conjunta para las variables aleatorias  $X_{t_1}, X_{t_2}, \dots, X_{t_n}$ . De esta manera, el proceso estocástico se puede describir a través de su distribución conjunta de probabilidad. Sin embargo, en la práctica se usan los momentos de primer y segundo orden, la media,  $\mu_t = E(X_t)$ , la varianza,  $\sigma_t^2 = E(X_t - \mu_t)^2$ , y la covarianza  $\gamma_{ts} = E[(X_t - \mu_t)(X_s - \mu_s)]$ . Un concepto esencial en la literatura de series de tiempo es el de estacionariedad. En sentido estricto, un proceso es estacionario cuando la función de distribución conjunta de  $(X_{t_1}, X_{t_2}, \dots, X_{t_n})$  y  $(X_{t_1+\tau}, X_{t_2+\tau}, \dots, X_{t_n+\tau})$  es la misma para todo  $(t_i, n, \tau) \in \mathbb{Z}$ . La estacionariedad estricta es un supuesto bastante restrictivo así que suele emplearse una forma más débil. La estacionariedad de segundo orden, o en sentido débil, se cumple si se satisfacen tres propiedades. La  $E(X_t) = \mu, \forall t \in \mathbb{Z}$ , es decir  $\mu_t = \mu$ . Las varianzas son constantes y finitas  $Var(X_t) = \sigma^2, \forall t \in \mathbb{Z}$ . Finalmente, la autocovarianza  $\gamma_X(\tau) = cov(X_t, X_{t+\tau})$  sólo es función de  $\tau$ .

El supuesto de estacionariedad implica que las propiedades estadísticas de una serie temporal no se ven afectadas si se realiza un cambio de origen en el tiempo. Por lo tanto, las relaciones estadísticas para  $n$  observaciones con origen  $t$  son las mismas que para el cambio de origen  $t + \tau$ . Algunas de estas relaciones importantes son la autocorrelación y la autocorrelación parcial. La autocorrelación mide el grado de asociación lineal entre  $X_t$  y la variable trasladada  $\tau$  lugares,  $X_{t+\tau}$ . La autocorrelación parcial cuantifica la correlación ajustada por el efecto de los retardos intermedios. Tanto la función de autocovarianza como las funciones de autocorrelación son estimadas mediante los valores muestrales  $\{x_t\}_{t=1}^{T-\tau}$  y los valores rezagados  $\{x_{t+\tau}\}_{t=1}^{T-\tau}$ .

Otra función importante es el espectro o función de densidad espectral,  $f_X(\omega)$ , que mide la cantidad de oscilación en diferentes frecuencias  $\omega \in (-\pi, \pi)$ . Particularmente,  $f_X(\omega) d\omega$  cuantifica la contribución de la varianza total de  $X_t$  para frecuencias en el rango  $(\omega, \omega + d\omega)$ . El espectro se relaciona con la autocovarianza mediante la relación de Fourier

$$f_X(\omega) = (2\pi)^{-1} \sum_{\tau=-\infty}^{\infty} \gamma_X(\tau) \exp(-i\omega\tau). \quad (2-1)$$

## 2.2. Linealidad y no linealidad de series de tiempo

En la literatura existen diversas definiciones sobre no linealidad. [Campbell et al., 1997] parten de una serie de tiempo,  $\{x_t\}$ , que es función de una sucesión de innovaciones  $\varepsilon_t$ , independientes e idénticamente distribuidas (*iid*)

$$x_t = f(\varepsilon_t, \varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-j}). \quad (2-2)$$

La serie se puede descomponer mediante una expansión de Taylor alrededor de  $\varepsilon_t = 0$ , para  $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots$  dados, como

$$x_t = f(0, \varepsilon_{t-1}, \dots) + \varepsilon_t \frac{d}{dt} f(0, \varepsilon_{t-1}, \dots) + \frac{1}{2} \varepsilon_t^2 \frac{d^2}{dt^2} f(0, \varepsilon_{t-1}, \dots) + \dots \quad (2-3)$$

Si la magnitud de la innovación  $\varepsilon_t$  es pequeña, los términos de orden superior resultan despreciables, y (2-3) se puede aproximar por

$$x_t = g(0, \varepsilon_{t-1}, \dots) + \varepsilon_t h(0, \varepsilon_{t-1}, \dots), \quad (2-4)$$

donde  $g(0, \varepsilon_{t-1}, \dots) = f(0, \varepsilon_{t-1}, \dots)$  y  $h(0, \varepsilon_{t-1}, \dots) = \frac{d}{dt} f(0, \varepsilon_{t-1}, \dots)$ . Dado que  $E[x_t | x_{t-1}, x_{t-2}, \dots] = g(\varepsilon_{t-1}, \varepsilon_{t-2}, \dots)$ , la función  $g$  representa la media condicional del proceso y  $h^2$  la varianza condicional [Teräsvirta et al., 1994]. Los modelos que usan la función  $g$  no lineal se denominan modelos no lineales en media, mientras que los modelos que usan una función  $h^2$  no lineal, son llamados modelos no lineales en varianza [Tsay, 2001].

En series no lineales la media condicional se puede descomponer en una componente lineal y una no lineal, expandiendo la función  $g$  en una serie de Volterra en la forma

$$g(\varepsilon_{t-1}, \varepsilon_{t-2}, \dots) = \sum_{i=1}^{\infty} a_i \varepsilon_{t-i} + \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} b_{ij} \varepsilon_{t-i} \varepsilon_{t-j} + \sum_{i=1}^{\infty} \sum_{j=i}^{\infty} \sum_{k=j}^{\infty} c_{ijk} \varepsilon_{t-i} \varepsilon_{t-j} \varepsilon_{t-k} + \dots \quad (2-5)$$

El primer término es una media móvil lineal, mientras que los demás son no lineales. Un test para probar dependencia no lineal en la media condicional es el test propuesto por [Brock et al., 1996] denominado test BDS. El test fue propuesto originalmente para determinar si las observaciones de una serie de tiempo son independientes e idénticamente distribuidas (*iid*) detectando todo tipo de dependencia lineal o no lineal. Si  $\{x_t\}$  es una serie de tiempo compuesta por observaciones independientes e idénticamente distribuidas (*iid*), la probabilidad de que una historia de dos observaciones estén a una distancia menor o igual a  $\varepsilon$  la una de la otra, está dada por

$$C_2(\varepsilon) = P(|x_t - x_j| \leq \varepsilon, |x_{t-1} - x_{j-1}| \leq \varepsilon) \quad (2-6)$$

$$= P(|x_t - x_j| \leq \varepsilon) P(|x_{t-1} - x_{j-1}| \leq \varepsilon) \quad (2-7)$$

$$= P(|x_t - x_j| \leq \varepsilon)^2 \quad (2-8)$$

$$= [C_1(\varepsilon)]^2. \quad (2-9)$$

Generalizando (2-9) a un espacio de fase  $m$ -dimensional, se tiene

$$C_m(\varepsilon) = [C_1(\varepsilon)]^m, \quad (2-10)$$

donde  $C_m(\varepsilon)$  es la integral de correlación de [Gassberger and Procaccia, 1983] cuyo estimador, para una serie de longitud  $n$ , es

$$C_{mn}(\varepsilon) = \frac{2}{(n-m+1)(n-m)} \sum_{s=1}^{n-m+1} \sum_{t=1}^{n-m+1} \prod_{s=1}^{m-1} H(\varepsilon - |x_{s+j} - x_{t+j}|), \quad (2-11)$$

donde  $H(R) = \begin{cases} 1 & \text{si } \varepsilon - |x_{s+j} - x_{t+j}| \geq 0 \\ 0 & \text{eoc} \end{cases}$ , es la función heaviside que enumera el número de  $m$ -dimensiones que cumplen con la condición de cercanía. De este modo el estadístico BDS se define como

$$W_m(\varepsilon) = \sqrt{T} \frac{C_{mn}(\varepsilon) - [C_1(\varepsilon)]^m}{\hat{\sigma}_m(\varepsilon)}, \quad (2-12)$$

donde  $\hat{\sigma}_m(\varepsilon)$  es el estimador asintótico de la desviación estándar de  $C_{mn}(\varepsilon) - [C_1(\varepsilon)]^m$ . Bajo la hipótesis nula de datos *iid*, el test BDS converge a una distribución normal estándar. Una descripción detallada del test y sus propiedades se puede encontrar en [Brock et al., 1996]. Para probar dependencia no lineal el BDS se calcula sobre los residuales de un modelo que remueva la dependencia lineal de los datos.

## 2.3. Retornos logarítmicos

En la literatura de series de tiempo financieras, es común investigar los retornos de los activos en lugar de los precios. En general, el retorno es el rendimiento del activo con capitalización continua y proporciona toda la información sobre la decisión de invertir. La serie de los retornos presenta ciertas propiedades estadísticas que la caracterizan [Campbell et al., 1997].

- La media muestral es cercana a cero. Es común que las autocorrelaciones no sean significativas.

- La distribución incondicional presenta colas pesadas, es leptocúrtica y asimétrica.
- La distribución no es la misma para diferentes escalas de tiempo; a medida que la escala es mayor se aproxima a una distribución normal.
- Existe una volatilidad abrupta que tiende a agruparse; días de mayor o menor volatilidad son seguidos por días con las mismas características.
- Los retornos absolutos y los retornos al cuadrado presentan un decrecimiento lento de las autocorrelaciones.

## 2.4. Algunos modelos para series de tiempo

### 2.4.1. Modelo ARIMA Y ARFIMA

Un modelo empleado tradicionalmente en el pronóstico de series de tiempo ha sido el propuesto por [Box and Jenkins, 1970] conocido como Modelo Autorregresivo de Medias Móviles Integrado (ARIMA). Para una serie de tiempo  $\{X_t, t = 1, \dots, T\}$ , el modelo  $ARIMA(p, d, q)$  se define como

$$\phi_p(L)(1-L)^d X_t = \theta_q(L)\epsilon_t, \quad (2-13)$$

donde  $L$  denota el operador de retardo,  $d$  el parámetro de diferenciación,  $\phi_p(L) = 1 - \phi_1 L - \dots - \phi_p L^p$  y  $\theta_q(L) = 1 - \theta_1 L - \dots - \theta_q L^q$  son los polinomios autorregresivos y de medias móviles, respectivamente. Si  $\{X_t\}$  es estacionario e invertible, las raíces de ambos polinomios igualados a cero se encuentran fuera del círculo unidad, además los errores aleatorios  $\epsilon_t$  son independientes e idénticamente distribuidos normales, con media cero y varianza  $\sigma^2$ .

El parámetro de diferenciación es un número entero positivo, generalmente  $d \leq 2$ . Este parámetro surge como consecuencia de la existencia de series que no satisfacen la condición de estacionariedad en media. Sin embargo, algunas series pueden volverse estacionarias al ser diferenciadas una o más veces. Es decir, si se diferencia una vez la serie no estacionaria  $X_t$ , se puede obtener una nueva serie que cumpla esta propiedad.

Si  $d = 1$ , el proceso puede ser persistente. La persistencia es entendida como el efecto permanente de un cambio unitario en la innovación sobre el nivel futuro (pronóstico) de la serie <sup>1</sup>. En este caso, se dice que el proceso tiene memoria infinita. Si  $d = 0$ , el modelo se reduce a un  $ARMA(p, q)$  con memoria corta. Para determinar los órdenes de los polinomios autorregresivos y de medias móviles se sigue el procedimiento de modelación proporcionado por [Box and Jenkins, 1970].

<sup>1</sup> Como lo señala [Cochrane, 1991], la existencia de una raíz unitaria no es sinónimo de persistencia. En este sentido, de manera informal sería más adecuado decir que una serie que tenga una probabilidad mayor de rechazar la prueba de raíz unitaria generalmente puede tener menos persistencia.

Como lo señalan algunos autores ([Granger, 1980] y [Castaño et al., 2008b]), para algunas series de tiempo, la diferenciación entera puede generar que la componente asociada a la baja frecuencia desaparezca de la serie original, deteriorando el pronóstico de largo plazo. [Granger and Joyeux, 1980] y [Hosking, 1981] introducen los modelos de memoria larga *ARFIMA*. Este proceso sigue la misma forma de (2-13) pero el parámetro de diferenciación toma un valor fraccional en un intervalo continuo de números reales. El operador de diferencia fraccional se especifica como

$$(1 - L)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-L)^k = \sum_{k=0}^{\infty} \frac{\Gamma(-d + k)}{\Gamma(k + 1) \Gamma(-d)} L^k, \quad (2-14)$$

donde  $\Gamma(\cdot)$  denota la función Gamma. Si  $-\frac{1}{2} < d < \frac{1}{2}$  el proceso es estacionario e invertible. Si  $0 < d < \frac{1}{2}$ , el proceso es estacionario con memoria larga, y si  $-\frac{1}{2} < d < 0$  el proceso será estacionario con memoria corta y tendrá una fuerte reversión a la media.

[Hosking, 1981] demuestra que cuando  $0 < d < 0,5$  la función de autocorrelación se aproxima a

$$\rho_k \approx C k^{2d-1}. \quad (2-15)$$

Por lo tanto, la función de autocorrelación decae hiperbólicamente hacia cero y  $\sum_{k=-\infty}^{\infty} |\rho_k|$  no converge. Cuando  $d < 0,5$ , la función de densidad espectral es

$$f_X(\omega) = |1 - e^{-i\omega}|^{-2d} f_W(\omega) = \left[ 2 \operatorname{sen} \left( \frac{\omega}{2} \right) \right]^{-2d} f_W(\omega) \quad 0 \leq \omega \leq \pi, \quad (2-16)$$

donde  $f_W(\omega) = \frac{\sigma_a^2}{2\pi} \left| \frac{\theta(e^{-i\omega})}{\phi(e^{-i\omega})} \right|^2$  es la densidad espectral del proceso *ARMA*( $p, q$ ),  $W_t = (1 - L)^d X_t$ . Como lo señala [Castaño et al., 2008b], para valores  $0 < d < 0,5$  la densidad espectral es una función no acotada en el origen, que decae con  $\omega$  y estará dominada en las frecuencias bajas. Para  $d < 0$  la densidad espectral se anulará en el origen y estará dominada por las frecuencias altas.

[Reisen, 1994] señala la dificultad de estimar el parámetro de diferenciación fraccional cuando existen componentes de memoria corta, ver [Smith et al., 1997]. [Boes et al., 1989] señalan que la especificación incorrecta en los órdenes del modelo *ARMA* implicaría una estimación sesgada del parámetro de diferenciación fraccional. En general, los métodos de estimación se pueden agrupar en dos grupos, métodos paramétricos que involucran la función de verosimilitud y los semiparamétricos.

El método paramétrico requiere de la especificación completa del modelo. Por otro lado, la estimación semiparamétrica se basa en un modelo de regresión para el logaritmo del periodograma especificado en la ecuación (2-16). En este trabajo se usan dos métodos semiparamétricos tradicionales; el estimador propuesto por [Geweke and Porter-Hudak, 1983], GPH, y su versión modificada propuesta por [Reisen, 1994]. Como método de estimación paramétrico se usa el propuesto por [Castaño et al., 2008b]

### 2.4.1.1. Estimador GPH del parámetro de diferenciación

Tomando logaritmos en la ecuación (2-16) y reemplazando  $\omega$  por las frecuencias discretas de Fourier  $\omega_j = \frac{2\pi j}{T}$ ,  $j = 1, 2, \dots, \left[\frac{T}{2}\right]$ , se tiene

$$\ln f_X(\omega) = \ln f_W(0) + d \ln |1 - e^{-i\omega}|^{-2} + \ln \left( \frac{f_W(\omega)}{f_W(0)} \right). \quad (2-17)$$

Agregando el periodograma de  $X_t$ ,  $I_X(\omega_j)$ , en ambos lados de la ecuación, se obtiene

$$\ln I_X(\omega_j) = \ln f_W(0) + d \ln |1 - e^{-i\omega_j}|^{-2} + \ln \left( \frac{f_W(\omega_j)}{f_W(0)} \right) + \ln \left( \frac{I_X(\omega_j)}{f_X(\omega_j)} \right), \quad (2-18)$$

donde  $I_X(\omega_j) = \frac{1}{2\pi} \left( \hat{\gamma}(0) + 2 \sum_{l=1}^{T-1} \hat{\gamma}(l) \cos(l\omega_j) \right)$  y  $\hat{\gamma}(k) = \frac{1}{T} \sum_{t=1}^{T-k} (X_t - \bar{X})(X_{t+k} - \bar{X})$  para  $k = 0, 1, 2, \dots, T-1$  corresponde a la autocovarianza muestral del proceso  $X_t$ . La expresión  $\ln \left( \frac{f_W(\omega_j)}{f_W(0)} \right) \approx 0$  para valores de  $\omega_j$  cercanos a cero. Por lo tanto se puede obtener el modelo del tipo

$$\gamma_j = c + dZ_j + e_j, \quad (2-19)$$

donde  $\gamma_j = \ln I_X(\omega_j)$ ,  $c = \ln f_W(0)$ ,  $Z_j = |1 - e^{-i\omega_j}|^{-2}$  y  $e_j = \ln \left( \frac{I_X(\omega_j)}{f_X(\omega_j)} \right)$  son variables aleatorias *iid*. El número de ecuaciones en la regresión será función del tamaño muestral  $T$ , de modo que  $m = T^\alpha$  con  $0 < \alpha < 1$ . La ecuación (2-19) puede ser estimada por mínimos cuadrados ordinarios. El estadístico de prueba para contrastar la hipótesis nula de memoria corta,  $H_0 : d = 0$ , se define como

$$GPH = \frac{\hat{d} - d}{\widehat{\text{var}}(\hat{d})} \rightarrow N(0, 1). \quad (2-20)$$

### 2.4.1.2. Estimador de Reisen

Si existe una componente autorregresiva o de medias móviles con parámetro cercano a la unidad para  $\varepsilon_t$ , el GPH estará sesgado a rechazar la hipótesis de memoria larga, aun siendo cierta, [Agiakloglou et al., 1993]. Además, como el periodograma no es un estimador consistente de la función de densidad espectral, ver [Brockwell and Davis, 2006], [Reisen, 1994] propone como alternativa sustituir la función periodograma por la función periodograma suavizado, basado en el núcleo de Parzen, para obtener estimaciones menos sesgadas y más eficientes, de modo que

$$I_s(\omega) = \frac{1}{2\pi} \sum_{j=-v}^v k \left( \frac{j}{v} \right) \hat{\gamma}(j) \cos(j\omega), \quad (2-21)$$

donde  $k(\cdot)$  es el núcleo de Parzen

$$k(u) = \begin{cases} 1 - 6u^2 + 6|u|^3 & \text{si } |u| \leq 1/2 \\ 2(1 - |u|)^3 & \text{si } 1/2 < |u| < 1/2 \\ 0 & \text{eoc.} \end{cases} \quad (2-22)$$

La estimación se obtiene mediante mínimos cuadrados ordinarios. El punto de truncamiento  $v = T^\beta$ , con  $0 < \beta < 1$ . Sin embargo, [Reisen, 1994] señala pérdida de eficiencia cuando existen componentes autorregresivas o de medias móviles de primer orden.

### 2.4.1.3. Estimador de Castaño

[Castaño et al., 2008b] proponen un método paramétrico para estimar a  $d$  a partir del modelo *ARFIMA*  $(p^*, d, 0)$ . Reespecificando el modelo *ARFIMA* en (2-13) como

$$(1 - L)^d \Pi(L) X_t = \varepsilon_t, \quad (2-23)$$

donde  $\Pi(L) = \theta_q^{-1}(L) \phi_p(L) = 1 - \Pi_1 L - \Pi_2 L^2 - \dots$ , es la componente dual autorregresiva del modelo de corto plazo *ARMA*  $(p, q)$  del modelo *ARFIMA*  $(p, d, q)$ . [Castaño et al., 2008a] proponen aproximar el polinomio infinito  $\Pi(L)$  por medio de un polinomio finito  $\Pi^*(L) = 1 - \Pi_1^* L - \Pi_2^* L^2 - \dots - \Pi_p^* L^p$  para un orden adecuado de  $p^*$ . La estimación se realiza mediante máxima verosimilitud y el estadístico para probar  $H_0 : d = 0$ , contra  $H_a : d > 0$  se define

$$t_d = \frac{\hat{d} - d}{se(\hat{d})} \rightarrow N(0, 1) \quad (2-24)$$

[Castaño et al., 2008a] demuestran que la potencia promedio de esta prueba es superior a las dos presentadas anteriormente. Asimismo, proponen usar una aproximación autorregresiva de  $p^* = \lceil T^{\frac{1}{4}} \rceil$ . Este resultado concuerda con el trabajo de [Cheung et al., 2005] que muestra la superioridad de los métodos de máxima verosimilitud sobre los métodos semiparamétricos. [Castaño, 2016] muestra a través de un estudio de simulación que el procedimiento paramétrico tiene, en general, a nivel agregado un mejor desempeño, y en [Castaño et al., 2008a] se muestra que la prueba puede emplearse en el caso en que los errores no son correlacionados serialmente pero son dependientes, como es el caso de los modelos GARCH.

### 2.4.1.4. Identificación modelo ARFIMA

Algunos autores como [Reisen, 1994], [Castaño et al., 2008a], [Castaño, 2016], proponen un procedimiento de identificación del modelo ARFIMA que comparte ciertas características con el sugerido por [Box and Jenkins, 1970] para el modelo ARMA. En particular, [Castaño, 2016] proponen el siguiente procedimiento iterativo:

1. Estimar a  $d$  en el modelo *ARFIMA*  $(p^*, d, 0)$ , Denotando esta estimación como  $\hat{d}$ .
2. Calcular  $\hat{U}_t = (1 - L)^{\hat{d}} X_t$ .

3. Usar el procedimiento de Box-Jenkins para identificar y estimar los ordenes  $p$  y  $q$ , ver [Box and Jenkins, 1970]. Estimar el modelo  $ARMA(p, q)$ ,  $\phi_p(L)\widehat{U}_t = \theta_q(L)a_t$ . El uso de criterios de información para determinar los órdenes de los modelos también es extendido.
4. Calcular la serie filtrada  $\widehat{Y}_t = \frac{\widehat{\phi}(L)}{\widehat{\theta}(L)}X_t$ .
5. Estimar a  $d$  en el modelo  $ARFIMA(0, d, 0)$ ,  $(1 - L)^{\widehat{d}}\widehat{Y}_t = a_t$ . El valor obtenido de  $\widehat{d}$  en esta etapa es el nuevo estimador para  $d$ .
6. Repetir los pasos de 2 a 5 hasta que los parámetros converjan.

Como lo señalan los autores, en la mayoría de los casos sólo son necesarias las estimaciones obtenidas en la primera interacción, pasos 1-3.

## 2.5. Modelos GARCH

A menudo el supuesto de varianza constante en las dos especificaciones anteriores es bastante restrictivo, los retornos de los activos financieros tienen una naturaleza heteroscedástica. Lo anterior, hace necesario no solo modelar la media condicional del proceso, sino también la varianza condicional. El modelo de heteroscedasticidad más común en la modelización de series de tiempo es el Modelo de Heteroscedasticidad Condicional Autorregresiva Generalizada (GARCH) propuesto por [Bollerslev, 1986] el cual es una generalización del modelo ARCH introducido por [Engle, 1982]. Dada una serie corregida por la media,  $a_t = X_t - u_t$ , donde  $u_t$  sigue una especificación ARMA o ARFIMA, el modelo  $GARCH(m, s)$  se define como

$$a_t = \sigma_t \epsilon_t, \quad (2-25)$$

$$\sigma_t^2 = \alpha_0 + \alpha(L)a_t^2 + \beta(L)\sigma_t^2,$$

donde  $\epsilon_t$  es una variable aleatoria independiente e idénticamente distribuida,  $\alpha_0 > 0$ ,  $\beta(L) = \beta_1 L + \dots + \beta_s L^s$  y  $\alpha(L) = \alpha_1 L + \dots + \alpha_m L^m$ . Cuando  $s = 0$ , se obtiene el modelo ARCH. Una condición necesaria y suficiente para que el proceso sea estacionario es  $\sum_{i=1}^m \alpha_i + \sum_{i=0}^s \beta_i < 1$ . Esta condición implica que la varianza incondicional es finita.

En muchos trabajos empíricos se ha detectado que no se cumple la condición de estacionariedad. Para resolver esta limitación, [Engle and Bollerslev, 1986] introducen el modelo GARCH Integrado (IGARCH). Asumiendo que  $n_t = a_t^2 - \sigma_t^2$ , la expresión (2-25) puede reescribirse como

$$(1 - \alpha(L) - \beta(L))a_t^2 = \alpha_0 + (1 - \beta(L))n_t, \quad (2-26)$$

donde  $n_t$  tiene media cero y no está serialmente correlacionada. Si el polinomio autorregresivo,  $1 - \alpha(L) - \beta(L)$ , tiene una raíz unitaria, se dice que el proceso es integrado. El proceso *IGARCH* ( $p, q$ ) es

$$\phi(L)(1-L)a_t^2 = \alpha_0 + (1 - \beta(L))n_t, \quad (2-27)$$

donde  $\phi(L) = (1 - \alpha(L) - \beta(L))(1 - L)^{-1}$ . Todas las raíces de  $\phi(L)$  y  $(1 - \beta(L))$  están por fuera del círculo unidad. Una extensión natural de este proceso, es reemplazar el operador de diferencia en (2-27) por el operador de diferencia fraccional definido en (2-14). Este modelo es conocido en la literatura como modelo GARCH fraccionalmente integrado (*FIGARCH*) y fue propuesto originalmente por [Baillie et al., 1996]. En la literatura se reporta evidencia de memoria larga en volatilidad, [Ding and Granger, 1996] para el índice bursátil de Japón (Nikkei) y [Liu, 2000] para el S&P 500.

La varianza condicional del *FIGARCH* ( $m, d, s$ ) está dada por

$$\sigma_t^2 = \alpha_0 [1 - B(L)]^{-1} \left\{ 1 - [1 - B(L)]^{-1} \phi(L)(1-L)^d \right\} \varepsilon_t^2 \quad (2-28)$$

ó  $\sigma_t^2 = \omega^* + \sum_{i=1}^{\infty} \lambda_i L^i \varepsilon_t^2 = \omega^* + \lambda(L) \varepsilon_t^2$ , con  $0 \leq d \leq 1$ . Las condiciones suficientes para asegurar que la varianza condicional del *FIGARCH* ( $1, d, 1$ ) es estacionaria son:  $\alpha_0 > 0$ ,  $\beta_1 - d \leq \phi_1 \leq \frac{2-d}{2}$  y  $d(\phi_1 - \frac{1-d}{2}) \leq \beta_1(\phi_1 - \beta_1 + d)$ , ver [Bollerslev and Mikkelsen, 1996]. La amplitud del proceso *FIGARCH* es  $S = \lambda(1) = 1$ . La amplitud indica la dimensión de un shock en la varianza condicional. Consecuentemente, el segundo momento de la distribución incondicional de  $\varepsilon_t$  es infinito y el proceso no será estacionario en covarianza, ver [Baillie et al., 1996] para más detalles. [Davidson, 2004] argumenta que la longitud de la memoria se incrementa cuando  $d$  se aproxima a cero. Esto contrasta con la interpretación convencional que sugiere que la memoria se incrementa conforme  $d$  aumenta. [Davidson, 2004] demuestra que este comportamiento inusual es producto de la restricción en la amplitud igual a uno y propone una generalización del proceso conocido como GARCH Hiperbólico *HYGARCH*. En la especificación anterior  $\lambda(1)$  es reemplazado por  $1 - [1 - B(L)]^{-1} \phi(L) \left\{ 1 + \alpha \left[ (1 - L)^d \right] \right\}$ .

La amplitud del proceso *HYGARCH* será  $S = 1 - \frac{\phi}{B}(1 - \alpha)$ . El modelo soluciona algunas limitaciones de la especificación anterior. El modelo hiperbólico es estacionario en covarianza, modela la amplitud y la memoria de forma separada. Finalmente, la memoria aumenta cuando  $d$  aumenta. El proceso será estacionario en covarianza, si satisface las restricciones de estacionariedad para la componente *GARCH* y  $\ln(\alpha) < 0$ , ver [Davidson, 2004] para más detalles.

Un hecho estilizado en los retornos de activos financieros es que su distribución incondicional tiene colas pesadas. La distribución Gaussiana es incapaz de capturar este fenómeno. Por lo tanto, los procesos de la clase GARCH han sido combinados con distribuciones que capturan este fenómeno. En este trabajo, además de la distribución normal, se emplea la distribución *t*-Student y la Distribución de Error Generalizado (*GED*).

Los modelos son estimados usando máxima verosimilitud, expresando la media condicional como

$$X_t = E(X_t | \Omega_{t-1}) + \epsilon_t, \quad \epsilon_t = z_t \sigma_t, \quad (2-29)$$

donde  $\Omega_{t-1}$  es el conjunto de información hasta el período  $t - 1$ , el logaritmo de la función de verosimilitud para la distribución normal será

$$L_{norm} = -\frac{1}{2} \sum_{t=1}^T [\ln(2\pi) + \ln(\sigma_t^2) + z_t^2], \quad (2-30)$$

donde  $T$  es el número de observaciones. Para la distribución  $t$ -Student se define como

$$L_{ts} = T \left\{ \ln \Gamma \left( \frac{v+1}{2} \right) - \ln \left( \frac{v}{2} \right) - \frac{1}{2} \ln(\pi(v-2)) \right\} - \frac{1}{2} \sum_{t=1}^T \left[ \ln(\sigma_t^2) + (1+v) \ln \left( 1 + \frac{z_t^2}{v-2} \right) \right], \quad (2-31)$$

donde  $v$  son los grados de libertad,  $2 < v < \infty$ , y  $\Gamma(\cdot)$  es la función *gamma*. Finalmente, para la función *GED*, tenemos

$$L_{GED} = \sum_{t=1}^T \left[ \ln \left( \frac{v}{\lambda_v} \right) - 0,5 \left| \frac{z_t}{\lambda_v} \right|^v - (1+v^{-1}) \ln(2) - \ln \Gamma \left( \frac{1}{v} \right) - 0,5 \ln(\sigma_t^2) \right], \quad (2-32)$$

donde  $0 < v < \infty$  y  $\lambda_v = \sqrt{\frac{\Gamma(\frac{1}{v}) 2^{-\frac{2}{v}}}{\Gamma(\frac{3}{v})}}$ . Esta distribución se reduce a la normal si  $v = 2$  y tiene colas más pesadas si  $v < 2$ .

## 2.6. Generalidades sobre redes neuronales

Los modelos que se han presentado hasta ahora no capturan las dinámicas no lineales que presentan algunas series de tiempo en la media condicional. Un modelo no lineal usado ampliamente en la predicción de series de tiempo es la red neuronal perceptrón multicapa (MLP) con una sola capa oculta. Las redes neuronales artificiales son modelos semiparamétricos que, dada su estructura, imitan el funcionamiento del cerebro humano. Están constituidas por unidades básicas de procesamiento de información denominadas neuronas que se interconectan entre sí y se organizan en varias capas: de entrada, ocultas y de salida. Ver gráfico **2-1**.

La red neuronal opera de la siguiente manera: la capa de entrada recibe un conjunto de variables del mundo externo  $x_i$ ,  $i = 1, \dots, k$ . Cada variable es ponderada por un factor de intensidad denotado  $\gamma_{ji}$  de forma que el input total que entra a la unidad oculta esta dado por  $u_j = \sum_i \gamma_{ji} x_i$  (o  $\mathbf{x}' \boldsymbol{\gamma}_j$ , donde  $\mathbf{x} = (1, x_1, \dots, x_k)'$  y  $\boldsymbol{\gamma}_j = (\gamma_{j0}, \gamma_{j1}, \dots, \gamma_{jk})$ ). Cada neurona

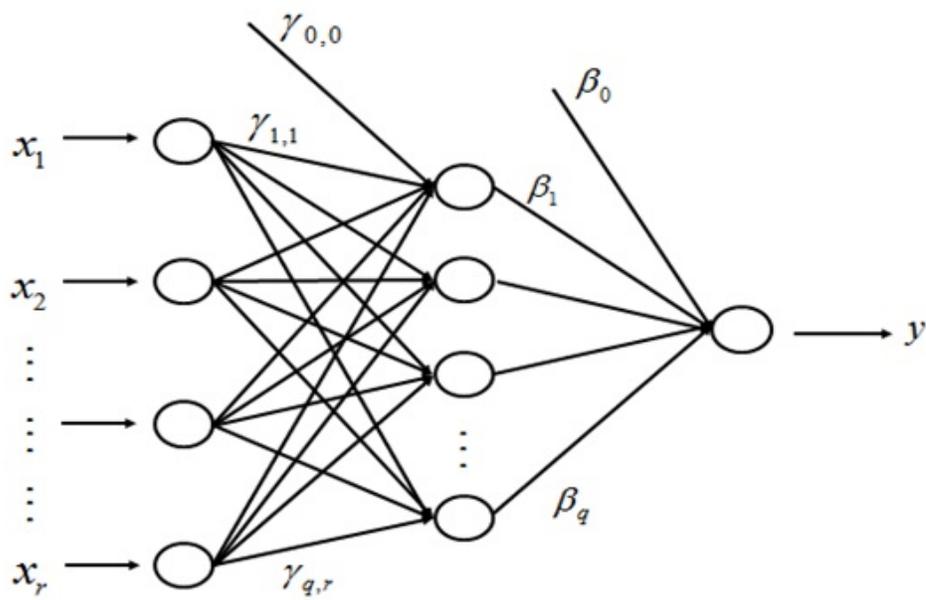


Figura 2-1: Arquitectura de una Red neuronal

de la capa oculta produce como output  $h_j = \psi(\sum_i \gamma_{ji}x_i)$ , es decir, una transformación no lineal del input total mediante el empleo de una función de activación diferenciable, acotada y no lineal. Las neuronas de la capa de salida reciben el input de la capa previa y, repitiendo el proceso antes descrito, producen el resultado final:

$$f(\mathbf{x}, \boldsymbol{\theta}) = \varphi\left(\beta_0 + \sum_{j=1}^q \beta_j \psi\left(\gamma_{j0} + \sum_i \gamma_{ji}x_i\right)\right), \quad (2-33)$$

donde  $\varphi$  es la función de activación de la capa de salida,  $\psi$  es la función de activación de la capa oculta, de tipo logístico  $\psi(u) = \frac{1}{(1+e^{-u})}$ ,  $\gamma_{ji}$  es el peso de la neurona  $i$  de la capa de entrada, a la neurona  $j$  de la capa oculta.  $\beta_j$  representa la fuerza de la conexión de la neurona  $j$ , de la capa oculta, a la capa de salida.  $n$  es el número de inputs,  $q$  es el número de unidades ocultas y  $\beta_0$ ,  $\gamma_{j0}$  corresponden a los sesgos de un modelo tradicional. Generalmente se usa la función identidad para  $\varphi$ , en este caso (2-33) se puede representar como

$$f(\mathbf{x}, \boldsymbol{\theta}) = \beta_0 + \sum_{i=1}^q \frac{\beta_j}{1 + e^{-(\gamma_{j0} + \sum_i \gamma_{ji}x_i)}} \quad (2-34)$$

Si se asume una conexión directa entre la capa de entrada y la de salida, (2-33) se puede definir como

$$f(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{x}'\boldsymbol{\delta} + \sum_{j=1}^q \beta_j \psi(\mathbf{x}'\boldsymbol{\gamma}_j), \quad (2-35)$$

donde  $\boldsymbol{\delta} = (\delta_1, \delta_2, \dots, \delta_k)$ . Tomando como variable de entrada retardos de la serie  $\{X_t\}$ , de modo que  $\mathbf{x} = (X_{t-1}, X_{t-2}, \dots, X_{t-p})'$  y el número de unidades ocultas  $q = 0$ , tendremos un modelo autorregresivo  $AR(p)$  de la forma  $f(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{x}'\boldsymbol{\delta} + \epsilon_t$ .

La red que se acaba de describir, se conoce como red *feedforward* dado que las unidades de cada capa se conectan con las unidades de la capa siguiente. Más detalles sobre este tipo de redes se pueden encontrar en [Haykin, 1994].

La red neuronal es estimada minimizando la siguiente función de costo

$$S(\boldsymbol{\theta}) = \sum_{t=1}^n |x_t - f(\mathbf{x}, \boldsymbol{\theta})|^2. \quad (2-36)$$

De modo que los parámetros  $\boldsymbol{\theta} = \{\beta_0, \dots, \beta_q, \delta_1, \dots, \delta_k, \gamma'_1, \dots, \gamma'_q\}$  se ajustan recursivamente hasta minimizar la suma de residuales al cuadrado y obtener el mejor modelo que capture el comportamiento de la serie bajo estudio. Los pesos iniciales se asignan de forma aleatoria. En cada iteración el vector es actualizado de acuerdo a una regla de aprendizaje

$$\widehat{\boldsymbol{\theta}}^{(r+1)} = \widehat{\boldsymbol{\theta}}^{(r)} + \Delta^{(r)}. \quad (2-37)$$

La definición de esta regla conduce a diversas formas de optimización. Un método común es el de gradiente descendente que tiene la forma

$$\widehat{\boldsymbol{\theta}}^{(r+1)} = \widehat{\boldsymbol{\theta}}^{(r)} - \lambda A(\boldsymbol{\theta}^{(r)})^{-1} \nabla S(\boldsymbol{\theta}^{(r)}), \quad (2-38)$$

donde el parámetro  $\lambda$  es la tasa de aprendizaje y  $A(\boldsymbol{\theta}^{(r)})$  es la matriz Hessiana de derivadas parciales de la función de costo respecto a los parámetros. En este trabajo se emplea el algoritmo BFGS sugerido por [Franses and van Dijk, 2000] para calcular dicha matriz. Además, para aumentar la capacidad de generalización de la red se emplea una componente que penalice la función de costo original. Esta penalización garantiza que cada conexión tiende al valor de cero a menos que sea importante. Para ello, se asigna una componente de regularización correspondiente al cuadrado de los pesos.

[Soltani, 2002] señala dos problemas recurrentes en el uso de redes neuronales:

- Elegir el orden del modelo: cuando el orden es pequeño, el entrenamiento es relativamente simple pero la información pasada no es suficiente para predecir adecuadamente. Cuando el orden es alto, el entrenamiento es complejo y el error de aproximación decrece lentamente.
- Complejidad del estimador: un estimador demasiado complejo conduce a un ajuste bueno dentro de la muestra de entrenamiento pero pobre para nuevos datos. Esto se conoce como sobreentrenamiento. Asimismo, un estimador muy simple puede no ser capaz de capturar la dinámica del sistema.

Como lo señala [Soltani, 2002], el desafío en la predicción de series de tiempo consiste en que el modelo sea capaz de capturar las dinámicas rápidas, correspondientes a las altas frecuencias, y cancelar el ruido simultáneamente. El sobreentrenamiento está relacionado directamente con el aprendizaje de ruido. En este sentido, [Soltani, 2002] propone el uso de filtros wavelet combinados con redes neuronales. Los wavelet actúan como un filtro pasa-bajo y pasa-alta que permiten descomponer la serie en distintas escalas, separando las dinámicas rápidas de las más lentas. Por lo tanto, el entrenamiento para la serie de tendencia y la serie de detalles, en los primeros niveles de descomposición, será más simple debido a que la razón señal-ruido será más alta comparada con la serie original.

# 3 Modelación de series de tiempo usando el análisis wavelet

## 3.1. Análisis multiescala

El análisis de wavelets se fundamenta, en gran parte, en el análisis multiescala. A continuación se ofrece una breve introducción basados en [Nason, 2008]. Se comienza desde una secuencia de datos y posteriormente se generaliza al concepto de función.

Dada una secuencia de números reales  $y = (y_1, y_2, \dots, y_n)$ , donde  $n$  es una potencia de 2, de modo que  $n = 2^J$  (diádica), el análisis multiescala busca extraer información del detalle en diferentes escalas y localizaciones. El detalle es entendido como la variación de las observaciones en una escala y localización particular.

La operación para extraer el detalle se define como

$$d_k = y_{2k} - y_{2k-1} \quad \text{para } k = 1, \dots, \frac{n}{2}, \quad (3-1)$$

de modo que si  $y_{2k}$  y  $y_{2k-1}$  son similares, el coeficiente  $d_k$  correspondiente será cercano a cero. De esta manera,  $d_k$  codifica la diferencia entre sucesivos pares de observaciones en diferentes localizaciones. Ahora bien,  $d_k$  sólo proporciona información entre los vecinos más cercanos, lo que se conoce como la escala más fina de detalle. En este punto es conveniente introducir el índice de escala  $j$ . La secuencia original contenía  $n = 2^J$  observaciones, el nivel más fino de detalle,  $d_k$ ,  $\frac{n}{2} = 2^{J-1}$  observaciones. Por lo tanto, la escala más fina de detalle se denotará con  $j = J - 1$  y  $d_k$  como  $d_{J-1,k}$  ( $d_{j,k}$ ).

Para obtener información en escalas más gruesas es necesario introducir otro operador

$$c_{J-1,k} = y_{2k} + y_{2k-1}, \quad \text{para } k = 1, \dots, \frac{n}{2}. \quad (3-2)$$

Esta secuencia puede ser vista como un promedio local escalado entre dos observaciones adyacentes. Para obtener el próximo detalle más grueso se repite la operación (3-1) sobre la secuencia  $c_{J-1,k}$  de modo que

$$d_{J-2,l} = c_{J-1,2l} - c_{J-1,2l-1}, \quad l = 1, 2, \dots, n/4. \quad (3-3)$$

Reemplazando  $c_{J-1,2l}$ , la expresión puede ser reescrita como

$$d_{J-2,l} = (y_{4l} + y_{4l-1}) - (y_{4l-2} + y_{4l-3}), \quad l = 1, 2, \dots, n/4. \quad (3-4)$$

Esta secuencia puede ser vista como una diferencia escala dos: diferencia de dos componentes que son el promedio de dos puntos adyacentes. Repitiendo la operación (3-2) para la secuencia  $c_{J-1,l}$ , obtenemos

$$c_{J-2,l} = c_{J-1,2l} + c_{J-1,2l-1}, \text{ para, } l = 1, 2, \dots, n/4. \quad (3-5)$$

De esta manera, repitiendo las operaciones (3-1) y (3-2) se pueden generar todos los coeficientes de detalle y suavizados en escalas progresivamente más gruesas. A medida que la escala aumenta, el número de coeficientes se reduce en una escala de dos. La escala más gruesa se produce en  $j = 0$ . Esta forma de calcular los coeficientes se conoce en la literatura como algoritmo piramidal. El procedimiento descrito anteriormente es un tipo de transformación wavelet discreta. Los coeficientes de detalle,  $d_{1,k}$ , se conocen como coeficientes wavelet y los coeficientes  $c_{j,k}$  se conocen como coeficientes de escalamiento o wavelet padre.

Para conservar que la norma del input,  $\|y\| = \sum_{i=1}^n y_i^2$ , sea igual a la norma del output,  $d_k$  y  $c_k$ , se redefinen las operaciones (3-1) y (3-2) como

$$d_k = \alpha (y_{2k} - y_{2k-1}) \quad (3-6)$$

$$c_k = \alpha (y_{2k} + y_{2k-1}), \quad (3-7)$$

de modo que la norma es igual a

$$d_k^2 + c_k^2 = 2\alpha^2 (y_{2k}^2 + y_{2k-1}^2) \quad (3-8)$$

Por lo tanto,  $\alpha = 2^{-1/2}$  y las operaciones serán

$$d_k = \sum_{l=-\infty}^{\infty} g_l y_{2k-l}, \text{ donde } g_l = \begin{cases} 2^{-1/2}, & l = 0 \\ -2^{-1/2}, & l = 1 \\ 0 & eoc \end{cases}, \quad (3-9)$$

$$c_k = \sum_{l=-\infty}^{\infty} h_l y_{2k-l}, \text{ donde } h_l = \begin{cases} 2^{-1/2}, & l = 0 \\ 2^{-1/2}, & l = 1 \\ 0 & eoc \end{cases}. \quad (3-10)$$

Esta nueva transformada se conoce como transformada wavelet Haar discreta. Finalmente, la transformada wavelet puede realizarse mediante una multiplicación matricial en lugar del procedimiento piramidal. De esta manera  $d = wy$ ,  $w$  generalmente es una matriz ortogonal.

## 3.2. Transformación Haar sobre funciones

En la sección precedente se presentó el análisis multiescala para una secuencia discreta de datos. A continuación se extiende el análisis para una función  $f(x)$  que se define, por simplicidad, en el intervalo  $x \in [0, 1]$ . En el caso discreto, el análisis multiescala implicaba la resta de vecinos cercanos, en el caso de funciones esto se generaliza a la resta de integrales de la función sobre pares consecutivos de intervalos.

La función wavelet Haar padre en la escala más fina,  $2^J$ , se define como  $\phi(2^J x)$ , donde<sup>1</sup>

$$\phi(x) = \begin{cases} 1, & x \in [0, 1] \\ 0 & \text{eoc} \end{cases} . \quad (3-12)$$

Los coeficientes wavelet padre se definen como

$$c_{J,k} = \int_0^1 f(x) \phi_{Jk}(x) dx = \int_0^1 f(x) 2^{J/2} \phi(2^J x - k) dx = \langle f, \phi_{Jk} \rangle. \quad (3-13)$$

Teniendo en cuenta (3-12) se puede mostrar que

$$\phi_{J,k}(x) = \begin{cases} 2^{J/2} & x \in [2^{-J}k, 2^{-J}(k+1)] \\ 0 & \text{eoc} \end{cases} , \quad (3-14)$$

es decir,  $\phi_{Jk}(x)$  es una función constante a lo largo del intervalo  $I_{J,k} = [2^{-J}k, 2^{-J}(k+1)]$  y nula en caso contrario. Además, como  $f(x)$  solo está definida en el intervalo  $[0, 1]$ , para que  $I_{Jk}$  se solape en ese intervalo es necesario que  $k$  esté definido desde 0 a  $2^J - 1$ .

Por lo tanto, (3-13) no es más que la integral de  $f(x)$  sobre el intervalo  $I_{J,k}$ ; se puede interpretar como el valor promedio de la función en dicho intervalo y representa una aproximación para  $f(x)$  como

$$f_J(x) = \sum_{k=0}^{2^J-1} c_{J,k} \phi_{J,k}(x). \quad (3-15)$$

Para computar los coeficientes en una escala más gruesa  $J-1$ , se debe calcular la integral  $f(x)$  sobre el intervalo  $I_{J-1,k}$ , como

$$c_{J-1,k} = \int_{2^{-(J-1)}}^{2^{-(J-1)}(k+1)} f(x) \phi_{J-1,k}(x) dx. \quad (3-16)$$

<sup>1</sup>Dada una función  $p(x)$ ,  $x \in R$ , su versión trasladada y escalada, díadicamente, se define como

$$p_{j,k} = 2^{j/2} p(2^j x - k), \quad j, k \text{ son enteros.} \quad (3-11)$$

$p_{j,k}$  estará centrada alrededor de  $2^{-j}k$  y el factor  $2^{j/2}$  garantiza que tenga la misma norma que  $p(x)$ .

Sin embargo, existe un vínculo interescala entre wavelets padre tomados en escalas diádicas adyacentes, de modo que

$$\phi(2^{J-1}x - k) = \phi(2^Jx - 2k) + \phi(2^Jx - 2k - 1). \quad (3-17)$$

Esta última relación permite reescribir (3-16) como

$$c_{J-1,k} = 2^{-\frac{1}{2}}(c_{J,2k} + c_{J,2k+1}). \quad (3-18)$$

Por la tanto, para computar los valores  $c_{J-2,l}$  hasta la escala más gruesa  $c_{0,m}$ , sólo se requieren los valores de los coeficientes en la escala más fina  $c_{J-1,l}$ . Cabe resaltar que los coeficientes en escalas más finas capturan la información asociada a frecuencias altas y las escalas más gruesas corresponden a frecuencias más bajas. Como se mencionó,  $f_J(x)$  representa una aproximación a la función  $f(x)$  en la escala  $J$ . Si se toman las dos aproximaciones en la escala más gruesa, se tiene

$$f_0(x) = c_{0,0}\phi(x), \quad (3-19)$$

$$f_1(x) = c_{0,0}2^{1/2}\phi(2x) + c_{1,1}\phi(2x - 1). \quad (3-20)$$

La diferencia entre las dos aproximaciones puede escribirse agrupando términos y empleando (3-18) en la forma

$$f_1(x) - f_0(x) = d_{0,0}\psi(x), \quad (3-21)$$

$$\text{donde } d_{0,0} = 2^{-\frac{1}{2}}(c_{1,1} - c_{1,0}) \text{ y } \psi(x) = \phi(2x) - \phi(2x - 1) = \begin{cases} 1 & \text{si } 0 \leq x < 1/2 \\ -1 & \text{si } 1/2 \leq x < 1 \\ 0 & \text{eoc} \end{cases}.$$

La función  $\psi(x)$  se conoce como wavelet madre Haar. De esta manera la aproximación más fina en el nivel 1 se obtiene desde la aproximación más gruesa, más un nivel de detalle encapsulado en  $d_{0,0}$ . Generalizando este resultado para todas las escalas, se tiene

$$f_{j+1}(x) = f_j(x) + \sum_{k=0}^{2^j-1} d_{jk}\psi_{jk}(x) = \sum_{k=0}^{2^j-1} c_{jk}\phi_{jk}(x) + \sum_{k=0}^{2^j-1} d_{jk}\psi_{jk}(x). \quad (3-22)$$

### 3.3. Transformación wavelet discreta

Para tener un marco teórico un poco más sólido en la construcción de wavelets, se deben extender los resultados de la sección previa a funciones más generales que la función Haar. Dicho marco es el estudio de los espacios escala de funciones, conocido como análisis por multiresolución. Como lo señala [Nason, 2008], la transformada wavelet discreta descompone una función  $f(x)$  en distintos niveles de resolución. El análisis por multiresolución es una sucesión creciente de subespacios cerrados  $\{V_j, j \in \mathbb{Z}\}$  que aproximan a  $L^2(\mathbb{R})$ , esto es

$$i) \dots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \dots$$

$$ii) \bigcup_{j \in \mathbb{Z}} V_j \text{ es denso en } L^2(\mathbb{R}).$$

$$iii) \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$$

A medida que  $J$  aumenta  $V_j$  contiene funciones con un nivel de detalle más alto. Existe una función de escalamiento en  $V_0$ ,  $\phi(x)$ , denominada wavelet padre cuya versión trasladada,  $\{\phi(x-k), k \in \mathbb{Z}\}$ , es una base ortonormal para  $V_0$ . Asimismo, su versión trasladada y escalada,  $\{\phi_{jk}(x)\}, k \in \mathbb{Z}$ , forma una base ortonormal para  $V_j$ .

De esta manera, para un nivel de resolución  $j$  la función  $f(x)$  puede ser aproximada mediante la combinación lineal

$$f(x) \approx f_j(x) = \sum_{k \in \mathbb{Z}} c_{j,k} \phi_{j,k}(x), \quad (3-23)$$

donde  $\phi_{jk}$  es una función más general que la wavelet Haar. La propiedad de ser ortogonal permite calcular los coeficientes  $c_{j,k}$  como

$$c_{j,k} = \int_{-\infty}^{\infty} f(x) \phi_{j,k}(x) dx = \langle f, \phi_{jk} \rangle. \quad (3-24)$$

Dado que  $V_0$  es un subespacio de  $V_1$  y  $\phi(x) \in V_0$ , entonces  $\{\phi_{1,k}, k \in \mathbb{Z}\}(x)$  es una base para  $V_1$  y  $\phi(x)$  se puede construir como una combinación lineal de su versión a doble escala de si misma, de la forma

$$\phi(x) = \sum_{k \in \mathbb{Z}} h_k \phi_{1k}(x). \quad (3-25)$$

Esta ecuación es conocida en la literatura como ecuación de dilatación y es la generalización de (3-17)<sup>2</sup>. La ecuación (3-23) puede ser interpretada como la proyección de la función  $f(x)$  sobre el espacio  $V_j$ . [Daubechies, 1988] introduce el operador proyección,  $P_j f$ , de modo que

$$f_j(x) = \sum_{k \in \mathbb{Z}} c_{jk} \phi_{jk}(x) = P_j f. \quad (3-26)$$

Dado que la proyección (3-26) representa una aproximación menos detallada de  $f(x)$  para el nivel de resolución  $j$ , es necesario usar proyecciones sobre otras funciones denominadas wavelets madre. [Daubechies, 1992] proporciona un resultado clave para su construcción. Si  $V_j$  con  $\phi$  forma un análisis por multiresolución de  $L^2(\mathbb{R})$ , entonces existe una base ortonormal,  $\psi_{j,k}$ , para  $L^2(\mathbb{R})$  tal que para  $j, k \in \mathbb{Z}$

$$P_{j+1} f = P_j f + \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}. \quad (3-27)$$

<sup>2</sup>En el caso de la función wavelet Haar es igual a:  $h_0 = h_1 = 1/\sqrt{2}$

Una posibilidad para la construcción de la wavelet  $\psi(x)$  es

$$\widehat{\psi}(u) = e^{\frac{iu}{2}} m_0\left(\frac{u}{2} + \pi\right) \widehat{\phi}(u/2), \quad (3-28)$$

donde  $\widehat{\psi}$  y  $\widehat{\phi}$  son las transformadas de Fourier para  $\psi$  y  $\phi$ , respectivamente;  $m_0(u) = \frac{1}{\sqrt{2}} \sum_n h_n e^{-inu}$ , o en forma equivalente,

$$\psi(x) = \sum (-1)^{n-1} \overline{h_{-n-1}} \phi_{1,n}(x). \quad (3-29)$$

La función  $\psi(x)$  se conoce como wavelet madre y los coeficientes en (3-29) pueden ser denotados como

$$g_n = (-1)^{n-1} h_{1-n}. \quad (3-30)$$

Por lo tanto, la diferencia entre las dos proyecciones  $(P_{j+1} - P_j)f$  puede ser expresada como la combinación lineal de funciones wavelets<sup>3</sup>.

A medida que el nivel de resolución  $j$  tienda a  $\infty$ ,  $f(x)$  puede ser representada generalizando (3-22) como

$$f(x) = \sum_{k \in \mathbb{Z}} c_{j_0,k} \phi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k \in \mathbb{Z}} d_{j,k} \psi_{j,k}(x), \quad (3-31)$$

donde los coeficientes  $d_{j,k} = \int_{-\infty}^{\infty} f(x) \psi_{j,k}(x) dx$ . De la misma manera que para una secuencia discreta los coeficientes wavelets en escalas gruesas son calculados de forma recursiva desde la escala más fina, también se puede generalizar este resultado para la función  $f(x) \in L^2(\mathbb{R})$ . Los coeficientes wavelet en la escala  $J-1$  se definen como

$$c_{J-1,k} = \int_{-\infty}^{\infty} f(x) \phi_{J-1,k}(x) dx \quad (3-32)$$

Desde la ecuación de dilatación se tiene

$$\phi_{J-1,k}(x) = 2^{\frac{(J-1)}{2}} \phi(2^{J-1}x - k) = \sum h_n \phi_{J,n+2k}(x). \quad (3-33)$$

Sustituyendo este resultado en (3-32) se obtiene la expresión

$$c_{J-1,k} = \sum_n h_n c_{J,n+2k}, \quad (3-34)$$

o de una manera más usual

$$c_{J-1,k} = \sum_n h_{n-2k} c_{J,n}. \quad (3-35)$$

<sup>3</sup>En el caso de las wavelets Haar  $g_0 = g_1 = 1/\sqrt{2}$

Por otra parte, a partir de la ecuación de escalado (3-30) se pueden derivar de forma similar los coeficientes wavelet

$$d_{J-1,k} = \sum_n g_{n-2k} c_{J,n}. \quad (3-36)$$

Para la escala más fina los coeficientes se calculan de forma determinística como sigue. Considere la función en un dominio discreto, tal que

$$f(n) = \sum_k \langle f, \phi_{0,k} \rangle \phi(n-k), \quad n \in \mathbb{Z} \text{ y } f \in V_0. \quad (3-37)$$

Aplicando la transformada de Fourier y ordenando los términos, se tiene

$$\sum_n f(n) e^{-iun} = \Theta(u) \sum_k \langle f, \phi_{0,k} \rangle e^{-iuk}, \quad (3-38)$$

donde  $\Theta(u) = \sum_m \phi(m) e^{-ium}$ ,  $m = (n-k)$ , es la transformada discreta de Fourier de  $\phi(n-k)$ . Despejando términos en (3-38),

$$\sum_k \langle f, \phi_{0,k} \rangle e^{-iuk} = \Theta^{-1}(u) \sum_n f(n) e^{-iun}, \quad (3-39)$$

y aplicando la transformada inversa de Fourier

$$\langle f, \phi_{0,k} \rangle = \sum_n a_{n-k} f(n) = c_{0,k}, \quad (3-40)$$

donde  $a_m = (2\pi)^{-1} \int_0^{2\pi} e^{-ium} \Theta^{-1}(u) du$ .

La DWT para una serie de tiempo,  $X = \{x_t, t = 1, \dots, N\}$ , se define como la transformación ortogonal

$$\omega = WX \quad (3-41)$$

de donde,

$$X = W^T \omega, \quad (3-42)$$

$\omega = \{\omega_n, n = 0, 1, \dots, N-1\}$  es un vector columna de coeficientes wavelet de dimensión  $N = 2^J$  y  $W \in \mathbb{R}^{N \times N}$  es una matriz ortogonal. En términos de filtros, la DWT es un filtro  $\{(h_n, n \in \mathbb{Z})\}$  con a lo más  $L$  valores no nulos. Las filas de la matriz  $W$  se componen de diferentes  $h_j$  y ceros.

Escribiendo  $\omega = (\omega_1, \dots, \omega_J, v_j)'$ , donde los primeros  $\frac{N}{2}$  elementos se denotan como  $\omega_1 = \{\omega_{1,t} = 0, 1, \dots, \frac{N}{2} - 1\}$  se define

$$\omega_{1,t} = \sum_{j=0}^{L-1} h_j x_{2t+1-j} \quad (3-43)$$

y

$$2t + 1 = j(\text{mod}N). \quad (3-44)$$

Los coeficientes de detalle se pueden definir como  $d_j = W^T \omega_j$ ,  $j = 1, 2, \dots, J$  y la serie  $x_t$  se puede expresar en términos de una combinación lineal de los coeficientes de detalle como

$$x_t = \sum_{j=1}^{J+1} d_{j,t}, \quad t = 0, \dots, N - 1. \quad (3-45)$$

Esta descomposición será útil cuando se defina la metodología híbrida. [Daubechies, 1988] construye wavelets de soporte compacto (funciones no nulas solo en el intervalo  $[0, 1]$ ) más suavizados que las wavelets Haar. A partir de la solución a la ecuación de dilatación (3-25), propone una familia de wavelets ortonormales. Cada wavelet es indexado por un número  $N$  que hace referencia al número de momentos de desvanecimiento. Una función  $\psi \in L^2(\mathbb{R})$  se dice que tiene  $m$  momentos de desvanecimiento si satisface

$$\int_{-\infty}^{\infty} x^l \psi(x) dx = 0 \quad \text{para } l = 0, \dots, m - 1. \quad (3-46)$$

Esta propiedad es importante porque todos los coeficientes wavelet para un polinomio de grado  $m$  o menos, serán iguales a cero. Lo anterior permite una compresión de los datos que tiene ventajas en el momento de la estimación. La longitud de los coeficientes  $h_n$ , corresponde a dos veces el número de momentos de desvanecimiento. En la Figura **3-1** se muestran algunas wavelets Daubechies calculados para distintos momentos de desvanecimiento  $p$ , cada filtro tendrá un tamaño  $N = 2p$ . En la parte izquierda se muestra la wavelet madre y en la derecha la wavelet padre. Puede apreciarse que  $p = 1$  representa la wavelet Haar<sup>4</sup>.

Como puede observarse, a medida que aumenta el número de coeficientes mayor es el número de momentos nulos y más suave será la curva de la wavelet. [Daubechies, 1992] construye wavelets que buscan minimizar el grado de asimetría conservando la propiedad de ser ortogonales. En la Figura **3-2** se muestra dicha familia.

Para la elección de la base wavelet se recomienda un número de momentos de desvanecimiento que sea razonable. Si se elige un número pequeño se puede originar variaciones bruscas

<sup>4</sup>La familia Daubechies fue generada desde la librería wavethresh de R Project. En este trabajo, excepto donde se especifique, se emplea el entorno R Project para computar las distintas pruebas y estimar los modelos. En cada sección se referenciarán las librerías usadas.

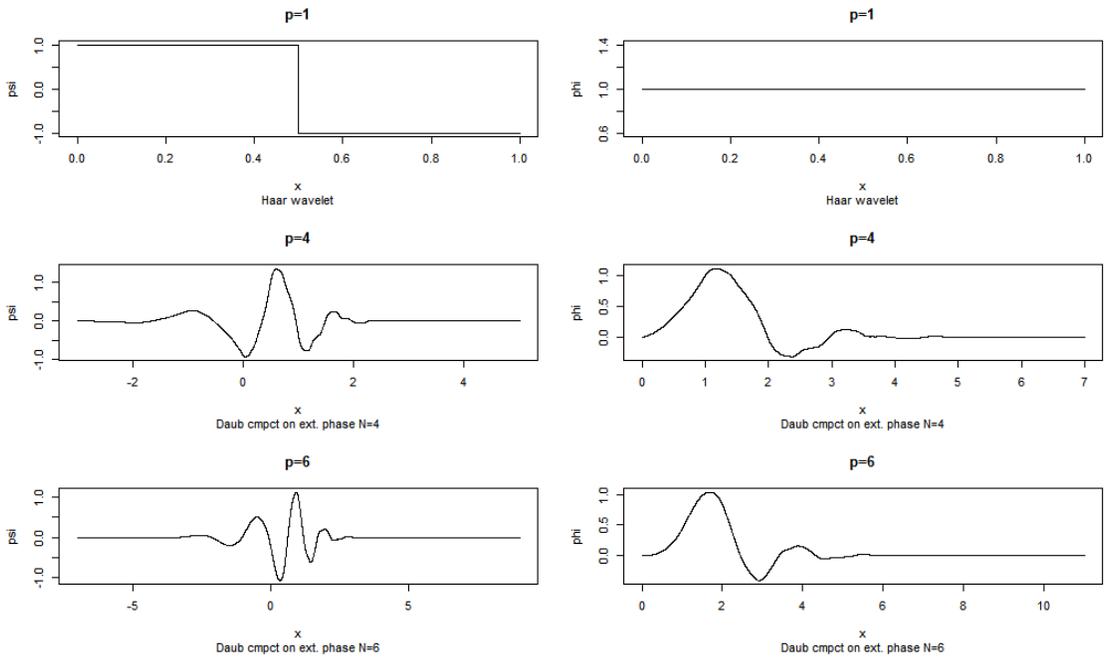


Figura 3-1: Familia Daubechies

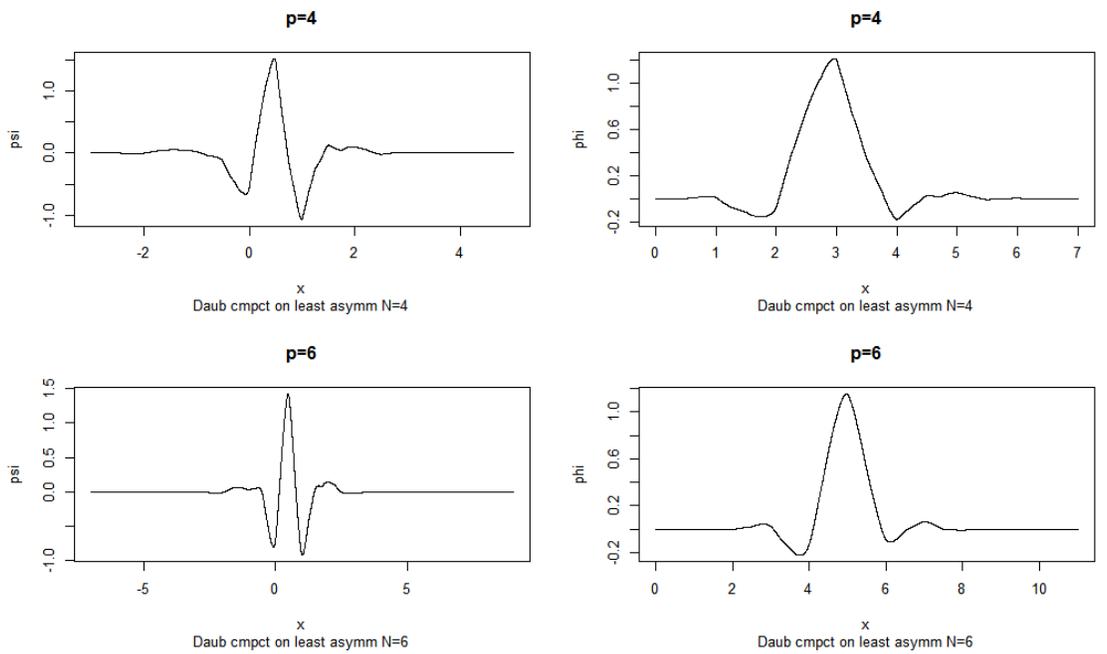


Figura 3-2: Familia Daubechies mínima asimetría

artificiales en el momento de reconstruir la serie. Por el contrario, si el número de coeficientes es muy alto, estarán influenciados por las condiciones de borde y el esfuerzo computacional podría ser mayor. [Fernandez, 2007] señala que una elección común en la literatura es el filtro de mínima asimetría de orden  $N = 8$ .

### 3.3.1. Transformación Wavelet No-Diezmada

Las operaciones descritas en (3-35) y (3-36) pueden expresarse de forma diferente. Como señala [Nason, 2008], si se realiza la convolución

$$c_{j-1,k}^* = \sum_n h_{n-k} c_{j,n}, \quad (3-47)$$

entonces se puede extraer  $c_{j-1,k}$  mediante la operación  $c_{j-1,k} = c_{j-1,2k}^*$ , lo que se conoce como diezmado diádico. Usando el operador de diezmado introducido por [Nason and Silverman, 1995] se tiene que<sup>5</sup>:

$$c_{j-1} = D_0 H c_j, \quad (3-48)$$

y

$$d_{j-1} = D_0 G c_j, \quad (3-49)$$

donde  $H$  y  $G$  denotan las operaciones de filtrado regular. Note que, el diezmado solamente selecciona los elementos pares de la secuencia. Sin embargo, se pueden seleccionar sólo los elementos impares eligiendo una base ortonormal distinta. Introduciendo el operador de diezmado impar  $(D_1 x)_l = x_{2l+1}$  se pueden calcular los coeficientes wavelets madre y padre. En general, para cada nivel  $j$  se puede elegir un tipo de diezmado par o impar lo que da lugar a distintas descomposiciones ortogonales de la secuencia original. Por lo tanto, una base particular puede representarse como  $\epsilon = \epsilon_{J-1} \epsilon_{J-2} \dots \epsilon_0$ , donde  $\epsilon_j$  es uno si se emplea  $D_1$  para calcular el nivel, o cero en caso contrario. Esta variante se conoce como diezmado- $\epsilon$ . Según [Nason, 2008], si se desea extraer toda la información de  $f(x)$ , empleando el análisis por multiresolución, es necesario conservar tanto el conjunto de coeficientes wavelets original como el generado por el diezmado impar. Sin embargo, la información será redundante y la transformada estará sobredeterminada. Este último método se conoce como transformación no diezmada. Una de la transformaciones no diezmadas más comunes es la Transformación wavelet de máximo traslape (MODWT).

<sup>5</sup>El operador de diezmado se define para una secuencia  $x_l$  como

$$(D_o x)_l = x_{2l}$$

### 3.4. Proceso Wavelet Localmente Estacionario (LSW) y pronóstico

Los hechos estilizados que siguen los retornos logarítmicos, enumerados en la introducción, frecuentemente son modelados mediante especificaciones que siguen el supuesto de estacionariedad como los modelos GARCH. Sin embargo, esas regularidades empíricas también pueden ser generadas por series que no son estacionarias.

Una de las consecuencias de la no estacionariedad es el hecho de que la función de autocorrelación (ACF) muestral no es una herramienta válida para estimar la ACF poblacional. Por lo tanto, el efecto de memoria larga reportado en la literatura sobre los retornos y los retornos al cuadrado, pueden ser un efecto espureo producto de la no estacionariedad.

Un proceso que sea estacionario tiene la siguiente representación espectral de Fourier

$$X_t = \int_{(-\pi, \pi)} A(u) \exp(iut) dZ(u), \quad t \in \mathbb{Z}, \quad (3-50)$$

donde  $Z(u)$ , es un proceso estocástico con incrementos ortonormales. Es decir,  $\text{cov}(dZ(u), dZ(u')) = \delta(u - u') du$ . Por lo tanto, el proceso  $X_t$  es la suma de una colección de funciones sinusoidales  $\exp(iut)$ , en distintas frecuencias  $u \in (-\pi, \pi)$ . La oscilación en la frecuencia  $u^*$  es aumentada por la amplitud  $A(u)$ , el proceso será estacionario si la amplitud no depende del tiempo.  $A(u)$  se relaciona con el espectro mediante  $f(u) = |A(u)|^2$ .

[Fryzlewicz et al., 2003] señala que una gran cantidad de series de tiempo presenta una estructura de segundo orden (en varianza o covarianza) que varía en función del tiempo y, por lo tanto, su estructura espectral es variable. Para este tipo de procesos es necesario que la amplitud dependa del tiempo. En la literatura se destacan dos alternativas de análisis

- Asumir estacionariedad por partes y encontrar tramos de homogeneidad de forma óptima, pero bajo este enfoque, se introducen discontinuidades artificiales, [Mallat et al., 1998].
- Reemplazar la representación de Fourier por funciones base más generales. [Nason et al., 2000] usan una representación basada en wavelets no diezmados. Por lo tanto, se pasa de una representación tiempo-frecuencia a una representación tiempo-escala. Esta metodología ofrece una representación multiescala de la covarianza y su estimación se construye a partir del periodograma wavelet.

[Nason et al., 2000] introduce una clase de procesos estocásticos denominados Wavelets Localmente Estacionarios (LSW). Esta metodología es útil en la modelización y análisis de series de tiempo que exhiben no estacionariedad.

Como se mencionó antes, es posible obtener los coeficientes wavelet madre y padre en escalas gruesas desde la escala más fina. Este resultado puede invertirse de modo que se puedan

obtener los coeficientes en escalas finas desde la escala más gruesa. Dada la ecuación (3-18) y generalizando (3-21), se tiene

$$c_{j-1,2k} = \frac{(c_{j,2k} + c_{j,2k+1})}{\sqrt{2}} \quad (3-51)$$

y

$$d_{j-1,k} = \frac{(c_{j,2k} - c_{j,2k+1})}{\sqrt{2}}, \quad (3-52)$$

de donde

$$c_{j,2k} = \frac{(c_{j-1,k} + d_{j-1,k})}{\sqrt{2}}, \quad (3-53)$$

$$c_{j,2k+1} = \frac{(c_{j-1,k} - d_{j-1,k})}{\sqrt{2}}. \quad (3-54)$$

[Mallat, 1989] muestra que para wavelets generales, esta relación puede definirse como

$$c_{j,n} = \sum_k h_{n-2k} c_{j-1,k} + \sum_k g_{n-2k} d_{j-1,k}, \quad (3-55)$$

donde  $h_n$  y  $g_n$  se conocen como filtros en espejo y se definen como antes. Dados estos dos filtros, [Nason et al., 2000] propone la construcción de las wavelets  $\psi_j = (\psi_{j,0}, \dots, \psi_{j,(N_j-1)})$ , para la escala  $j = -1, -2, \dots, -J$ , mediante una transformación

$$\psi_{-1,n} = \sum_k g_{n-2k} \delta_{0,k} = g_n \quad \text{para } n = 0, \dots, N_{-1} - 1, \quad (3-56)$$

$$\psi_{(j-1),n} = \sum_k h_{n-2k} \psi_{j,k} \quad \text{para } n = 0, \dots, N_{j-1} - 1,$$

con  $N_j = (2^{-j} - 1)(N_h - 1) + 1$ ,  $\delta_{0,k}$  el delta de kronecker y  $N_h$  el número de elementos no cero en  $h_k$ . Por ejemplo, para la wavelet Haar  $N_h = 2$  y por lo tanto  $N_j = 2^{-j}$ . En este sentido, los wavelets  $\psi_j$  definidos en (3-56) se representan, para esta familia wavelet, como

$$\psi_{-1,n} = g_n \quad \text{para } n = 0, 1, \quad (3-58)$$

$$\psi_{(j-1),n} = \sum_k h_{n-2k} \psi_{j,k} \quad \text{para } n = 0, \dots, 2^{-(j-1)} - 1.$$

Para las escalas  $j = -1, -2$  y  $-3$   $\psi_{-1} = (g_0, g_1)$ ,  $\psi_{-2} = (h_0g_0, h_1g_0, h_0g_1, h_1g_1)$ ,  $\psi_{-3} = (h_0^2g_0, h_0h_1g_0, h_0h_1g_0, h_1^2g_0, h_0^2g_1, h_0h_1g_1, h_0h_1g_1, h_1^2g_1)$  y así sucesivamente.

El proceso LSW para una serie de tiempo,  $\{X_{t,T}, t = 0, \dots, T-1; T = 2^J \geq 1\}$ , se representa como

$$X_{t,T} = \sum_{j=-J}^{-1} \sum_{k=0}^{T-1} \omega_{j,k;T} \psi_{j,k}(t) \varepsilon_{j,k}, \quad (3-60)$$

donde  $\varepsilon_{j,k}$  es una secuencia aleatoria que incrementa ortonormalmente, de esta manera,  $X_{t,T}$  puede representarse como una combinación lineal de funciones oscilatorias  $\psi_{j,k}$  con amplitudes aleatorias  $\omega_{j,k;T} \varepsilon_{j,k}$ . En el proceso LSW,  $X_{t,T}$  también tiene media cero,  $E(\varepsilon_{j,k}) = 0$  y  $cov(\varepsilon_{j,k}, \varepsilon_{l,m}) = \delta_{j,l} \delta_{k,m}$ , siendo  $\delta_{j,l}$  el delta Kronecker. Para cada  $j \leq -1$  existe una función continua Lipschitz,  $W_j(z)$ , para  $z \in (0, 1)$  que controla la velocidad de evolución de  $\omega_{j,k;T}$  y satisface que

$$\sup_k \left| \omega_{j,k;T} - W_j\left(\frac{k}{T}\right) \right| \leq \frac{C_j}{T}, \quad (3-61)$$

donde los  $C_j$  son constantes tales que  $\sum_{j=-\infty}^{-1} C_j < \infty$ .  $W_j(z)$  debe satisfacer ciertas condiciones de regularidad y determina como la varianza se distribuye a través de cada escala  $j$  y localización  $z$ . Por lo tanto, el espectro wavelet (EWS) se define como

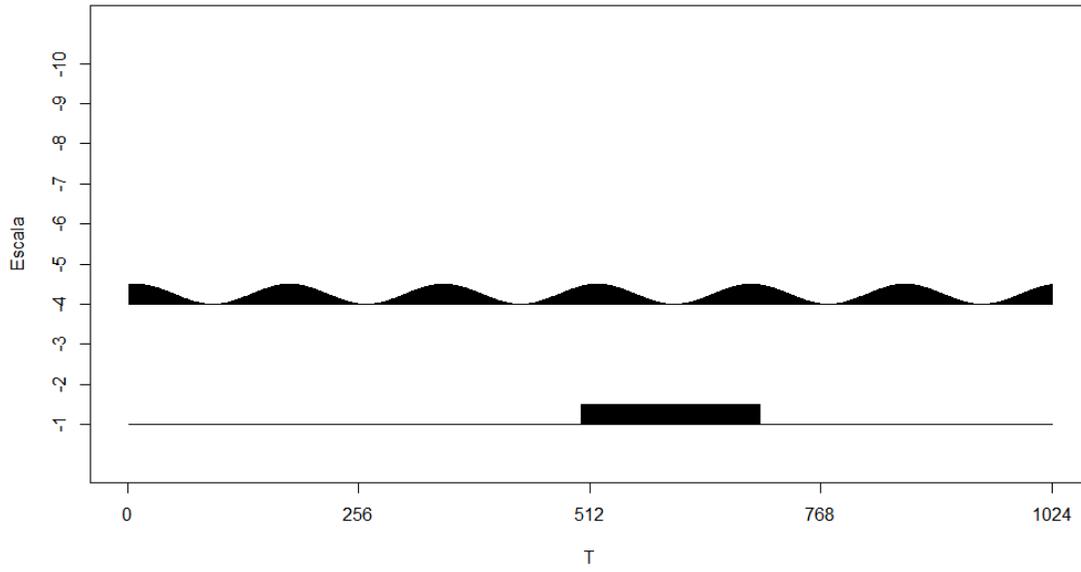
$$S_j(z) = |W_j(z)|^2 \text{ para } j = -1, -2, \dots, -J(T). \quad (3-62)$$

Las funciones  $S_j(z)$  y  $W_j(z)$  se definen sobre el intervalo  $z \in (0, 1)$  y no sobre el tiempo observado  $(0, \dots, T-1)$ . Por lo que  $z$  se conoce como tiempo reescalado y se define como el mapeo  $t = [zT]$ .

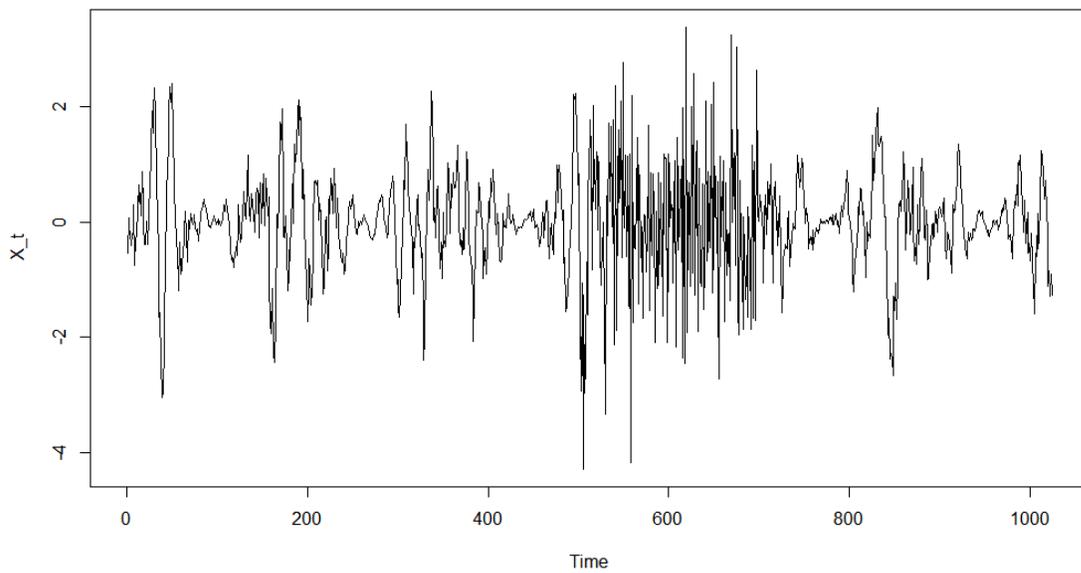
En la Figura **3-3**, para una serie de longitud  $T = 1024$ , se muestra el espectro  $S_j(z)$  que varía en función del tiempo reescalado  $z = k/T$ , donde  $k = 1, \dots, T$ . Este espectro se define como

$$S_j(z) = \begin{cases} \cos^2(6\pi z), & \text{para } j = -4, z \in (0, 1) \\ 1, & \text{para } j = -1, z \in (500/1024, 700/1024) \\ 0 & \text{eoc} \end{cases} \quad (3-63)$$

para  $j = -1, \dots, -J$ , donde  $J = \log_2(T) = 4$ . A partir de este espectro se simula un proceso LSW<sup>6</sup>, ver Figura **3-4**. El cambio de la naturaleza sinusoidal del espectro en la escala  $j = -4$ ,



**Figura 3-3:** Espectro  $S_j(z)$



**Figura 3-4:** Proceso LSW simulado

hacia un valor constante en la escala  $j = -1$ , se ve reflejado en el proceso LSW simulado como un cambio de alta frecuencia en el segmento (500, 700).

De la misma manera que existe una relación entre el espectro y la autocovarianza en el caso estacionario, también existe una relación entre el espectro wavelet y la autocorrelación wavelet. La función de autocovarianza del proceso LSW depende del tiempo, y se representa como

$$\text{cov}(z, \tau) = \sum_{j=-\infty}^{-1} S_j(z) \Psi_j(\tau), \quad (3-64)$$

donde  $\Psi_j(\tau) = \sum_k \psi_{j,k} \psi_{j,k}(\tau)$ , para  $j < 0$ ,  $\tau \in \mathbb{Z}$  y  $z \in (0, 1)$ , es la función de autocorrelación wavelet. Se supone que  $\sum_{\tau=0}^{\infty} \sup |cov(z, \tau)| < \infty^7$ . Por lo tanto, la varianza local se define como  $\sigma^2(z) = cov(z, 0) = \sum_{j=-\infty}^{-1} S_j(z)$ . [Fryzlewicz et al., 2003] asumen que las innovaciones  $\varepsilon_{j,k}$  son *iid*  $N(0, 1)$  y muestra que si la función de autocorrelación wavelet es construida a partir de un wavelet Daubechies (incluyendo el wavelet Haar), se tiene que

$$\sum_{j=-\infty}^{-1} 2^j \Psi_j(\tau) = \delta_0(\tau). \quad (3-65)$$

Dos casos particulares de un proceso LSW es el ruido blanco y el ruido blanco gaussiano modulado temporalmente (TMWN). El primero se define como  $X_{t,T} = Z_t$ , donde  $Z_t$  es *iid*  $N(0, 1)$  con  $S_j = 2^j$  y el proceso TMWN depende de  $Z_t$  en la forma

$$X_{t,T} = \sigma\left(\frac{t}{T}\right) Z_t. \quad (3-66)$$

La desviación estándar  $\sigma(z)$  es una función Lipschitz cuyas contantes deben satisfacer ciertas condiciones, ver [Fryzlewicz, 2005].

[Nason et al., 2000] proponen para la estimación del espectro usar el periodograma wavelet

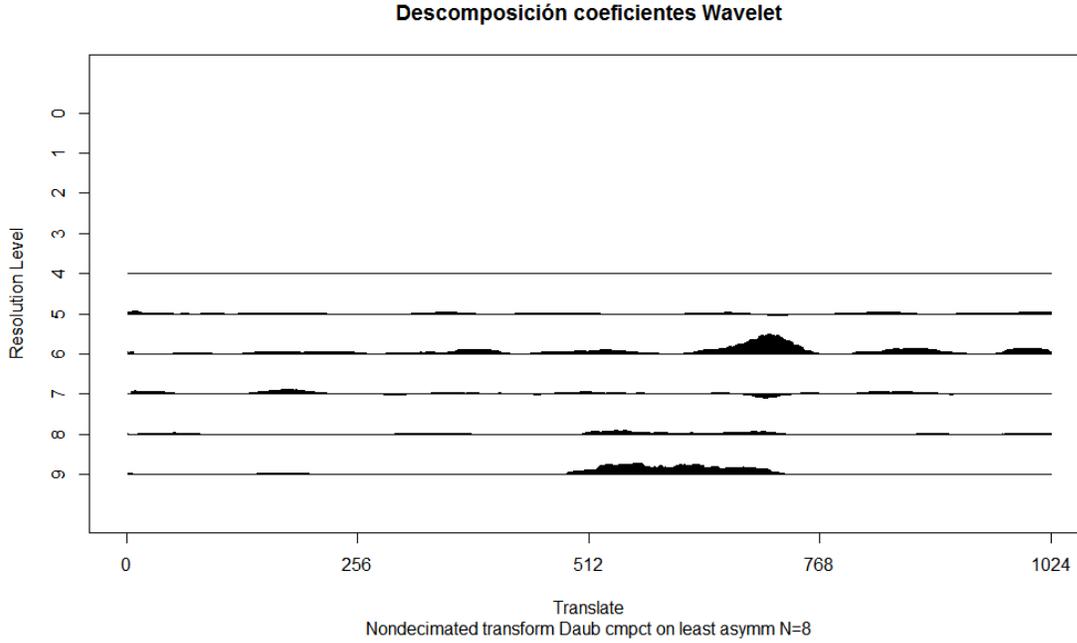
$$I_{jk} = \left| \sum_t X_{t,T} \psi_{j,p-t} \right|^2, j = -1, -2, \dots, -J \text{ y } k = 0, 1, 2, \dots, T-1. \quad (3-67)$$

La  $E[I_{jp}] = \sum_{i=-\infty}^{-1} S_i\left(\frac{p}{T}\right) A_{ij} + O\left(\frac{2^{-j \log T}}{T}\right)$ , donde  $A_{ij} = \sum_{\tau} \Psi_i(\tau) \Psi_j(\tau) = \langle \Psi_i(\tau), \Psi_j(\tau) \rangle$ . [Fryzlewicz et al., 2003] señalan que para asegurar la invertibilidad de  $A$ ,  $\Psi_j$  debe ser construida usando la wavelet Haar. El periodograma es un estimador sesgado del espectro. Sin embargo, el sesgo puede ser corregido mediante la operación

$$\mathbf{L}_p = A^{-1} \mathbf{I}_p, \quad (3-68)$$

<sup>6</sup>Para simular el proceso LSW se emplea la librería wavethresh.

<sup>7</sup>Por lo tanto, la covarianza es absolutamente sumable para cada  $z$  y el proceso será de memoria corta.



**Figura 3-5:** Descomposición Coeficientes Wavelet

donde  $\mathbf{I}_p = \{I_{jp}\}_{j=-1, \dots, -J}$  es el vector de periodogramas wavelet. De esta manera se obtiene un estimador aproximadamente insesgado. Sin embargo, la varianza es

$$\text{var}(I_{jp}) = 2 \left( \sum_i S_i \left( \frac{p}{T} \right) A_{ij} \right)^2 + O \left( \frac{2^{-j} \log T}{T} \right), \quad (3-69)$$

es decir, el estimador no es consistente. Para solucionar este problema, [Nason et al., 2000] propone suavizar el periodograma wavelet.

En la Figura 3-5 se muestra el periodograma wavelet estimado para la serie simulada <sup>8</sup>. Como puede observarse, la naturaleza cambiante del espectro wavelet se ve reflejada en la parte inferior del gráfico, escala  $j = 9$ . Esta escala corresponde en la notación presentada antes a  $j = -1$ . En esta escala, se observa que el espectro es dominado en el segmento temporal (500, 700). Asimismo, se observa para la escala  $j = 6$  ( $j = -4$  en nuestra notación) que el espectro es relativamente homogéneo.

### 3.4.1. Detección de no estacionariedades

[Nason, 2013] propone un test para probar estacionariedad de segundo orden utilizando los coeficientes Haar. A partir de la expresión  $\beta_l(z) = \sum_{l=-J}^{-1} S_l(z) A_{lj}$ , donde  $J = -\log_2(T)$ ,

<sup>8</sup>Para la descomposición de los coeficientes wavelets, se emplea la librería locits.

se puede determinar la estacionariedad de la serie si para todo  $l \in N$ ,  $\beta_l(z)$  es una función constante de  $z \in (0, 1)$ . Para esto, se examinan sus coeficientes Haar, de modo que

$$v_{i,p}^{(l)} = \int_0^1 \beta_l(z) \psi_{i,p}^H(z) dz \quad (3-70)$$

para  $i = -1, -2, \dots, -J$ ,  $p = 1, \dots, 2^i - 1$  y  $\psi_{i,p}^H$  son los wavelets Haar usuales. Si cualquier  $v_{i,p}^{(l)}$  no es cero, entonces  $\beta_l(z)$  no será constante, la serie no es estacionaria y existirá evidencia de un proceso LSW. Ya que  $\beta_l(z)$  no es conocido y es reemplazado por el estimador de  $I_{l,k}$  para obtener la estimación de  $\tilde{v}_{i,p}^{(l)}$  y probar la hipótesis del tipo

$$H_0 : v_{i,p}^{(l)} = 0 \text{ para distintos valores de } l, i, p \quad H_1 : v_{i,p}^{(l)} \neq 0 \quad (3-71)$$

En la práctica, los coeficientes wavelet estimados para cada escala  $l$  del periodograma, están dados por

$$\tilde{v}_{i,p}^{(l)} = 2^{-\frac{i}{2}} \left( \sum_{r=0}^{2^{i-1}-1} I_{l,2^i p-r} - \sum_{q=2^{i-1}}^{2^i-1} I_{l,2^i p-q} \right) \quad (3-72)$$

para el mismo rango de  $i$  y  $p$ . El estadístico de prueba se define como

$$T_{i,p}^{(l)} = \tilde{v}_{i,p}^{(l)} \tilde{\sigma}_{ip}^{(l)-1}, \quad (3-73)$$

donde  $\tilde{\sigma}_{ip}^{(l)2} = 2T^{-1} I_{l,<1,T>}^2$ , con  $I_{l,<1,T>}^2 = T^{-1} \sum_t I_{l,t}^2$ . El estadístico de prueba es comparado con los valores críticos de una distribución normal. La prueba requiere realizar múltiples hipótesis  $H_0 : v_{i,p}^{(l)}$  para distintos valores de  $l, i, p$ . Para lo anterior, [Nason, 2013] propone el uso de la Tasa de Descubrimientos Falsos (FDR), método propuesto por [Benjamini and Hochberg, 1995]. La localización y la extensión de la no estacionariedad puede representarse gráficamente sobre la serie de tiempo original. En este sentido se identifica tanto la escala  $l$  del periodograma como la escala wavelet  $i$  donde se realiza el test.

### 3.4.2. Pronóstico de un proceso LSW

[Fryzlewicz et al., 2003] proponen realizar el pronóstico de una serie de tiempo a partir de un proceso LSW. Dada una secuencia de observaciones  $X_{0,T}, X_{1,T}, \dots, X_{t-1,T}$  definen el predictor  $h$  pasos adelante como

$$\hat{X}_{t-1+h,T} = \sum_{s=0}^{t-1} b_{t-1-s,T}^{(h)} X_{s,T}, \quad (3-74)$$

donde los coeficientes  $b_{t-1-s}^{(h)}$  minimizan el Error Cuadrático Medio de Predicción (MSPE) definido como:<sup>9</sup>

$$MSPE \left( \hat{X}_{t+h-1,T}, X_{t+h-1,T} \right) = E \left( \hat{X}_{t+h-1,T} - X_{t+h-1,T} \right)^2 = b'_{t+h-1} \Sigma_{t+h-1,T} b_{t+h-1}, \quad (3-75)$$

donde  $\Sigma_{t+h-1,T}$  es la matriz de covarianza de  $X_{0,T}, \dots, X_{t+h-1,T}$  y  $b_{t+h-1}$  es el vector  $\left( b_{t-1}^{(h)}, \dots, b_0^{(h)}, b_{-1}^{(h)}, \dots, b_{-h}^{(h)} \right)$  con  $b_{-1}^{(h)}, \dots, b_{-h+1}^{(h)} = 0$  y  $b_{-h}^{(h)} = -1$ . El MSPE puede ser aproximado como  $b'_{t+h-1} B_{t+h-1,T} b_{t+h-1}$ , donde  $B_{t+h-1,T}$  es una matriz de dimensión  $(t+h) \times (t+h)$  cuyo  $(m, n)$ -ésimo elemento está dado por

$$\sum_{j=-J}^{-1} S_j \left( \frac{n+m}{2T} \right) \Psi_j(n-m). \quad (3-76)$$

Sin embargo, la matriz  $\Sigma_{t+h-1,T}$  depende de  $\omega_{j,k;T}^2$ , el cual no puede ser estimado dado que no es identificable. Por lo tanto, se requiere la estimación de  $S_j(z)$ . Para encontrar los valores óptimos de  $b_s^{(h)}$  se resuelve el sistema

$$\sum_{m=0}^t b_m^t \left\{ \sum_{j=-J}^{-1} S_j \left( \frac{n+m}{2T} \right) \Phi_j(m-n) \right\} = \sum_{j=-J}^{-1} S_j \left( \frac{n+t}{2T} \right) \Phi_j(t-n). \quad (3-77)$$

Usando (3-64), la expresión puede escribirse como

$$\sum_{m=0}^t b_m^t cov \left( \frac{m+n}{2T}, m-n \right) = cov \left( \frac{t+n}{2T}, t-n \right) \quad (3-78)$$

para  $n = 0, 1, \dots, t$ . [Fryzlewicz, 2005] propone una estimación no suavizada de la autocovarianza local como

$$\widehat{cov}(z, \tau) = \sum_{j=-J}^{-1} \left( \sum_i A_{ij}^{-1} I_{i,[z\tau]} \right) \Phi_j(\tau) \quad (3-79)$$

Posteriormente, la estimación es suavizada usando un kernel gaussiano con un ancho de banda adecuado. A medida que se incorporan más observaciones en (3-74), el error global en la estimación de los coeficientes de predicción,  $b_t = b_s^{(t)}$ ,  $s = 0, 1, \dots, t$ , aumenta debido a la no estacionariedad; sin embargo, el error de predicción teórico del MSPE decrece. Para lograr un equilibrio entre estos dos tipos de errores, [Fryzlewicz, 2005] propone acortar el predictor lineal en algún rezago en el pasado, de modo que

$$\hat{X}_{t+1,T}^{(p)} = \sum_{s=t-p+1}^t b_s^{(t)} X_{s,T}. \quad (3-80)$$

<sup>9</sup>Note que para un horizonte de predicción  $h$ , la última observación se denota como  $X_{t-1,T} = X_{t-h-1,T}^T$ , mientras que el último pronóstico posible será  $X_{T-1,T}$ . Por lo tanto, en el tiempo reescalado, el espectro wavelet evolutivo  $S_T(z)$  solo puede ser estimado en el dominio  $[0, 1 - \frac{h+1}{T}]$ .

Por lo tanto, en el momento de realizar el pronóstico se deben elegir dos parámetros, el parámetro de rezago  $p$  y el ancho de banda  $h$  para suavizar la autocovarianza local. [Fryzlewicz, 2005] recomienda el siguiente procedimiento adaptativo para la elección de los parámetros. Se elige un par inicial  $(p_0, h_0)$  y sólo se utiliza un segmento de entrenamiento hasta el período  $t - s$ ,  $(X_{0,T}, X_{1,T}, \dots, X_{t-s,T})$ . Con ese par se pronostica  $X_{t-s+1,T}$ . Asimismo, se realizan otros 8 pronósticos alternativos, usando vecinos cercanos del par inicial, definidos como  $(p_0 \pm 1, h_0 \pm \delta), (p_0, h_0 \pm \delta), (p_0 \pm 1, h_0)$ , para un  $\delta$  fijo. Dado que se conoce  $X_{t-s+1,T}$ , se escoge el par con mejor pronóstico  $(p_1, h_1)$ . En el segundo paso se escoge el par  $(p_1, h_1)$  y sus 8 vecinos cercanos, para pronosticar  $X_{t-s+2,T}$  y se repite de nuevo el proceso. El procedimiento continúa hasta alcanzar  $X_{t,T}$  cuando se obtiene el par  $(p_s, h_s)$  y se realiza el pronóstico  $X_{t+1,T}$ . El criterio para elegir el par óptimo es

$$d(x, p, h) = \frac{|X_{kT} - \widehat{X}_{kT}^{(p)}(h)|}{\widehat{P}_{kT}^{(p)}(h)}, \quad (3-81)$$

donde  $\widehat{X}_{kT}^{(p)}(h)$  es el pronóstico de  $X_{kT}$ , usando el par  $(p, h)$ ,  $\widehat{P}_{kT}^{(p)}(h)$  es la longitud del intervalo de predicción correspondiente. A medida que el segmento de entrenamiento sea más grande,  $(p_s, h_s)$  no deberá depender de forma significativa de los parámetros iniciales. Además, se recomienda escoger un segmento al final de la serie que no presente valores extremos visibles.

### 3.5. Metodología híbrida basada en wavelets

Como se mencionó en la introducción, en la literatura de series de tiempo las wavelets se han utilizado en conjunto con otros modelos tradicionales de pronóstico. [Zhu et al., 2014], descomponen la expresión (3-45) para una serie de tiempo  $X = \{x_t, t = 0, \dots, T - 1\}$ , como

$$x_t = A_{J,t} + \sum_{j=1}^J D_{j,t}, \quad t = 0, \dots, T - 1, \quad (3-82)$$

donde  $A_{J,t}$  es una subserie que representa la tendencia del proceso, mientras que  $D_{j,t}$  presenta los detalles locales de la serie. Para realizar la proyección  $\widehat{X}_{N+h}$ , con  $N = T - 1$ , es necesario evaluar  $\widehat{A}_{J,N+h}$  y  $\widehat{D}_{j,N+h}$ , así que se definen los siguientes modelos para ambas componentes

$$\widehat{A}_{J,N+h} = f_0(A_{J,N}, A_{J,N-1}, \dots, A_{J,N-p_0}), \quad (3-83)$$

$$\widehat{D}_{j,N+h} = f_j(D_{j,N}, D_{j,N-1}, \dots, D_{j,N-p_j}), \quad j = 1, \dots, J, \quad (3-84)$$

donde cada modelo  $f_j (j = 0, \dots, J)$  tiene su propio orden  $p_j$ . La elección de  $f_j$  está relacionada con el comportamiento dinámico de cada subserie.

En este trabajo se propone usar el modelo general ARFIMA-HYGARCH y, debido al uso frecuente en la literatura de modelos no paramétricos, se usa una red neuronal. Para la descomposición de la serie, se emplea la transformación wavelet de máximo traslape (MODWT). Como se mencionó antes, es una transformación no diezmada y a diferencia de la DWT no requiere que  $N = 2^J$ . Además [Zhu et al., 2014], usando la serie MacKey-Glass y una serie de precipitaciones diarias, muestran que la transformación MODWT es superior en términos de pronóstico a la transformación DWT. Con la transformación MODWT para cada nivel de descomposición, hay  $T$  coeficientes wavelet y de escalamiento. La definición de la MODWT se puede obtener directamente desde la ecuación (3-55). Dados los filtros en espejo  $h_{j,n}$  y  $g_{j,n}$  para cada escala  $j = -1, -2, \dots, -J$ , el filtro wavelet MODWT,  $\tilde{h}_{j,n}$ , y el filtro escalar  $\tilde{g}_{j,n}$  se definen como

$$\tilde{h}_{j,n} = \frac{h_{j,n}}{2^{j/2}}, \quad \tilde{g}_{j,n} = \frac{g_{j,n}}{2^{j/2}}, \quad (3-85)$$

donde  $n = 1, 2, \dots, N_h$  es la longitud del filtro. Los coeficientes wavelet en cada nivel  $j$ , serán la convolución de la serie de tiempo,  $X = \{x_t, t = 0, \dots, T-1\}$ , y los filtros MODWT

$$\tilde{W}_{j,t} = \sum_{n=0}^{N_j-1} \tilde{h}_{j,n} x_{t-n \bmod T}, \quad \tilde{V}_{j,t} = \sum_{n=0}^{N_j-1} \tilde{g}_{j,n} x_{t-n \bmod T} \quad (3-86)$$

donde  $N_j = (2^j - 1)(N_h - 1) + 1$ . En notación matricial se pueden definir como

$$\tilde{W}_j = \tilde{w}_j X, \quad \tilde{V}_j = \tilde{v}_j X, \quad (3-87)$$

donde,

$$\tilde{w}_j = \frac{1}{2^k} \begin{bmatrix} \tilde{h}_{j,0} & \tilde{h}_{j,T-1} & \tilde{h}_{j,T-2} & \dots & \tilde{h}_{j,2} & \tilde{h}_{j,1} \\ \tilde{h}_{j,1} & \tilde{h}_{j,0} & \tilde{h}_{j,T-1} & \dots & \tilde{h}_{j,3} & \tilde{h}_{j,2} \\ \tilde{h}_{j,2} & \tilde{h}_{j,1} & \tilde{h}_{j,0} & \dots & \tilde{h}_{j,4} & \tilde{h}_{j,3} \\ \vdots & \vdots & \vdots & \dots & \vdots & \vdots \\ \tilde{h}_{j,T-2} & \tilde{h}_{j,T-3} & \tilde{h}_{j,T-4} & \dots & \tilde{h}_{j,0} & \tilde{h}_{j,T} \\ \tilde{h}_{j,T-1} & \tilde{h}_{j,T-2} & \tilde{h}_{j,T-3} & \dots & \tilde{h}_{j,1} & \tilde{h}_{j,0} \end{bmatrix}.$$

$\tilde{v}_j$  se define de forma similar reemplazando  $\tilde{h}_{j,n}$  por  $\tilde{g}_{j,n}$ . La serie original puede reconstruirse como

$$X_t = \tilde{v}_J^T \tilde{V}_J + \sum_{j=1}^J \tilde{w}_j^T \tilde{W}_j = \tilde{S}_{J,t} + \sum_{j=1}^J \tilde{D}_{j,t}. \quad (3-88)$$

[Zhu et al., 2014] demuestran que  $\tilde{W}_{j,t} = \sum_{n=0}^{N_j-1} \tilde{h}_{j,n} X_{t-n}$ , basado en un filtro wavelet Daubechies, será un proceso estacionario y de memoria corta. Según este resultado, los modelos pueden definirse de la siguiente manera:

La subserie de tendencia  $\tilde{S}_J = \{\tilde{S}_{J,t}, t = 1, \dots, T\}$ , seguirá el proceso ARFIMA-HYGARCH

$$\phi_{p_J}(L)(1-L)^{d_m}\tilde{S}_J = \theta_{q_J}(L)\epsilon_t, \quad \epsilon_t \sim N(0, \sigma_t^2), \quad (3-89)$$

$$\sigma_t^2 = \omega [1 - B(L)]^{-1} \left\{ 1 - [1 - B(L)]^{-1} \phi_v(L) \left\{ 1 + \alpha [(1-L)]^{d_v} \right\} \right\} \epsilon_t^2. \quad (3-90)$$

Los polinomios autorregresivos,  $\phi_{p_J}$ , y de medias móviles,  $\theta_{q_J}$ , se definen como en (2-13). Los parámetros de diferenciación fraccional para la media condicional,  $d_m$ , y la varianza condicional,  $d_v$ , se definen como en (2-14).

Las subseries de detalles,  $\tilde{D}_j = \{\tilde{D}_{j,t}, t = 1, \dots, N\}$  seguirán el proceso ARMA-HYGARCH

$$\phi_{p_j}(L)\tilde{D}_j = \theta_{q_j}(L)\epsilon_t, \quad \epsilon_t \sim N(0, \sigma_t^2), \quad (3-91)$$

y  $\sigma_t^2$  seguirá un proceso HYGARCH definido como antes.

La descomposición en sucesivas subseries puede generar dependencias no lineales en la estructura temporal de los datos por lo que algunos autores han combinado la metodología wavelet con modelos no lineales, ver [Soltani, 2002]. En este trabajo se emplea una red neuronal con una sola capa oculta. Así, para la serie de tendencia el modelo de redes neuronales se puede definir como

$$f(\tilde{Y}_i, \theta) = \beta_0^J + \sum_{k=1}^{q^J} \beta_k^J \psi(\gamma_{k0}^J) + \sum_i \gamma_{ki}^J \tilde{Y}_i, \quad (3-92)$$

donde  $\psi(\cdot)$  representa la función de activación logística,  $q^J$  es el número de unidades en la capa oculta y  $\tilde{Y}_i = (\tilde{S}_{J,t-1}, \tilde{S}_{J,t-2}, \dots, \tilde{S}_{J,t-p})'$  son las variables explicativas. De forma análoga, para las subseries de detalles los modelos considerados son

$$f(\tilde{Z}_i, \theta) = \beta_0^j + \sum_{k=1}^{q^j} \beta_k^j \psi(\gamma_{k0}^j) + \sum_i \gamma_{ki}^j \tilde{Z}_i, \quad (3-93)$$

donde  $\tilde{Z}_i = (\tilde{D}_{j,t-1}, \tilde{D}_{j,t-2}, \dots, \tilde{D}_{j,t-p})'$ .  $\psi(\cdot)$  y  $q^j$  se define como antes. [Soltani, 2002] señala que las subseries correspondientes a los detalles, en los niveles más bajos, están contaminadas con un alto nivel de ruido y, por lo tanto, pueden ser omitidas (pronóstico cero) en el momento de realizar la predicción agregada. Esto puede verse como una medida adicional para evitar sobreentrenamiento.

Para la evaluación del pronóstico fuera de la muestra se compara la capacidad predictiva de la metodología ARFIMA-HYGARCH, el modelo LSW y el modelo híbrido basado en

wavelets. Asimismo, estas metodologías se comparan con el modelo referente en la literatura, el paseo aleatorio. El pronóstico se evalúa usando el RMSE (Raíz del error cuadrático medio). Además, se usa el test de [Diebold and Mariano, 1995] para determinar si los errores de predicción de los modelos propuestos son menores a los del paseo aleatorio.

El estadístico de contraste de [Diebold and Mariano, 1995] examina la hipótesis nula de que la función de pérdida esperada entre los pronósticos sea la misma. Si  $e_{1t}$  y  $e_{2t}$  representan los errores de predicción correspondientes al paseo aleatorio y al modelo propuesto, respectivamente, el estadístico se define como

$$DM = \frac{\bar{d}}{\sqrt{\frac{LRV_{\bar{d}}}{T}}}, \quad (3-94)$$

donde  $\bar{d} = \frac{1}{T-t_0} \sum_{t=t_0}^T (e_{1t}^2 - e_{2t}^2)$  es el diferencial medio de la función de pérdida y  $LRV_{\bar{d}}$  es un estimador consistente del error estándar. [Diebold and Mariano, 1995] muestran que el estadístico se distribuye asintóticamente normal estándar bajo la hipótesis nula. El test es robusto al incumplimiento de supuestos estándar sobre los errores de previsión como normalidad y no autocorrelación. Un valor significativo y positivo de  $DM$  muestra una mayor precisión del predictor propuesto con relación al paseo aleatorio.

Una aproximación adicional a la evaluación puntual, consiste en predecir si el tipo de cambio sufrirá una apreciación o depreciación en el futuro, ver [Walzack, 2001] y [Álvarez and Álvarez, 2004]. Este enfoque es adecuado particularmente en el análisis del mercado cambiario porque la existencia de errores predictivos muy pequeños, pero en la dirección equivocada, puede originar importantes pérdidas de capital. Para lo anterior, se emplea el test de acierto direccional (DA) de [Pesaran and Timmermann, 1992] que prueba la hipótesis nula de independencia entre el retorno observado  $X_t$  y el pronosticado  $\hat{X}_t$ . Si  $P_x = P_r(X_t > 0)$ ,  $P_{\hat{x}} = P_r(\hat{X}_t > 0)$  y  $\hat{P}$  es la tasa de acierto de signo, el estadístico  $DA$  se define como

$$DA = \frac{\hat{P} - \hat{P}_*}{\sqrt{var(\hat{P}) - var(\hat{P}_*)}}, \quad (3-95)$$

donde  $\hat{P} = \sum_{t=t_0}^T \theta(X_t, \hat{X}_t)$  y  $\hat{P}_* = \hat{P}_x \hat{P}_{\hat{x}} + (1 - \hat{P}_x)(1 - P_{\hat{x}})$ .  $\theta(X)$  es la función heaviside definida como

$$\theta(X) = \begin{cases} 1 & \text{si } X_t \hat{X}_t > 0 \\ 0 & \text{si } X_t \hat{X}_t < 0. \end{cases} \quad (3-96)$$

Bajo la hipótesis nula el estadístico  $DA$  se distribuye normal estándar.

## 4 Resultados empíricos

Los datos empleados en este estudio corresponden a los retornos logarítmicos de la TRM (tasa de cambio representativa del mercado) del Peso Colombiano (COP) respecto al Dólar Estadounidense (USD) <sup>1</sup>. Sólo se consideraron días comerciales. Los datos tienen una longitud de 1074 observaciones comenzando desde el 4 de junio de 2012. El período de estudio se divide en dos partes. Un conjunto de 1024 observaciones destinadas para la estimación y 50 observaciones por fuera de la muestra para llevar a cabo la predicción. En la Figura 4-1 se observa la evolución de los retornos y sus cuadrados a lo largo del período bajo estudio. La inspección visual parece indicar estacionariedad en media. Se observa claramente que existe un período de mayor volatilidad. Este hecho sugiere un proceso de segundo orden no estacionario.

### 4.1. Estadísticas preliminares

Las estadísticas básicas de la serie se muestran en la Tabla 4-1. El estadístico de Ljung-Box muestra evidencia de correlación serial significativa. La prueba de Jarque-Bera rechaza la hipótesis de normalidad, se observa exceso de curtosis característico de funciones de distribución leptocúrticas.

En la Figura 4-2 se muestra el ajuste no paramétrico de la función de distribución de probabilidad. Se observa que los retornos no se ajustan a una distribución normal y en cambio presentan un apuntalamiento elevado y colas pesadas.

Los correlogramas muestrales, para los primeros cien rezagos, se muestran en la Figura 4-3. No existen decaimientos exponenciales o hiperbólicos propios de procesos con memoria para los retornos. Sin embargo, algunos rezagos son significativos. Por lo tanto, en el caso de que exista un proceso fraccional, el parámetro de diferenciación tendrá un valor cercano a cero. En cuanto a los retornos al cuadrado, la prueba de Ljung-Box muestra que existe correlación serial significativa hasta el rezago 20. Además, el correlograma muestral presenta un decaimiento lento propio de procesos integrados.

---

<sup>1</sup>Fuente: Los datos fueron tomados de [www.superfinanciera.gov.co](http://www.superfinanciera.gov.co).

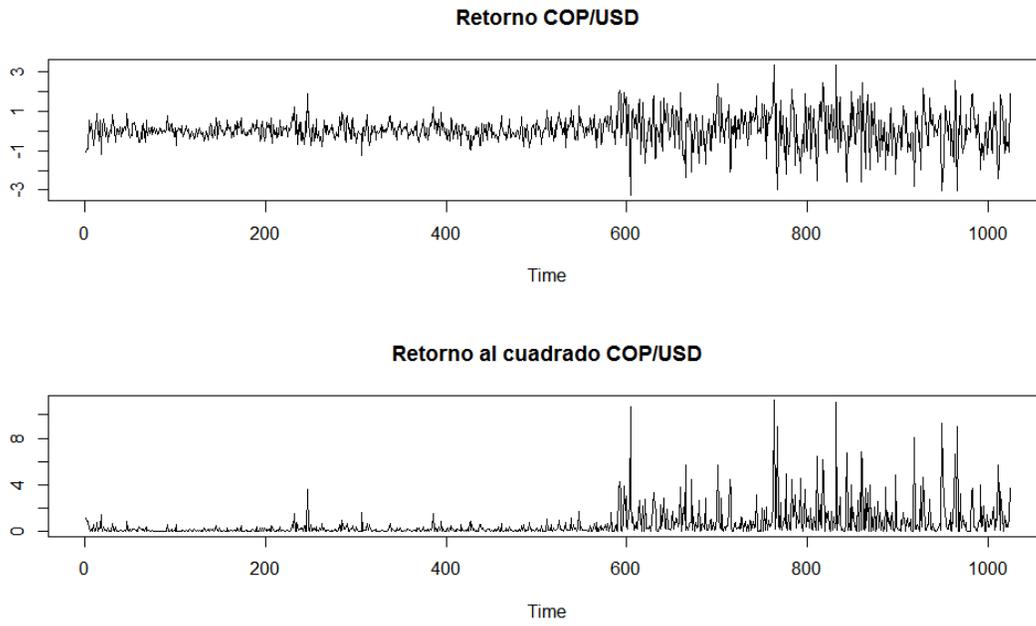


Figura 4-1: Retornos del tipo de cambio

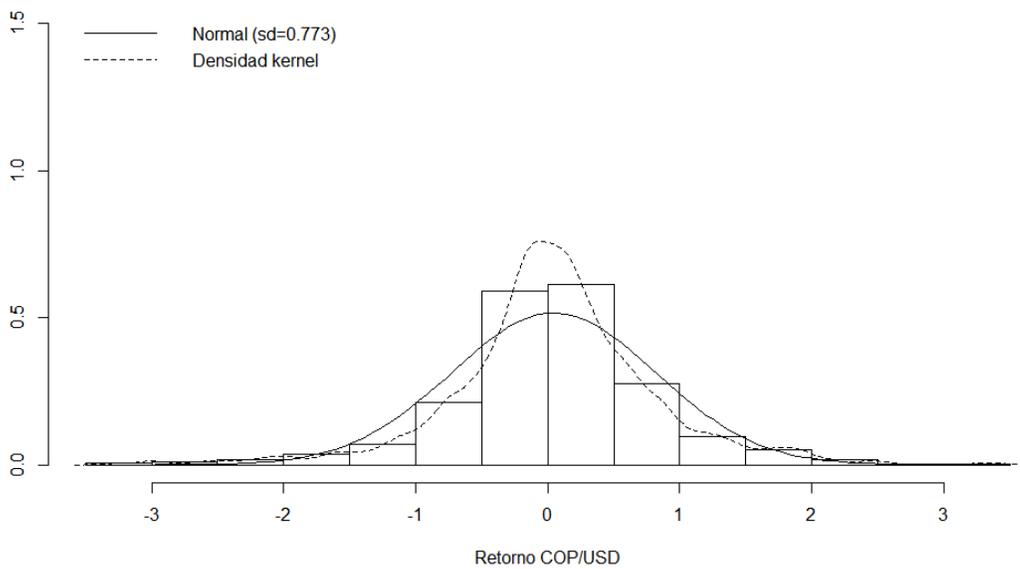


Figura 4-2: Histograma y densidad Kernel para los retornos

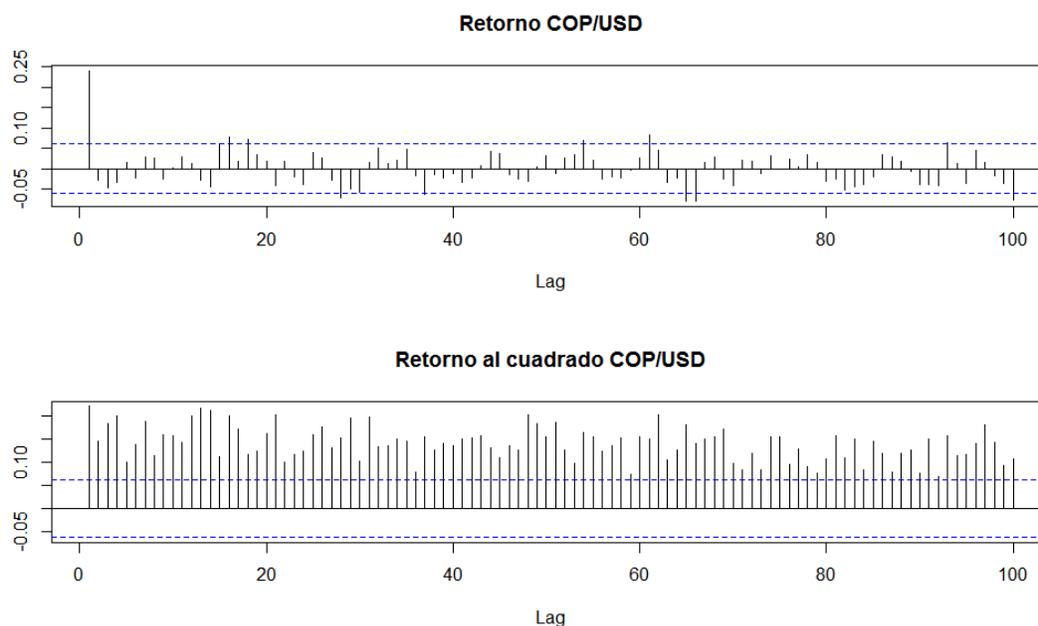
**Tabla 4-1:** Estadísticas básicas para los retornos

Estadísticas	USDCOP
Media	0,045
Mediana	0,037
Máximo	3,352
Mínimo	-3,278
Desviación Estandar	0,774
Asimetría	-0,176
Exceso de Curtosis	2,441
Primer Cuartil	-0,318
Tercer Cuartil	0,421
Jarque-Bera	259,49 (0,000)
$Q(20)$	87,876 (0,000)
$Q2(20)$	581,19 (0,000)

Fuente: elaboración propia.

USDCOP representa los retornos logarítmicos de la TRM del Peso Colombiano respecto al Dólar Estadounidense.

$Q(20)$  y  $Q2(20)$  representan los estadísticos del contraste de Pormanteau de Ljung-Box hasta el rezago 20 sobre los retornos y sus cuadrados, respectivamente. Entre paréntesis se encuentran los  $p$  – valores.

**Figura 4-3:** Correlogramas para los retornos y sus cuadrados

**Tabla 4-2:** Estimación parámetro de diferenciación fraccional,  $d$ 

Variables	GPH			REISEN		
	$\lfloor T^{0,5} \rfloor$	$\lfloor T^{0,6} \rfloor$	$\lfloor T^{0,7} \rfloor$	$\lfloor T^{0,5} \rfloor$	$\lfloor T^{0,6} \rfloor$	$\lfloor T^{0,7} \rfloor$
USDCOP						
$X_t$	0,154	0,042	-0,014	0,157	0,041	-0,007
	(1,150)	(0,473)	(-0,240)	(2,882)	(1,140)	(-0,268)
	{0,250}	{0,636}	{1,190}	{0,003}	{0,254}	{1,211}
$X_t^2$	0,390	0,231	0,121	0,497	0,334	0,181
	(2,899)	(2,565)	( 1,990)	( 9,117)	( 9,163)	(7,299)
	{0,004}	{0,010}	{0,047}	{0,000}	{0,000}	{0,000}

Fuente: Elaboración propia.

$\lfloor T^* \rfloor$  representa el ancho de banda usado para cada prueba. Los estadísticos para contrastar  $H_0 : d = 0$  están entre paréntesis y los  $p$  – valores entre llaves.

## 4.2. Ajuste modelo ARFIMA-HYGARCH

Una vez realizado un análisis descriptivo de la serie bajo estudio, se procede a modelar la media condicional. Para ello, se emplean dos pruebas para evaluar la existencia de memoria larga <sup>2</sup>. Se emplea el GPH y el test de Reisen. En la literatura, se recomienda el uso de distintos anchos de banda para que el análisis sea más robusto. En este sentido, en este trabajo se usa  $\lfloor T^{0,5} \rfloor$ ,  $\lfloor T^{0,6} \rfloor$  y  $\lfloor T^{0,7} \rfloor$ , donde  $\lfloor \cdot \rfloor$  denota la parte entera del argumento.

En la Tabla 4-2 se muestra el parámetro de diferenciación fraccional estimado junto con el estadístico de prueba para la  $H_0 : d = 0$ . En la mayoría de las pruebas, los resultados sugieren que no hay evidencia de memoria larga. Sin embargo, para un ancho de banda  $\lfloor T^{0,5} \rfloor$  el test de Reisen rechaza la hipótesis nula y sugiere la existencia de un parámetro de diferenciación fraccional igual a 0,157. Para la prueba de Reisen se emplea como punto de truncamiento  $v = T^\beta$ , con  $\beta = 0,9$ .

Como se mencionó antes, ambos estadísticos son sensibles a la existencia de una componente autorregresiva o de medias móviles que conduce a rechazar la hipótesis de memoria larga, por lo que se emplea la prueba propuesta por [Castaño et al., 2008b]. Para aproximar el orden del polinomio autorregresivo se emplea  $p^* = \lfloor T^{\frac{1}{4}} \rfloor$  como lo sugieren los autores. Los resultados se muestran en la Tabla 4-3. Como puede observarse, los resultados muestran que el parámetro de diferenciación es igual a 0,073 y es estadísticamente significativo. Además, el modelo parece estar bien especificado en el sentido que todas las raíces están por fuera del círculo unidad y los residuales no están correlacionados.

Estas pruebas también fueron aplicadas en los retornos al cuadrado para determinar la exis-

<sup>2</sup>Las pruebas de memoria larga son calculadas usando la librería fracdiff.

**Tabla 4-3:** Contraste paramétrico

Variables	Castaño et al.	
	$X_t$	$X_t^2$
	0,073	0,218
USDCOP	(9,442)	(17,416)
	{0,000}	{0,000}

Fuente: Elaboración propia.

El orden autorregresivo del modelo ARFIMA para computar la prueba de Castaño et al. (2008) se escogió mediante  $p^* = \lfloor T^{\frac{1}{4}} \rfloor = 6$ . Los estadísticos para contrastar  $H_0 : d = 0$  están entre paréntesis y los  $p$  - valores entre llaves.

tencia de persistencia en volatilidad. En este caso, todas las pruebas conducen al mismo resultado, un parámetro de diferenciación fraccional distinto de cero y ampliamente significativo.

#### 4.2.1. Ajuste del modelo ARFIMA

Dado que el contraste paramétrico de [Castaño et al., 2008b] sugiere la presencia de memoria larga se procede a realizar la estimación del modelo  $ARFIMA(p, d, q)$  <sup>3</sup>. Para la identificación del modelo se emplean los pasos propuestos en la sección 2.4.1.4. En el análisis estadístico inicial se encontró evidencia de que los retornos se ajustan a una distribución leptocúrtica, para la estimación del modelo se emplea una distribución GED. Los resultados del modelo se muestran en la Tabla 4-4. El modelo que mejor se ajusta a los datos es el  $ARFIMA(1, d, 2)$ . A pesar de que en la estimación conjunta el parámetro de diferenciación fraccional no resulta significativo, cuando se compara el criterio de información de Shibata del modelo  $ARFIMA$ , respecto al modelo  $ARMA$ , resulta levemente superior. Como puede observarse, los residuales no están autocorrelacionados. Sin embargo, el test de Ljung-Box sobre los residuales al cuadrado rechaza la hipótesis nula de no autocorrelación. Además, la prueba de multiplicadores de Lagrange de Engle rechaza hipótesis nula de no existencia de efectos ARCH.

#### 4.2.2. Ajuste del modelo en varianza condicional

Para modelar la varianza condicional se ajusta el modelo general ARFIMA-HYGARCH. Los resultados, para la varianza condicional, se presentan en la Tabla 4-5. Como puede observarse, los parámetros no resultan significativos, así que se procede a modelar el pro-

<sup>3</sup>La estimación y pronóstico de los modelos ARFIMA y de la familia GARCH es llevada a cabo usando la librería rugarch. Además, para la estimación de los modelos FIGARCH e HYGARCH se emplea Ox y se enlaza con la consola R Project.

**Tabla 4-4:** Estimación de la media condicional para los retornos

Retornos	USDCOP
Especificación	ARFIMA (1,d, 2)
Media condicional	
$u$	-
$d$	0,150 (1,829) {0,067}
$\phi_1$	0,653 (20,217) {0,000}
$\theta_1$	-0,592 (-19,465) {0,000}
$\theta_2$	-0,1659 (-5,004) {0,000}
$\sigma$	0,748 (29,414) {0,000}
$GED$	1,023 (16,687) {0,000}
Log-L	-1087,933
Akaike	2,136
Bayes	2,1655
Shibata	2,137
Q(5)	4,405 (0,493)
Q(10)	8,173 (0,611)
Q(15)	16,178 (0,370)
Q2(5)	153,12 (0,000)
Q2(10)	287,38 (0,000)
Q2(15)	481,06 (0,000)
ARCHLM(5)	95,36 (0,000)

Fuente: Elaboración propia.

Q() y Q2() hacen referencia a la prueba de Ljung Box para los retornos y sus cuadrados, respectivamente. ARCHLM hace referencia a la prueba de Multiplicadores de Lagrange de Engle.

**Tabla 4-5:** Estimación de la varianza condicional para los retornos

Retornos	USDCOP		
Especificación	HYGARCH(1,1)	FIGARCH(1,1)	GARCH(1,1)
Varianza condicional			
$\alpha_0$	0,003 {0,256}	0,001 {0,241}	0,000 {0,286}
$d$	0,451 {0,160}	1,000 {0,000}	-
$\alpha_1$	0,372 {0,028}	0,032 {0,763}	0,0678 {0,000}
$\beta_1$	0,755 {0,000}	0,937 {0,000}	0,934 {0,000}
$GED$	1,668 {0,000}	1,678 {0,000}	1,671 {0,000}
$HY$	0,089 {0,618}	-	-

Fuente: elaboración propia.

Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

ceso de memoria larga FIGARCH(1,d,1). De nuevo, la no significancia de los parámetros no respalda esta especificación. El paso siguiente es modelar el proceso GARCH(1,1), sin embargo, los parámetros estimados no satisfacen la restricción  $\alpha_1 + \beta_1 < 1$  para garantizar estacionariedad y que la varianza incondicional exista. Por lo tanto, se procede a estimar el modelo integrado ARFIMA-IGARCH(1,1). Los resultados se presentan en la Tabla 4-6. Esta última especificación presenta parámetros estimados estadísticamente significativos y, ni los residuales estandarizados, ni sus cuadrados, exhiben algún tipo de autocorrelación. A pesar de que el parámetro de diferenciación fraccional en la media condicional no es significativo, esta especificación cuando se compara con un modelo ARMA en la media condicional tiene mejor criterio de información. Por lo tanto, se conserva dicho parámetro dentro del modelo. En cuanto a la estabilidad de los parámetros, el test de Nyblom sugiere que todos los parámetros son estables. El proceso IGARCH es de memoria corta, por lo tanto, un choque sobre el proceso de volatilidad desaparece en forma exponencial. Una manera de determinar la existencia de dependencias no lineales en los retornos del tipo de cambio, es aplicar el test BDS sobre los residuales del modelo ARFIMA-IGARCH. En la Tabla 4-7 se muestran los resultados. Se emplean dimensiones de inmersión que van de 2 a 5 y rangos de cercanía que oscilan entre 0,5 y 2 veces la desviación estándar. Como puede observarse, en todos los casos no se rechaza la hipótesis nula de datos iid. Por lo tanto el modelo es adecuado para pronosticar por fuera de la muestra. Los resultados se presentan en la Figura 4-9.

### 4.3. Ajuste del proceso LSW

Como se mencionó antes, un análisis alternativo para modelar procesos que exhiben una estructura de segundo orden que varía en función del tiempo es el proceso LSW. En este sentido, [Fryzlewicz, 2005], señala que una de las consecuencias de la no estacionariedad, es

**Tabla 4-6:** Estimación modelo ARFIMA-IGARCH para los retornos

Retornos	USDCOP
Especificación	ARFIMA (1, <i>d</i> ,2)- IGARCH(1,1)
Media condicional	
<i>u</i>	-
<i>d</i>	0,166 (1,374) {0,169}
$\phi_1$	0,669(6,424) {0,000}
$\theta_1$	-0,603 (-4,868) {0,000}
$\theta_2$	-0,144 (-5,004) {0,000}
Varianza condicional	
$\alpha_0$	0,001 (1,754) {0,079}
$\alpha_1$	0,066 (5,310) {0,000}
$\beta_1$	0,933
<i>GED</i>	1,683 (16,056) 0,000
Log-L	-916,257
Akaike	1,803
Bayes	1,836
Shibata	1,803
Q(5)	1,875 {0,866}
Q(10)	5,366 {0,865}
Q(15)	9,984 {0,821}
Q2(5)	3,817 {0,576}
Q2(10)	5,488 {0,856}
Q2(15)	12,107 {0,670}
ARCHLM(5)	2,149 {0,439}
Nyblom	1,639 (1,9)

Fuente: Elaboración propia.

Q() y Q2() hacen referencia a la prueba de Ljung Box para los retornos y sus cuadrados, respectivamente. Para la estimación se impone la restricción  $\alpha_1 + \beta_1 = 1$ .

**Tabla 4-7:** TEST BDS RESIDUALES MODELO ARFIMA-IGARCH

$m$	$\varepsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A, Residuales				
2	0,601 (0,548)	0,833 (0,404)	0,717 (0,473)	0,713 (0,475)
3	0,947 (0,344)	1,107 (0,267)	0,813 (0,416)	0,689 (0,491)
4	1,377 (0,168)	1,182 (0,237)	0,790 (0,429)	0,589 (0,556)
5	1,515 (0,130)	1,254 (0,210)	0,738 (0,460)	0,357 (0,721)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\varepsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p$ -valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

la generación de un efecto espureo de memoria larga sobre los retornos al cuadrado. Para justificar el uso de este modelo, se prueba si efectivamente los retornos analizados exhiben este tipo de estructura. Para esto, se emplea el test propuesto por [Nason, 2013]<sup>4</sup>, los resultados se muestran en el Figura 4-4.

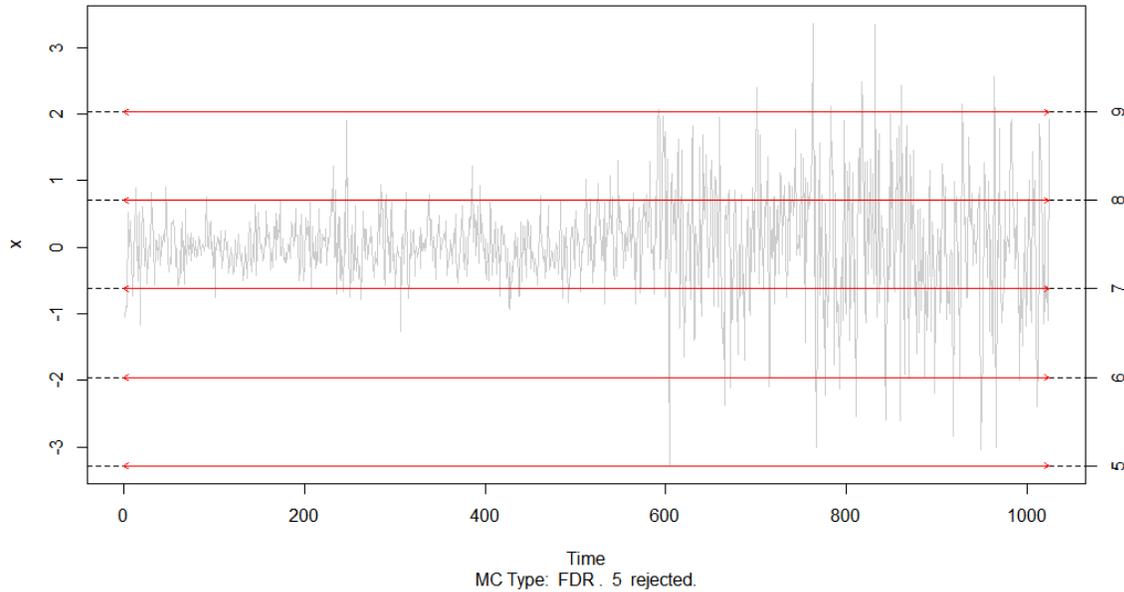
En 5 pruebas se rechaza la hipótesis nula de estacionariedad dado que se encuentran coeficientes Haar estadísticamente significativos de acuerdo al criterio FDR. La localización de dichos coeficientes se muestran en la Figura 4-4. La flecha horizontal indica la localización de la no estacionariedad mientras que la posición vertical muestra la escala  $l$  del periodograma y la escala  $i$  de la transformada wavelet Haar donde se efectuó el test. En el lado derecho del eje se codifica la escala  $l$  y la escala wavelet  $i$  no se muestra explícitamente pero se ordena de forma ascendente. La longitud de la flecha muestra el período sobre el cual la no estacionariedad fue detectada.

Para los retornos USDCOP el test de [Nason, 2013] sugiere la presencia de no estacionariedades en  $I_{l,k}$ ,  $l = 5, 6, 7, 8, 9$  para wavelets Haar centrados en  $k = 512$  de longitud  $1024$ <sup>5</sup>. Todos los coeficientes Haar significativos ocurren en el nivel  $i = 0$ , lo que significa que la extensión de la no estacionariedad existe sobre  $1024/2^0 = 1024$  puntos en el tiempo.

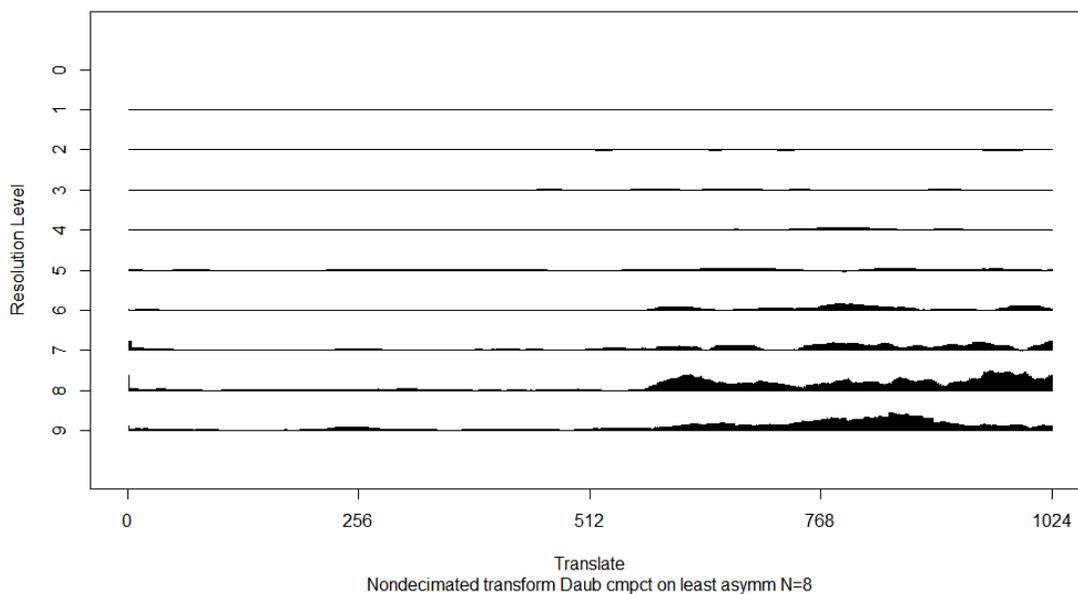
En la Figura 4-5 se muestra el Espectro Wavelet Evolutivo, la versión suavizada del periodograma wavelet. El EWS se estima desde la escala  $j = -1$  hasta  $j = -10$ . Se emplea la wavelet Daubechies de mínima asimetría de orden  $N = 8$ . Se emplea como umbral de

<sup>4</sup>Para computar la prueba de Nason se emplea de la librería locits.

<sup>5</sup>En términos de la notación presentada en el capítulo 3  $l = 9$  y  $l = 1$  corresponden a  $l = -1$  y  $l = -9$ , respectivamente.



**Figura 4-4:** No estacionariedades USDCOP. En el eje derecho se muestra la escala  $l$  del periodograma donde se detectaron las no estacionariedades. Todos los coeficientes son estadísticamente significativos.

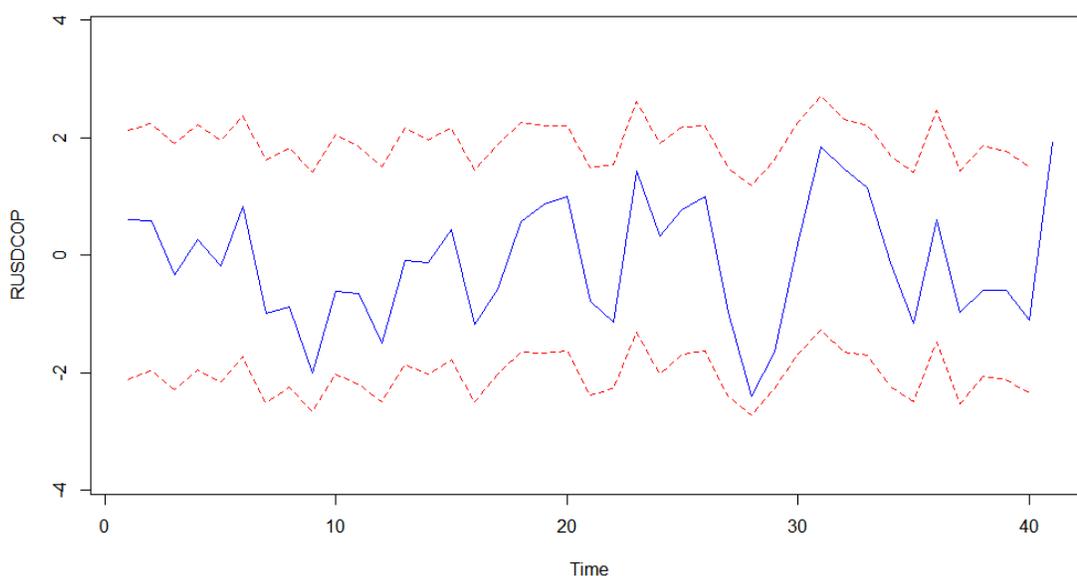


**Figura 4-5:** Descomposición coeficientes Wavelet. En el eje izquierdo se muestra la escala. Se usa la función de Daubechies de mínima asimetría ( $N = 8$ ).

**Tabla 4-8:** Ajuste parámetro inicial  $h_0$ 

$h_0$	10	20	30	40	50	60	70	80	90	100	110
Ajuste	85 %	87,5 %	90 %	90 %	92,5 %	92,5 %	92,5 %	92,5 %	92,5 %	97,5 %	97,5 %

Fuente: Elaboración propia.



**Figura 4-6:** Ajuste segmento de entrenamiento proceso LSW. La línea sólida corresponde a la predicción para los retornos USDCOP. Las líneas punteadas corresponden al intervalo de predicción  $\{\pm 1,96 * valorpredicho\}$ .

suavizamiento  $\lambda = \hat{\sigma}_t \sqrt{2 \log T}$ , donde  $\hat{\sigma}_t$  se estima empleando la desviación absoluta mediana y  $T$  es el tamaño muestral. Como se observa en la Figura 4-5, para las escalas más finas existen magnitudes lo suficientemente grandes que permiten sospechar que dominan ciertos segmentos de tiempo. Especialmente, para la segunda mitad de la serie temporal.

Una vez detectada una estructura de segundo orden variante, se procede a estimar el proceso y realizar el pronóstico por fuera de la muestra. Para esto, se emplean la metodología presentada en la sección 3.4.2 Pronóstico LSW<sup>6</sup>. La longitud del segmento de entrenamiento para ajustar los parámetros  $p$  y  $h$  se definen como: en el primer paso,  $(X_{0,T}, X_{1,T}, \dots, X_{t-s,T})$  se usa para pronosticar  $X_{t-s+1,T}$ , en el último,  $(X_{0,T}, X_{1,T}, \dots, X_{t-1,T})$  es usado para pronosticar  $X_{t,T}$ . Se usa un kernel normal para suavizar la covarianza local.

Dado que el modelo puede ser sensible a la elección inicial del parámetro  $h_0$ , se ensayan dis-

<sup>6</sup>Para la estimación y pronóstico del proceso LSW se emplea el código de Fryzlewicz que se puede encontrar en la página <http://stats.lse.ac.uk/fryzlewicz/flsw/flsw.html>. Con algunos ajustes corre bajo R Project.

tintos valores iniciales. En la Tabla 4-8 se muestran los resultados. El ajuste hace referencia al porcentaje de observaciones del segmento de entrenamiento que se encuentran dentro del intervalo de predicción  $\{\pm 1,96 * \text{valorpredicho}\}$  sugerido por [Fryzlewicz et al., 2003]. De acuerdo a estos resultados, empleamos  $h_0 = 100$ . Para el parámetro de rezago se emplea  $p_0 = 2$ . En la Figura 4-6 se muestra el ajuste del modelo para las últimas 40 observaciones, intervalo reservado como segmento de entrenamiento. Como podemos observar el 97,5% de las observaciones se encuentran dentro del intervalo de predicción.

## 4.4. Metodología Híbrida

Finalmente se lleva a cabo la metodología híbrida. Dado que existe evidencia de que, en múltiples escalas, hay una variabilidad significativa de la serie en el horizonte temporal (ver periodograma wavelet Figura 4-5), la serie se descompone en distintos niveles <sup>7</sup>. Se emplea como base wavelet la función Daubechies de mínima Asimetría de orden  $N = 8$ . Esta familia fue propuesta por [Daubechies, 1992] para reducir la asimetría de las wavelets originales. Esta familia tiene las mismas características de la familia original en cuanto a que posee  $p$  momentos de desvanecimiento, soporte en  $[-p + 1, p]$  y es ortogonal. En la Figura 4-7 se muestran las subseries correspondientes a los detalles o los componentes de alta frecuencia. Se muestran desde la escala 1 hasta la escala 6. Asimismo, en la Figura 4-8 se muestra la serie de tendencia resultante de la descomposición.

A cada subserie de detalle se le aplica la metodología ARFIMA-HYGARCH. Se realiza la prueba de Dickey-Fuller para garantizar que las subseries sean estacionarias, ver Tabla 4-9. Asimismo, se emplea la prueba de [Castaño et al., 2008b] para evaluar la existencia de memoria larga, aunque para algunas subseries el parámetro de diferenciación es significativo se presentan múltiples raíces dentro del círculo unidad, ver Tabla 4-10. Por lo tanto, para las subseries de detalle se emplea una especificación ARMA en la media condicional.

La subserie de tendencia se modela como un proceso fraccionalmente integrado ARFIMA-HYGARCH.

En la Tabla 4-11, columna uno, se muestran las especificaciones resultantes. Los resultados de los modelos, aplicados a cada subserie, se encuentran en el Anexo 6.1. Sin embargo, un hecho común surge en todos los modelos, ninguna especificación logra cumplir que sus residuales sean incorrelacionados. La prueba de Ljung-Box rechaza la hipótesis nula de no autocorrelacion para distintos rezagos. Además, el test BDS fue aplicado sobre los residuales, encontrándose evidencia de dependencia no lineal en la mayoría de los casos. Los resultados se muestran en el Anexo 6.2. Por lo tanto, se concluye que la metodología ARFIMA-IGARCH no es adecuada para modelar las subseries de detalle.

En cuanto a la subserie de tendencia, aplicando la prueba KPSS el estadístico para probar hipótesis nula de estacionariedad alrededor de una tendencia es igual a 0,22 con un

<sup>7</sup>Para la descomposición de la serie original se emplea la librería wavelets.

**Tabla 4-9:** Prueba de Dickey-Fuller Aumentada

Subserie	DF
D1	(-26,927) {0,01}
D2	(-17,842) {0,01}
D3	(-15,045) {0,01}
D4	(-12,522) {0,01}
D5	(-9,375) {0,01}
D6	(-6,770) {0,01}

Fuente: Elaboración propia.

El estadístico DF para contrastar hipótesis nula de no-estacionariedad se encuentra entre paréntesis, los  $p$  – valores entre llaves. El rezago del modelo autorregresivo  $p = 10$ .

**Tabla 4-10:** Prueba de Castaño para subseries de detalle

Especificacion	Subserie					
	D1	D2	D3	D4	D5	D6*
$d$	0,000 {0,999}	0,000 {0,999}	0,210 {0,000}	0,483 {0,000}	0,497 {0,000}	0,05 {0,018}
$\phi_1$	-1,442 {0,000}	1,243 {0,000}	1,688 {0,000}	2,006 {0,000}	2,164 {0,000}	2,711 {0,000}
$\phi_2$	-1,824 {0,000}	-2,268 {0,000}	-1,182 {0,000}	-2,079 {0,000}	-1,793 {0,000}	-2,716 {0,000}
$\phi_3$	-1,731 {0,000}	1,759 {0,000}	0,379 {0,000}	1,766 {0,000}	1,030 {0,000}	1,276 {0,000}
$\phi_4$	-1,359 {0,000}	-1,894 {0,000}	-0,981 {0,000}	-1,482 {0,000}	-1,060 {0,000}	0,0274 {0,000}
$\phi_5$	-0,079 {0,000}	0,084 {0,000}	1,330 {0,000}	1,071 {0,000}	1,149 {0,000}	
$\phi_6$	-0,035 {0,000}	-0,057 {0,000}	-0,742 {0,000}	-0,521 {0,000}	-0,534 {0,000}	
Log-L	13,64	589,3	1525	2625	4059	5623
Akaike	-11,275	-1162,671	-3033,772	-5235	-8102,145	-11233

Fuente: Elaboración propia.

El orden autorregresivo del modelo ARFIMA para computar la prueba de Castaño et al. (2008) se escogió mediante  $p^* = \lfloor T^{\frac{1}{4}} \rfloor = 6$ . Los  $p$  – valores para  $H_0 : d = 0$  están entre llaves. \*En la subserie D6 se emplea un orden autorregresivo menor para poder computar la matriz de correlación.

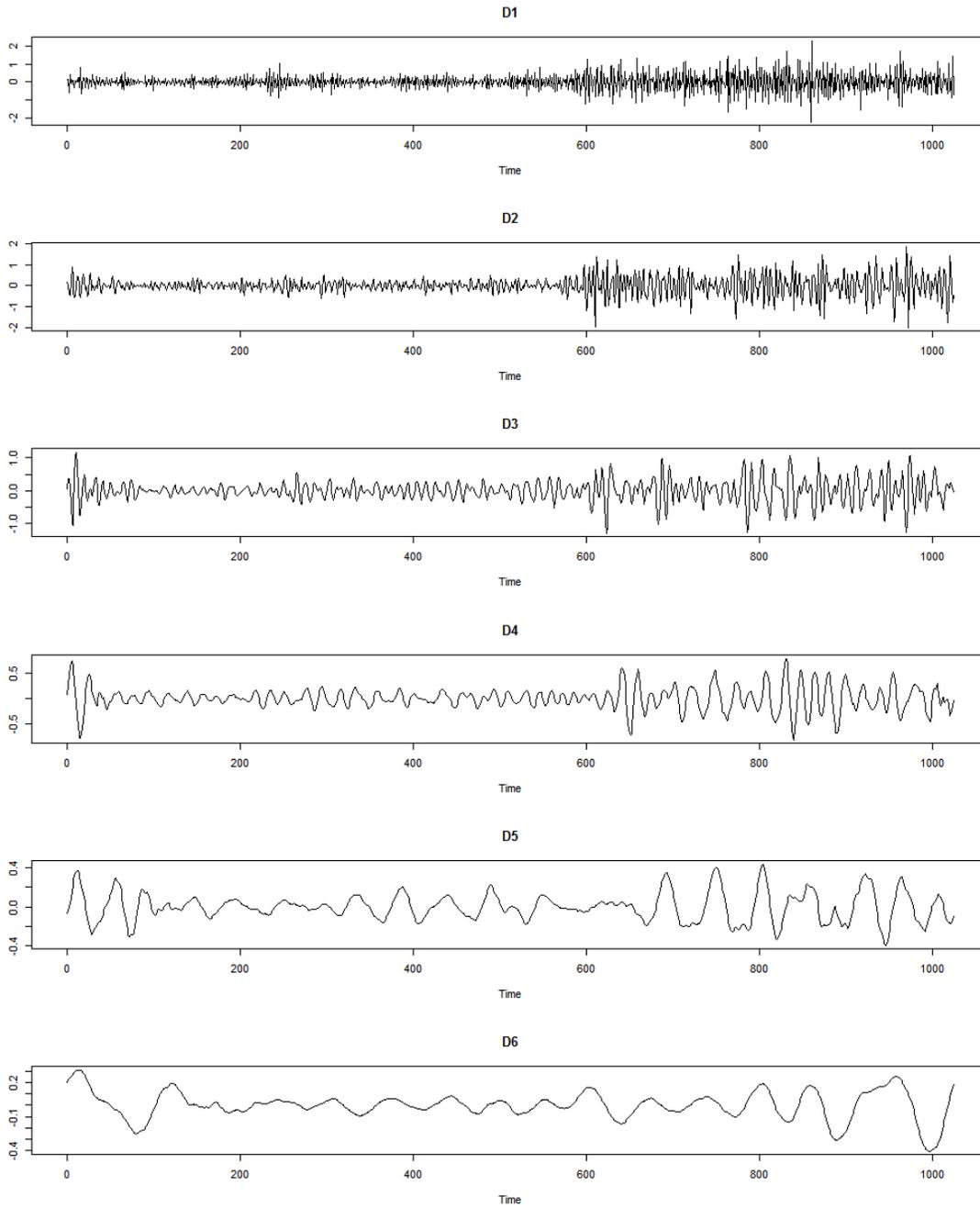
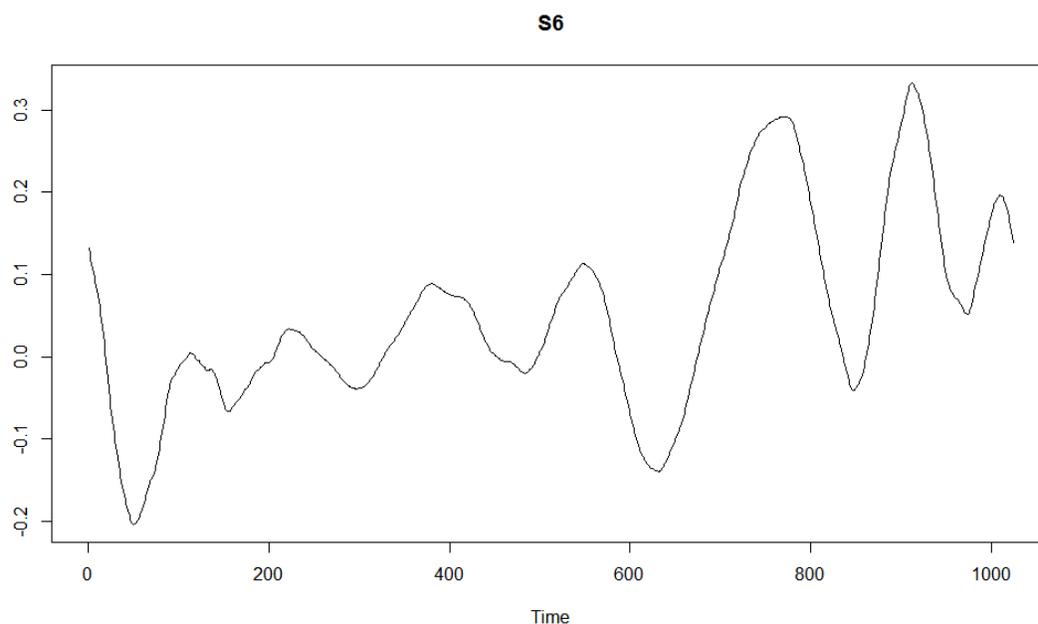


Figura 4-7: Subseries de detalles



**Figura 4-8:** Subserie de tendencia

**Tabla 4-11:** Prueba de Ljung Box sobre los residuales

Subserie	Especificación	Q(5)	Q(10)	Q(15)	Q(20)
D1	ARMA(2,4)-IGARCH	28,385 (0,000)	31,477 (0,000)	34,221 (0,003)	37,792 (0,009)
D2	ARMA(1,0)-IGARCH	614,13 (0,000)	626,33 (0,000)	629,47 (0,000)	634,37 (0,000)
D3	ARMA(2,2)*-IGARCH	502,99 (0,000)	669,36 (0,000)	672,15 (0,000)	676,27 (0,000)
D4	ARMA(1,0)-IGARCH	891,02 (0,000)	2231,3 (0,000)	2352,7 (0,000)	2568,7 (0,000)
D5	ARMA(3,0)*-IGARCH	211,46 (0,000)	246,67 (0,000)	329,42 (0,000)	539,23 (0,000)
D6	ARMA(0,0)-GARCH	4528,1 (0,000)	7009 (0,000)	7695,9 (0,000)	7740,2 (0,000)
S6	ARIMA(0,1,0)-GARCH	4480,7 (0,000)	7812,1 (0,000)	10134 (0,000)	11601 (0,000)

Fuente: Elaboración propia.

$Q()$  representa el estadístico de Ljung-Box sobre los retornos, entre paréntesis se encuentran los  $p$  – valores. (\*) Indica que existen raíces dentro del círculo unidad.

**Tabla 4-12:** Estimación modelo ARFIMA prueba de Castaño subserie de tendencia

Retornos	USDCOP
Especificación	ARFIMA (6, $d$ ,0)
$d$	0,225 (10,35) {0,000}
$\phi_1$	1,898(9428,39) {0,000}
$\phi_2$	-1,544 (-7706,76) {0,000}
$\phi_3$	1,115 (5625,84) {0,000}
$\phi_4$	-1,083 (-5487,26) {0,000}
$\phi_5$	1,150 (5782,77) {0,000}
$\phi_6$	-0,547 (5625,84) {0,000}
$\phi_7$	1,115 (-2760,69) {0,000}
Log-L	7272
Akaike	-14528,82

Fuente: Elaboración propia.

$p$  – valor de 0,01, se concluye entonces que el proceso es integrado. Después de diferenciar la serie, se procede a evaluar la existencia de memoria larga mediante el procedimiento de [Castaño et al., 2008b]. Sin embargo, a pesar de que en el modelo el parámetro de diferenciación fraccional estimado es ampliamente significativo, la especificación no es adecuada debido a la existencia de múltiples raíces dentro del círculo unidad, ver Tabla 4-12. Para la especificación final estimada, se encuentra que los residuales están correlacionados y el test BDS señala la existencia de dependencias no lineales. Los resultados se muestran en el Anexo 6.2.

Por lo anterior, se modela cada subserie mediante el uso de redes neuronales. Para determinar el orden de rezagos de la variable input, se realiza un procedimiento stepwise de dos vías y se selecciona el número de rezagos óptimos de acuerdo al criterio de información AIC. Además, dado que las redes neuronales son modelos semiparamétricos, se realiza un proceso de validación cruzada para determinar el número de unidades ocultas y el valor del parámetro de regularización. Para ello, se fija un rango de valores posibles para ambos parámetros y se dividen los datos en cinco submuestras del mismo tamaño. Por turno, una de las submuestras se usa como conjunto de prueba y las cuatro restantes como conjunto de entrenamiento. Posteriormente se promedian los errores obtenidos en los cinco turnos para cada valor del parámetro y se selecciona el que tenga el mejor desempeño. Este proceso se repite cinco veces. En el Anexo 6.3 se muestran los resultados <sup>8</sup>.

Una vez determinada la estructura de la red neuronal se procede a la estimación. Para evitar los mínimos locales en el proceso de optimización, se ajustan múltiples redes cambiando el valor de los pesos iniciales. Estos pesos se eligen de forma aleatoria y la red es evaluada en

<sup>8</sup>La validación cruzada de la red neuronal es llevada a cabo usando la librería caret de R Project.

**Tabla 4-13:** Redes neuronales ajustadas

Subserie	Arquitectura	Regularización	Rmse
D1	(8-1-1)	0,01	0,2182
D2	(8-3-1)	0,01	0,1118
D3	(8-10-1)	0,01	0,0473
D4	(8-10-1)	0,01	0,0167
D5	(8-2-1)	0,01	0,0063
D6	(8-1-1)	0,01	0,0035
S6	(8-9-1)	0,01	0,0008

Fuente: elaboración propia.

La arquitectura indica el número de unidades de entrada, unidades en la capa oculta y de salida. El RMSE fue calculado sobre el total de la muestra [1 : 1024].

el segmento de entrenamiento de acuerdo al RMSE <sup>9</sup>. En la Tabla **4-13** se presentan las arquitecturas de las redes neuronales óptimas encontradas para cada subserie. Además se muestra que, para todos los modelos, el parámetro de regularización obtenido fue de 0,01. Como puede apreciarse, para las subseries D1 y D2 el ajuste dentro de la muestra fue el más pobre en términos del RMSE. Siguiendo la recomendación de [Soltani, 2002] dado que estas series contienen un alto nivel de ruido no se tienen en cuenta en el momento de realizar el pronóstico agregado.

## 4.5. Evaluación de los pronósticos

Para todos los modelos, se realiza el pronóstico un paso hacia adelante en un horizonte de 50 observaciones. El pronóstico sigue un esquema rolling en el sentido de que el modelo inicial se usa para generar el pronóstico un paso hacia adelante, posteriormente se reestiman los parámetros de cada modelo considerando como conjunto de información aquella que reúne la muestra inicial con la observación inmediatamente siguiente <sup>10</sup>. Este proceso se repite hasta recorrer todo el horizonte de pronóstico.

En la Figura **4-9** y **4-10** se muestra el pronóstico un paso hacia adelante para los modelos propuestos. Además, en la Tabla **4-14** se muestran las medidas de evaluación. En términos del RMSE, los modelos ARFIMA-IGARCH, LSW y paseo aleatorio (RW) tienen un desempeño similar. Para el modelo híbrido MODWT-RED, en términos del RMSE, tiene el peor desempeño. Sin embargo, de acuerdo al test de Diebold y Mariano (DM), la diferencia entre los errores de predicción respecto a los errores del paseo aleatorio de todos los métodos no son significativos. La hipótesis nula de que los dos métodos tienen la misma exactitud de pronóstico no puede ser rechazada en ningún caso. Por otro lado, cuando se evalúa el porcen-

<sup>9</sup>Para la estimación y pronóstico de la red neuronal se emplea la librería `rminer`.

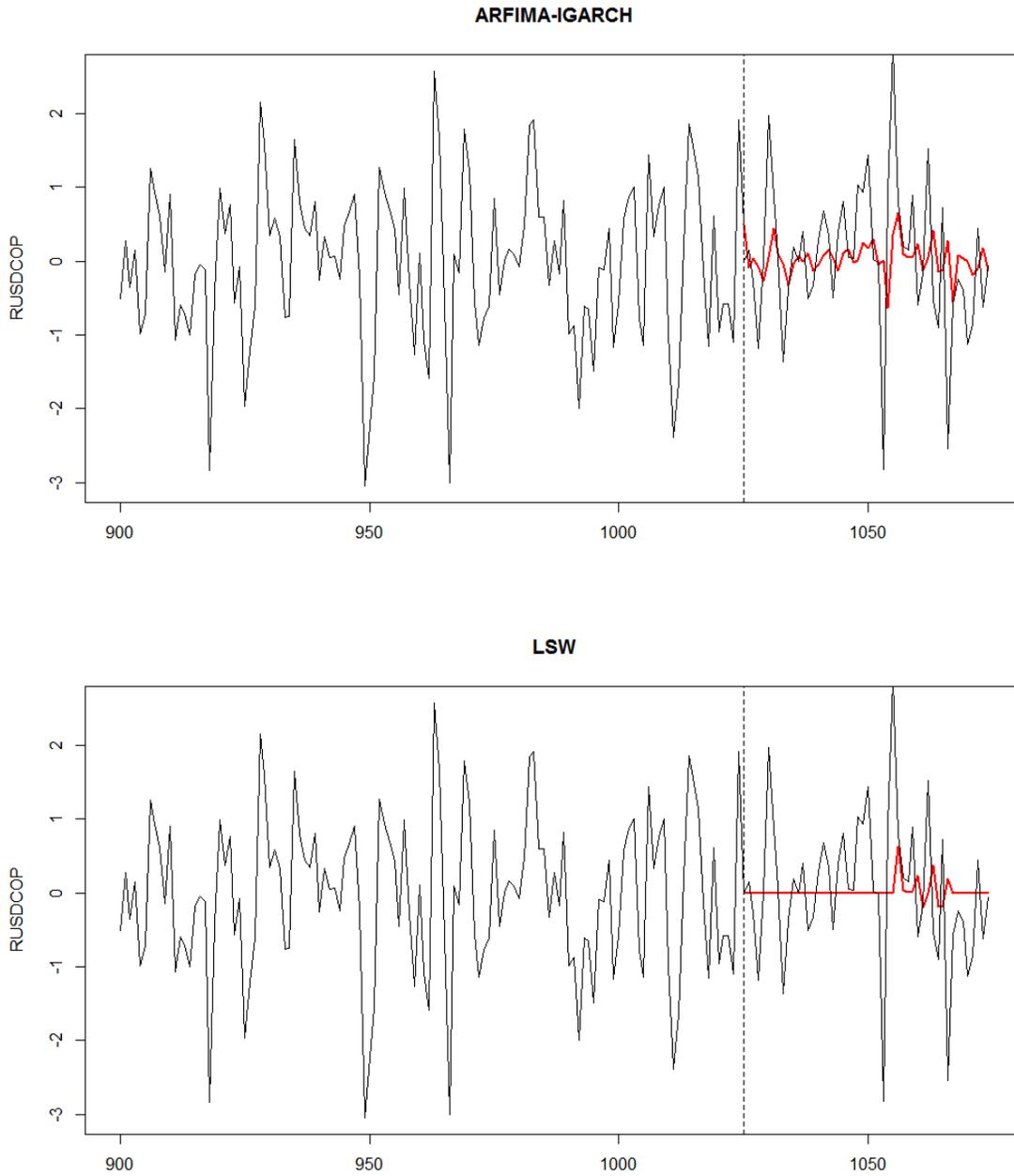
<sup>10</sup>Para conservar el tamaño muestral de 1024 observaciones se van eliminando de la muestra las primeras observaciones

**Tabla 4-14:** Resultados predicción RMSE, DA y DM

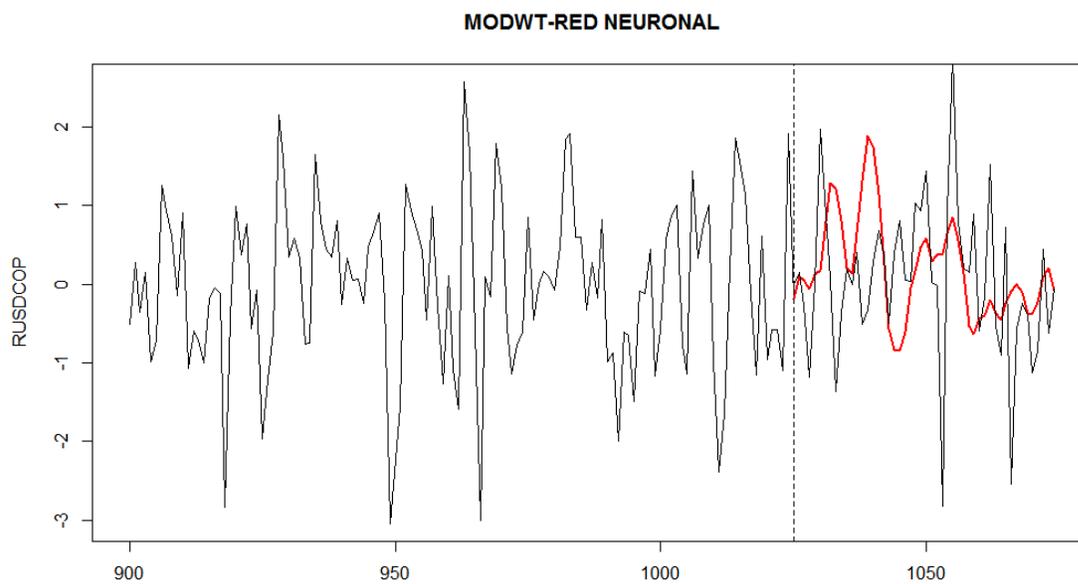
Especificación	RMSE	DA	DM
ARFIMA-IGARCH	0,961	56 % (0,778){0,218}	-0,2092 (0,835)
LSW	0,981	12 % (7,253){1,0}	0,838 (0,406)
MODWT-RED	1,107	64 % (1,940){0,026}	1,434 (0,158)
RW	0,969	0 %	-

Fuente: elaboración propia.

taje de aciertos de signo y el contraste estadístico de [Pesaran and Timmermann, 1992], el único modelo que rechaza la hipótesis nula de independencia entre los valores pronosticados y observados en un nivel de significancia del 5 %, es el modelo MODWT-RED. El porcentaje de aciertos de signo es significativamente superior al que se obtendría bajo el modelo de paseo aleatorio. Por lo tanto, la metodología híbrida que combina wavelets y redes neuronales es capaz de predecir una eventual apreciación o depreciación del tipo de cambio.



**Figura 4-9:** Pronóstico modelo ARFIMA-IGARCH y proceso LSW. La línea negra corresponde a la serie original y la línea roja al pronóstico, por fuera de la muestra, de cada especificación.



**Figura 4-10:** Pronóstico modelo híbrido MODWT-Red neuronal. La línea negra corresponde a la serie original y la línea roja al pronóstico por fuera de la muestra.

## 5 Conclusiones

El uso de modelos no lineales en series de tiempo económicas representa un complemento al modelamiento macroeconómico tradicional, adecuado para trabajar con series procedentes de sistemas lineales simples y con principios fundamentales que subyacen en poco nivel de ruido. En la literatura, se sugiere que el proceso generador de datos de la tasa de cambio en un proceso integrado de orden uno; por lo tanto, su cambio no estaría correlacionado y no sería linealmente predecible. Lo anterior, ha generado que se use un paseo aleatorio para pronosticar su valor. Sin embargo, este análisis no incorpora algunas características empíricas detectadas para esta serie como memoria larga, cluster de volatilidad y dependencias no lineales.

En este trabajo se ha comparado el pronóstico de dos modelos basados en wavelets respecto al modelo tradicional estacionario ARFIMA-HYGARCH y el modelo referente en la literatura, el paseo aleatorio. La serie bajo estudio fue el tipo de cambio USD/COP. Particularmente, se emplea el modelo no estacionario LSW, basado en la wavelet Haar, y un modelo híbrido que combina la MODWT, basada en wavelets Daubechies, con un modelo de pronóstico no lineal de redes neuronales. El uso de la red neuronal se debe a que las subseries resultantes de la descomposición wavelet no se ajustaron bien a especificaciones lineales y parecen exhibir dependencias no lineales. En términos del pronóstico, el modelo híbrido, basado en wavelets y redes neuronales, resulta superior que las demás especificaciones para pronosticar una apreciación o depreciación del tipo de cambio. Como trabajo futuro debe investigarse el desempeño de los modelos wavelet para otras series de tiempo.

## 6 Anexo

### 6.1. Modelos ARFIMA-HYGARCH para subseries

En esta sección se presentan los modelos estimados para las subseries de detalles y de tendencia en el marco de la metodología ARFIMA-HYGARCH. Como puede observarse, varias especificaciones no cumplen el supuesto de que las raíces de los polinomios autorregresivos y de medias móviles estén por fuera del círculo unidad. Los resultados se muestran desde la tabla 6-1 a la 6-8

### 6.2. Test BDS sobre residuales ARFIMA-IGARCH para subseries

A continuación se presentan los resultados del test BDS sobre los residuales de los modelos ajustados, dentro de la metodología ARFIMA-HYGARCH, a las subseries de detalles y de tendencia, ver desde la tabla 6-9 a la tabla 6-15.

### 6.3. Validación cruzada redes neuronales

En el gráfico 6.3 se muestra como se determinan las arquitecturas de las redes neuronales, para el número de unidades ocultas y el parámetro de regularización.

### 6.4. Código: Análisis descriptivo de la serie

Se realiza un análisis estadístico preliminar que permita caracterizar la serie. Se prueba normalidad y la existencia de autocorrelación. Asimismo, se analiza que clase de distribución tienen los datos.

```
library(TSA)
```

```
library(forecast)
```

```
serie=read.table("D:/RTC2/USDCOP.txt", header=TRUE)
```

**Tabla 6-1:** Estimación modelo ARMA-IGARCH para D1

Retornos	D1
Especificación	ARMA (2,4)- IGARCH(1,1)
Media condicional	
$u$	0,000
$\phi_1$	0,094(2,893) {0,000}
$\phi_2$	-0,352(-11,245) {0,000}
$\theta_1$	-2,397 (-67385,972) {0,000}
$\theta_2$	1,635 (42586,522){0,000}
$\theta_3$	-0,002 (-55,959){0,000}
$\theta_4$	-0,230 (42586,522){0,000}
Varianza condicional	
$\omega$	0,000 (4,911) {0,000}
$\alpha_1$	0,229 (28,265) {0,000}
$\beta_1$	0,770
$GED$	1,521 (16,198) {0,000}
Log-L	570,3531
Akaike	-1,094
Bayes	-1,046
Shibata	-1,095

Fuente: elaboración propia.

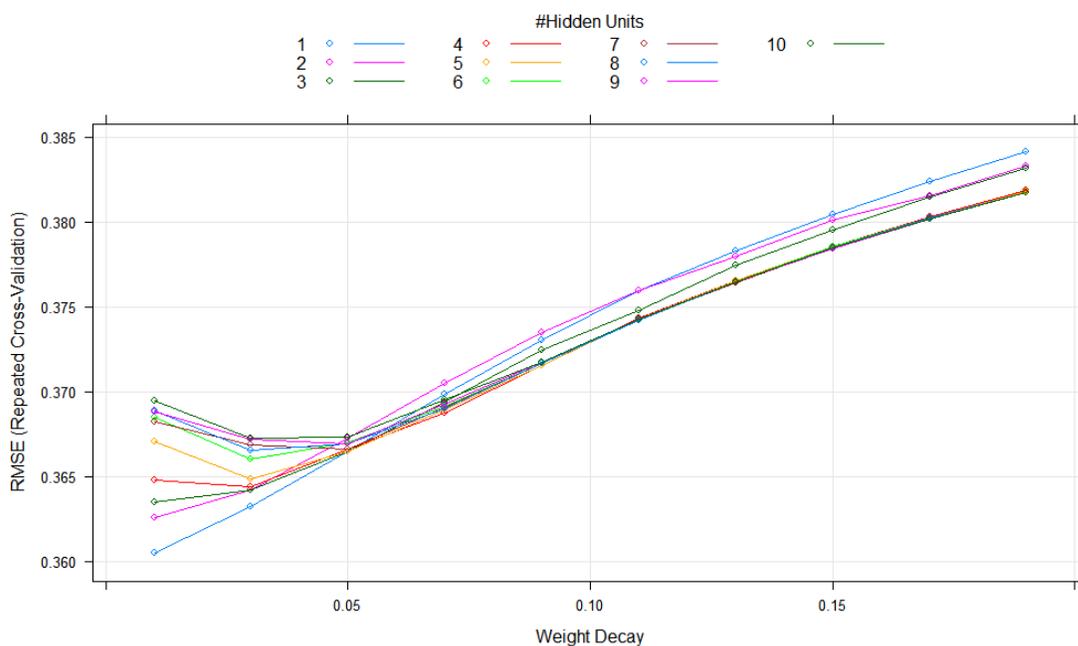
Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

**Tabla 6-2:** Estimación modelo ARMA-IGARCH para D2

Retornos	D2
Especificación	ARMA (1,0)- IGARCH(1,1)
Media condicional	
$u$	0,000 (0,0000) {1,000}
$\phi_1$	0,348(11,498) {0,000}
Varianza condicional	
$\alpha_0$	0,002 (2,743) {0,006}
$\alpha_1$	0,232(5,666) {0,000}
$\beta_1$	0,767
$GED$	1,765 (15,119) {0,000}
Log-L	-176,400
Akaïke	0,3543
Bayes	0,3783
Shibata	0,3542

Fuente: elaboración propia.

Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

**Figura 6-1:** Validación cruzada D1

**Tabla 6-3:** Estimación modelo ARMA-IGARCH para D3

Retornos	D3
Especificación	ARMA (2,2)- IGARCH(1,1)
Media condicional	
$u$	0,000 (0,0000) {1,000}
$\phi_1$	0,854(22,182) {0,000}
$\phi_2$	-0,285(-7,630) {0,000}
$\theta_1$	1,800 (18894,12) {0,000}
$\theta_2$	0,820 (10471,827) {0,000}
Varianza condicional	
$\alpha_0$	0,000 (3,495) {0,000}
$\alpha_1$	0,249(11,859) {0,000}
$\beta_1$	0,750
$GED$	1,890(13,458) {0,000}
Log-L	1347,431
Akaike	-2,616
Bayes	-2,578
Shibata	-2,616

Fuente: elaboración propia.

Los  $p - valores$  para probar la significancia estadística de los parámetros están entre llaves.

**Tabla 6-4:** Estimación modelo ARMA-IGARCH para D3

Retornos	D3
Especificación	ARMA (2,2)- IGARCH(1,1)
Media condicional	
$u$	0,000 (0,0000) {1,000}
$\phi_1$	0,854(22,182) {0,000}
$\phi_2$	-0,285(-7,630) {0,000}
$\theta_1$	1,800 (18894,12) {0,000}
$\theta_2$	0,820 (10471,827) {0,000}
Varianza condicional	
$\alpha_0$	0,000 (3,495) {0,000}
$\alpha_1$	0,249(11,859) {0,000}
$\beta_1$	0,750
$GED$	1,890(13,458) {0,000}
Log-L	1347,431
Akaike	-2,616
Bayes	-2,578
Shibata	-2,616

Fuente: elaboración propia.

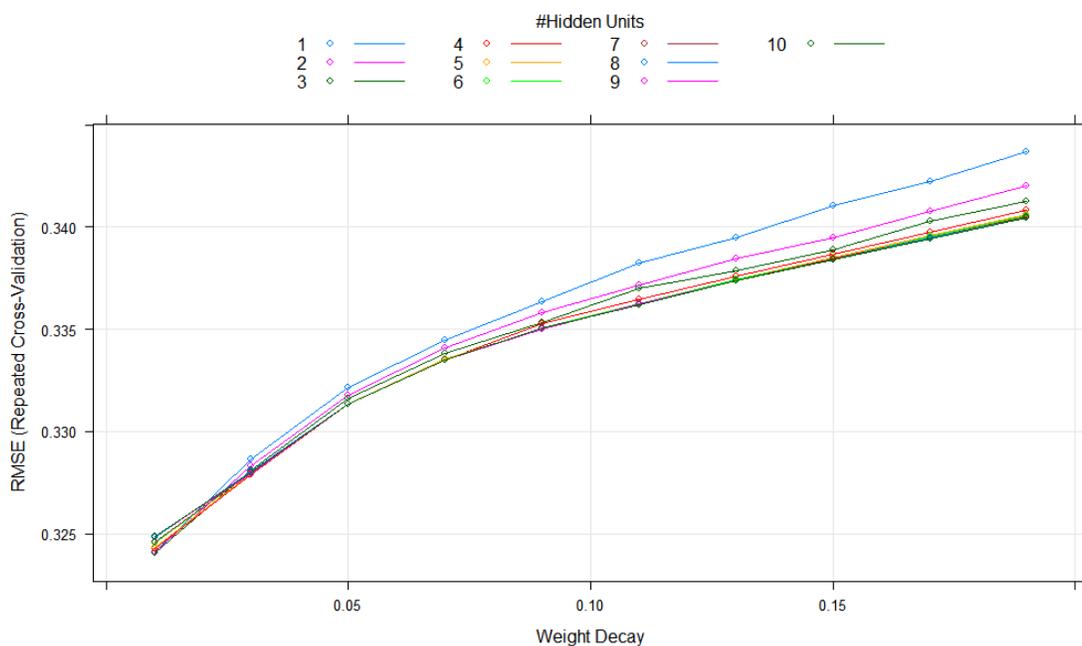
Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

**Tabla 6-5:** Estimación modelo ARMA-IGARCH para D4

Retornos	D4
Especificación	ARMA (1,0)- IGARCH(1,1)
Media condicional	
$u$	0,000 (0,0000) {1,000}
$\phi_1$	0,921(92,987) {0,000}
Varianza condicional	
$\alpha_0$	0,000 (6,328) {0,000}
$\alpha_1$	0,769(18,482) {0,000}
$\beta_1$	0,231
$GED$	2,265(17,881) {0,000}
Log-L	1608,863
Akaike	-3,133
Bayes	-3,108
Shibata	-3,133

Fuente: elaboración propia.

Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

**Figura 6-2:** Validación cruzada D2

**Tabla 6-6:** Estimación modelo ARMA-IGARCH para D5

Retornos	D5
Especificación	ARMA (3,0)- IGARCH(1,1)
Media condicional	
$u$	0,000 (0,0000) {1,000}
$\phi_1$	2,398(1535,478) {0,000}
$\phi_2$	-1,881(-1008,537) {0,000}
$\phi_3$	0,470(208,957) {0,000}
Varianza condicional	
$\alpha_0$	0,000 (0,221) {0,825}
$\alpha_1$	0,099(4,322) {0,000}
$\beta_1$	0,901
<i>GED</i>	1,053(28,486) {0,000}
Log-L	4012,749
Akaike	-7,823
Bayes	-7,7900
Shibata	-7,823

Fuente: elaboración propia.

Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

**Tabla 6-7:** Estimación modelo ARMA-IGARCH para D6

Retornos	D6
Especificación	ARMA (0,0)-GARCH(1,1)
Media condicional	
$u$	0,000 (0,0000)1,000
Varianza condicional	
$\alpha_0$	0,000 (1,479) {0,139}
$\alpha_1$	0,097(4,473) {0,000}
$\beta_1$	0,875(23,725){0,000}
<i>GED</i>	3,727(12,811) {0,000}
Log-L	1319
Akaike	-2,566
Bayes	-2,542
Shibata	-2,566

Fuente: elaboración propia.

Los  $p$  – valores para probar la significancia estadística de los parámetros están entre llaves.

**Tabla 6-8:** Estimación modelo ARFIMA-IGARCH para S6

Retornos	D6
Especificación	ARFIMA (0,1,0)- GARCH(1,1)
Media condicional	
$u$	0,171 (29668)0,000
Varianza condicional	
$\alpha_0$	0,000 (1,856) {0,063}
$\alpha_1$	0,512(42,789) {0,000}
$\beta_1$	0,000(0,000){0,999}
$GED$	12,626(16,414) {0,000}
Log-L	5075,736
Akaike	-9,903
Bayes	-9,880
Shibata	-9,904

Fuente: elaboración propia.

Los  $p - valores$  para probar la significancia estadística de los parámetros están entre llaves.

**Tabla 6-9:** TEST BDS RESIDUALES MODELO ARMA-IGARCH D1

$m$	$\epsilon$			
	0,5 $\sigma$	$\sigma$	1,5 $\sigma$	2 $\sigma$
A. Residuales				
2	-0,980 (0,327)	-1,457 (0,145)	-1,362 ( 0,173)	-1,507 (0,132)
3	-1,396 (0,163)	-1,330 (0,183)	-1,270 (0,204)	-1,236 (0,216)
4	-0,346 (0,729)	-0,638 (0,522)	-0,739 (0,460)	-0,734 (0,463)
5	1,189 (0,235)	-0,138 (0,890)	-0,354 (0,723)	-0,418 (0,676)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p - valores$  para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

**Tabla 6-10:** TEST BDS RESIDUALES MODELO ARMA-IGARCH D2

$m$	$\epsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A. Residuales				
2	0,480 (0,632)	0,455 (0,649)	0,880 (0,379)	1,500 (0,134)
3	20,042 (0,000)	15,907 (0,000)	13,093 (0,000)	10,962 (0,000)
4	28,920 (0,000)	20,351 (0,000)	16,072 (0,000)	13,500 (0,000)
5	42,795 (0,000)	23,244 (0,000)	16,843 (0,000)	13,765 (0,000)

Fuente: Elaboración propia,

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar, Los  $p$  – valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis,

**Tabla 6-11:** TEST BDS RESIDUALES MODELO ARMA-IGARCH D3

$m$	$\epsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A. Residuales				
2	0,0141 (0,989)	-0,302 (0,762)	-0,770 (0,441)	-0,801 (0,423)
3	-1,814 (0,070)	-1,227 (0,219)	-1,836 (0,066)	-1,713 (0,087)
4	-1,705 (0,088)	-1,301 (0,193)	-1,830 (0,067)	-1,425 (0,154)
5	5,274 (0,000)	5,221 (0,000)	4,293 (0,000)	4,383 (0,000)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p$  – valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

**Tabla 6-12:** TEST BDS RESIDUALES MODELO ARMA-IGARCH D4

$m$	$\epsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A. Residuales				
2	158,672 (0,000)	103,288 (0,000)	56,709 (0,000)	35,771 (0,000)
3	219,508 (0,000)	110,202 (0,000)	53,238 (0,000)	31,712 (0,000)
4	306,932 (0,000)	119,312 (0,000)	50,182 (0,000)	27,686 (0,000)
5	449,530 (0,000)	135,274 (0,000)	48,545 (0,000)	24,532 (0,000)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p$  – valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

**Tabla 6-13:** TEST BDS RESIDUALES MODELO ARMA-IGARCH D5

$m$	$\epsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A. Residuales				
2	-0,189 (0,850)	0,685 (0,493)	2,298 ( 0,022)	7,682 (0,000)
3	0,215 (0,830)	0,412 (0,681)	1,519 ( 0,129)	6,198 (0,000)
4	0,292 (0,771)	0,247 (0,805)	1,053 (0,292)	5,151 (0,000)
5	2,396 (0,017)	2,093 (0,036)	2,474 (0,013)	5,803 (0,000)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p$  – valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

**Tabla 6-14:** TEST BDS RESIDUALES MODELO ARMA-IGARCH D6

$m$	$\epsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A. Residuales				
2	-1235,881 (0,000)	686,597 (0,000)	181,033 ( 0,000)	115,071 (0,000)
3	2242,300 (0,000)	890,804 (0,000)	198,115 (0,000)	113,763 (0,000)
4	4426,456 (0,000)	1193,340 (0,000)	219,542 (0,000)	113,030 (0,000)
5	9626,266 (0,000)	1667,577 (0,000)	249,669 (0,000)	114,421 (0,000)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p$  – valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

**Tabla 6-15:** TEST BDS RESIDUALES MODELO ARFIMA-IGARCH S6

$m$	$\epsilon$			
	$0,5\sigma$	$\sigma$	$1,5\sigma$	$2\sigma$
A. Residuales				
2	341,888 (0,000)	5186,192 (0,000)	280,192 ( 0,000)	108,347 (0,000)
3	507,319 (0,000)	6796,490 (0,000)	333,448 (0,000)	111,628 (0,000)
4	793,518 (0,000)	9221,028 (0,000)	406,199 (0,000)	115,865 (0,000)
5	1321,541 (0,000)	13093,861 (0,000)	512,371 (0,000)	123,072 (0,000)

Fuente: Elaboración propia.

$m$  representa la dimensión de inmersión,  $\epsilon$  el rango de cercanía y  $\sigma$  la desviación estándar. Los  $p$  – valores para contrastar la hipótesis nula de datos *iid* se encuentran entre paréntesis.

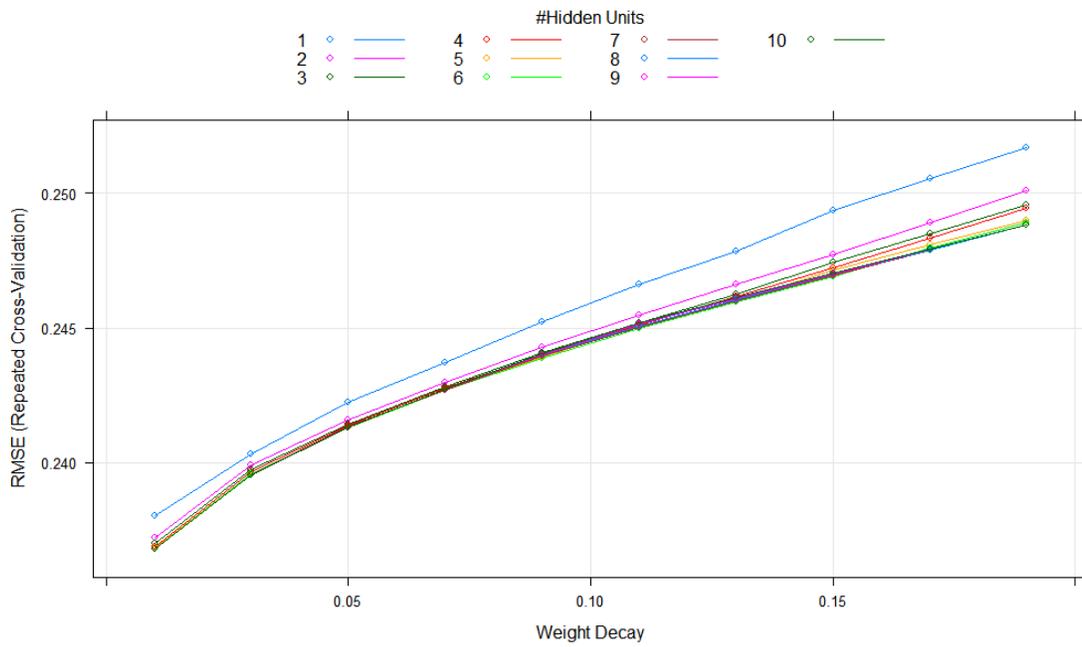


Figura 6-3: Validación cruzada D3

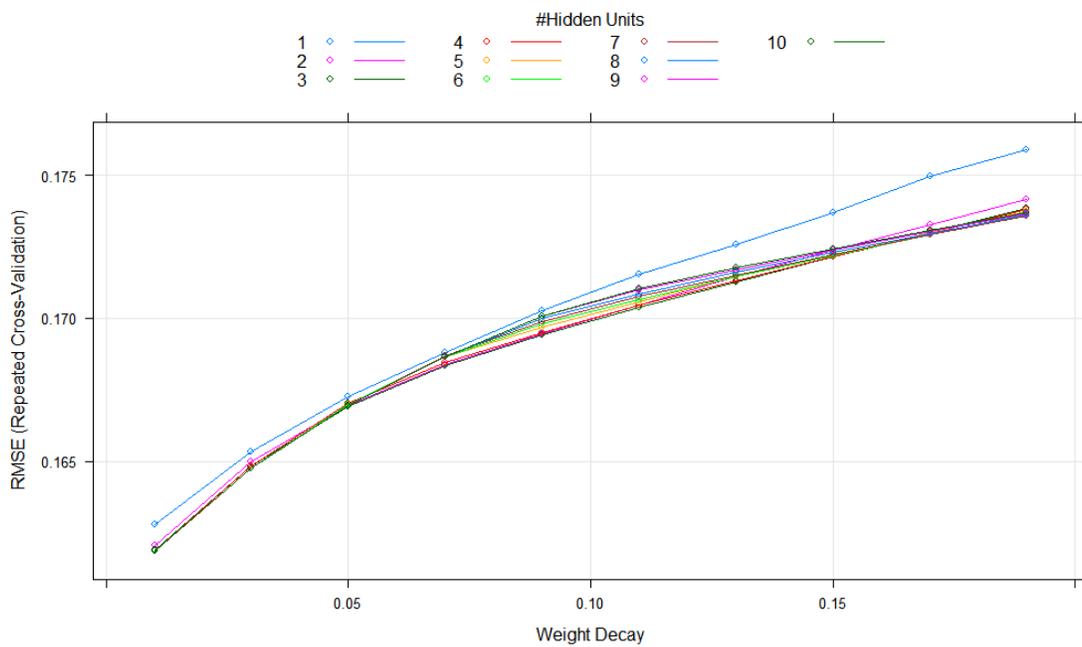


Figura 6-4: Validación cruzada D4

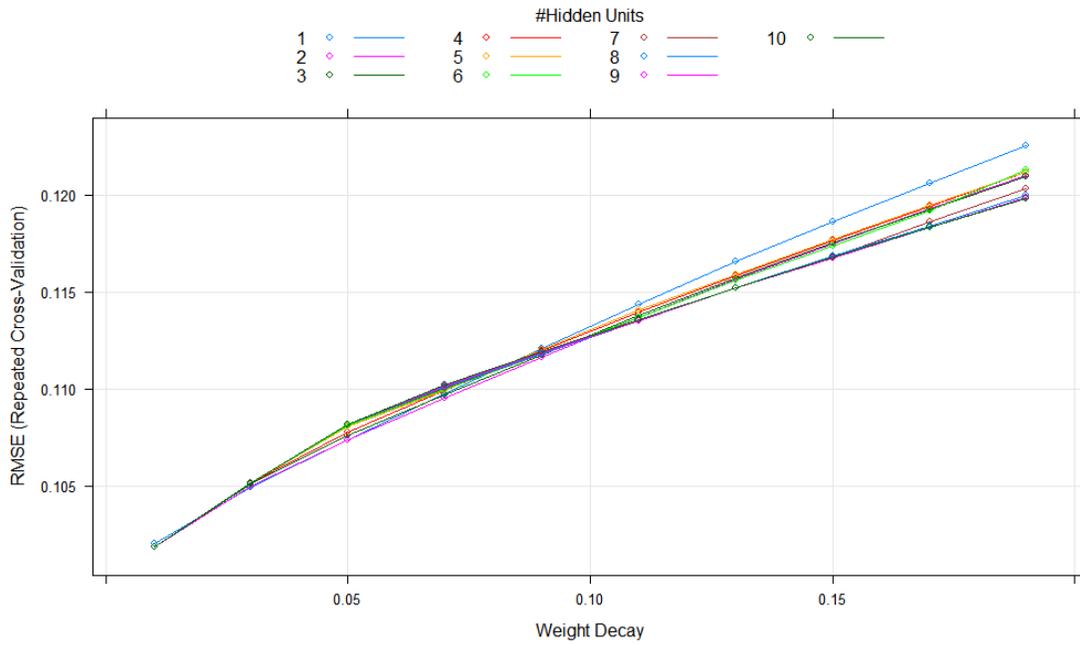


Figura 6-5: Validación cruzada D5

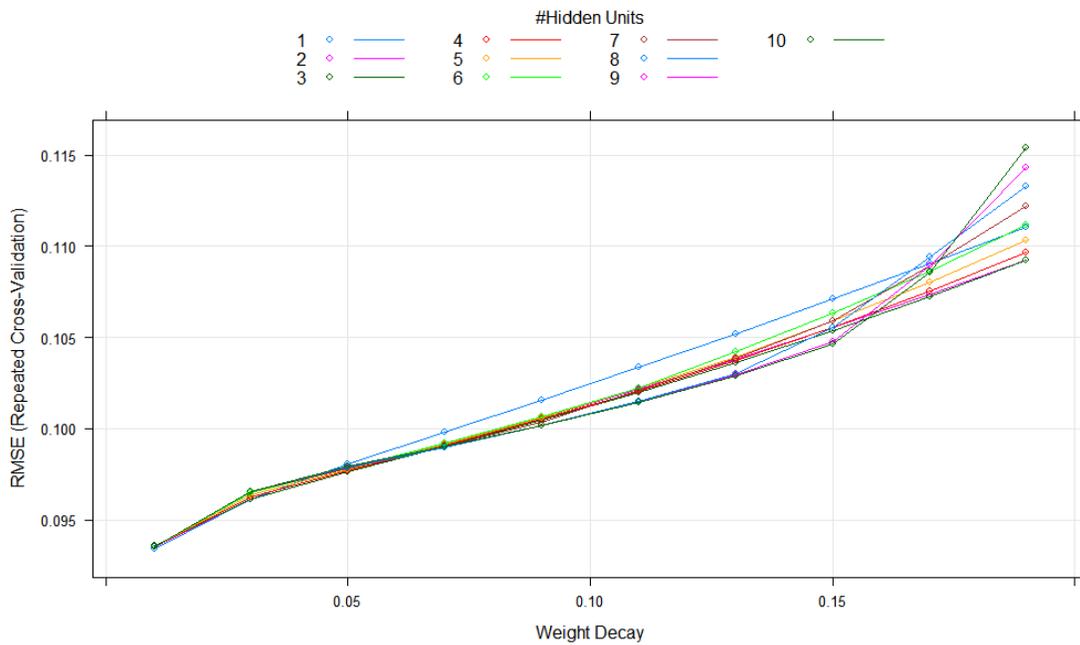


Figura 6-6: Validación cruzada D6

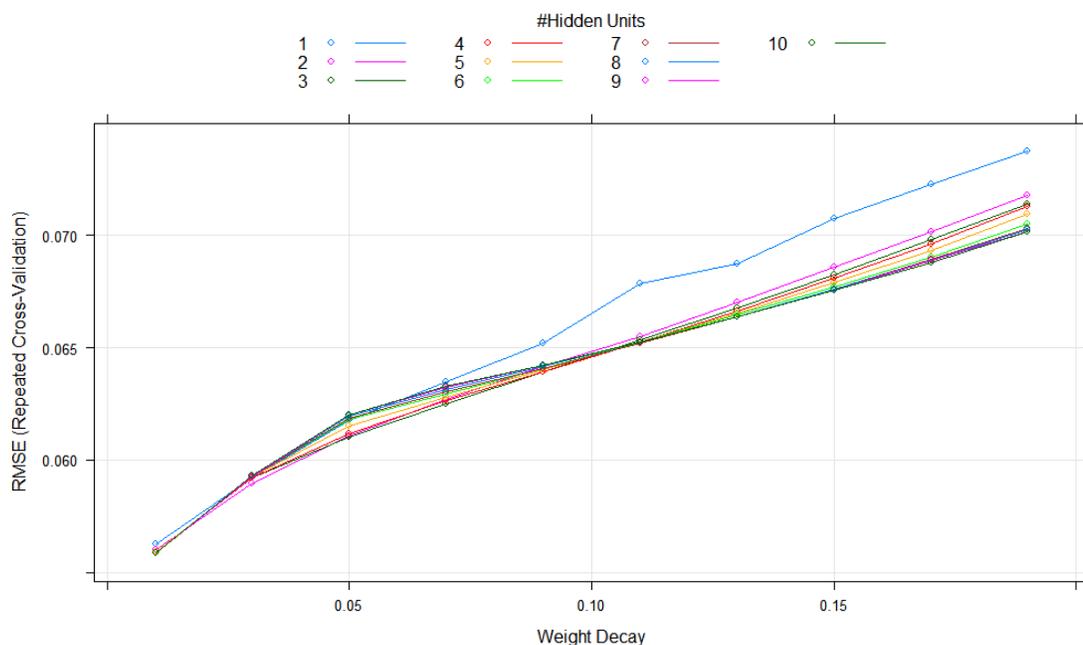


Figura 6-7: Validación cruzada S6

```

attach(serie)
summary(RUSDCOP[1:1024])
sd(RUSDCOP[1:1024])
kurtosis(RUSDCOP[1:1024])
skewness(RUSDCOP[1:1024])
jarque.bera.test(RUSDCOP[1:1024])
Box.test(RUSDCOP[1:1024], lag=20, type = c("Ljung-Box"))
RUSDCOP2=RUSDCOP^2
Box.test(RUSDCOP2[1:1024], lag=20, type = c("Ljung-Box"))
par(mfrow=c(2,1))
plot.ts(RUSDCOP[1:1024],ylab=" ",main="Retorno COP/USD")
plot.ts(RUSDCOP2[1:1024],ylab=" ",main="Retorno al cuadrado COP/USD")

par(mfrow=c(2,1))
acf(RUSDCOP[1:1024],ylab=" ",main="Retorno COP/USD", lag.max=100)
acf(RUSDCOP2[1:1024],ylab=" ", main="Retorno al cuadrado COP/USD", lag.max=100)

hist(RUSDCOP[1:1024], breaks=20, ylab=" ", xlab="Retorno COP/USD", main=" ",
      ylim=c(0,1.5), freq=FALSE,prob = TRUE)
curve(dnorm(x, mean= 0.04531, sd=0.773),add=TRUE)
lines(density(RUSDCOP[1:1024], bw = 0.1), lty=2, lwd=1)

```

```
legend("topleft", c("Normal (sd=0.773)", "Densidad kernel"), lty=1:2, bty="n")
```

## 6.5. Código: análisis de memoria larga

Para determinar la existencia de memoria larga se usa la librería *fracdiff*. Para la prueba GPH se usa el comando *fdGPH()*. Para la prueba de Reissen se usa *fdSperio*. Para cada prueba se usan distinto anchos de banda, *bandw.exp*. Para la prueba de castaño se ajusta un modelo ARFIMA usando *fracdiff()*. Se aproxima la componente autorregresiva en  $nar = 1$ .

```
##GPH
```

```
library(fracdiff)
```

```
PDF_COP_0.5=fdGPH(RUSDCOP[1:1024], bandw.exp=0.5)  
t_GPHCOP_0.5=PDF_COP_0.5$d/PDF_COP_0.5$sd.as  
qnorm(0.05)  
2*pnorm(t_GPHCOP_0.5, lower.tail = FALSE)
```

```
PDF_COP2_0.5=fdGPH(RUSDCOP2[1:1024], bandw.exp=0.5)  
t_GPHCOP2_0.5=PDF_COP2_0.5$d/PDF_COP2_0.5$sd.as  
qnorm(0.05)  
2*pnorm(t_GPHCOP2_0.5, lower.tail = FALSE)
```

```
PDF_COP_0.7=fdGPH(RUSDCOP[1:1024], bandw.exp=0.7)  
t_GPHCOP_0.7=PDF_COP_0.7$d/PDF_COP_0.7$sd.as  
qnorm(0.05)  
2*pnorm(t_GPHCOP_0.7, lower.tail = FALSE)
```

```
PDF_COP2_0.7=fdGPH(RUSDCOP2[1:1024], bandw.exp=0.7)  
t_GPHCOP2_0.7=PDF_COP2_0.7$d/PDF_COP2_0.7$sd.as  
qnorm(0.05)  
2*pnorm(t_GPHCOP2_0.7, lower.tail = FALSE)
```

```
PDF_COP_0.6=fdGPH(RUSDCOP[1:1024], bandw.exp=0.6)
```

```
t_GPHCOP_0.6=PDF_COP_0.6$d/PDF_COP_0.6$sd.as
qnorm(0.05)
2*pnorm(t_GPHCOP_0.6, lower.tail = FALSE)
```

```
PDF_COP2_0.6=fdGPH(RUSDCOP2[1:1024], bandw.exp=0.6)
t_GPHCOP2_0.6=PDF_COP2_0.6$d/PDF_COP2_0.6$sd.as
qnorm(0.05)
2*pnorm(t_GPHCOP2_0.6, lower.tail = FALSE)
```

```
##Reissen
```

```
RDF_COP_0.5=fdSperio(RUSDCOP[1:1024], bandw.exp = 0.5, beta = 0.9)
t_FDSCOP_0.5=RDF_COP_0.5$d/RDF_COP_0.5$sd.as
2*pnorm(t_FDSCOP_0.5, lower.tail = FALSE)
```

```
RDF_COP2_0.5=fdSperio(RUSDCOP2[1:1024], bandw.exp = 0.5, beta = 0.9)
t_FDSCOP2_0.5=RDF_COP2_0.5$d/RDF_COP2_0.5$sd.as
2*pnorm(t_FDSCOP2_0.5, lower.tail = FALSE)
```

```
RDF_COP_0.6=fdSperio(RUSDCOP[1:1024], bandw.exp = 0.6, beta = 0.9)
t_FDSCOP_0.6=RDF_COP_0.6$d/RDF_COP_0.6$sd.as
2*pnorm(t_FDSCOP_0.6, lower.tail = FALSE)
```

```
RDF_COP2_0.6=fdSperio(RUSDCOP2[1:1024], bandw.exp = 0.6, beta = 0.9)
t_FDSCOP2_0.6=RDF_COP2_0.6$d/RDF_COP2_0.6$sd.as
2*pnorm(t_FDSCOP2_0.6, lower.tail = FALSE)
```

```
RDF_COP_0.7=fdSperio(RUSDCOP[1:1024], bandw.exp = 0.7, beta = 0.9)
t_FDSCOP_0.7=RDF_COP_0.7$d/RDF_COP_0.7$sd.as
2*pnorm(t_FDSCOP_0.7, lower.tail = FALSE)
```

```
RDF_COP2_0.7=fdSperio(RUSDCOP2[1:1024], bandw.exp = 0.7, beta = 0.9)
t_FDSCOP2_0.7=RDF_COP2_0.7$d/RDF_COP2_0.7$sd.as
2*pnorm(t_FDSCOP2_0.7, lower.tail = FALSE)
```

```

###Castano

mod_arfiCOP=fracdiff(RUSDCOP[1:1024],nar=6)
summary(mod_arfiCOP)

mod_arfiCOP2=fracdiff(RUSDCOP2[1:1024],nar=1)
summary(mod_arfiCOP2)

resid_mod_arfiCOP=resid(mod_arfiCOP)
Box.test(resid_mod_arfiCOP, lag=10, type = c("Ljung-Box"))
Box.test(resid_mod_arfiCOP, lag=15, type = c("Ljung-Box"))
Box.test(resid_mod_arfiCOP, lag=20, type = c("Ljung-Box"))

```

## 6.6. Código: identificación modelo ARFIMA

Para la identificación del modelo ARFIMA, se diferencia la serie de retornos usando el parámetro estimado mediante la prueba de castaño. Para la serie diferenciada se usa la función de autocorrelación extendida *eacf* y la función *auto.arima* para determinar los órdenes de ambos polinomios. Posteriormente se estima el modelo ARFIMA identificando, usando *arfimafit* de la librería *rugarch*.

```

RUSCOP_DIF=diffseries(RUSDCOP[1:1024], mod_arfiCOP$d)

eacf(RUSCOP_DIF, ar.max = 7, ma.max = 13)
auto.arima(RUSCOP_DIF, d = NA, D = NA, max.p = 5, max.q = 5)

mod_arfiCOP_final=fracdiff(RUSDCOP[1:1024],nar=1,nma=2)
mod_arfiCOP_final2=fracdiff(RUSDCOP[1:1024],nar=2,nma=2)
summary(mod_arfiCOP_final2)

SPEC_ARMA=arfimaspec(mean.model = list(armaOrder = c(1, 2),
include.mean = FALSE,arfima = TRUE),distribution.model = "ged")

ARFIMA=arfimafit(SPEC_ARMA, RUSDCOP[1:1024])

```



```
mean.model= list(armaOrder = c(1, 2), arfima = TRUE, include.mean = FALSE),
distribution.model = "ged")
```

```
ARMAIGARCH_RUSDCOP <- ugarchfit(spec = SPEC_ARMAIGARCH ,
data =RUSDCOP[1:1024], solver='hybrid')
```

```
residualsts_AG=residuals(ARMAIGARCH_RUSDCOP,standardize=TRUE)
Box.test(residualsts_AG, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_AG, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_AG, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_AG, lag=15, type = c("Ljung-Box"))
Box.test(residualsts, lag=20, type = c("Ljung-Box"))
```

```
Box.test(residualsts_AG^2, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_AG^2, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_AG^2, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_AG^2, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_AG^2, lag=20, type = c("Ljung-Box"))
```

```
bds.test(residualsts_AG, m=5, eps = seq(0.5 * sd(residualsts_AG),
2 * sd(residualsts_AG), length = 4))
```

```
nsteps <- 50
fCOPUSD_garch <- rep(0, nsteps)
```

```
for(i in 1:nsteps) {
```

```
ARMAIGARCH_RUSDCOP <- ugarchfit(spec = SPEC_ARMAIGARCH ,
data =RUSDCOP[(i):(1023+i)], solver='hybrid')
```

```
forARMAIGARCH_RUSDCOP=ugarchforecast(ARMAIGARCH_RUSDCOP ,
data = NULL, n.ahead =1, n.roll = 0,out.sample = 0)
```

```
fCOPUSD_garch[i]=fitted(forARMAIGARCH_RUSDCOP)
}
```

```
fCOPUSD_garch
```

```
plot(seq(900,1074),c(RUSDCOP[900:1024],fCOPUSD_garch),type='l',
ylab="RUSDCOP", xlab=" ", main="ARFIMA-IGARCH")
```

```
lines(seq(1025,1074),fCOPUSD_garch,type='l',col='red',lwd=2)
lines(seq(1025,1074),RUSDCOP[1025:1074])
abline(v =1025,lty=2)
```

```
'rmse' <- function(forecast, true)
{
  if (length(forecast) != length(true))
    stop("RMSE: the lengths of input vectors must be the same.")
  err = sqrt(mean((true - forecast)^2))
  return(round(err, 6))
}
```

```
rmse(fCOPUSD_garch,RUSDCOP[1025:1074])
DACTest(fCOPUSD_garch, RUSDCOP[1025:1074], test = "PT", conf.level = 0.99)
```

```
error_fCOPUSD_GARCH=fCOPUSD_garch-RUSDCOP[1025:1074]
error_RW=rep(0,50)-RUSDCOP[1025:1074]
dm.test(error_fCOPUSD_GARCH,error_RW)
```

```
"garchOxFit" <-
function (formula.mean = ~arma(0, 0), formula.var = ~garch(1, 1),
  series = x, cond.dist = c("gaussian", "t", "ged", "skewed-t"),
  include.mean = TRUE, include.var=TRUE, truncation = 100,
  trace = TRUE, title = NULL, arch.in.mean=0)
{
  fit = list()
  fit$x = series
  ## include.var = TRUE
  ## include.var=FALSE (FTS 2008, This command can be used to estimate
  ## volatility model without the constant term,e.g. RiskMetric models.
```

```

##      RST(4/27/2008)
## As an alternative, I moved "include.var" command into the argument.
##
fit$csts = c(include.mean, include.var)
distrib = 0:3
names(distrib) = c("gaussian", "t", "ged", "skewed-t")
distrib = distrib[cond.dist[1]]
fit$cond.dist = cond.dist[1]
if (missing(formula.mean)) {
  fit$formula.mean = ~arma(0, 0)
  fit$arma.orders = c(0, 0)
}
else {
  fit$arma.orders = as.numeric(strsplit(strsplit
(strsplit(as.character(formula.mean),
        "\\("[[2]][2], "\\)"[[1]], ",")[[1]])
  )
}
arfima = TRUE
fit$arfima = as.integer(arfima)
if (missing(formula.var)) {
  fit$formula.var = ~garch(1, 1)
  fit$garch.orders = c(1, 1)
}
else {
  fit$garch.orders = as.numeric(strsplit(
strsplit(strsplit(as.character(formula.var),
        "\\("[[2]][2], "\\)"[[1]], ",")[[1]])
  )
}
#   arch.in.mean = 0
fit$arch.in.mean = arch.in.mean
models = 1:11
names(models) = c("garch", "egarch", "gjr", "aparch", "igarch",
  "figarch.bbm", "figarch.chung", "fiagarch", "fiaparch.bbm",
  "fiaparch.chung", "hygarch")
selected = strsplit(as.character(formula.var), "\\("[[2]][1]
fit$model = models[selected]
nt = length(series)
ident = paste(selected, as.character(floor(runif(1) * 10000)),
  sep = "")
parameters = c(csts = fit$csts, distrib = distrib, arma = fit$arma.orders,

```

```

    arfima = fit$arfima, garch = fit$garch.orders, model = fit$model,
    inmean = fit$aarch.in.mean, trunc = truncation, nt = nt)
write(x = parameters, file = "OxParameter.txt")
write(x = "X", file = "OxSeries.csv", ncolumns = 1)
write(x=series, file = "OxSeries.csv", ncolumns = 1, append = TRUE)
command = "D:\\OxMetrics6\\Ox\\bin\\oxl.exe
D:\\OxMetrics6\\Ox\\lib\\GarchOxModelling.ox"
fit$ox = system(command, show.output.on.console = trace,
    invisible = TRUE)
fit$model = selected
fit$call = match.call()
fit$residuals = scan("OxResiduals.csv", skip = 1, quiet = TRUE)
fit$condvars = scan("OxCondVars.csv", skip = 1, quiet = TRUE)
fit$coef = matrix(scan("OxParameters.csv", skip = 1, quiet = TRUE),
    byrow = TRUE, ncol = 3)
fit$title = title
if (is.null(title))
    fit$title = "GARCH Ox Modelling"
fit$description = as.character(date())
class(fit) = "garchOx"
fit
}

```

## 6.8. Código: prueba de no estacionariedad y proceso LSW

Se usa la prueba de no estacionariedad basada en un proceso LSW. Para esto, se emplea la librería *locits*. Se emplea un filtro wavelet Daubechies de mínima asimetría, "*DaubLeAsymm*", de orden 8. Para el proceso LSW se emplea el código de Fryzlewicz disponible en su página personal. Acá se reproduce el código. Las funciones principales son *adjust* y *pred*. La primera función estima los parámetros del modelo. Dado que puede ser sensible a la elección inicial del parámetro  $g$  se prueban distintos valores iniciales y se escoge el que tenga mejor ajuste.

```
library(locits)
```

```

Test_Nason_RUSDCOP=hwtos2(RUSDCOP[1:1024])
plot(Test_Nason_RUSDCOP,main="")
spec_LSW= ewspect3(RUSDCOP[1:1024], filter.number = 8, family = "DaubLeAsymm"
,smooth.filter.number = 8,binwidth = 50)

```

```
plot(spec_LSW$$S,main="")

#Pronostico en base a un proceso LSW

## Busqueda g inicial

searchs=c(seq(10,150,10))

for(i in searchs){
myproRUSDCOP <- adjust(RUSDCOP[1:1024],start.p = 2,s = 40,start.g =c(i),
limits.g = c(10, 400),attit ="conservative")
cat("g:",i,"FIT:",myproRUSDCOP$percent.fits.in,"\n")
}

myproRUSDCOP <- adjust(RUSDCOP[1:1024],start.p = 2,s = 40,start.g =100,
limits.g = c(10, 400),attit ="conservative")

results1 <- myproRUSDCOP$xhat + 1.96*myproRUSDCOP$sig
results2 <- myproRUSDCOP$xhat - 1.96*myproRUSDCOP$sig

ylim <- c(min(-1.96*RUSDCOP[984:1024]), max(1.96*RUSDCOP[984:1024]))

plot.ts(RUSDCOP[984:1024] , col="blue", ylim=ylim,xlim=NULL,
ylab="RUSDCOP",main="")

lines(results1,col="red", lty=2, lwd=1)
lines(results2, col="red", lty=2, lwd=1)

## pronostico

nsteps <- 50
ppredRUSDCOP <- rep(0, nsteps)
pseRUSDCOP <- rep(0,nsteps)

for(i in 1:nsteps) {
```

```

myproRUSDCOP <- adjust(RUSDCOP[(i):(1023+i)],start.p = 2,s = 40,start.g =100,
limits.g = c(10, 400),attit ="conservative")
pRUSDCOP.pred <- pred(RUSDCOP[(i):(1023+i)], p = myproRUSDCOP$p[41],
g=myproRUSDCOP$g[41], h=1)

ppredRUSDCOP[i]=pRUSDCOP.pred$mean
pseRUSDCOP[i]=pRUSDCOP.pred$std.err
}

plot(seq(900,1074),c(RUSDCOP[900:1024],ppredRUSDCOP),type='l',ylab="RUSDCOP",
xlab=" ", main="LSW")
lines(seq(1025,1074),ppredRUSDCOP,type='l',col='red',lwd=2)
lines(seq(1025,1074),RUSDCOP[1025:1074])
abline(v =1025,lty=2)

rmse(ppredRUSDCOP,RUSDCOP[1025:1074])
DACTest(ppredRUSDCOP, RUSDCOP[1025:1074], test = "PT", conf.level = 0.99)

error_ppredRUSCOP=ppredRUSDCOP-RUSDCOP[1025:1074]
dm.test(error_ppredRUSCOP,error_RW)

```

## 6.9. Código: metodología híbrida: Modelo ARFIMA-HYGARCH

La serie de retornos se descompone en 6 niveles de descomposición. Para ello, se emplea el comando para la Transformación Wavelet de Máximo Traslape *modwt()* de la librería *wavelets*. Para cada subserie resultante, se emplea la misma estrategia de identificación empleada para la serie original.

```

library(wavelets)

mraout=modwt(RUSDCOP[1:1024], filter = "la8",
n.levels = floor(log(length(RUSDCOP[1:1024]))), fast = TRUE)

WaveletSeries <- cbind(do.call(cbind, mraout@W),
mraout@V[[floor(log(length(RUSDCOP[1:1024])))]])

```

```
ts_1 <- as.numeric(WaveletSeries[,1])
ts_2 <- as.numeric(WaveletSeries[,2])
ts_3 <- as.numeric(WaveletSeries[,3])
ts_4 <- as.numeric(WaveletSeries[,4])
ts_5 <- as.numeric(WaveletSeries[,5])
ts_6 <- as.numeric(WaveletSeries[,6])
ts_7 <- as.numeric(WaveletSeries[,7])

par(mfrow=c(3,1))

plot.ts(ts_1,ylab=" ",main="D1")
plot.ts(ts_2,ylab=" ",main="D2")
plot.ts(ts_3,ylab=" ",main="D3")

par(mfrow=c(3,1))
plot.ts(ts_4,ylab=" ",main="D4")
plot.ts(ts_5,ylab=" ",main="D5")
plot.ts(ts_6,ylab=" ",main="D6")

plot.ts(ts_7,ylab=" ",main="S6")

adf.test(ts_1)
adf.test(ts_2)
adf.test(ts_3)
adf.test(ts_4)
adf.test(ts_5)
adf.test(ts_6)

mod_arfiCOP_ts_1=fracdiff(ts_1 ,nar=6)
mod_arfiCOP_ts_2=fracdiff(ts_2 ,nar=6)
mod_arfiCOP_ts_3=fracdiff(ts_3 ,nar=6)
mod_arfiCOP_ts_4=fracdiff(ts_4 ,nar=6)
mod_arfiCOP_ts_5=fracdiff(ts_5 ,nar=6)
mod_arfiCOP_ts_6=fracdiff(ts_6 ,nar=4)

k <- kpss.test(ts_7, null="Trend")
```

```
ts_1 <- as.numeric(WaveletSeries[,1])
auto.arima(ts_1, d = NA, D = NA, max.p = 10, max.q = 10)

SPEC_ARMA_ts_1=arfimaspec(mean.model = list(armaOrder = c(2, 4),
include.mean = TRUE, arfima = FALSE),distribution.model = "norm")

ARFIMA_ts1=arfimafit(SPEC_ARMA_ts_1, ts_1[1:1024], solver='hybrid')
residualsts_1=residuals(ARFIMA_ts1)

Modelts_1_hygarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1),series=residualsts_1,cond.dist = c("ged"))

Modelts_1_figarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=residualsts_1,cond.dist = c("ged"))

Modelts_1_garch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=residualsts_1,cond.dist = c("ged"))

Modelts_1_igarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ igarch(1, 1), series=residualsts_1,cond.dist = c("ged"))

SPEC_ARMAIGARCH_ts_1 <- ugarchspec(variance.model = list(model = "iGARCH",
garchOrder = c(1, 1),
submodel = NULL,
external.regressors = NULL,
variance.targeting = FALSE),
mean.model= list(armaOrder = c(2, 4), arfima = FALSE),
distribution.model = "ged")

ARMAIGARCH_RUSDCOP_ts_1 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_1 ,
data = ts_1, solver='hybrid')

residualsts_1=residuals(ARMAIGARCH_RUSDCOP_ts_1
,standardize=TRUE)
```

```
Box.test(residualsts_1, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_1, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_1, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_1, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_1, lag=20, type = c("Ljung-Box"))

bds.test(residualsts_1, m=5, eps = seq(0.5 * sd(residualsts_1),
2 * sd(residualsts_1), length = 4))

ts_2 <- as.numeric(WaveletSeries[,2])
auto.arima(ts_2, d = NA, D = NA, max.p = 10, max.q = 10)

SPEC_ARMA_ts_2=arfimaspec(mean.model = list(armaOrder = c(1, 0),
include.mean = TRUE,arfima = FALSE),distribution.model = "norm")

ARFIMA_ts2=arfimafit(SPEC_ARMA_ts_2, ts_2,solver='hybrid')
residualsts_2=residuals(ARFIMA_ts2)

Modelts_2_hygarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1), series=residualsts_2,cond.dist = c("norm"))

Modelts_2_figarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=residualsts_2,cond.dist = c("ged"))

Modelts_2_garch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=residualsts_2,cond.dist = c("ged"))

Modelts_2_igarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ igarch(1, 1), series=residualsts_2,cond.dist = c("ged"))

SPEC_ARMAIGARCH_ts_2 <- ugarchspec(variance.model = list(model = "iGARCH",
garchOrder = c(1, 1),
submodel = NULL,
external.regressors = NULL,
variance.targeting = FALSE),
mean.model= list(armaOrder = c(1, 0), arfima = FALSE),
```

```
distribution.model = "ged")

ARMAIGARCH_RUSDCOP_ts_2 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_2 ,
data = ts_2, solver='hybrid')

residualsts_2=residuals(ARMAIGARCH_RUSDCOP_ts_2,standardize=TRUE)
Box.test(residualsts_2, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_2, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_2, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_2, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_2, lag=20, type = c("Ljung-Box"))

bds.test(residualsts_2, m=5, eps = seq(0.5 * sd(residualsts_2),
 2 * sd(residualsts_2), length = 4))

ts_3 <- as.numeric(WaveletSeries[,3])
auto.arima(ts_3, d = NA, D = NA, max.p = 10, max.q = 10)
SPEC_ARMA_ts_3=arfimaspec(mean.model = list(armaOrder = c(2, 2),
include.mean = TRUE,
arfima = FALSE),distribution.model = "norm")

ARFIMA_ts3=arfimafit(SPEC_ARMA_ts_3, ts_3,solver='hybrid')
residualsts_3=residuals(ARFIMA_ts3)

Modelts_3_hygarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1), series=residualsts_3,cond.dist = c("ged"))

Modelts_3_figarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=residualsts_3,cond.dist = c("ged"))

Modelts_3_garch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=residualsts_3,cond.dist = c("ged"))

Modelts_3_igarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ igarch(1, 1), series=residualsts_3,cond.dist = c("ged"))

SPEC_ARMAIGARCH_ts_3 <- ugarchspec(variance.model = list(model = "iGARCH",
garchOrder = c(1, 1),
submodel = NULL,
```

```

                                external.regressors = NULL,
                                variance.targeting = FALSE),
mean.model= list(armaOrder = c(2, 2), arfima = FALSE),
distribution.model = "ged")

ARMAIGARCH_RUSDCOP_ts_3 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_3 ,
data = ts_3, solver='hybrid')

residualsts_3=residuals(ARMAIGARCH_RUSDCOP_ts_3,standardize=TRUE)
Box.test(residualsts_3, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_3, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_3, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_3, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_3, lag=20, type = c("Ljung-Box"))

bds.test(residualsts_3, m=5, eps = seq(0.5 * sd(residualsts_3),
2 * sd(residualsts_3), length = 4))

ts_4 <- as.numeric(WaveletSeries[,4])
auto.arima(ts_4, d = NA, D = NA, max.p = 10, max.q = 10)

SPEC_ARMA_ts_4=arfimaspec(mean.model = list(armaOrder = c(1, 0),
include.mean = TRUE,arfima = FALSE),distribution.model = "norm")

ARFIMA_ts4=arfimafit(SPEC_ARMA_ts_4, ts_4,solver='hybrid')
residualsts_4=residuals(ARFIMA_ts4)

Modelts_4_hygarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1), series=residualsts_4,cond.dist = c("ged"))

Modelts_4_figarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=residualsts_4,cond.dist = c("ged"))

Modelts_4_garch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=residualsts_4,cond.dist = c("ged"))

Modelts_4_igarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ igarch(1, 1), series=residualsts_4,cond.dist = c("ged"))

SPEC_ARMAIGARCH_ts_4 <- ugarchspec(variance.model = list(model = "iGARCH",

```

```

                                garchOrder = c(1, 1),
                                submodel = NULL,
                                external.regressors = NULL,
                                variance.targeting = FALSE),
mean.model= list(armaOrder = c(1, 0), arfima = FALSE),
distribution.model = "ged")

ARMAIGARCH_RUSDCOP_ts_4 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_4 ,
data = ts_4, solver='hybrid')
residualsts_4=residuals(ARMAIGARCH_RUSDCOP_ts_4
,standardize=TRUE)
Box.test(residualsts_4, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_4, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_4, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_4, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_4, lag=20, type = c("Ljung-Box"))

bds.test(residualsts_4, m=5, eps = seq(0.5 * sd(residualsts_4),
2 * sd(residualsts_4), length = 4))

ts_5 <- as.numeric(WaveletSeries[,5])
auto.arima(ts_5, d = NA, D = NA, max.p = 10, max.q = 10)

SPEC_ARMA_ts_5=arfimaspec(mean.model = list(armaOrder = c(3, 0),
include.mean = TRUE,
arfima = FALSE),distribution.model = "norm")

ARFIMA_ts5=arfimafit(SPEC_ARMA_ts_5, ts_5,solver='hybrid')
residualsts_5=residuals(ARFIMA_ts5)

Modelts_5_hygarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1), series=residualsts_5,cond.dist = c("ged"))

Modelts_5_figarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=residualsts_5,cond.dist = c("ged"))

Modelts_5_garch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=residualsts_5,cond.dist = c("ged"))

Modelts_5_igarch=garchOxFit(formula.mean = ~ arma(0,0),

```

```
formula.var =~ igarch(1, 1), series=residualsts_5,cond.dist = c("ged"))

SPEC_ARMAIGARCH_ts_5 <- ugarchspec(variance.model = list(model = "iGARCH",
                                                       garchOrder = c(1, 1),
                                                       submodel = NULL,
                                                       external.regressors = NULL,
                                                       variance.targeting = FALSE),
mean.model=list(armaOrder = c(3, 0), arfima = FALSE),
distribution.model = "ged")

ARMAIGARCH_RUSDCOP_ts_5 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_5 ,
data = ts_5, solver='hybrid')

residualsts_5=residuals(ARMAIGARCH_RUSDCOP_ts_5,standardize=TRUE)
Box.test(residualsts_5, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_5, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_5, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_5, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_5, lag=20, type = c("Ljung-Box"))

bds.test(residualsts_5, m=5, eps = seq(0.5 * sd(residualsts_5),
2 * sd(residualsts_5), length = 4))

ts_6 <- as.numeric(WaveletSeries[,6])
auto.arima(ts_6, d = NA, D = NA, max.p = 10, max.q = 10)

Modelts_6_hygarch=garch0xFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1), series=ts_6 ,cond.dist = c("ged"))

Modelts_6_figarch=garch0xFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=ts_6 ,cond.dist = c("ged"))

Modelts_6_garch=garch0xFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=ts_6 ,cond.dist = c("ged"))

Modelts_6_igarch=garch0xFit(formula.mean = ~ arma(0,0),
formula.var =~ igarch(1, 1), series=ts_6 ,cond.dist = c("ged"))
```

```

SPEC_ARMAIGARCH_ts_6 <- ugarchspec(variance.model = list(model = "sGARCH",
                                                    garchOrder = c(1, 1),
                                                    submodel = NULL,
                                                    external.regressors = NULL,
                                                    variance.targeting = FALSE),
mean.model=list(armaOrder = c(0, 0), arfima = FALSE),
distribution.model = "ged")

ARMAIGARCH_RUSDCOP_ts_6 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_6,
data = ts_6, solver='hybrid')

residualsts_6=residuals(ARMAIGARCH_RUSDCOP_ts_6,standardize=TRUE)
Box.test(residualsts_6, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_6, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_6, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_6, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_6, lag=20, type = c("Ljung-Box"))
bds.test(residualsts_6, m=5, eps = seq(0.5 * sd(residualsts_6),
2 * sd(residualsts_6), length = 4))

ts_7 <- as.numeric(WaveletSeries[,7])

dts_7=diff(ts_7)
mod_arfiCOP=fracdiff(dts_7,nar=6)
summary(mod_arfiCOP)

auto.arima(ts_7, d = NA, D = NA, max.p = 10, max.q = 10)

Modelts_7_hygarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ hygarch(1, 1), series=ts_7 ,cond.dist = c("ged"))

Modelts_7_figarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ figarch.bbm(1, 1), series=ts_7 ,cond.dist = c("ged"))

Modelts_7_garch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ garch(1, 1), series=ts_7 ,cond.dist = c("ged"))

Modelts_7_igarch=garchOxFit(formula.mean = ~ arma(0,0),
formula.var =~ igarch(1, 1), series=ts_7 ,cond.dist = c("ged"))

```

```
SPEC_ARMAIGARCH_ts_7 <- ugarchspec(variance.model = list(model = "sGARCH",
                                                    garchOrder = c(1, 1),
                                                    submodel = NULL,
                                                    external.regressors = NULL,
                                                    variance.targeting = FALSE),
mean.model=list(armaOrder = c(0, 0), arfima = TRUE),
distribution.model = "ged",fixed.pars=list(arfima=1))

ARMAIGARCH_RUSDCOP_ts_7 <- ugarchfit(spec = SPEC_ARMAIGARCH_ts_7 ,
data = ts_7, solver='hybrid')

residualsts_7=residuals(ARMAIGARCH_RUSDCOP_ts_7,standardize=TRUE)
bds.test(residualsts_7, m=5, eps = seq(0.5 * sd(residualsts_7),
2 * sd(residualsts_7), length = 4))

Box.test(residualsts_7, lag=1, type = c("Ljung-Box"))
Box.test(residualsts_7, lag=5, type = c("Ljung-Box"))
Box.test(residualsts_7, lag=10, type = c("Ljung-Box"))
Box.test(residualsts_7, lag=15, type = c("Ljung-Box"))
Box.test(residualsts_7, lag=20, type = c("Ljung-Box"))

##No linealidad

bds.test(residualsts_1, m=5, eps = seq(0.5 * sd(residualsts_1),
2 * sd(residualsts_1), length = 4))

bds.test(residualsts_2, m=5, eps = seq(0.5 * sd(residualsts_2),
2 * sd(residualsts_2), length = 4))

bds.test(residualsts_3, m=5, eps = seq(0.5 * sd(residualsts_3),
2 * sd(residualsts_3), length = 4))

bds.test(residualsts_4, m=5, eps = seq(0.5 * sd(residualsts_4),
2 * sd(residualsts_4), length = 4))

bds.test(residualsts_5, m=5, eps = seq(0.5 * sd(residualsts_5),
2 * sd(residualsts_5), length = 4))
```

```
bds.test(residualsts_6, m=5, eps = seq(0.5 * sd(residualsts_6),
2 * sd(residualsts_6), length = 4))
```

```
bds.test(residualsts_7, m=5, eps = seq(0.5 * sd(residualsts_6),
2 * sd(residualsts_6), length = 4))
```

## 6.10. Código: metodología híbrida: Modelo de redes neuronales

### 6.10.1. Selección de variables

Se emplea la librería *MASS* para seleccionar las variables rezagadas que sean más importantes. Para ello, se emplea el comando *stepAIC* para aplicar la metodología stepwise.

```
variablesUSDCOP=CasesSeries(RUSDCOP[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOP[1:1024,])
step <- stepAIC(fit, direction="both")
```

```
#ts1
```

```
variablesUSDCOPts_1=CasesSeries(ts_1[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_1[1:1024,])
stepts_1 <- stepAIC(fit, direction="both")
stepts_1$anova
```

```
#ts2
```

```
variablesUSDCOPts_2=CasesSeries(ts_2[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_2[1:1024,])
stepts_2 <- stepAIC(fit, direction="both")
stepts_2$anova
```

```
#ts3
```

```
variablesUSDCOPts_3=CasesSeries(ts_3[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_3[1:1024,])
stepts_3 <- stepAIC(fit, direction="both")
```

```
stepts_3$anova

#ts4

variablesUSDCOPts_4=CasesSeries(ts_4[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_4[1:1024,])
stepts_4 <- stepAIC(fit, direction="both")
stepts_4$anova

#ts5

variablesUSDCOPts_5=CasesSeries(ts_5[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_5[1:1024,])
stepts_5 <- stepAIC(fit, direction="both")
stepts_5$anova

#ts6

variablesUSDCOPts_6=CasesSeries(ts_6[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_6[1:1024,])
stepts_6 <- stepAIC(fit, direction="both")
stepts_6$anova

#ts7

variablesUSDCOPts_7=CasesSeries(ts_7[1:1024],c(1,2,3,4,5,6,7,8))
fit=lm(y~.,variablesUSDCOPts_7[1:1024,])
stepts_7 <- stepAIC(fit, direction="both")
stepts_7$anova
```

### 6.10.2. Código: validación cruzada redes neuronales

Se realiza un análisis de validación cruzada para determinar cuales son los parámetros adecuados para ajustar el modelo. El procedimiento se lleva a cabo usando la librería *caret*. Para entrenar las redes neuronales se emplea la librería *nnet*. Para la validación cruzada se emplean K-iteraciones. Los datos se dividen en k subconjuntos, uno de los subconjuntos se utiliza como datos de prueba y el resto como datos de entrenamiento. En nuestro caso, se emplean 5 folds y se repite durante 5 veces para cada uno de los posibles subconjuntos de

prueba.

```
library(caret)
```

```
d_RUSDCOPts_1=CasesSeries(ts_1[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_2=CasesSeries(ts_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_3=CasesSeries(ts_3[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_4=CasesSeries(ts_4[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_5=CasesSeries(ts_5[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_6=CasesSeries(ts_6[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_7=CasesSeries(ts_7[1:1024],c(1,2,3,4,5,6,7,8))
```

```
fitControl <- trainControl(
  method = "repeatedcv",
  number = 5,
  repeats = 5)
```

```
fitControl <- trainControl(method = "repeatedcv",
  number = 5,
  repeats = 5,
)
```

```
nnetGrid <- expand.grid(size = seq(from = 1, to = 10, by = 1),
  decay = seq(from = 0.01, to = 0.2, by = 0.02))
```

```
set.seed(45567)
```

```
netFit_ts_1 <- train(x=d_RUSDCOPts_1[,1:8],y=d_RUSDCOPts_1[,9],
  method = "nnet",
  metric = "RMSE",
  trControl = fitControl,
  tuneGrid = nnetGrid,
  verbose = FALSE)
```

```
netFit_ts_1
```

```
plot(netFit_ts_1)
```

```
plot(varImp(object=netFit_ts_1),main="GBM - Variable Importance")
```

```
MFOR_RUSDCOPTs_1=lforecast(netFit_ts_1,d_RUSDCOPTs_1,1017,50)

set.seed(45567)
netFit_ts_2 <- train(x=d_RUSDCOPTs_2[,1:8],y=d_RUSDCOPTs_2[,9],
                    method = "nnet",
                    metric = "RMSE",
                    trControl = fitControl,
                    tuneGrid = nnetGrid,
                    verbose = FALSE)

netFit_ts_2
plot(netFit_ts_2)
plot(varImp(object=netFit_ts_2),main="GBM - Variable Importance")

MFOR_RUSDCOPTs_2=lforecast(netFit_ts_2,d_RUSDCOPTs_2,1017,50)

set.seed(45567)
netFit_ts_3 <- train(x=d_RUSDCOPTs_3[,1:8],y=d_RUSDCOPTs_3[,9],
                    method = "nnet",
                    metric = "RMSE",
                    trControl = fitControl,
                    tuneGrid = nnetGrid,
                    verbose = FALSE)

netFit_ts_3
plot(netFit_ts_3)
plot(varImp(object=netFit_ts_3),main="GBM - Variable Importance")

MFOR_RUSDCOPTs_3=lforecast(netFit_ts_3,d_RUSDCOPTs_3,1017,50)

set.seed(45567)
netFit_ts_4 <- train(x=d_RUSDCOPTs_4[,1:8],y=d_RUSDCOPTs_4[,9],
                    method = "nnet",
                    metric = "RMSE",
                    trControl = fitControl,
                    tuneGrid = nnetGrid,
```

```
        verbose = FALSE)

netFit_ts_4
plot(netFit_ts_4)
plot(varImp(object=netFit_ts_4),main="GBM - Variable Importance")

MFOR_RUSDCOPts_4=lforecast(netFit_ts_4,d_RUSDCOPts_4,1017,50)

set.seed(45567)
netFit_ts_5 <- train(x=d_RUSDCOPts_5[,1:8],y=d_RUSDCOPts_5[,9],
                    method = "nnet",
                    metric = "RMSE",
                    trControl = fitControl,
                    tuneGrid = nnetGrid,
                    verbose = FALSE)

netFit_ts_5
plot(netFit_ts_5)
plot(varImp(object=netFit_ts_5),main="GBM - Variable Importance")

MFOR_RUSDCOPts_5=lforecast(netFit_ts_5,d_RUSDCOPts_5,1017,50)

set.seed(45567)
netFit_ts_6 <- train(x=d_RUSDCOPts_6[,1:8],y=d_RUSDCOPts_6[,9],
                    method = "nnet",
                    metric = "RMSE",
                    trControl = fitControl,
                    tuneGrid = nnetGrid,
                    verbose = FALSE)

netFit_ts_6
plot(netFit_ts_6)
plot(varImp(object=netFit_ts_6),main="GBM - Variable Importance")

MFOR_RUSDCOPts_6=lforecast(netFit_ts_6,d_RUSDCOPts_6,1017,50)

set.seed(45567)
```

```

netFit_ts_7 <- train(x=d_RUSDCOPts_7[,1:8],y=d_RUSDCOPts_7[,9],
                    method = "nnet",
                    metric = "RMSE",
                    trControl = fitControl,
                    tuneGrid = nnetGrid,
                    verbose = FALSE)

netFit_ts_7
plot(netFit_ts_7)
plot(varImp(object=netFit_ts_7),main="GBM - Variable Importance")

MFOR_RUSDCOPts_7=lforecast(netFit_ts_7,d_RUSDCOPts_7,1017,50)

```

### 6.10.3. Código: pesos iniciales redes neuronales

Se entrenan diferentes redes eligiendo distintos pesos iniciales para evitar caer en mínimos locales. Las redes son entrenadas con el 80% de las observaciones, el restante 20% se usa para la validación. Se usa el comando *fit* para estimar la red y se emplean los parámetros obtenidos en la fase de validación cruzada.

```

nstep <- 100
model_semilla_ts_1=rep(0,nstep)
rmse_ts_1=rep(0,nstep)

searchs=seq(1,nstep,1)

for(i in searchs){

set.seed(c(i))

M_RUSDCOPts_1=fit(y~.,d_RUSDCOPts_1[1:792,],model="mlpe",size=1,decay=0.01,
scale="all",rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)

model_semilla_ts_1[i]=i
rmse_ts_1[i]=mmetric(lforecast(M_RUSDCOPts_1,d_RUSDCOPts_1[1:792,],793,224),
d_RUSDCOPts_1$y[793:1016], "RMSE")
}

```

```
pesos_ts_1=cbind(model_semilla_ts_1,rmse_ts_1)

model_semilla_ts_2=rep(0,nstep)
rmse_ts_2=rep(0,nstep)

for(i in searchs){

set.seed(c(i))

M_RUSDCOPts_2=fit(y~.,d_RUSDCOPts_2[1:792,],model="mlpe",size=3,decay=0.01,
  scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
model_semilla_ts_2[i]=i
rmse_ts_2[i]=mmetric(lforecast(M_RUSDCOPts_2,d_RUSDCOPts_2[1:792,],793,224),
d_RUSDCOPts_2$y[793:1016],"RMSE")
}

pesos_ts_2=cbind(model_semilla_ts_2,rmse_ts_2)

model_semilla_ts_3=rep(0,nstep)
rmse_ts_3=rep(0,nstep)

for(i in searchs){

set.seed(c(i))

M_RUSDCOPts_3=fit(y~.,d_RUSDCOPts_3[1:792,],model="mlpe",size=10,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
model_semilla_ts_3[i]=i
rmse_ts_3[i]=mmetric(lforecast(M_RUSDCOPts_3,d_RUSDCOPts_3[1:792,],793,224),
d_RUSDCOPts_3$y[793:1016],"RMSE")
}

pesos_ts_3=cbind(model_semilla_ts_3,rmse_ts_3)

model_semilla_ts_4=rep(0,nstep)
rmse_ts_4=rep(0,nstep)
```

```
for(i in searches){

set.seed(c(i))

M_RUSDCOPts_4=fit(y~.,d_RUSDCOPts_4[1:792,],model="mlpe",size=10,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
model_semilla_ts_4[i]=i
rmse_ts_4[i]=mmetric(lforecast(M_RUSDCOPts_4,d_RUSDCOPts_4[1:792,],793,224),
d_RUSDCOPts_4$y[793:1016], "RMSE")
}

pesos_ts_4=cbind(model_semilla_ts_4,rmse_ts_4)

model_semilla_ts_5=rep(0,nstep)
rmse_ts_5=rep(0,nstep)

for(i in searches){

set.seed(c(i))

M_RUSDCOPts_5=fit(y~.,d_RUSDCOPts_5[1:792,],model="mlpe",size=2,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
model_semilla_ts_5[i]=i
rmse_ts_5[i]=mmetric(lforecast(M_RUSDCOPts_5,d_RUSDCOPts_5[1:792,],793,224),
d_RUSDCOPts_5$y[793:1016], "RMSE")
}

pesos_ts_5=cbind(model_semilla_ts_5,rmse_ts_5)

model_semilla_ts_6=rep(0,nstep)
rmse_ts_6=rep(0,nstep)

for(i in searches){

set.seed(c(i))
```

```

M_RUSDCOPts_6=fit(y~.,d_RUSDCOPts_6[1:792,],model="mlpe",size=1,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
model_semilla_ts_6[i]=i
rmse_ts_6[i]=mmetric(lforecast(M_RUSDCOPts_6,d_RUSDCOPts_6[1:792,],793,224),
d_RUSDCOPts_6$y[793:1016],"RMSE")
}

pesos_ts_6=cbind(model_semilla_ts_6,rmse_ts_6)

model_semilla_ts_7=rep(0,nstep)
rmse_ts_7=rep(0,nstep)

for(i in searchs){

set.seed(c(i))

M_RUSDCOPts_7=fit(y~.,d_RUSDCOPts_7[1:792,],model="mlpe",size=9,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
model_semilla_ts_7[i]=i
rmse_ts_7[i]=mmetric(lforecast(M_RUSDCOPts_7,d_RUSDCOPts_7[1:792,],793,224),
d_RUSDCOPts_7$y[793:1016],"RMSE")
}

pesos_ts_7=cbind(model_semilla_ts_7,rmse_ts_7)

pesos_ts_1[which.min(pesos_ts_1[,2]), ]
pesos_ts_2[which.min(pesos_ts_2[,2]), ]
pesos_ts_3[which.min(pesos_ts_3[,2]), ]
pesos_ts_4[which.min(pesos_ts_4[,2]), ]
pesos_ts_5[which.min(pesos_ts_5[,2]), ]
pesos_ts_6[which.min(pesos_ts_6[,2]), ]
pesos_ts_7[which.min(pesos_ts_7[,2]), ]

##pronostico

z=matrix(data = NA, nrow = 50, ncol = 1)

```

```
searchs=c(1:50)

for(i in searchs){
mraout_2=modwt(RUSDCOP[(i):(1023+i)], filter = "la8",
n.levels = floor(log(length(RUSDCOP[6:1029]))), fast = TRUE)
WaveletSeries_2 <- cbind(do.call(cbind, mraout_2@W),
mraout_2@V[[floor(log(length(RUSDCOP[6:1029])))]])

ts_1_2 <- as.numeric(WaveletSeries_2[,1])
ts_2_2 <- as.numeric(WaveletSeries_2[,2])
ts_3_2 <- as.numeric(WaveletSeries_2[,3])
ts_4_2 <- as.numeric(WaveletSeries_2[,4])
ts_5_2 <- as.numeric(WaveletSeries_2[,5])
ts_6_2 <- as.numeric(WaveletSeries_2[,6])
ts_7_2 <- as.numeric(WaveletSeries_2[,7])

d_RUSDCOPts_1_2=CasesSeries(ts_1_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_2_2=CasesSeries(ts_2_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_3_2=CasesSeries(ts_3_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_4_2=CasesSeries(ts_4_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_5_2=CasesSeries(ts_5_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_6_2=CasesSeries(ts_6_2[1:1024],c(1,2,3,4,5,6,7,8))
d_RUSDCOPts_7_2=CasesSeries(ts_7_2[1:1024],c(1,2,3,4,5,6,7,8))

set.seed(20)
M_RUSDCOPts_1_2=fit(y~.,d_RUSDCOPts_1_2[1:1016,],model="mlpe",size=1,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
set.seed(96)
M_RUSDCOPts_2_2=fit(y~.,d_RUSDCOPts_2_2[1:1016,],model="mlpe",size=3,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
set.seed(75)
M_RUSDCOPts_3_2=fit(y~.,d_RUSDCOPts_3_2[1:1016,],model="mlpe",size=10,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
set.seed(92)
M_RUSDCOPts_4_2=fit(y~.,d_RUSDCOPts_4_2[1:1016,],model="mlpe",size=10,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
```

```
maxit = 400, MaxNWts = 500)
set.seed(93)
M_RUSDCOPts_5_2=fit(y~.,d_RUSDCOPts_5_2[1:1016,],model="mlpe",size=2,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
set.seed(98)
M_RUSDCOPts_6_2=fit(y~.,d_RUSDCOPts_6_2[1:1016,],model="mlpe",size=1,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)
set.seed(90)
M_RUSDCOPts_7_2=fit(y~.,d_RUSDCOPts_7_2[1:1016,],model="mlpe",size=9,
decay=0.01, scale="all", rang = 0.7,abstol = 1.0e-3, reltol = 1.0e-3,
maxit = 400, MaxNWts = 500)

MFOR_RUSDCOPts_1_2=lforecast(M_RUSDCOPts_1,d_RUSDCOPts_1_2,1017,1)
MFOR_RUSDCOPts_2_2=lforecast(M_RUSDCOPts_2,d_RUSDCOPts_2_2,1017,1)
MFOR_RUSDCOPts_3_2=lforecast(M_RUSDCOPts_3,d_RUSDCOPts_3_2,1017,1)
MFOR_RUSDCOPts_4_2=lforecast(M_RUSDCOPts_4,d_RUSDCOPts_4_2,1017,1)
MFOR_RUSDCOPts_5_2=lforecast(M_RUSDCOPts_5,d_RUSDCOPts_5_2,1017,1)
MFOR_RUSDCOPts_6_2=lforecast(M_RUSDCOPts_6,d_RUSDCOPts_6_2,1017,1)
MFOR_RUSDCOPts_7_2=lforecast(M_RUSDCOPts_7,d_RUSDCOPts_7_2,1017,1)

TOTAL=MFOR_RUSDCOPts_3_2+MFOR_RUSDCOPts_4_2+
MFOR_RUSDCOPts_5_2+MFOR_RUSDCOPts_6_2+MFOR_RUSDCOPts_7_2

z[i,1]=TOTAL}

rmse(z,RUSDCOP[1025:1074])

DACTest(z, RUSDCOP[1025:1074], test = "PT", conf.level = 0.99)

plot(seq(900,1074),c(RUSDCOP[900:1024],z),type='l',ylab="RUSDCOP",
xlab=" ", main="MODWT-RED NEURONAL")
lines(seq(1025,1074),z,type='l',col='red',lwd=2)

lines(seq(1025,1074),RUSDCOP[1025:1074])
abline(v =1025,lty=2)
```

```
error_forred=z-RUSDCOP[1025:1074]  
dm.test(error_forred,error_RW)
```

# Bibliografía

- [Agiakloglou et al., 1993] Agiakloglou, C., Newbold, P., and Wohar, M. (1993). Bias in an estimator of the fractional difference parameter. *Journal of Time Series Analysis*, 14:235–246.
- [Álvarez and Álvarez, 2004] Álvarez, M. and Álvarez, A. (2004). Predicción no lineal de los tipos de cambio. aplicación de un algoritmo genético. *Revista de Economía Aplicada*, 12(35):55–64.
- [Bachelier, 1964] Bachelier, L. (1964). *The random character of stock market price*, chapter Theory of speculation. P. Cootner (ed.).
- [Baillie et al., 1996] Baillie, R., Bollerslev, T., and Mikkelsen, H. (1996). Fractionally integrated generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 74:3–30.
- [Ben Mabrouk et al., 2008] Ben Mabrouk, A., Ben Abdallah, N., and Z., D. (2008). Wavelet decomposition and autoregressive model for time series prediction. *Applied Mathematics and Computation*, 199:334–40.
- [Benjamini and Hochberg, 1995] Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. (57):289–300.
- [Boes et al., 1989] Boes, D., Davis, R., and Gupta, S. (1989). Parameter estimation in low order fractionally differenced arma processes. *Stochastic Hydrology and Hydraulics*, 3:97–110.
- [Bollerslev, 1986] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327.
- [Bollerslev and Mikkelsen, 1996] Bollerslev, T. and Mikkelsen, H. (1996). Modelling and pricing long memory in stock market volatility. (73):151–184.
- [Box and Jenkins, 1970] Box, G. and Jenkins, G. (1970). *Time series analysis, forecasting and control*. San Francisco, CA:.
- [Brock et al., 1996] Brock, W., Dechert, D., Scheinkman, J., and Lebaron, B. (1996). A test for independence based on the correlation dimension. Vol 15(3):197–235.

- [Brockwell and Davis, 2006] Brockwell, P. J. and Davis, R. A. (2006). *Time Series: Theory and Methods*. Springer-Verlag, New York, 2 edition.
- [Campbell et al., 1997] Campbell, J., Lo, A., and MacKinlay, C. (1997). *The Econometrics of Financial Markets*.
- [Castaño, 2016] Castaño, E. (2016). Identificación de modelos arfima. *Revista de la Facultad de Ciencias*, 5(1):12–37.
- [Castaño et al., 2008a] Castaño, E., Gómez, K., and Gallón, S. (2008a). Pronóstico y estructuras de volatilidad multiperíodo de la tasa de cambio del peso colombiano. *Cuadernos de Economía*, XXVII(48):241–266.
- [Castaño et al., 2008b] Castaño, E., Gómez, K., and Gallón, S. (2008b). Una nueva prueba para el parámetro de diferenciación fraccional. *Revista Colombiana de Estadística*, 31, no. 1:67–84.
- [Chang and Ting, 2000] Chang, K. and Ting, K. (2000). A variance ratio test of the random walk hypothesis for taiwanâs stock market. *Applied Financial Economics*, 10:525–532.
- [Cheung et al., 2005] Cheung, Y.-W., Menzie, C., and Garcia, A. (2005). *Recent exchange rate models: in-sample fit and out-of-sample performance*. Cambridge U. Press.
- [Cochrane, 1991] Cochrane, J. H. (1991). [pitfalls and opportunities: What macroeconomists should know about unit roots]: Comment. 6:201–210.
- [Conejo et al., 2005] Conejo, A. J., Plazas, M. A., Espínola, R., and Molina, A. B. (2005). Day-ahead electricity price forecasting using the wavelet transform and arima models. *IEEE TRANSACTIONS ON POWER SYSTEMS*, 20:1035–1042.
- [Daubechies, 1988] Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Comms Pure Appl. Math.*, 41:909–996.
- [Daubechies, 1992] Daubechies, I. (1992). *Ten Lectures on Wavelets*. SIAM, Philadelphia.
- [Davidson, 2004] Davidson, J. (2004). Moment and memory properties of linear conditional heteroscedasticity models, and a new model. *Journal of Business and Economics Statistics*, 22:16–29.
- [Diebold and Mariano, 1995] Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 20(1):134–144.
- [Ding and Granger, 1996] Ding, Z. and Granger, C. (1996). Modelling volatility persistence of speculative returns: A new approach. pages 185–215.

- [Engle, 1982] Engle, E. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica*, 50:987–1008.
- [Engle and Bollerslev, 1986] Engle, R. and Bollerslev, T. (1986). Modelling the persistence of conditional variances. *Econometric Reviews*, 5:1–50.
- [Fernandez, 2007] Fernandez, V. (2007). Wavelet-and svm-based forecasts: An analysis of the u.s. metal and materials manufacturing industry. *Resources Policy*, 32:80–89.
- [Franses and van Dijk, 2000] Franses, P. and van Dijk, D. (2000). *Non-linear time series models in empirical finance*.
- [Fryzlewicz, 2005] Fryzlewicz, P. (2005). Modelling and forecasting financial log-returns as locally stationary wavelet processes. *Journal of Applied Statistics*, 32:503–528.
- [Fryzlewicz et al., 2003] Fryzlewicz, P., Van Bellegem, S., and von Sachs, R. (2003). Forecasting nonstationary time series by wavelet process modelling. *Ann. Inst. Statist.Math.*, 55:737–764.
- [Gassberger and Procaccia, 1983] Gassberger, P. and Procaccia, I. (1983). Measuring the strangeness of strange attractors. Vol. 9:189–208.
- [Geweke and Porter-Hudak, 1983] Geweke, J. and Porter-Hudak, S. (1983). The estimation and application of long memory time series models. *Journal of Time Series Analysis*, 4(4):221–238.
- [Granger and Joyeux, 1980] Granger, C. W. J. and Joyeux, R. (1980). An introduction to long-memory time series models and fractional differencing. *Journal of Time Series Analysis*, 1:15–39.
- [Granger, 1980] Granger, W. (1980). Long memory relationships and the aggregation of dynamic models. *Journal of Econometrics*, 14:227–238.
- [Groen, 1999] Groen, J. (1999). Long horizon predictability of exchange rates: is it for real? *Empirical Economics*, 24:451–469.
- [Guo et al., 2008] Guo, X., Sun, L., Li, G., and Wang, S. (2008). A hybrid wavelet analysis and support vector machines in forecasting development of manufacturing. *Expert Systems with Applications*, 35:415–422.
- [Haykin, 1994] Haykin, S. (1994). *Neuronal Networks: A Comprehensive Foundation*.
- [Hosking, 1981] Hosking, J. R. M. (1981). Fractional differencing. *Biometrika*, 68:165–176.

- [Ismail et al., 2016] Ismail, M. T., B., A., and M., T. M. (2016). Volatility forecasting with the wavelet transformation algorithm garch model: Evidence from african stock markets. pages 125–135.
- [Jalil and Misas, 2007] Jalil, M. and Misas, M. (2007). Evaluación de pronósticos del tipo de cambio utilizando redes neuronales y funciones de pérdida asimétricas. *Revista Colombiana de Estadística*, 30(1):143–161.
- [Karthikeyan and Nagesh Kumar, 2013] Karthikeyan, L. and Nagesh Kumar, D. (2013). Predictability of nonstationary time series using wavelet and emd based arma models. *Journal of Hydrology*, 502:103–119.
- [Kilian and Taylor, 2003] Kilian, L. and Taylor, M. (2003). Why is it so difficult to beat the random walk forecast of exchange rates? *Journal of International Economics*, 60(1):85–107.
- [Liu, 2000] Liu, M. (2000). Modelling long memory in stock market volatility,. pages 139–171.
- [Lo and C., 1988] Lo, A. W. and C., M. (1988). Stock market prices do not follow random walks: Evidence from a simple specification test. *Review of Financial Studies*, 1:41–66.
- [Mallat et al., 1998] Mallat, S., Papanicolaou, G., and Zhang, Z. (1998). Adaptive covariance estimation of locally stationary processes. *Ann. Statist.*, 26:1–47.
- [Mallat, 1989] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 11:674–693.
- [Maya and Gómez, 2008] Maya, C. and Gómez, K. (2008). What exactly is  $\hat{\alpha}$  bad news in foreign exchange markets? evidence from latin american markets. *Cuadernos de Economía*, 45:161–183.
- [Meese and K.S., 1983] Meese, R. and K.S., R. (1983). Empirical exchange rate models of the seventies: Do they fit out of sample? (14):3–24.
- [Montenegro, 2010] Montenegro, R. (2010). Medición de la volatilidad en series de tiempo financieras. una evaluación a la tasa de cambio representativa del mercado (trm) en colombia. *Revista Finanzas y Política Económica*, 2(1):125–132.
- [Nason, 2008] Nason, G. (2008). *Wavelet Methods in Statistics with R*.
- [Nason, 2013] Nason, G. (2013). A test for second-order stationarity and approximate confidence intervals for localized autocovariances for locally stationary time series. B(75):879–904.

- [Nason and Silverman, 1995] Nason, G. P. and Silverman, B. W. (1995). *The stationary wavelet transform and some statistical applications*, volume 103 of *Lecture Notes in Statistics*, chapter Wavelets and Statistics, pages 281–229. Springer-Verlag, New York.
- [Nason et al., 2000] Nason, G. P., von Sachs, R., and Kroisandt, G. (2000). Wavelet processes and adaptive estimation of evolutionary wavelet spectra. *J. Roy. Statist. Soc. Ser. B*, 62:271–292.
- [Osborne, 1964] Osborne, M. F. (1964). *The random character of stock market price*, chapter Brownian motion in the stock market. The MIT Press, Cambridge, Massachusetts.
- [Parra, 2014] Parra, A. D. (2014). Estimación de la volatilidad de la tasa de cambio peso-dólar a través de un modelo de volatilidad estocástica. Master’s thesis.
- [Patiño and Alonso, 2005] Patiño, C. I. and Alonso, J. C. (2005). Evaluación de pronósticos para la tasa de cambio en colombia. (96):13–29.
- [Pesaran and Timmermann, 1992] Pesaran, M. and Timmermann, A. (1992). A simple non-parametric test of predictive performance. *Journal of Business and Economic Statistics*, 4:461–465.
- [Reisen, 1994] Reisen, V. A. (1994). Estimation of the fractional difference parameter in the arima (p, d, q) model using the smoothed periodogram. pages 335–350.
- [Rowland, 2002] Rowland, P. (2002). Forecasting the usd/cop exchange rate: A random walk with a variable drift. *Borradores de Economía*, 254.
- [Singh and Dey, 2002] Singh, J. P. and Dey, P. (2002). Risk measurement, nonlinearities, and chaos. *Singapore Management Review*, 24(2):47–58.
- [Smith et al., 1997] Smith, J., Taylor, N., and Yadav, S. (1997). Comparing the bias and misspecification in arfima models. *Journal of Time Series Analysis*, 18(5):507–527.
- [Soltani, 2002] Soltani, S. (2002). On the use of the wavelet decomposition for time series prediction. *Neurocomputing*, 48:267–277.
- [Stock, 2001] Stock, J. H. (2001). Forecasting economic time series.
- [Tan et al., 2010] Tan, Z., Zhang, J., Wang, J., and Xu, k. (2010). Day-ahead electricity price forecasting using wavelet transform combined with arima and garch models. *Applied Energy*, 87:3606–3610.
- [Teräsvirta et al., 1994] Teräsvirta, T., Tjøstheim, D., and Granger, C. W. J. (1994). *Aspects of modelling nonlinear time series*.
- [Tsay, 2001] Tsay, R. (2001). *Analysis of Financial Time Series*. New York: Wiley.

- 
- [Walzack, 2001] Walzack, S. (2001). An empirical analysis of data requirements for financial forecasting with neural networks. *Journal of management Information Systems*, 4:203–222.
- [Wong et al., 2003] Wong, H., Ip, W., Xie, Z., and Zhongjie, X. (2003). Modelling and forecasting by wavelets, and the application to exchange rates. *Journal of Applied Statistics*, 30:537–553.
- [Xie et al., 2009] Xie, Y., Yu, J., and Ranney, B. (2009). Forecasting using locally stationary wavelet processes. 79.
- [Zhu et al., 2014] Zhu, L., Wang, Y., and Fan, Q. (2014). Modwt-arma model for time series prediction. *Applied Mathematical Modelling*, 38:1859–1865.