TESIS DOCTORAL

2023

INGENIERÍA DE CARACTERÍSTICAS EN INTELIGENCIA ARTIFICIAL PARA DATOS FINANCIEROS DE ALTA FRECUENCIA

PABLO MANTILLA GÓMEZ

PROGRAMA DE DOCTORADO EN INGENIERÍA DE SISTEMAS Y CONTROL

DIRECTOR: SEBASTIÁN DORMIDO CANTO

En la memoria de esta tesis doctoral se ha tratado de proporcionar una traducción de todos los términos anglosajones utilizados, pero en aquellos casos donde la traducción podría conducir a confusión se ha optado por mantener el término anglosajón.

La investigación relativa a esta tesis doctoral ha sido objeto de la siguiente publicación científica:

> Mantilla, P., Dormido-Canto, S. (2023). A novel feature engineering approach for high-frequency financial data. *Engineering Applications of Artificial Intelligence, 125*, 106705. https://doi.org/10.1016/j.engappai.2023.106705

Contenido

Resumen **vii**

1 Introducción **1**

2 Objetivos **6**

3 Estado del arte **10**
3.1 Segmentación 11
3.2 Volatilidad intradiaria y duración 13
3.3 Inteligencia artificial en predicción direccional 17

4 Metodología **24**
4.1 Planteamiento del problema 25
4.2 Propuesta de resolución 25
4.2.1 Doble segmentación 26
4.2.2 Procedimiento de evaluación de la segmentación . . 35
4.2.3 Extracción de características 38

5 Aplicación **45**
5.1 Etiquetado 46
5.2 Incrustación 47
5.3 Modelización 48
5.4 Métricas de rendimiento 51

6 Experimentación **56**
6.1 Datos financieros de alta frecuencia 57
6.2 Reconstrucción del libro de órdenes límite 69
6.3 Segmentación 75
6.4 Selección de variables 80
6.5 Aprendizaje automático 93
6.6 Recursos computacionales 96

7 Resultados y discusión **98**

8 Conclusiones y líneas futuras **112**

Lista de figuras

4.1 Esquema de ingeniería de características 25
4.2 Agregación de períodos 28
4.3 BIC y RSS de la segmentación 34
4.4 Ejemplo de segmentación 34
4.5 Estado del LOB en un instante 40
4.6 Proceso de extracción de características del LOB 42
4.7 Características de los segmentos 43

5.1 Esquema de aprendizaje automático 47
5.2 Esquema de incrustación 48

6.1 Procesado previo de los datos 59
6.2 Series temporales de operaciones negociadas. Transaction time 65
6.3 Series temporales de operaciones negociadas I. Tick time . 66
6.4 Series temporales de operaciones negociadas II. Tick time . 67
6.5 Órdenes límite por día. Datos brutos 70
6.6 Ciclo de vida de las órdenes 72
6.7 Promedio de posiciones de métodos de segmentación . . . 79
6.8 Evolución de la duración 84
6.9 Evolución del retorno por segundo 87
6.10 Evolución de la volatilidad por unidad de tiempo 89

7.1 Tiempo de ejecución total de trabajos de segmentación . . 101
7.2 BIC diario por activo financiero y método 102
7.3 Volatilidad. Importancia de variables 109
7.4 Duración. Importancia de variables 110
7.5 Dirección. Importancia de variables 111

Lista de tablas

6.1 Variables de los datos de operaciones negociadas 58
6.2 Operaciones simultáneas. Datos brutos 60
6.3 Observaciones y valores atípicos. Datos limpios 64
6.4 Operaciones negociadas diarias 68
6.5 Variables de los datos de órdenes 69
6.6 Observaciones diarias por períodos de agregación 76
6.7 Test de Friedman 77
6.8 Test de Nemenyi 78
6.9 Matrices de Nemenyi 78
6.10 Segmentos por día 80
6.11 Selección de características 82
6.12 Segmentos y casos 94

7.1 Resultados segmentación 100
7.2 Resultados predicción volatilidad 105
7.3 Resultados predicción duración 106
7.4 Resultados predicción dirección 107

Algoritmos

1 Double segmentation with period aggregation 29
2 LOB features extraction 74

Abreviaturas

AIC	Akaike Information Criterion
ACD	Autoregressive Conditional Duration
ARCH	Autoregressive Conditional Heteroskedastic
BIC	Bayesian Information Criterion
B3	Brasil, Bolsa, Balcão
CNN	Convolutional Neural Networks
DSPA	Double Segmentation with Period Aggregation
XGBoost	Extreme Gradient Boosting
KNN	K-nearest Neighbors
LOB	Limit Order Book
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
OBI	Order Book Imbalance
POS	Porcentaje de Operaciones Simultáneas
POSDP	Porcentaje de Operaciones Simultáneas con Distinto Precio
RF	Random Forest
RSS	Residual Sum of Squares
SVM	Support Vector Machines
UHF-GARCH	Ultra-High-Frequency Generalized Autoregressive Conditional Heteroskedastic

Resumen

Ingeniería de características en inteligencia artificial para datos financieros de alta frecuencia

Los datos financieros de alta frecuencia están formados por series temporales multivariantes que se registran con precisiones mínimas del orden de milisegundos y a intervalos irregulares, generándose grandes volúmenes de datos. Estos datos provienen de las órdenes de compra y venta que llegan al sistema del mercado financiero, donde algunas de éstas se cruzan, dando lugar a las series de operaciones negociadas, y otras permanecen en cola, cuyo procesamiento consiste en la reconstrucción del denominado libro de órdenes límite.

Las series de operaciones negociadas experimentan tendencias intradiarias de duración irregular, por lo que están compuestas de un número variable de observaciones. El objetivo de la investigación es extraer características de este tipo de movimientos y de los estados del libro de órdenes límite en cada instante de las tendencias, con la finalidad de que puedan analizarse y utilizarse como input de modelos de inteligencia artificial para

predecir el comportamiento de variables respuesta en tendencias futuras. Este problema puede resolverse de forma satisfactoria con una ingeniería de características específica, cuyo primer paso consiste en obtener secuencias temporales de subconjuntos con un número de observaciones variable, basándose en el criterio de que cada subconjunto debe contener una tendencia intradiaria de la serie de partida. Para ello, es preciso dividir la serie original en fragmentos que contengan las tendencias citadas.

La técnica apropiada para fragmentar estas series en tendencias es la segmentación de series temporales. El propósito es que los segmentos contengan movimientos direccionales claramente definidos, por lo que dichos movimientos deben delimitarse de la forma más precisa posible. El método de segmentación que la literatura científica reconoce como más preciso es el denominado método óptimo o exacto, el cual consiste en ajustar las tendencias de la serie temporal con líneas rectas. El inconveniente de este método es su complejidad algorítmica, que es de orden cuadrático, por lo que no estaría indicada su aplicación directa para las series temporales de alta frecuencia con mayor número de observaciones del mercado financiero, ya que los tiempos de ejecución de los trabajos de segmentación serían tan elevados que resultaría inviable el empleo del método. Para salvar este obstáculo, se diseñó un método preciso y viable para segmentar las series temporales de alta frecuencia citadas, que se denominó *double segmentation with period aggregation (DSPA)*. Este método está basado en el método óptimo o exacto y consiste en realizar una agregación previa de los datos a una frecuencia inferior a la inicial. De esta forma, se reduce considerablemente el número de observaciones de la serie de partida. A continuación, se segmenta la serie agregada con el método óptimo o exacto y se obtienen los puntos de ruptura entre un segmento y el siguiente,

los cuales se trasladan a la serie original, obteniéndose una primera partición de los datos. Sobre los segmentos resultantes, se realiza una segunda partición, obteniéndose los segmentos finales, los cuales constituyen los subconjuntos sobre los que se van a extraer las características respectivas.

Con el fin de proporcionar una aplicación de la técnica desarrollada, se planteó la predicción de tres variables respuesta utilizando la ingeniería de características diseñada, tomando como regresores una selección de variables. Para ello, se construyeron múltiples modelos de aprendizaje automático basados en el algoritmo *extreme gradient boosting*, con el objeto de predecir la volatilidad, la duración y la dirección asociadas a tendencias intradiarias futuras.

La experimentación se realizó con 26 activos cotizados de la Bolsa de Valores de Brasil. Los experimentos relativos a la segmentación se ejecutaron con 6 valores localizados entre las últimas posiciones de los 150 activos más negociados del mercado, los cuales tienen un número de observaciones suficientemente pequeño para permitir ejecutar la segmentación con el método óptimo o exacto y poder realizar una comparación con la alternativa propuesta. Los 20 activos restantes se encuentran entre los más operados del mercado, y se emplearon para mostrar la aplicación de la ingeniería de características diseñada.

Los resultados del método de segmentación DSPA se evaluaron estadísticamente, comparando el método óptimo con tres modalidades de agregación de períodos, en términos de tiempo de ejecución, precisión y un criterio de selección de modelos. Los tres tipos de agregación consiguieron mayor rapidez de ejecución que el método óptimo, además de proporcionar

un menor error total.

El método de doble segmentación consigue la segmentación de series temporales de alta frecuencia de forma precisa y en un tiempo razonable, y permite extraer características de las tendencias intradiarias para su análisis. Asimismo, la ingeniería de características basada en dicho método permite predecir variables respuesta vinculadas a tendencias intradiarias de alta frecuencia, mediante la utilización de métodos de inteligencia artificial.

La aplicación de la ingeniería de características a la predicción de la volatilidad, la duración y la direccionalidad obtuvo precisiones superiores en las predicciones de las dos primeras variables que en la última. La importancia de las variables seleccionadas se obtuvo a partir de los modelos basados en el algoritmo de inteligencia artificial *extreme gradient boosting*, y se determinó, entre otros aspectos, que las variables vinculadas a las series de precios de cotización explican mejor la varianza de las variables respuesta que las variables asociadas al libro de órdenes límite, con el método y los datos utilizados.

1

Introducción

En la nueva era de la tecnología, la inteligencia artificial se ha convertido en el tema central de la ciencia de datos, con aplicaciones en numerosos campos de conocimiento. Uno de estos campos es el de las finanzas, donde las técnicas de aprendizaje automático (*machine learning*) han surgido como herramientas poderosas para el tratamiento de datos financieros de alta frecuencia, a la vez que se han convertido en un área de investigación cada vez más importante para abordar el análisis y la predicción con este tipo de datos. Por otra parte, el avance tecnológico de los últimos años y la llegada de los sistemas electrónicos a los mercados bursátiles han permitido almacenar grandes volúmenes de datos de alta frecuencia para su procesamiento, previamente registrados con la alta precisión con la que fueron generados. Dicha coyuntura ha provocado un auge en el desarrollo y la aplicación de algoritmos de aprendizaje a este tipo de datos, como se

Pablo Mantilla Gómez
Programa de Doctorado en Ingeniería de Sistemas y Control

muestra en la completa revisión presentada en Ntakaris et al. (2018).

Estos datos presentan la particularidad de la alta velocidad con la que se generan, en marcos temporales de milisegundos o incluso a frecuencias superiores, lo que resulta en un elevado volumen de datos, si se compara con el originado a bajas frecuencias. Por otro lado, no se trata de las series temporales de cotizaciones habituales, ya que estos datos provienen de órdenes de compra y venta que llegan al sistema del mercado, donde algunas de éstas se cruzan, dando lugar a las denominadas operaciones negociadas (*trades*), mientras que otras órdenes permanecen en cola, cuyo procesamiento consiste en la reconstrucción del llamado libro de órdenes límite (*limit order book* [LOB]). Los datos finales contienen múltiples variables que evolucionan con el tiempo, lo cual ha contribuido a que estos datos hayan sido objeto de multitud de trabajos de investigación desde el punto de vista estadístico (Hautsch, 2012). Sin embargo, la propia naturaleza de los datos financieros de alta frecuencia provoca que sus distribuciones no se mantengan en el tiempo, por lo que esta circunstancia ha creado el marco de trabajo propicio para que la resolución de problemas predictivos utilizando estos datos se aborde con otro tipo de métodos, basados en inteligencia artificial, en los que no es necesario hacer suposiciones sobre las distribuciones de los datos (Nousi et al., 2019).

El progresivo interés científico por la combinación de los campos de conocimiento relativos al análisis de datos financieros de alta frecuencia y a la utilización de métodos de aprendizaje automático con fines predictivos es evidente en las múltiples referencias citadas en esta tesis doctoral, la mayoría de las cuales han sido publicadas en años recientes. Este creciente cuerpo de literatura reconoce la importancia de usar métodos basados

en inteligencia artificial para analizar, modelizar y pronosticar el comportamiento de los mercados financieros. Además, existe un fuerte interés en el sector privado por este tema de investigación, debido al hecho de que los bancos de inversión, los *hedge funds* y otras firmas de gestión de patrimonios compiten para obtener predicciones exitosas y conseguir mayores ganancias con menores riesgos.

Con la finalidad de aplicar técnicas de aprendizaje automático sobre los datos brutos que provienen de los mercados financieros y resolver los múltiples problemas predictivos existentes en alta frecuencia, es necesario extraer características relevantes de estos datos, por lo que deben ser previamente analizados y tratados, utilizando técnicas que permitan alcanzar dicho objetivo. Los algoritmos de aprendizaje automático pueden utilizar como entrada las variables originales de los datos, pero también pueden emplear una transformación de dichas variables, con objeto de capturar información adicional relevante que permita identificar patrones presentes en los datos. Se trata de extraer las características de los datos que mejor expliquen la variación en la variable a predecir. En este sentido, la ingeniería de características contempla los procesos que permiten utilizar el conocimiento para extraer características de los datos brutos y proporcionar una mejora en el rendimiento de los modelos de inteligencia artificial construidos.

La literatura revisada contempla la extracción de características en conjuntos de observaciones con un número constante de elementos, haciendo predicciones sobre un horizonte de predicción fijo. En esta tesitura se encuentran los problemas de predicción habituales en este campo de investigación, como por ejemplo la predicción de la volatilidad de alta fre-

cuencia y la predicción de la direccionalidad asociada a los movimientos de las cotizaciones. En dichos casos, las predicciones se realizan sobre la base de un esquema de muestreo y horizonte de predicción definidos previamente. Sin embargo, la volatilidad es una variable que presenta *clustering* en sus valores, además de no comportarse de forma simétrica con los movimientos alcistas y bajistas de los mercados (Engle & Patton, 2001), por lo que se considera razonable agrupar la volatilidad por movimientos tendenciales del precio de cotización de los activos financieros. Por otra parte, las cotizaciones de alta frecuencia suelen experimentar tendencias intradiarias de duración irregular. Estas particularidades despiertan el interés por conocer cuál sería la evolución de una volatilidad relativa vinculada a la duración de tendencias intradiarias, o qué dirección o duración tendría una futura tendencia intradiaria. Pero el problema que se plantea no se limita a dichas variables, existen muchas otras cuya evolución podría vincularse a la duración de estos movimientos que se mantienen durante períodos de tiempo variables.

Revisado el estado del arte en el que se encuadra esta investigación doctoral, no consta que se hayan publicado trabajos enfocados en la extracción de características exclusivamente de tendencias de alta frecuencia cuyo objetivo haya sido el diseño de una ingeniería de características que permita predecir variables relativas a futuras tendencias utilizando inteligencia artificial. Este hueco en la literatura ha servido de motivación para proponer un nuevo problema no abordado previamente y enfrentarse al gran desafío de segmentar con precisión series temporales de alta frecuencia con decenas de miles de observaciones diarias, con la finalidad de obtener tendencias de alta frecuencia y extraer características de éstas, así como de las órdenes límite asociadas a las mismas.

Con esta motivación, se desarrolló una metodología que proporciona una respuesta a los interrogantes anteriormente formulados y que se fundamenta, se desarrolla y se aplica de acuerdo a lo expuesto en las siguientes páginas de esta tesis doctoral, cuya estructura general se compone de ocho capítulos, incluida esta parte introductoria. El Capítulo 2 recoge el objetivo principal y los objetivos específicos perseguidos en la investigación realizada. El Capítulo 3 contiene una revisión de la literatura relacionada con la investigación desarrollada. En el Capítulo 4, se plantea el problema de investigación y su propuesta de resolución. El Capítulo 5 recoge la aplicación de la metodología desarrollada a problemas predictivos, en los que se utiliza el algoritmo de inteligencia artificial denominado *extreme gradient boosting* (XGBoost). En dicho capítulo, se revisan conceptos teóricos del método de inteligencia artificial utilizado para aplicar la metodología diseñada, así como las métricas de rendimiento empleadas para evaluar la capacidad predictiva de los modelos construidos. El Capítulo 6 proporciona una descripción detallada de la laboriosa y extensa experimentación realizada, en la que se ejecuta y evalúa la metodología de segmentación diseñada y en la que se aplica la ingeniería de características desarrollada a tres problemas predictivos concretos. Los resultados se discuten en el Capítulo 7. Finalmente, en el Capítulo 8 se emiten las conclusiones finales y los posibles trabajos futuros.

2

Objetivos

El objetivo general de la investigación doctoral es extraer características de las tendencias que experimentan las series temporales de alta frecuencia relativas a las operaciones negociadas en los mercados financieros, así como de los estados del libro de órdenes límite en cada instante que se generan las operaciones de estos movimientos tendenciales.

Una vez planteado este objetivo, nos encontramos con varios problemas asociados para los cuales es preciso encontrar una solución. En primer lugar, surge la necesidad de fragmentar las series de operaciones negociadas en tendencias. El objetivo es que las tendencias se delimiten de forma precisa y en un tiempo razonable, que no exceda de unos días. Este no es un problema menor, ya que las series temporales financieras de alta frecuencia pueden tener decenas de miles de operaciones diarias, por lo que

desarrollar y automatizar un método de segmentación específico para este tipo de datos se convierte en un problema computacional, en el cual se deben valorar los recursos de computación necesarios y disponibles, la precisión del método de segmentación diseñado y los tiempos de espera hasta que se ejecuten los trabajos de segmentación en un supercomputador.

Para valorar adecuadamente un método de segmentación diseñado específicamente para este tipo de datos, se puede comparar con algún método existente que reúna, al menos, la característica de precisión requerida, y que pueda utilizarse con conjuntos de datos de series temporales de alta frecuencia. En este caso, se establecen las métricas a comparar y se utiliza un método comparación que determine cuál es el mejor método para lograr el objetivo establecido.

Una vez que se obtienen las tendencias utilizando un método de segmentación adecuado, es preciso delimitar los conjuntos de órdenes de compra y venta que se producen en cada una de dichas tendencias. Para ello, es necesario obtener los estados del libro de órdenes límite en cada instante que se producen las operaciones negociadas que integran las tendencias. Nuevamente, se plantea un problema computacional, ya que el número de órdenes diarias podría ser de cientos de miles para cada activo financiero, por lo que es necesario desarrollar un algoritmo específico para los datos utilizados que permita alcanzar el objetivo fijado.

Cumplidos los objetivos formulados, se pueden extraer características de tendencias de alta frecuencia a partir de los conjuntos de datos obtenidos para cada una de las mismas. La utilidad de la metodología desarrollada se muestra a partir de su aplicación a problemas concretos. En la

investigación doctoral, se planteó la predicción de tres variables respuesta vinculadas a tendencias de alta frecuencia, realizada a partir del diseño de modelos basados en inteligencia artificial. A estos efectos, se establece el objetivo de predecir la volatilidad, la duración y la dirección de tendencias de alta frecuencia futuras, planteándose problemas de clasificación para series temporales.

La metodología diseñada permite analizar variables sobre intervalos temporales en los que se producen tendencias intradiarias y también determinar cómo se comportaría una variable en intervalos futuros, utilizando valores de variables en intervalos previos, tratándolas como regresores que explican la varianza de la variable respuesta en modelos de inteligencia artificial.

El esquema de muestreo y predicción de la metodología desarrollada constituye un enfoque diferente al tradicional, cuya utilidad está dirigida a la resolución de un problema concreto con datos financieros de alta frecuencia, donde la finalidad es analizar el comportamiento de variables en intervalos temporales sobre los que se producen tendencias intradiarias, así como predecir el comportamiento de variables respuesta en intervalos futuros.

Por tanto, la principal contribución de esta investigación es la metodología para extraer características de subconjuntos de datos financieros de alta frecuencia de composición variable, además de proporcionar un método preciso y viable para segmentar series temporales de alta frecuencia con otras finalidades distintas de las que aquí se destacan. Adicionalmente, se diseñó un procedimiento para evaluar métodos de segmentación de

series temporales basado en comparación estadística múltiple. Por todo ello, se considera que se ha planteado un nuevo problema de inteligencia artificial con datos financieros de alta frecuencia, ya que no consta que este problema haya sido abordado previamente.

3

Estado del arte

Con carácter general, el estado del arte relativo a la investigación realizada se enmarca en el campo de conocimiento derivado de la aplicación de métodos de inteligencia artificial a datos financieros de alta frecuencia. En concreto, se trata de una ingeniería de características para datos de alta frecuencia, donde los enfoques revisados parten de subconjuntos de datos de composición fija, sobre los que se extraen las características correspondientes. Por el contrario, la ingeniería de características diseñada contempla un paso previo adicional, el cual consiste en la construcción de subconjuntos de datos de composición variable, de acuerdo a un criterio específico. Por otra parte, la ingeniería de características propuesta se basa en la segmentación de series temporales, y en este campo de conocimiento también se propone un nuevo método, enfocado a la segmentación de series temporales de alta frecuencia. Finalmente, la metodología de-

sarrollada se aplica a un problema predictivo en el que sus esquemas de muestreo y de predicción se fundamentan en la particularidad de la extracción de características diseñada. Por todo ello, se considera que el problema que se está planteando es totalmente novedoso desde el punto de vista metodológico. No obstante, aparte de trabajos previos de segmentación de series temporales, existen otros trabajos relacionados con la propuesta de investigación que se presenta, los cuales se comentan a continuación. En primer lugar, se revisan publicaciones referidas a la segmentación óptima de series temporales. Posteriormente, se relacionan aquellos artículos que han tratado, de alguna forma, la predicción de variables relacionadas con las variables respuesta seleccionadas para aplicar la ingeniería de características desarrollada.

3.1 Segmentación

La segmentación de series temporales es un problema que se ha investigado durante décadas. Su resolución permite obtener fragmentos de serie temporal con una estructura determinada. Revisiones de los avances conseguidos y de las múltiples técnicas empleadas se encuentran en los trabajos de Keogh et al. (2004) y de Lovrić et al. (2014).

En la investigación realizada, se destacó la importancia de delimitar las tendencias intradiarias de la forma más precisa posible, por lo que era necesario que el método de segmentación empleado cumpliese dicho requisito, además de permitir la obtención de segmentos en un tiempo de ejecución razonable. El método más preciso recogido por el estado del arte es el que utiliza la técnica *piecewise linear representation*, referida a la aproximación de una serie temporal T de n observaciones con K

líneas rectas. Para llevar a cabo dicha aproximación, se utilizó regresión lineal con algoritmos de programación dinámica, empleados inicialmente por Bellman & Roth (1969) y posteriormente por Bai & Perron (2003) y Zeileis et al. (2003).

El método citado, denominado por múltiples autores como método óptimo o exacto, se ha estado utilizando como *benchmark* por las diversas aproximaciones que han surgido, debido a su precisión. Las aproximaciones han tratado de obtener algoritmos con la precisión más próxima al *benchmark*, pero con un tiempo de ejecución menor, principalmente debido a que el método óptimo es un algoritmo con complejidad de orden cuadrático $O(n^2)$, en el cual el tiempo de ejecución aumenta como máximo de forma cuadrática con el número de observaciones n. Por lo tanto, este algoritmo, directamente aplicado sobre datos brutos, no es adecuado para segmentar series temporales con un gran número de observaciones, como es el caso de los datos de alta frecuencia, ya que sus tiempos de ejecución serían muy elevados, como se muestra en el Capítulo 6, relativo a la experimentación realizada.

Existen múltiples alternativas que recogen métodos aproximados del problema de la segmentación, los cuales no resultan adecuados para la resolución del problema que se plantea, debido a la pérdida de precisión. Evidencias de la precisión del método exacto se encuentran en Terzi & Tsaparas (2006), donde se consideró que la segmentación óptima se obtiene con algoritmos de programación dinámica, pero debido a la complejidad algorítmica de éstos, existen aproximaciones más rápidas. De forma similar, en Lemire (2007) se contempló como solución exacta la proporcionada por la programación dinámica. Además, en esta publicación se

recogen las características que debe tener un algoritmo de segmentación de series temporales, entre las que se encuentran la precisión y la rapidez, en términos de bondad del ajuste y tiempo de ejecución, respectivamente. En Chundi & Rosenkrantz (2009), consideraron que un planteamiento basado en programación dinámica proporciona la solución óptima, pero el inconveniente es su elevado tiempo de ejecución.

3.2 Volatilidad intradiaria y duración

En la investigación realizada, se aplicó la metodología diseñada a la predicción de la volatilidad intradiaria asociada a movimientos tendenciales de alta frecuencia de los precios de cotización de activos financieros. Se utilizó un *blocking scheme* dinámico sin solapamiento, es decir, la volatilidad se computó secuencialmente por intervalos de duración variable, que no comparten observaciones entre ellos. No consta que este tipo de esquema de muestreo se encuentre entre los recogidos por la literatura vinculada a la volatilidad de alta frecuencia. Por otra parte, se empleó una variable proxy de la volatilidad que está descrita en la literatura científica. En este apartado, se revisan múltiples modelos predictivos de la volatilidad intradiaria de alta frecuencia, así como los principales esquemas de predicción y variables proxies habituales.

Existen diferentes tipos de volatilidad y de modelos. En Engle & Patton (2001), se recogen una serie de hechos que caracterizan la volatilidad, y en Poon & Granger (2003) se hace una profunda revisión de la materia, donde se relatan múltiples variables proxies usadas para computar la volatilidad, así como detalles de la experimentación realizada con una multitud de modelos y diversos horizontes y esquemas de predicción. Se

destacan dos formas de cálculo de la volatilidad utilizadas con frecuencia, recogidas en Giot (2001) como formas habituales de computar la volatilidad intradiaria. Se trata de la varianza de los retornos en un intervalo de tiempo determinado (Ederington & Lee, 1993) y la media de los retornos al cuadrado sobre un intervalo temporal (Gwilym et al., 1997).

Los modelos de alta frecuencia para estimar la volatilidad intradiaria se clasifican en dos clases (Giot, 2001): modelos de series temporales de tipo *autoregressive conditional heteroskedastic (ARCH)* (Engle, 1982) adaptados a datos financieros de alta frecuencia (Andersen & Bollerslev, 1997), y los que utilizan modelos de duración de tipo *autoregressive conditional duration (ACD)* (Engle & Russell, 1997) y (Engle & Russell, 1998) o *Log-ACD* (Bauwens & Giot, 2000). Los primeros tratan con datos regularmente espaciados, muestreados sobre los datos brutos con una periodicidad determinada, en los que no se tiene en cuenta la información proporcionada por el tiempo que transcurre entre los eventos del mercado. En cuanto a los modelos que contemplan la duración entre observaciones, emplean datos irregularmente espaciados adaptados. Una revisión de los modelos de duración se encuentra en Pacurar (2008). También se revisan este tipo de modelos y los relativos a la volatilidad de alta frecuencia en Hautsch (2012).

En Engle (2000), se extendió el modelo ACD básico recogido en Engle & Russell (1997) y en Engle & Russell (1998), y se presentó el modelo *ultra-high-frequency generalized autoregressive conditional heteroskedastic (UHF-GARCH)*, el cual se combinó con el modelo ACD y se destinó a datos irregularmente espaciados, donde se estimó la volatilidad condicionada por unidad de tiempo, computada como la varianza de los retornos

logarítmicos divididos por la raíz cuadrada de la duración entre operaciones negociadas. Dicha volatilidad se propuso como la más adecuada para datos irregulares, ya que la métrica está ajustada por la duración entre transacciones. El modelo *UHF-GARCH* también se trató en Racicot et al. (2008).

En relación a los esquemas de predicción, en West (2006) se recogen las tres modalidades habituales en la literatura de predicción, también discutidas en Violante & Laurent (2012). Se trata del *recursive scheme*, el *rolling scheme* y el *fixed scheme*. Cada uno de los esquemas se compone de muestra de estimación y muestra de evaluación de la predicción u horizonte de predicción. Considerando el horizonte de predicción de un paso (*one-step ahead*), el primero de los esquemas parte de un número de observaciones sobre las que se construye el modelo, se realiza una predicción y se avanza un paso, diseñándose nuevamente el modelo con la observación adicional acumulada, y así sucesivamente. El *rolling scheme* consiste en una ventana deslizante de tamaño fijo, por lo que los parámetros del modelo siempre se estiman sobre un número constante de observaciones, que va variando según el avance establecido. Finalmente, el *fixed scheme* se caracteriza por construir el modelo una única vez, sobre un número de observaciones determinado, y las predicciones se realizan siempre con el mismo modelo.

La volatilidad también se ha tratado con modelos de inteligencia artificial. En dichos modelos, el esquema de predicción habitual con una variable autorregresiva es similar a los comentados, aunque las modalidades se pueden combinar en función de la etapa de construcción del modelo de aprendizaje automático. En primer lugar, se selecciona el esquema de

muestreo para extraer características de los datos. Se puede computar una variable proxy de la volatilidad con un *rolling scheme* de tamaño y avance fijo, o con un *blocking scheme*, el cual se diferencia del primero en que las observaciones sobre las que se calcula la volatilidad son siempre diferentes, no hay solapamiento. En segundo lugar, se elige el esquema de incrustación (*embedding*), referido a la representación y configuración vectorial de las variables que integran el input de los modelos de aprendizaje automático, que podría ser cualquiera de los comentados para los modelos tradicionales. Existen incrustaciones con un esquema *multistep*, empleando *recursive* o *rolling scheme*, pero la modalidad más habitual utiliza un *fixed scheme*, donde el conjunto de datos con las volatilidades calculadas se divide en dos partes, una se utiliza para la fase de entrenamiento de los modelos y otra para la fase de validación de éstos. Con el primer subconjunto se entrena el modelo, y con el segundo se evalúa. Es habitual utilizar un *rolling scheme* en cada uno de los subconjuntos de entrenamiento (*train*) y validación (*test*), consistente en una incrustación compuesta de casos (*samples*) con un tamaño de ventana fijo, donde el número de regresores k es igual al número de retardos (*lags*) seleccionado, y $k+1$ es la volatilidad que se pretende predecir. Un ejemplo del tratamiento de la volatilidad con modelos de aprendizaje automático se encuentra en Ramos-Pérez et al. (2019), donde emplearon los métodos *gradient descent boosting*, *random forest (RF)*, *support vector machines (SVM)* y *artificial neural networks* para predecir la volatilidad del índice S&P500. En dicha publicación, se utilizó como proxy de la volatilidad la desviación estándar de los retornos, calculada con una ventana de tamaño igual a cinco observaciones. También en Liu (2019) utilizaron SVM, redes neuronales *long short-term memory recurrent neural networks (LSTM RNNS)* y datos diarios del índice citado, además de datos relativos a la

acción *Apple Inc.* En este caso, el objetivo fue predecir la volatilidad en un horizonte de uno y tres días, y las redes LSTM RNNs se comportaron de forma similar a las SVM.

Aparte de los modelos estadísticos previamente referenciados, también se han desarrollado modelos de aprendizaje automático para predecir la volatilidad intradiaria. En Guo et al. (2018), se predijo la volatilidad de corto plazo utilizando la volatilidad histórica y el libro de órdenes límite. Los autores de la publicación utilizaron los métodos RF, XGBoost y redes neuronales LSTM. En Peng et al. (2018), se predijo la volatilidad de criptodivisas utilizando *support vector regression* y datos diarios e intradiarios, con un horizonte de predicción de una hora para el caso intradiario. En Doering et al. (2017), se predijo la volatilidad de alta frecuencia en un horizonte de 20 eventos, para lo cual se utilizó un modelo de clasificación binaria con *convolutional neural networks (CNN)*, con el que se emplearon datos de libros de órdenes límite de la acción *Barclays PLC*, la cual cotiza en la Bolsa de Londres.

3.3 Inteligencia artificial en predicción direccional

En el análisis de datos financieros de alta frecuencia, la predicción direccional es uno de los problemas habituales abordado con enfoques de aprendizaje automático. Este tema ha cobrado importancia a la luz de resultados recientes, que prueban que los métodos basados en inteligencia artificial son la opción más adecuada para enfrentarse a este problema.

Los estudios realizados durante la última década han proporcionado información importante sobre este desafío, en el que se hacen predicciones

direccionales basadas en datos históricos. La predicción direccional puede consistir en estimar el movimiento futuro del precio sobre el horizonte temporal de la siguiente observación, pero también puede intentar pronosticar la dirección en un horizonte de tiempo superior, sobre múltiples observaciones.

Los trabajos de investigación encontrados centraron su atención en predecir la dirección de la próxima observación futura, en un número fijo de observaciones o en un intervalo de tiempo constante, alimentando sus algoritmos de aprendizaje automático con características extraídas de datos de alta frecuencia sobre un número constante de observaciones o en un intervalo temporal fijo. Por lo tanto, no constan trabajos previos que tuviesen como objetivo el uso de técnicas de inteligencia artificial en datos financieros de alta frecuencia para predecir la dirección de la futura tendencia intradiaria extrayendo datos de subconjuntos formados por movimientos tendenciales previos. A continuación, se revisan múltiples trabajos que, de alguna forma, realizan predicción direccional con datos financieros de alta frecuencia y métodos de inteligencia artificial.

En Fletcher & Shawe-Taylor (2013), se propuso un método basado en SVM para clasificar la dirección del precio en tres clases y diferentes horizontes de predicción. El input consistió en características extraídas de las actualizaciones del libro de órdenes del EURUSD, con muestreo a frecuencia de 1 segundo. Desde entonces, otros artículos han tratado el uso de SVM para abordar el problema de la direccionalidad del precio, como en Kercheval & Zhang (2015), donde emplearon un SVM multiclase para pronosticar el precio medio y la dirección del *bid-ask spread*. Los datos consistieron en un día de negociación de 5 acciones cotizadas en

el NASDAQ. La entrada de los modelos estaba formada por múltiples características relacionadas con el precio y el volumen en diferentes niveles del libro de órdenes. Hicieron predicciones sobre un horizonte futuro de 5, 10, 15 y 20 eventos.

Otras publicaciones han planteado el empleo de SVM y de redes neuronales. En Tsantekidis et al. (2017a), se compararon métodos basados en CNN, *Multilayer Perceptron (MLP)* y SVM para realizar una clasificación multiclase de futuros movimientos del precio medio sobre tres horizontes de predicción diferentes: 10, 20 y 50. Realizaron el etiquetado comparando el promedio de los k precios medios anteriores con el promedio de los siguientes k precios medios, considerando un cierto umbral que determinaba cada una de las tres clases. Llevaron a cabo la experimentación con datos compuestos por 10 niveles del LOB de 5 acciones pertenecientes al mercado financiero finlandés, durante el período del 1 al 14 de junio de 2010. También en Nousi et al. (2019) se predijo la dirección del movimiento del precio medio con un modelo SVM multiclase y con redes neuronales. Se utilizaron horizontes de predicción iguales a 1, 5 y 10 precios medios futuros. En los dos últimos casos, predijeron el promedio del precio. Extrajeron características cada 10 eventos de órdenes límite y utilizaron datos de 5 acciones cotizadas del mercado financiero finlandés, relativos a 10 niveles del LOB y 10 días de negociación, a partir del 14 de junio de 2010. La entrada de los modelos de aprendizaje automático consistió en características extraídas del conjunto de datos brutos, además de otras obtenidas por modelos *autoencoders* y *bag-of-features*. En Tsantekidis et al. (2017b), se utilizaron redes LSTM para predecir la dirección de los futuros movimientos del precio medio. Los datos estaban compuestos por 10 niveles del LOB, correspondientes a 5 acciones per-

tenecientes al mercado finlandés, durante un período de negociación de 10 días. Realizaron el etiquetado en tres clases, comparando el promedio de precios medios anteriores con el siguiente promedio de precios medios, considerando un umbral específico. Realizaron una experimentación para tres horizontes de predicción diferentes: 10, 20 y 50. Finalmente, compararon los resultados con los obtenidos con un SVM lineal y con una red MLP, concluyendo que las redes LSTM se comportaron significativamente mejor para el problema planteado.

Diversas publicaciones han contemplado el uso de distintos tipos de redes neuronales para predecir la dirección de los movimientos del precio. En Dixon et al. (2017), se implementó una red neuronal profunda para clasificar la dirección futura de movimientos del precio durante un intervalo temporal futuro. Los datos correspondieron al precio medio en intervalos de 5 minutos, relativo a 43 futuros de materias primas y divisas, de marzo de 1991 a septiembre de 2014. La entrada consistió en características formadas por diferencias de precios y retardos de 1 a 100, medias móviles del precio con ventanas de tamaños de 5 a 100, correlaciones de pares entre retornos y retornos. En Doering et al. (2017), además de predecir la volatilidad, como se comenta en el apartado anterior, se eligió una CNN profunda para clasificar en tres clases la dirección de la tendencia del precio en un horizonte futuro de 20 eventos, con base en la estimación de la dirección de los retornos aritméticos. Los datos estaban formados por el libro de mensajes de *Barclays PLC*, activo que cotiza en la Bolsa de Valores de Londres. En total, fueron 217 días de negociación, entre junio de 2007 y junio de 2008. Los datos de entrada fueron los estados anteriores del libro de órdenes y el flujo de eventos. En Arévalo et al. (2018), se utilizó una red neuronal profunda con *wavelet* para predecir el siguiente

pseudo-log-return de un minuto, a partir del cual se obtuvo el precio promedio del minuto siguiente. Justificaron el uso de *wavelets* debido a que los datos de alta frecuencia presentan transacciones simultáneas, además de saltos de precio y baja varianza. Los *wavelets* tenían una longitud de ocho y comprimían las correspondientes transacciones *tick-by-tick* en un minuto determinado. La experimentación se realizó con datos *tick-by-tick* de 19 sociedades del índice *Dow Jones Industrial Average* seleccionadas al azar, durante el período de enero de 2015 a julio de 2017. La entrada de la red estaba formada por 27 variables: los *pseudo-log-returns* de un minuto y los vectores *wavelet* de datos *tick-by-tick* comprimidos en un minuto, ambos con retardo igual a 3. En Dixon et al. (2019), se empleó una red de aprendizaje profundo (*deep learning*) con múltiples capas para clasificar el movimiento del precio medio sobre un intervalo temporal futuro . Los datos correspondieron al libro de mensajes del futuro *E-mini S&P500* durante el mes de agosto de 2016. Se realizó un preprocesado con una red elástica y se extrajeron características de los estados del LOB relativos a 5 niveles. En Sirignano & Cont (2019), se alimentaron redes LSTM en secuencias de 100 y 5000 retardos con el precio histórico y el flujo de órdenes de múltiples valores del mercado bursátil estadounidense, con la finalidad de predecir si la siguiente dirección del precio era creciente o decreciente. Se construyeron diversos modelos utilizando datos de 1000 acciones del NASDAQ. En Ntakaris et al. (2019), utilizaron modelos MLP, CNN y LSTM para predecir la dirección del futuro movimiento del precio medio, además de estimar el número de eventos del libro de órdenes que se generan hasta que se produce el cambio de dirección. La entrada de las redes neuronales estaba compuesta por indicadores técnicos y cuantitativos, características sensibles e insensibles al tiempo, así como características extraídas automáticamente. Hicieron dos predicciones di-

ferentes. Por un lado, estimaron la dirección del precio, hacia arriba o hacia abajo, y cuándo ocurriría el próximo evento, mediante clasificación y regresión, respectivamente. Con este enfoque, extrajeron características con una ventana deslizante de tamaño igual a 10 eventos y avance de 1 evento. Por otro lado, hicieron una clasificación multiclase cada 10 eventos, para lo cual utilizaron el promedio de 10 eventos futuros, con extracción de características cada 10 observaciones sin superposición. El conjunto de datos correspondió a 10 días de negociación, 5 acciones del mercado finlandés y 2 acciones del mercado estadounidense, para 10 niveles de LOB y frecuencia de milisegundos. En Passalis et al. (2019), se combinó el método *bag-of-features* con redes neuronales profundas para resolver el problema de clasificación multiclase de la futura dirección del promedio del precio medio. Consideraron los 15 vectores de características más recientes para cada *timestep*, y utilizaron dos horizontes de predicción: 10 y 50 *timesteps*. Los datos estaban compuestos por 10 niveles del LOB de 5 acciones pertenecientes a la Bolsa de Helsinki, durante el período comprendido entre el 1 de junio de 2010 y el 14 de junio de 2010, lo que supone un total de 10 días de negociación. En Zhang et al. (2019), se diseñó un modelo de aprendizaje profundo basado en capas convolucionales y unidades LSTM para predecir futuros movimientos del precio sobre tres horizontes de predicción diferentes: 20, 50 y 100. Utilizaron dos conjuntos de datos distintos. Por un lado, 10 niveles del LOB relativos a 5 acciones de la Bolsa de Valores de Londres durante el período de 1 año. Por otro lado, hicieron uso del conjunto de datos públicos denominado FI-2010 (Ntakaris et al., 2018). Seleccionaron 40 características para cada evento: precio y volumen de cada nivel por cada uno de los dos lados del LOB de los 100 estados del libro de órdenes previos.

También se han utilizado otros métodos para resolver el mismo problema. En Felker et al. (2014), se presentó un método basado en la distancia euclídea de características ponderadas al centroide de un clúster de entrenamiento. Utilizaron múltiples indicadores técnicos para predecir un cambio en el precio, de 10 a 2000 milisegundos antes de que ocurriera. Sus predicciones se hicieron cuando la confianza de la clasificación era lo suficientemente alta, a partir del registro de cada nuevo evento de mercado. En Tran et al. (2017), se utilizó *multilinear discriminant analysis* y *weighted multichannel time-series regression* para predecir el movimiento del precio medio. El input estaba constituido por una representación tensorial de las series temporales. El etiquetado correspondió a tres clases del movimiento del precio medio de los siguientes 10 eventos de órdenes. Los datos estaban compuestos por el LOB de 5 acciones pertenecientes al mercado finlandés, relativo a 10 días de negociación.

4

Metodología

A lo largo de las páginas que integran este capítulo, se repasan los sucesivos pasos seguidos para obtener una solución viable al problema de alta frecuencia planteado, considerando los fundamentos teóricos recogidos en el estado del arte. Se realizó una investigación experimental que culmina con la discusión de los resultados obtenidos y las conclusiones derivadas de los mismos.

La solución del problema de investigación que se plantea tiene aplicación a problemas predictivos de alta frecuencia abordados con métodos de inteligencia artificial. Por ello, la solución obtenida se aplicó a los problemas de predicción de la volatilidad, la duración y la dirección asociadas a tendencias intradiarias futuras.

Pablo Mantilla Gómez
Programa de Doctorado en Ingeniería de Sistemas y Control

4.1 Planteamiento del problema

Las series temporales de alta frecuencia experimentan tendencias intradiarias de duración irregular, por lo que están compuestas de un número variable de observaciones. Se plantea el problema de extraer características de dichas tendencias y de los estados del libro de órdenes límite en cada instante de las mismas. La solución a este problema permite analizar múltiples variables en subconjuntos definidos por los intervalos temporales en los que se producen las tendencias citadas y también permite predecir el comportamiento de variables respuesta en intervalos temporales futuros, mediante la alimentación de modelos de inteligencia artificial con las características extraídas.

4.2 Propuesta de resolución

Se considera que este problema puede resolverse de forma satisfactoria por medio de una ingeniería de características compuesta de varias etapas, según se muestra en el esquema de la figura 4.1.

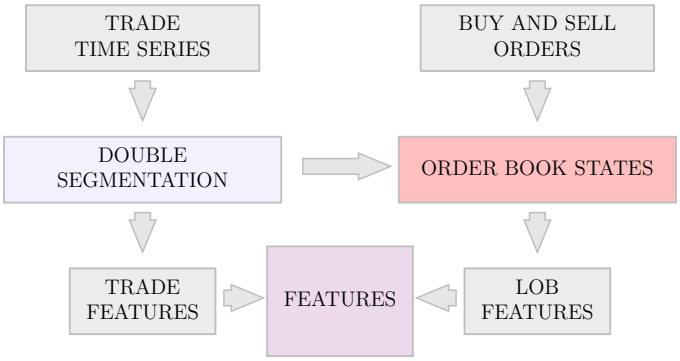


Figura 4.1: Esquema de ingeniería de características

El primer paso consiste en realizar una partición de la serie temporal de transacciones, obteniéndose segmentos con un número de observaciones

variable, determinados por la duración irregular de las tendencias intradiarias que se producen en los mismos. Los intervalos obtenidos forman subconjuntos de valores de las variables de los datos relativos a las operaciones negociadas. La siguiente etapa se basa en la sincronización de los estados del libro de órdenes con los instantes de la serie de cotizaciones, obteniéndose las variables asociadas a cada estado del libro de órdenes, las cuales forman el segundo subconjunto comprendido en el intervalo temporal citado. Sobre el conjunto total de variables de cada segmento, se realizan las correspondientes transformaciones para obtener las características que constituyen la entrada para la construcción de modelos de inteligencia artificial.

4.2.1 Doble segmentación

La técnica adecuada para obtener los intervalos de alta frecuencia mencionados es la segmentación de series temporales. En términos generales, se fragmenta la serie de cotizaciones para obtener los puntos de ruptura (*breakpoints*) en los cuales las tendencias intradiarias cambian su dirección. Los puntos de ruptura se corresponden con marcas temporales (*timestamps*) de las series, que proporcionan información del instante en el que se producen las observaciones, donde el intervalo temporal entre dos marcas temporales se denomina *timestep*. La idea es obtener intervalos temporales dinámicos, cuyos límites son los puntos de ruptura calculados. En dichos segmentos, se extraen características de las series temporales de cotizaciones y del libro de órdenes límite, las cuales están vinculadas a la dirección de cada tendencia intradiaria.

El objetivo es que estos intervalos contengan movimientos direccionales claramente definidos, por lo que dichos movimientos deben delimitarse

de la forma más precisa posible. La literatura científica recoge diferentes formas de abordar el problema. Basándonos en la literatura de referencia, la técnica que permite una solución óptima o exacta del problema de segmentación es la que se fundamenta en programación dinámica (Bai & Perron, 2003) (Zeileis et al., 2003), pero la principal desventaja de este tipo de segmentación es su elevado tiempo de ejecución.

Los datos que se pretenden fragmentar se registran por días, por lo que esta circunstancia permite segmentar cada día de forma independiente. No obstante, el número de observaciones diarias de las series de algunos activos financieros es tan elevado que, aunque se realice una segmentación diaria utilizando el método óptimo directamente sobre los datos originales, los tiempos de ejecución seguirían siendo tan elevados que no permitirían obtener los segmentos en un tiempo razonable.

Como solución, la propuesta de resolución del problema contempla la fragmentación inicial de los datos diarios, previamente a la segmentación que devuelve los intervalos de los que se extraen las características. La figura 4.2 muestra cuatro gráficos de la misma serie temporal de alta frecuencia con cuatro periodicidades (*timeframes*) diferentes.

En la figura, los datos originales están representados con precisión de milisegundos en el gráfico superior izquierdo, al que le siguen los otros tres gráficos, los cuales presentan datos agregados a segundos, minutos y cinco minutos, respectivamente. Las agregaciones se realizaron dividiendo el espacio temporal en intervalos constantes con cada una de las tres periodicidades mencionadas, para posteriormente extraer la última observación de cada intervalo, de tal forma que las observaciones de cada

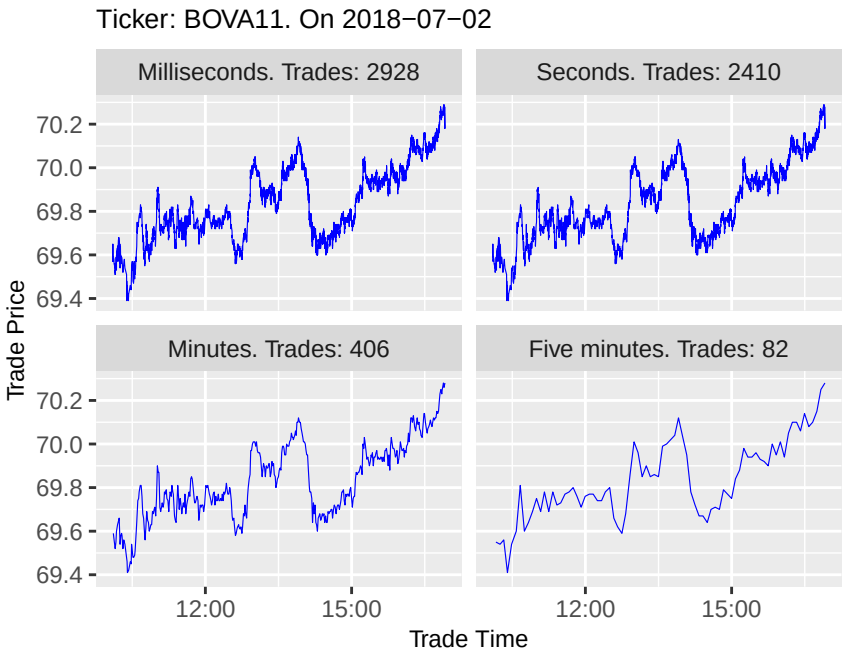


Figura 4.2: Agregación de períodos

una de las modalidades forman subconjuntos de los datos de partida. La serie original mantiene su forma a través de las cuatro representaciones y, aunque a medida que se reduce la frecuencia, las series pierden parte de la información inicial, los movimientos principales se conservan. Esta particularidad de conservar la forma se aprovechó para realizar una partición inicial de los datos.

Se aplicó el método de segmentación óptimo (Bai & Perron, 2003) (Zeileis et al., 2003) a datos agregados con frecuencia inferior a los originales, de acuerdo al procedimiento mostrado en el algoritmo 1, que recoge los pasos seguidos para ejecutar la solución propuesta.

Se desarrolló una doble segmentación con agregación de períodos, que se denominó *double segmentation with period aggregation (DSPA)*. En primer lugar, se obtiene una primera partición que no es arbitraria, ya que

Algoritmo 1: Double segmentation with period aggregation

Input : trade series
Output: segmentation matrix

```
1  h → minimum segment observations
2  function period_aggregation(trades):
3  |   return trade aggregation matrix
4  end
5
6  function break_points(trade aggregation, h):
7  |   return break timestamps → Segment Type I
8  end
9
10 Transfer break timestamps to input→ SegmentTypeImatrix
11 function break_points(SegmentTypeImatrix, h):
12 |   return break timestamps → Segment Type II
13 end
14
15 → Get matrix with Segment Type II
```

ésta respeta las tendencias principales de las series. Los puntos de ruptura obtenidos se trasladan a los datos originales, obteniéndose múltiples series concatenadas con precisión de milisegundos, las cuales poseen un número de observaciones relativamente reducido, en comparación con la serie original. Los *timestamps* de los extremos de estas series se corresponden con los puntos de ruptura obtenidos, donde cada una de estas series es un segmento, denominado *Segment Type I*. Sobre estos segmentos, se realizó una segunda partición, obteniéndose los segmentos finales, con denominación *Segment Type II*.

El método seleccionado para estimar los puntos de ruptura entre las distintas tendencias intradiarias se implementó en Zeileis et al. (2002). Se aplicó el mismo procedimiento para la estimación de los puntos de ruptura en las dos fases de la doble segmentación ejecutada. En Bai & Perron (2003) y en Zeileis et al. (2003) también se desarrollaron tests

para la detección de puntos de ruptura, referidos a los instantes de la serie temporal en los que se producen cambios estructurales. En casos específicos, sería recomendable realizar un test para determinar si en la serie temporal se produce uno o más puntos de ruptura, previamente a extraer los mismos. Sin embargo, en la experimentación ejecutada no se consideró, debido a que son series temporales financieras con miles de observaciones diarias que presentan estructura en media y varianza, por lo que se producen innumerables cambios y no se observó similitud con el caso previsto en las publicaciones citadas.

En Zeileis et al. (2002) y en Zeileis et al. (2003) se propuso un algoritmo de segmentación de series temporales basado en programación dinámica (Bai & Perron, 2003) y fundamentado en la detección de cambios estructurales en los denominados puntos de ruptura. El método considera la pérdida de estabilidad de los coeficientes de los modelos de regresión lineal que ajustan las observaciones de los segmentos temporales, determinando los puntos de ruptura en los cuales dichos coeficientes pasan de una relación estable a otra diferente. Los coeficientes son constantes en cada uno de los intervalos temporales estables. La estimación de dichos puntos se realiza minimizando la suma de los residuos al cuadrado (*residual sum of squares [RSS]*).

Sea una serie temporal con observaciones $i = 1, 2, ..., n$ y con (x_i, y_i) correspondientes al tiempo y al precio de cotización, respectivamente. Para un número de puntos de ruptura m, tendríamos $m + 1$ segmentos, donde los coeficientes de regresión serían constantes, y el modelo propuesto sería

$$
\begin{aligned}
y_i &= x_i^t \beta_j + u_i \\
i &= i_{j-1} + 1, ..., i_j \\
j &= 1, ..., m+1 \\
i_0 &= 0 \\
i_{m+1} &= n,
\end{aligned}
\tag{4.1}
$$

donde j es el índice del segmento y $\psi_{m,n} = i_1, ..., i_m$ es el conjunto de puntos de ruptura. Por tanto, tenemos segmentos con observaciones

$$
\begin{aligned}
1, 2, ..., i_1 \\
i_1 + 1, i_1 + 2, ..., i_2 \\
\vdots \\
i_m + 1, i_m + 2, ..., n
\end{aligned}
\tag{4.2}
$$

Entonces, cada uno de los segmentos tiene un número de observaciones

$$
e_j = i_j - i_{j-1} \tag{4.3}
$$

Los puntos de ruptura son desconocidos, por lo que tienen que estimarse a partir de los datos. El procedimiento para obtener los instantes

temporales en los que se producen los cambios citados se expone a continuación.

Sea una *m-partición* $i_1, ..., i_m$ con $m + 1$ segmentos o subconjuntos resultantes de la partición. Los coeficientes β_j de las regresiones se estiman por mínimos cuadrados, y la suma de residuos al cuadrado es

$$RSS(i_1, ..., i_m) = \sum_{j=1}^{m+1} rss(i_{j-1} + 1, i_j) \, , \tag{4.4}$$

donde la expresión dentro del sumatorio es el mínimo de la suma de residuos al cuadrado en los j segmentos respectivos. Se trata de obtener los puntos de ruptura $\hat{i_1}, ..., \hat{i_m}$ que minimizan la función objetivo

$$(\hat{i}_1, ..., \hat{i}_m) = \underset{(i_1,...,i_m)}{argmin} RSS(i_1, ..., i_m) \, , \tag{4.5}$$

sobre todas las particiones $(i_1, ..., i_m)$.

Para cualquier valor de m puntos de ruptura, la suma de los residuos al cuadrado de todos los segmentos de cualquier *m-partición* $i_1, ..., i_m$ es una combinación lineal de las sumas de los residuos al cuadrado de los segmentos de todas las particiones. La estimación de los m puntos de ruptura se obtiene con programación dinámica, comparando las posibles combinaciones lineales y seleccionando aquella *m-partición* con menor valor de la suma de los residuos al cuadrado de la totalidad de segmentos.

El procedimiento de programación dinámica comienza evaluando las particiones óptimas de un único punto de ruptura. El siguiente paso consiste en obtener las particiones óptimas de dos puntos de ruptura, analizando qué partición óptima de un punto de ruptura se puede insertar para obtener la suma de los residuos al cuadrado mínima, y así sucesivamente. La etapa final consiste en determinar cuál de las *(m-1)-particiones* tiene la menor suma de residuos al cuadrado al combinarse con un segmento adicional.

El *Bayesian information criterion (BIC)* y el *Akaike information criterion (AIC)* se usan para la selección de modelos entre un conjunto de éstos y evitar que se produzca sobreoptimización. Estos criterios introducen un término que penaliza el número de parámetros en el modelo, siendo dicho término mayor en el BIC que en el AIC. En este caso, estos criterios se emplean para seleccionar el número de puntos de ruptura m. En Bai & Perron (2003), se considera que el AIC suele sobreestimar el número de puntos de ruptura m, por lo que, en principio, el BIC sería una opción más apropiada para realizar la selección.

En el método de segmentación óptimo, la posición óptima de las rupturas de los intervalos temporales se determina minimizando la suma de los residuos al cuadrado, y el número óptimo de rupturas se obtiene minimizando el BIC. La figura 4.3 representa el punto óptimo de la segmentación del día 2 de julio de 2018 para el valor cuyo *ticker*, como se denomina el símbolo con el que se identifica el activo financiero, es ANIM3. La segmentación directa de las 503 observaciones de dicho día determina que el menor BIC es de -2112.17, lo que corresponde con un número óptimo de puntos de ruptura de 27 y un número de segmentos de 28.

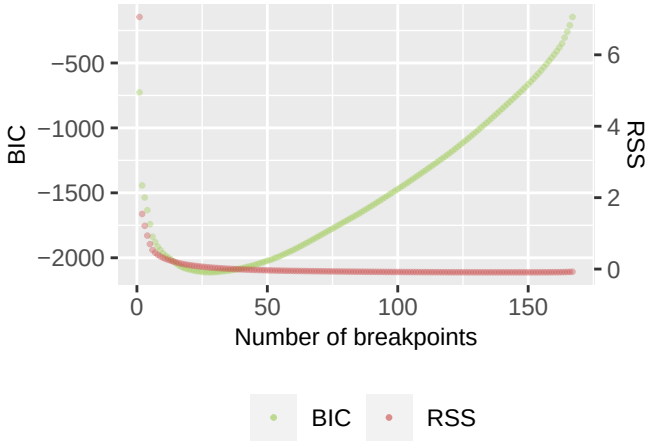


Figura 4.3: BIC y RSS de la segmentación

Un ejemplo del resultado final que se obtiene con la segmentación es el que se muestra en la figura 4.4, donde se representa en color negro un tramo de una serie temporal de alta frecuencia, formada por 687 observaciones. Los valores ajustados de las rectas de regresión se presentan en color verde, cuyos valores inicial y final de cada segmento corresponden a los *timestamps* que delimitan los intervalos temporales en los que se produce cada tendencia intradiaria.

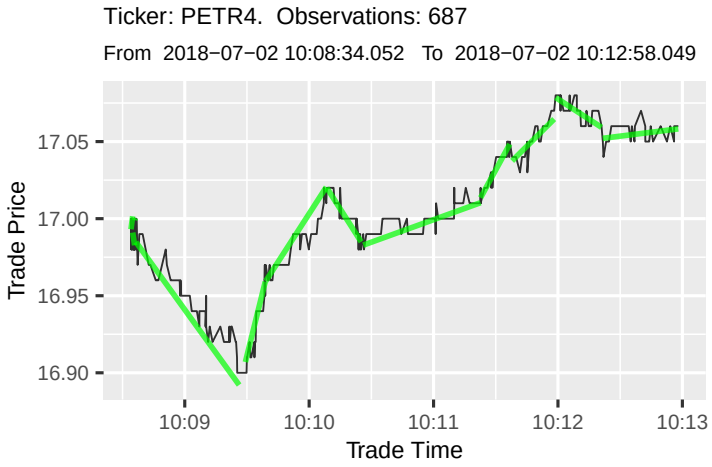


Figura 4.4: Ejemplo de segmentación

4.2.2 Procedimiento de evaluación de la segmentación

En Lemire (2007), se distinguieron las características que debe tener un buen algoritmo de segmentación de series temporales. Entre éstas, se encuentra la velocidad de ejecución y la precisión, esta última referida a la bondad del ajuste. Como se mencionó anteriormente, el método exacto obtiene la posición de los puntos de ruptura que minimizan el RSS, pero incorpora como restricción que el número de puntos de ruptura debe minimizar el BIC. Por esta razón, se considera que la evaluación de métodos de segmentación también debería considerar el BIC, además de la precisión y el tiempo de ejecución.

El AIC equilibra la bondad del ajuste del modelo y su complejidad, penalizando la primera con una función creciente del número de parámetros estimados $2k$, y se calcula con la siguiente expresión (Akaike, 1974).

$$AIC = -2log(\hat{L}) + 2k\,, \tag{4.6}$$

donde k representa el número de parámetros del modelo y $log(\hat{L})$ es el logaritmo del valor máximo de la función de verosimilitud del modelo.

El BIC, empleado con la misma finalidad que el AIC, tiene la misma expresión, pero el parámetro de penalización es $log(n)$ (Schwarz, 1978).

$$BIC = -2log(\hat{L}) + klog(n)\,, \tag{4.7}$$

donde n es el número de observaciones de la serie. Siendo m el número de puntos de ruptura de la segmentación, k tiene la siguiente expresión (Zeileis et al., 2002).

$$k = 2(m + 1) + m + 1 \tag{4.8}$$

El número de parámetros k o grados de libertad está formado por dos coeficientes de cada una de las rectas que ajusta los valores observados de cada intervalo, multiplicados por el número de segmentos $(m + 1)$, más el número de puntos de ruptura m, más uno por la varianza de los residuos. El logaritmo del valor máximo de la función de verosimilitud para el modelo se obtiene con la siguiente expresión (Zeileis et al., 2002).

$$log(\hat{L}) = -\frac{1}{2}n[log(RSS) + 1 - log(n) + 2log(2\pi)] \tag{4.9}$$

Finalmente, la suma de los residuos al cuadrado está referida a las rectas de ajuste de cada uno de los intervalos.

$$RSS = \sum_{i=1}^{n} \epsilon_i^2 \tag{4.10}$$

Las anteriores expresiones matemáticas permiten calcular el RSS y el BIC. Por otra parte, se mide el tiempo de computación empleado para ejecutar cada trabajo de segmentación. Estas métricas se pueden obtener para cada segmentación diaria ejecutada, pero para evaluar el compor-

tamiento de cada método en un conjunto de segmentaciones ejecutadas deben realizarse tests estadísticos, y dado que se pretenden comparar un número de k algoritmos sobre un número N de conjuntos de datos, se debe realizar una comparación estadística múltiple. En Demšar (2006), se revisan múltiples tipos de tests y procedimientos para realizar este tipo de comparación.

Se propone el test no paramétrico de *Friedman* (Friedman, 1937) para verificar si varios métodos de segmentación tienen diferencias en rendimiento, en términos de tiempo de ejecución, RSS y BIC. Este test clasifica los métodos analizados sobre cada uno de los conjuntos de datos utilizados, de forma que el método con mejor rendimiento en un conjunto de datos ocuparía la posición número 1, el segundo mejor la posición número 2 y así sucesivamente con todos los conjuntos de datos.

Sea r_i^j la posición j-ésima de k algoritmos en el conjunto i-ésimo de N conjuntos de datos. El promedio de posiciones de algoritmos tiene la siguiente expresión.

$$R_j = \frac{1}{N} \sum_i r_i^j \tag{4.11}$$

El test de Friedman compara dicho promedio, y su estadístico

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right] \tag{4.12}$$

se distribuye de acuerdo a χ_F^2 con $k-1$ grados de libertad, si N y k son suficientemente grandes ($N > 10$ and $k > 5$), y bajo la hipótesis nula H_o de que todos los métodos son equivalentes. En dicho caso, sus posiciones R_j serían iguales. Si el número de algoritmos o conjuntos de datos fuesen menores que los indicados, en Zar (1999) y en Sheskin (2000) se computaron valores críticos exactos.

Siguiendo lo contemplado en Demšar (2006), se propone analizar qué métodos son diferentes con un *post hoc test*. Se utilizó el test no paramétrico de Nemenyi (Nemenyi, 1963) para realizar una comparación de todos los métodos con cada uno de los otros. El test se basa en el valor absoluto de la diferencia del promedio de posiciones de los algoritmos. La hipótesis nula de que los métodos tienen el mismo rendimiento se rechaza, para un nivel de significancia α, si la diferencia entre el promedio de posiciones de dos métodos es mayor que la diferencia crítica, representada por la siguiente expresión.

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{4.13}$$

Los valores críticos q_α se basan en el estadístico de rango estudentizado dividido por $\sqrt{2}$, recogidos en Demšar (2006) para un número de algoritmos de 2 a 10 y niveles de significancia de 0.05 y 0.10.

4.2.3 Extracción de características

Partimos de la serie temporal de transacciones segmentada, por lo que tenemos el conjunto de $m+1$ segmentos, fragmentos o intervalos temporales

de la serie

$$S_1, S_2, ..., S_{m+1}, \tag{4.14}$$

donde cada uno de los intervalos tiene un número de observaciones variable e_j. Se extrae un conjunto de p características de la serie de operaciones negociadas en cada uno de los segmentos S_j

$$X_{j_1}, X_{j_2}, ..., X_{j_p} \tag{4.15}$$

Las características se obtienen de los precios y_i, tiempos x_i, transacciones T_i o volúmenes v_i contenidos en el intervalo S_j, tal que

$$Xj_t = f_t\left(\{y_i\}_{i=i_{j-1}+1}^{i_j},\ \{x_i\}_{i=i_{j-1}+1}^{i_j},\ \{T_i\}_{i=i_{j-1}+1}^{i_j},\ \{v_i\}_{i=i_{j-1}+1}^{i_j}\right), \tag{4.16}$$

donde cada característica X_{j_t} procedente de las series de operaciones negociadas en cada intervalo S_j y $t = 1, 2, ..., p$ es una función f_t de cualquier combinación de las variables: precios, tiempos, transacciones o volúmenes.

Otras características se extraen de los estados del LOB en los instantes determinados por los precios de transacción. Sea el conjunto formado por las q características del LOB

$$Z_{j_1}, Z_{j_2}, ..., Z_{j_q} \tag{4.17}$$

En la figura 4.5, se observa el estado del LOB en un instante x_i, con una profundidad de diez niveles por cada lado del mercado. Las mejores órdenes de compra y de venta, desde el punto de vista de ser fácilmente cruzadas, son aquellas con los precios más alto y más bajo, respectivamente. Dichas órdenes están situadas en la mitad del gráfico, separadas por lo que se denomina *spread*, que es la diferencia de precio entre dichas órdenes. El nivel 1 del LOB corresponde a estas órdenes, y los siguientes niveles están formados por los bloques que siguen a las mejores órdenes de compra y de venta, formando lo que se denomina profundidad del LOB.

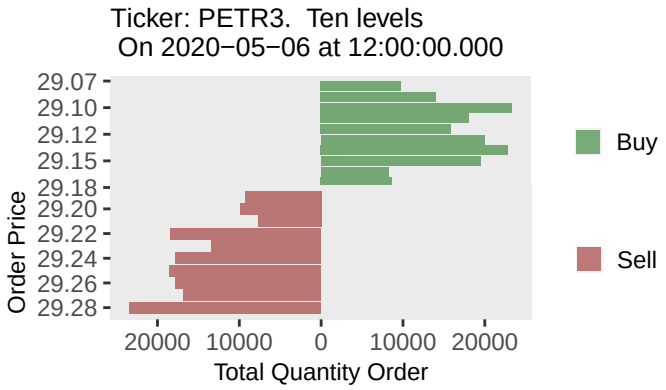


Figura 4.5: Estado del LOB en un instante

En términos matemáticos, el estado del LOB en cada instante x_i de la serie de operaciones negociadas es una matriz con los precios y los volúmenes de todos los niveles formados por las órdenes de compra y de venta, denominados profundidades de compra d_B y de venta d_S.

El proceso de extraer características del LOB se realiza por segmentos y empieza a partir del estado del LOB en cada instante de cada uno de los segmentos. Este proceso distingue entre las variables calculadas en los instantes de cada segmento y las características finales extraídas de cada segmento, que en el caso de la experimentación realizada corresponde a los valores promedio de las variables computadas en los instantes de cada segmento. Para realizar este proceso, partimos de los datos de las matrices que corresponden a los estados del LOB en los instantes de cada segmento y obtenemos el vector de variables en cada instante. Estos vectores forman la matriz de variables computadas en todos los instantes de un segmento. El resultado final es la reducción de esta matriz a un vector de características para cada segmento.

De acuerdo a la descripción anterior, el proceso de extracción de características de cada estado del LOB tiene tres etapas, como se muestra en la figura 4.6. Sea un intervalo S_j formado por las observaciones i de la serie de cotizaciones, y sean los estados i del LOB en cada instante x_i de cada una de las observaciones del segmento. Cada estado del LOB es una matriz con precios y volúmenes de órdenes de compra y de venta $(P_{i_{B_l}}\ V_{i_{B_l}}\ P_{i_{S_l}}\ V_{i_{S_l}})$ de cada uno de los niveles l existentes en un instante x_i. De los valores de esas cuatro variables, se obtienen otras variables que resumen con un único valor una característica de dicho instante, obteniéndose los vectores $u_{i_{j-1}+1}, u_{i_{j-1}+2}, ..., u_{i_j}$ para cada uno de los instantes x_i. El conjunto de vectores resultante forma la matriz M_j, que contiene las variables referidas a los instantes x_i del intervalo S_j. Finalmente, a través de medias aritméticas, se obtiene el conjunto de valores $Z_{j_1}, Z_{j_2}, ..., Z_{j_q}$ de características de dicho intervalo.

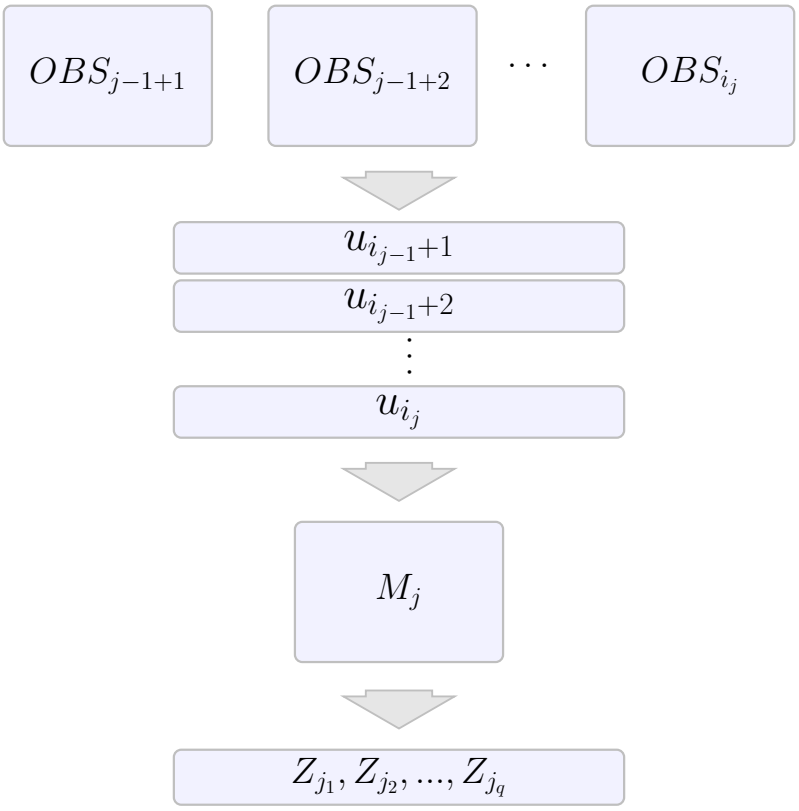


Figura 4.6: Proceso de extracción de características del LOB

En resumen, las características extraídas del LOB proceden de las variables que reducen los precios y volúmenes de las órdenes de compra y de venta, las cuales corresponden a todos los niveles de los instantes en que se producen las observaciones de cada segmento. En último término, las características dependen de los precios y volúmenes de las órdenes de compra y de venta, pero se establecen diferentes tipos de características del LOB, dependiendo de los datos de entrada. En términos generales, las características se obtienen de un subconjunto con un número de niveles de compra $1, 2, ..., d_B$ y de venta $1, 2, ..., d_S$ dentro del intervalo S_j. De acuerdo con este criterio, para $o = 1, 2, ..., q$ se define la función g_o que obtiene cada característica Z_{j_o} en un segmento S_j con observaciones i,

$$Z_{j_o} = g_o(\{\{P_{i_{B_l}}, V_{i_{B_l}}\}_{l=1}^{d_B},\ \{P_{i_{S_l}}, V_{i_{S_l}}\}_{l=1}^{d_S}\}_{i=i_{j-1}+1}^{i_j}) \,, \tag{4.18}$$

donde $P_{i_{B_l}}$ y $V_{i_{B_l}}$ son los precios y volúmenes de compra del nivel l del LOB para el instante x_i del intervalo S_j. De igual forma, $P_{i_{S_l}}$ y $V_{i_{S_l}}$ están referidos a las órdenes de venta.

Por tanto, en cada intervalo temporal se extraen dos conjuntos de características diferentes, de acuerdo al esquema representado en la figura 4.7.

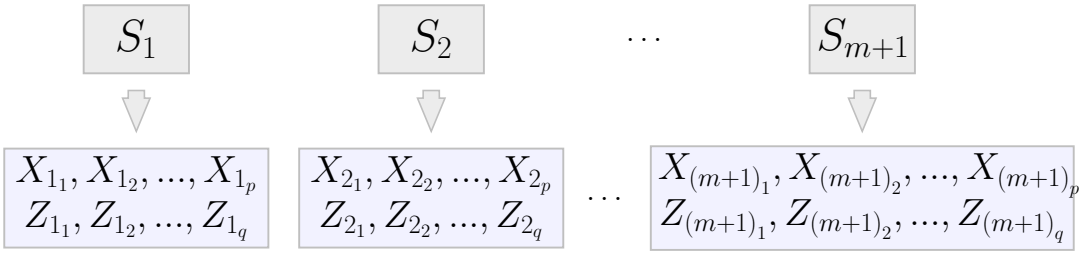


Figura 4.7: Características de los segmentos

En este capítulo, se ha descrito la metodología desarrollada para extraer características de las series de operaciones negociadas y de las órdenes de compra y venta vinculadas a tendencias del precio de activos financieros en alta frecuencia y, por tanto, alcanzar el objetivo general establecido. La utilidad de esta metodología se muestra en el siguiente capítulo, en el cual se describe cómo utilizar las características extraídas en modelos predictivos de clasificación para datos temporales.

5

Aplicación

Como se destacó en Kercheval & Zhang (2015), los métodos estadísticos a menudo imponen suposiciones matemáticas sobre los modelos, lo cual requeriría computación intensiva que complicaría su aplicación a problemas predictivos como los que se plantean en la aplicación de la metodología desarrollada, en los cuales se emplean grandes volúmenes de datos financieros de alta frecuencia. Otros autores también están de acuerdo en esta cuestión, como Nousi et al. (2019), quienes remarcaron que el diseño de modelos basados en métodos de aprendizaje automático no requiere hacer suposiciones sobre la distribución de los datos. Estos autores destacaron que las distribuciones de los datos financieros de alta frecuencia cambian rápidamente. Igualmente, durante el desarrollo de la metodología expuesta, también se ha estado de acuerdo con dichas afirmaciones, y por ello se considera que la aplicación de la misma debe enfocarse a la alimentación

de modelos basados en inteligencia artificial.

Por lo anteriormente expuesto, y con la finalidad de mostrar la aplicación de la metodología desarrollada, se construyeron modelos basados en un método de inteligencia artificial para problemas de clasificación, aunque la ingeniería de características diseñada podría emplearse con innumerables métodos de aprendizaje automático o aprendizaje profundo, como pueden ser los métodos: RF, SVM, KNN, MLP, CNN o LSTM.

El input formado por las características extraídas necesita una preparación previa para poder alimentar estos modelos, por lo que una vez obtenidas éstas, se pasa a las etapas de etiquetado (*labeling*) e incrustación, quedando el input preparado para utilizar en la modelización.

5.1 Etiquetado

Una vez que tenemos las series de operaciones negociadas segmentadas y se han extraído las características respectivas, los segmentos se clasifican de acuerdo con la variable respuesta correspondiente, en función de los cuantiles indicados en la figura 5.1, y según el siguiente procedimiento.

Para determinar los grupos en los que se van a clasificar los segmentos, se realiza un proceso de etiquetado de los mismos. Dicho etiquetado depende del valor de las variables respuesta, y está formado por diferentes etiquetas, cada una de las cuales está referida a una clase de dichas variables. Las clases están determinadas por las regiones de los valores que puede tomar la variable. Los umbrales o límites de dichas regiones se fijan de antemano sobre los valores de la variable en el subconjunto

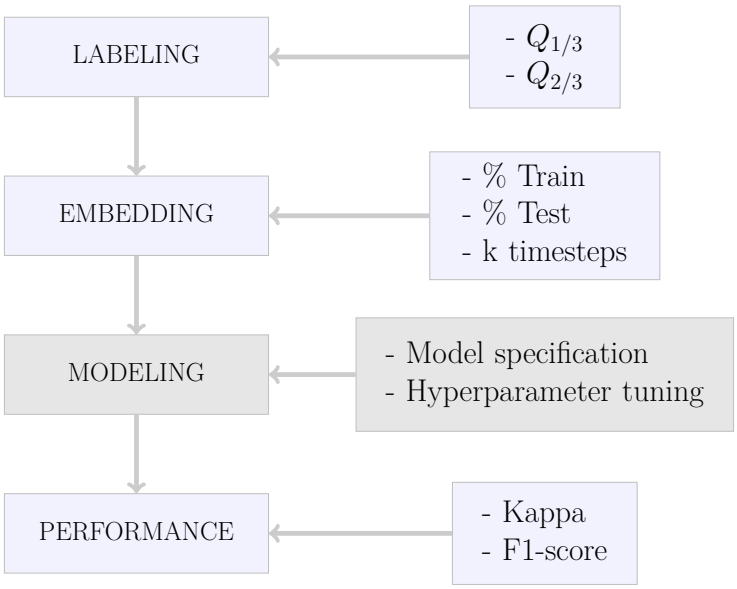


Figura 5.1: Esquema de aprendizaje automático

de entrenamiento, que representa un porcentaje mayor de los datos que el subconjunto de validación. El criterio se basó en la obtención de tres regiones equilibradas con los datos del subconjunto de entrenamiento, por lo que los límites están definidos por los cuantiles con probabilidades de 1/3 y 2/3, respecto a los valores de la variable respuesta.

5.2 Incrustación

Tenemos vectores de características que proceden de la serie de operaciones negociadas y del LOB, extraídas en cada uno de los intervalos temporales obtenidos mediante la segmentación de dicha serie. A partir de estos vectores, se construye el conjunto de casos para realizar la incrustación. Los casos están formados por los regresores y la variable respuesta respectiva. Tratándose de series temporales, es habitual incluir variables con retardos y que el avance sobre la serie de segmentos tenga un paso específico. Se trata de una ventana deslizante de tamaño igual al número

de retardos y avance igual al paso considerado. Se escogió un avance igual a 1 y un número de retardos igual a k, por lo que la estructura de los casos es la recogida en la figura 5.2.

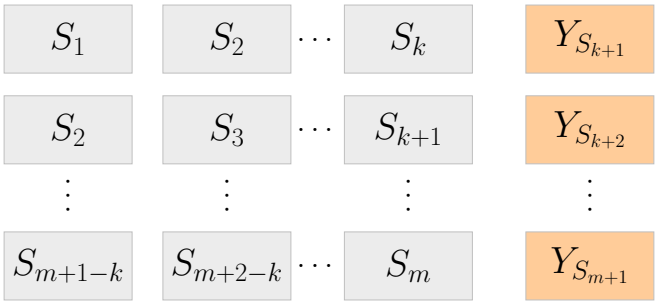


Figura 5.2: Esquema de incrustación

El primer caso es el vector formado por las características de los k primeros segmentos, las cuales representan la parte explicativa. La variable respuesta corresponde a la etiqueta que posee esta variable en el intervalo temporal $k+1$. Se avanza un paso en la serie de segmentos y se construye el segundo caso, comenzando con las características del segundo segmento y con variable respuesta igual a la etiqueta del segmento $k+2$, y así sucesivamente.

5.3 Modelización

El objetivo del Capítulo 5 es mostrar cómo se aplicaría la ingeniería de características desarrollada. Entre los innumerables métodos de aprendizaje automático que se podrían seleccionar para mostrar cómo funciona la metodología, se ha optado por el algoritmo XGBoost (Chen & Guestrin, 2016), debido a su velocidad y eficiencia y a los buenos resultados que ha obtenido en la competición de predicción M4, como se describe en Bojer & Meldgaard (2021). Sin embargo, como se indicó anteriormente, la metodología desarrollada podría utilizarse con cualquier otro método

de aprendizaje automático.

Esta sección revisa brevemente los fundamentos teóricos relativos al método XGBoost, el cual se basa en el método *gradient boosting* (Friedman, 2001) y constituye una implementación de los *gradient boosted trees*.

El objetivo es predecir la variable respuesta $\hat{Y}_i$, que en modelos de clasificación representa las etiquetas de cada una de las clases. Las variables explicativas X_i constituyen las b características extraídas de los datos, cada una de las cuales tiene un peso en el modelo, definido por los parámetros θ. El entrenamiento consiste en que el modelo aprenda y obtenga los parámetros θ que proporcionen el valor de variable respuesta más próximo al dado. La diferencia entre el valor estimado de la variable dependiente y el observado constituye el residuo.

Si tenemos un número de a casos y de b características, con $X_i \in \mathbb{R}^b$ e $Y_i \in \mathbb{R}$, la respuesta se predice por medio de un número K de funciones aditivas para un conjunto de árboles.

$$\hat{Y}_i = \phi(X_i) = \sum_{k=1}^{K} f_k(X_i)\,; \quad f_k \in \mathcal{F}\,, \tag{5.1}$$

donde $\mathcal{F}$ es el espacio de árboles, T el número de hojas de cada árbol y q la estructura de cada uno de éstos. Cada una de las funciones f_k se refiere a una estructura de árbol q y pesos de hoja w.

Se define la función que mide el error como la función de pérdida o de coste. Se trata de minimizar la función objetivo del método, que corres-

ponde a la suma de la función de pérdida y la función de regularización, la cual mide la complejidad del modelo. Se introduce esta última función con la finalidad de penalizar la sobreoptimización. Si el término regularización fuese cero, tendríamos el clásico *gradient tree boosting*.

$$\begin{aligned} \mathcal{L}(\phi) &= \sum_{i=1}^{a} l(\hat{Y}_i, Y_i) + \sum_{k=1}^{K} \Omega(f_k) \\ \Omega(f_k) &= \gamma T + \frac{1}{2}\|w\|^2, \end{aligned} \tag{5.2}$$

donde l es una función que mide la diferencia entre la predicción $\hat{Y}_i$ y el objetivo Y_i, y Ω es el término que penaliza la complejidad del modelo.

La optimización se realiza a partir de un entrenamiento de forma aditiva. Los *gradient boosted trees* combinan el *boosting* y el *gradient descent*. En el primero, los árboles de decisión se construyen de forma secuencial, de forma que sobre los residuos de cada árbol se construye el siguiente, reduciéndose el error en cada uno de los pasos. Finalmente, la función objetivo es la suma de las obtenidas en cada uno de los árboles.

El *gradient descent* es un método de optimización basado en la generalización vectorial de la derivada, cuyo objetivo es optimizar el mínimo total. Se trata de obtener los mejores parámetros θ que consigan que el residuo del ajuste sea el mínimo posible.

La importancia de las variables en el modelo se obtiene a partir de la

contribución relativa de cada una de las características al modelo XGBoost, basándose en la participación de cada característica en cada una de las divisiones del árbol, que se pondera sobre un valor total igual a 1. Si una característica X_t tiene una contribución superior a la que poseen otras características, entonces la característica X_t es más relevante para realizar predicciones. La suma total de las puntuaciones asociadas a cada una de las características es igual a 1.

5.4 Métricas de rendimiento

Como aplicación de la ingeniería de características desarrollada, se construyeron modelos de aprendizaje automático de clasificación multiclase para predecir las etiquetas de los futuros intervalos temporales. Con las etiquetas obtenidas, y aquellas correspondientes a los datos del conjunto de validación, se calcularon las matrices de confusión, comparando las clases observadas con las predicciones. La diagonal de la matriz de confusión recoge los casos con las predicciones correctas de cada clase, mientras que el resto de elementos de la matriz corresponde a los errores de los casos de cada clase. A partir de las matrices, se pueden calcular múltiples métricas para medir el rendimiento (*performance*) del modelo, entre las que se encuentran las métricas *kappa* y *F1-score*. La métrica *kappa* considera la precisión que podría obtenerse por casualidad, penalizando dicho aspecto en la precisión del modelo (Kuhn et al., 2013). Las métricas de rendimiento citadas se calculan con el procedimiento que se expone a continuación.

Sea A la matriz de confusión para un problema de clasificación de tres clases.

		Observados		
		High	Medium	Low
Predicciones	High	a_{11}	a_{12}	a_{13}
	Medium	a_{21}	a_{22}	a_{23}
	Low	a_{31}	a_{32}	a_{33}

Sea S_{ij} la suma de todos los elementos a_{ij} de A, $tr(A)$ la traza de A y sean O y E la precisión observada y la esperada, respectivamente.

$$S_{ij} = \sum_{i,j=1}^{3} a_{ij}\,; \quad tr(A) = \sum_{i=1}^{3} a_{ii} \tag{5.3}$$

La precisión observada O se obtiene a partir de la siguiente expresión.

$$O = \frac{tr(A)}{S_{ij}} \tag{5.4}$$

Utilizando la suma total y las sumas de los elementos de cada columna y cada fila, se obtiene la precisión esperada E.

$$E = \frac{S_{i1}}{S_{ij}}.\frac{S_{1j}}{S_{ij}} + \frac{S_{i2}}{S_{ij}}.\frac{S_{2j}}{S_{ij}} + \frac{S_{i3}}{S_{ij}}.\frac{S_{3j}}{S_{ij}} \tag{5.5}$$

A partir de la precisión observada y la esperada, se obtiene la *kappa de Cohen* (Cohen, 1960).

$$kappa = \frac{O - E}{1 - E}, \tag{5.6}$$

que puede tomar valores dentro del intervalo $[-1, 1]$. Un valor de cero significa que no hay concordancia entre las clases observadas y las predicciones. Un valor de uno representa la predicción perfecta, mientras que valores negativos indican que la predicción está en dirección contraria a la verdad.

Con la finalidad de disponer de una única medida por cada una de las clases, calculamos la $F_1 - score$ de la forma siguiente. Sean los verdaderos positivos (TP), falsos positivos (FP), verdaderos negativos (TN) y falsos negativos (FN) para la primera clase, donde los elementos que no pertenecen a la columna y fila uno son TN. El procedimiento es similar para las otras dos clases, salvo que los TP están en la diagonal de la matriz para cada una de las clases, por lo que los TN de la segunda clase corresponderían a los elementos no incluidos en la columna y fila 2, y así sucesivamente.

		Observados		
		High	Medium	Low
Predicciones	High	TP	FP_1	FP_2
	Medium	FN_1	TN_1	TN_2
	Low	FN_2	TN_3	TN_4

$$TP = a_{11} \tag{5.7}$$

$$FP = a_{12} + a_{13} \tag{5.8}$$

$$TN = a_{22} + a_{23} + a_{32} + a_{33} \tag{5.9}$$

$$FN = a_{21} + a_{31} \tag{5.10}$$

En primer lugar, obtenemos la métrica *precision*, que mide la proporción de predicciones correctas sobre el total de predicciones positivas,

$$Precision = \frac{TP}{TP + FP} \tag{5.11}$$

En segundo lugar, calculamos la métrica *recall*, la cual representa la proporción de positivos observados con una predicción correcta,

$$Recall = \frac{TP}{TP + FN} \tag{5.12}$$

Finalmente, obtenemos la $F_1 - score$, como la media armónica de las métricas *precision* y *recall*,

$$F1 = \frac{2PR}{P + R}\,, \tag{5.13}$$

donde P y R corresponden a las métricas de *precision* y *recall*, respecti-

vamente.

El mismo procedimiento se sigue para obtener la *F1-score* de las otras dos clases, con la particularidad de que los TP, TN, FP y FN se obtienen según las posiciones de la matriz respectivas para cada clase, de acuerdo a lo indicado anteriormente.

6

Experimentación

Se realizaron dos tipos de experimentación diferentes. La primera corresponde a la ejecución de los trabajos de segmentación y a la evaluación de ésta. Esta es la experimentación final más extensa, con un total de 4944 trabajos de computación enviados al supercomputador para su ejecución, a los que hay que añadir la infinidad de ejecuciones de pruebas. La segunda experimentación comprende los trabajos de entrenamiento de 20 modelos basados en inteligencia artificial, sobre los que se realizaron innumerables ejecuciones, probando con múltiples métodos de aprendizaje automático, arquitecturas e hiperparámetros. La experimentación final está compuesta por la ejecución de 20 trabajos.

Pablo Mantilla Gómez
Programa de Doctorado en Ingeniería de Sistemas y Control

6.1 Datos financieros de alta frecuencia

En la experimentación realizada, se emplearon datos de 26 activos de la Bolsa de Valores de Brasil (B3), identificados por sus *tickers*. Los trabajos de segmentación se realizaron con 6 *tickers*, los cuales corresponden a activos que se encuentran entre las últimas posiciones de los 150 valores más negociados del mercado citado, debido a que el criterio de selección se basó en la obtención de valores con un número de observaciones que permitiese ejecutar los trabajos de segmentación con el método óptimo en un espacio temporal máximo de varios días, con la finalidad de poder comparar dicha segmentación con las tres modalidades propuestas en un plazo razonable. Estos 6 valores se emplearon para evaluar 4 métodos de segmentación, según se expone más adelante. La experimentación con métodos basados en inteligencia artificial se ejecutó sobre los datos de 20 *tickers* entre los activos más negociados del mercado.

Los datos y la descripción de cada tipo de archivo se obtuvieron del servidor ftp://ftp.bmf.com.br/MarketData/ del mercado bursátil citado en mayo de 2019. Estos datos estaban almacenados en dos tipos de archivos de texto comprimidos. Uno de éstos está referido a las operaciones negociadas, mientras que el otro tipo corresponde a archivos de órdenes de compra y de venta. El período utilizado en la experimentación realizada empieza el 2 de julio de 2018 y finaliza el 6 de mayo de 2019, lo que supone un total de 206 días de negociación. Las operaciones negociadas y las órdenes de compra y de venta se registraron con precisión de milisegundos, dependiendo del momento en el que se originaron, por lo que los intervalos temporales entre una observación y la siguiente no tienen la misma duración, se trata de datos irregularmente espaciados. Las características

de este tipo de datos se tratan en profundidad en Dacorogna et al. (2001) y en Russell & Engle (2010). Únicamente se utilizaron datos correspondientes a las horas de negociación: de 10:00 a 16:55 horas. En Perlin & Ramos (2016), se recogen los diferentes horarios del mercado financiero brasileño y una revisión de trabajos realizados sobre este mercado.

Los datos de transacciones contienen múltiples variables. La especificación de éstas, así como las vinculadas a los datos de órdenes, se encontraban en los archivos con extensión .txt del servidor citado: NEG_LAYOUT, OFER_CPA_LAYOUT y OFER_VDA_LAYOUT. Adicionalmente, se obtuvo más información en B3 (2018). La tabla 6.1 recoge la denominación de las variables utilizadas en la investigación doctoral.

Tabla 6.1: Variables de los datos de operaciones negociadas

Session Date	Instrument Symbol
Trade Price	Traded Quantity
Trade Time	

La definición de estas variables se recoge a continuación:

- *Session Date*: fecha del día de negociación bursátil.
- *Instrument Symbol*: *ticker* del activo negociado.
- *Trade Price*: precio de negociación.
- *Traded Quantity*: volumen o cantidad negociada.
- *Trade Time*: instante temporal en que se ejecuta cada operación negociada.

Los datos financieros de alta frecuencia contienen múltiples errores (Dacorogna et al., 2001) (Falkenberry, 2002) (Hautsch, 2012). En Barndorff-Nielsen et al. (2009), se describe un procedimiento para la limpieza de

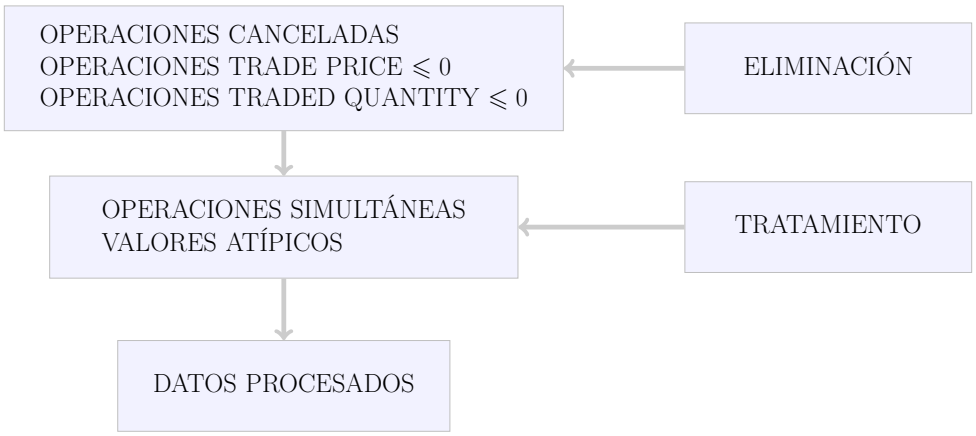


Figura 6.1: Procesado previo de los datos

datos financieros de alta frecuencia. En la presente investigación, el procesado previo de los datos se realizó según un procedimiento compuesto por la secuencia mostrada en la figura 6.1, incluyendo los pasos establecidos en Brownlees & Gallo (2006), es decir, un primer paso de *data cleaning*, para detectar y eliminar las operaciones canceladas y aquellas con precio o cantidad nula, y una segunda etapa de *data management*, para construir las series temporales que se utilizaron para alcanzar los objetivos del análisis. Se realizó una exploración inicial de las variables y se llevaron a cabo las operaciones que se describen a continuación. Con las variables indicadas, y con datos en el período diario de negociación, se efectuó un filtro para eliminar operaciones canceladas y órdenes no ejecutadas. También se rechazaron aquellas con *Trade Price* y *Traded Quantity* menores o iguales a cero (Hautsch, 2012). Adicionalmente, se realizó una transformación de las variables *Trade Price* y *Traded Quantity*, debido a que para un *timestamp* determinado pueden existir múltiples operaciones con sus respectivos precios y cantidades. Se evaluó la incidencia de esta particularidad y el análisis arrojó los resultados que se muestran en la tabla 6.2.

Tabla 6.2: Operaciones simultáneas. Datos brutos

Ticker	Trades	POS (%)	POSDP (%)
Group 1			
ANIM3	243834	40.93	6.39
BRPR3	337064	53.61	1.03
LEVE3	257078	47.74	4.70
PARD3	239697	41.71	5.86
SAPR4	318154	53.45	1.59
VULC3	285162	44.39	2.21
Group 2			
B3SA3	4621343	51.31	1.69
BBAS3	5097897	55.63	2.42
BBDC3	2014462	53.29	1.87
BBSE3	3038689	63.04	1.12
BRFS3	3425237	53.48	1.84
BRML3	3240840	53.88	0.45
CIEL3	4477212	60.94	0.65
CSNA3	2734771	55.46	0.73
GGBR4	3817163	50.72	1.00
GOAU4	2593701	38.67	0.35
HYPE3	1655824	46.09	1.73
JBSS3	4014349	58.22	0.50
KROT3	4023300	55.04	0.56
LAME4	2828343	48.91	1.20
MGLU3	1159513	58.39	8.53
PETR3	3165573	44.01	1.80
RAIL3	3595665	55.71	0.79
RENT3	3370796	58.86	1.26
TIMP3	1975100	45.22	0.62
USIM5	3078855	48.63	0.50

El número de observaciones de cada *ticker* es variable, dependiendo del número de operaciones negociadas durante el período de análisis. Las observaciones de las series temporales de operaciones negociadas de algunos *tickers* del primer grupo superan las 300000, mientras que en el segundo grupo el orden es de millones. En las columnas POS y POSDP, se refleja el porcentaje de operaciones simultáneas y el porcentaje de operaciones simultáneas con distinto precio, respectivamente. Hay porcentajes de operaciones simultáneas superiores al 60% en algunos valores, mientras que el de aquellas que tienen distinto precio no alcanza el 9%.

Con respecto a esta particularidad de los datos, en Brownlees & Gallo (2006) se propuso tomar la mediana del precio, aunque se consideró que en su lugar también podría tomarse una media ponderada. La transformación realizada para tratar las operaciones simultáneas se resume de la forma que se expone a continuación. Sean y_i los precios, v_i las cantidades negociadas e $i = 1, 2, ..., k$ la secuencia de *timestamps* iguales. Sea un *timestamp* x con varias operaciones negociadas simultáneas. Entonces, para el *timestamp* x, tenemos que

$$y_x = \frac{\Sigma_{i=1}^{k} y_i v_i}{\Sigma_{i=1}^{k} v_i} \qquad v_x = \Sigma_{i=1}^{k} v_i \tag{6.1}$$

El resultado es un *Trade Price* medio ponderado por volumen y una suma total de la variable *Traded Quantity* para cada grupo de operaciones simultáneas. Con el objeto de perder la menor información posible, se creó una variable con el número de transacciones simultáneas de cada grupo, denominada *Transactions*, como se recoge en Brownlees & Gallo (2006).

Finalmente, las variables seleccionadas para la extracción de características de las series de operaciones negociadas fueron las siguientes: *Trade Time*, *Trade Price*, *Traded Quantity* y *Transactions*.

Se utilizó *transaction data* con los 6 primeros *tickers* citados y *tick data* o *tick time* con los 20 últimos (Griffin & Oomen, 2008). El primer término se refiere a los datos sin ningún tipo de transformación, generados en el instante en el que se producen las operaciones negociadas. Los datos de alta frecuencia brutos contienen secuencias en las que el precio no varía, por lo que la serie asociada de retornos para dichas secuencias estaría

compuesta por ceros. Si dichas secuencias se reducen al primer valor, se obtienen los *tick data* o *tick time*, que es la denominación empleada cuando entre una operación negociada y la siguiente hay una diferencia de al menos un *tick*, definido como la variación más pequeña que se puede producir en el precio de cotización (Hautsch, 2012). Los datos *tick time* utilizados representan una reducción próxima al 75% con respecto a los *transaction data*, de acuerdo a las cifras del número de operaciones negociadas que se muestran en las tablas 6.2 y 6.3, relativas a los datos brutos y limpios, respectivamente.

El grupo de los 20 últimos *tickers* se obtuvo a partir de los tickers procesados previamente. Todas las secuencias de *Trade Price* constante se redujeron al primer valor, pero las variables *Transaction* y *Traded Quantity* son la suma del total de valores de cada una de las secuencias.

La siguiente operación que se efectuó fue la detección y eliminación de valores atípicos (*outliers*) de forma muy conservadora, respetando la particular dinámica de los datos financieros de alta frecuencia, y de esta manera reducir la pérdida de información.

El tratamiento de valores atípicos de alta frecuencia utilizando filtros se trató en Verousis & Gwilym (2010), donde se comparó el algoritmo que proponen los autores con métodos de limpieza de datos existentes previamente. En la investigación doctoral, se adoptó el filtro adaptativo que se expone a continuación, propuesto en Brownlees & Gallo (2006) para detectar inconsistencias en los datos. Entonces, para una serie temporal con observaciones $i = 1, 2, ..., n$ y precios y_i, tenemos

$$|y_i - \overline{y}_i(k)| < 3s_i(k) + \gamma \,, \tag{6.2}$$

donde $\overline{y}_i(k)$ y $s_i(k)$ representan la media δ-recortada muestral y la desviación estándar muestral de un vecindario (*neighborhood*) de k observaciones alrededor de i. El parámetro γ se denomina granularidad (*granularity*).

Este filtro permite que la observación verdadera i se mantenga, mientras que la observación falsa se elimine. El vecindario de la primera observación de cada serie diaria se compone de las primeras k observaciones, el vecindario de la última observación lo constituyen las k últimas transacciones y para la observación que se encuentre en el medio de la serie se computan las $k/2$ transacciones previas y las $k/2$ observaciones posteriores, y así sucesivamente. De esta forma, la referencia para determinar la validez de una observación son las observaciones válidas más cercanas. Se tomó un porcentaje de recorte δ igual a 0.1 y se escogió una longitud de la ventana k igual a 30.

La granularidad tiene por objeto evitar la varianza cero que se origina cuando se producen secuencias de k precios iguales. De acuerdo a Brownlees & Gallo (2006), γ debería tomarse como un múltiplo de la variación mínima del precio para cada activo. Siguiendo esta premisa, se seleccionó γ como la media de los valores absolutos de los cuantiles del 5% y 95% de la serie en diferencias del precio. Utilizando el método propuesto en Brownlees & Gallo (2006), se detectaron los valores atípicos que figuran en la tabla 6.3, en la que también se indica el número de observaciones de cada serie temporal empleada en la experimentación.

Tabla 6.3: Observaciones y valores atípicos. Datos limpios

Ticker	Trades	Outliers	Ticker	Trades	Outliers
Transaction Time					
ANIM3	179008	85	PARD3	173149	66
BRPR3	207808	53	SAPR4	204239	41
LEVE3	174936	67	VULC3	197106	46
Tick Time					
B3SA3	1244902	32	HYPE3	530391	64
BBAS3	1341834	14	JBSS3	765707	18
BBDC3	582892	67	KROT3	820254	18
BBSE3	621448	46	LAME4	762381	41
BRFS3	874510	17	MGLU3	463454	11
BRML3	622923	38	PETR3	859144	17
CIEL3	785091	15	RAIL3	731211	42
CSNA3	511758	3	RENT3	836196	61
GGBR4	885488	4	TIMP3	472230	20
GOAU4	574962	1	USIM5	633536	1

Las series con los datos limpios se representan en la figura 6.4. Los 6 primeros *tickers* corresponden a datos *transaction time*, mientras que los 20 últimos son datos *tick time*. Con el objeto de aligerar los gráficos de esta figura, se redujo el número de observaciones de cada serie mediante una agregación de los datos a un minuto, cuyo resultado son series con un número de observaciones inferior a 60000 y a 84000, para los 6 primeros tickers y para los 20 últimos, respectivamente. Las series corresponden a 206 días de negociación.

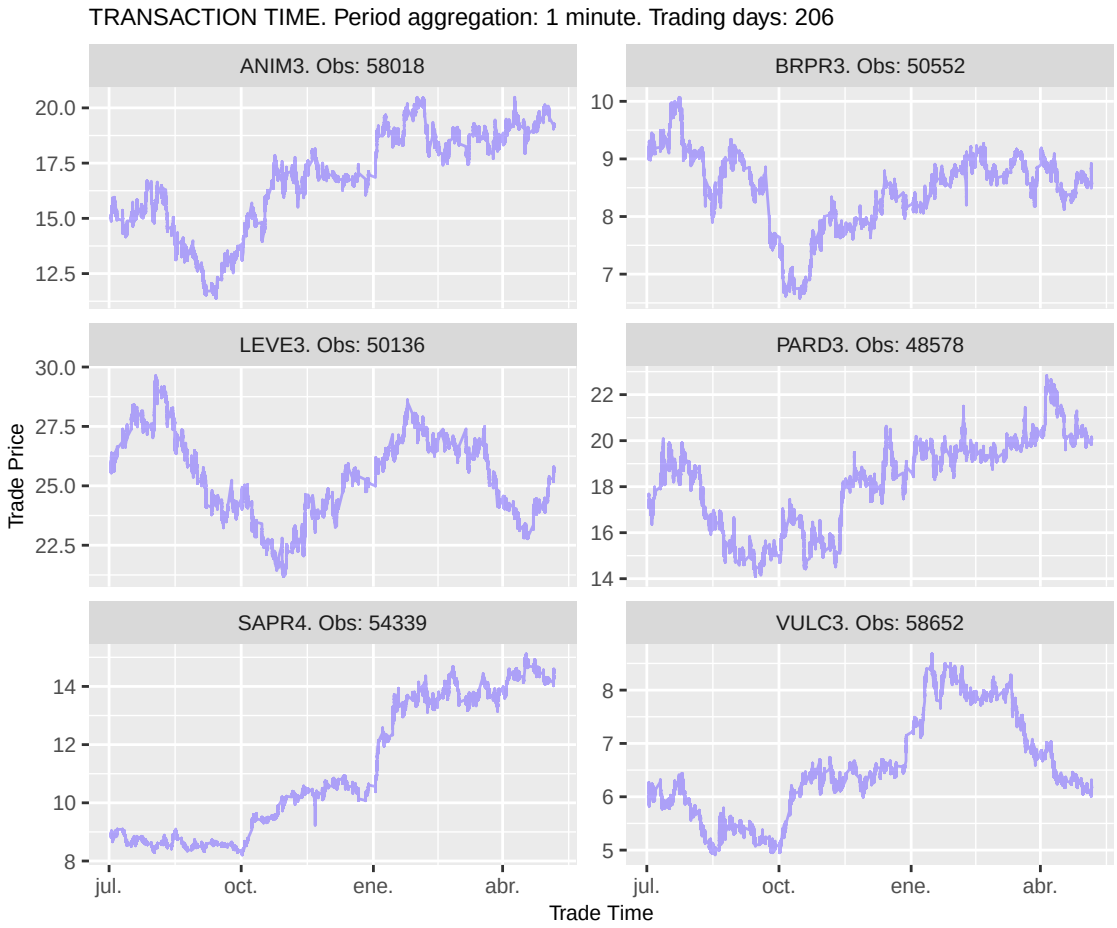


Figura 6.2: Series temporales de operaciones negociadas. Transaction time

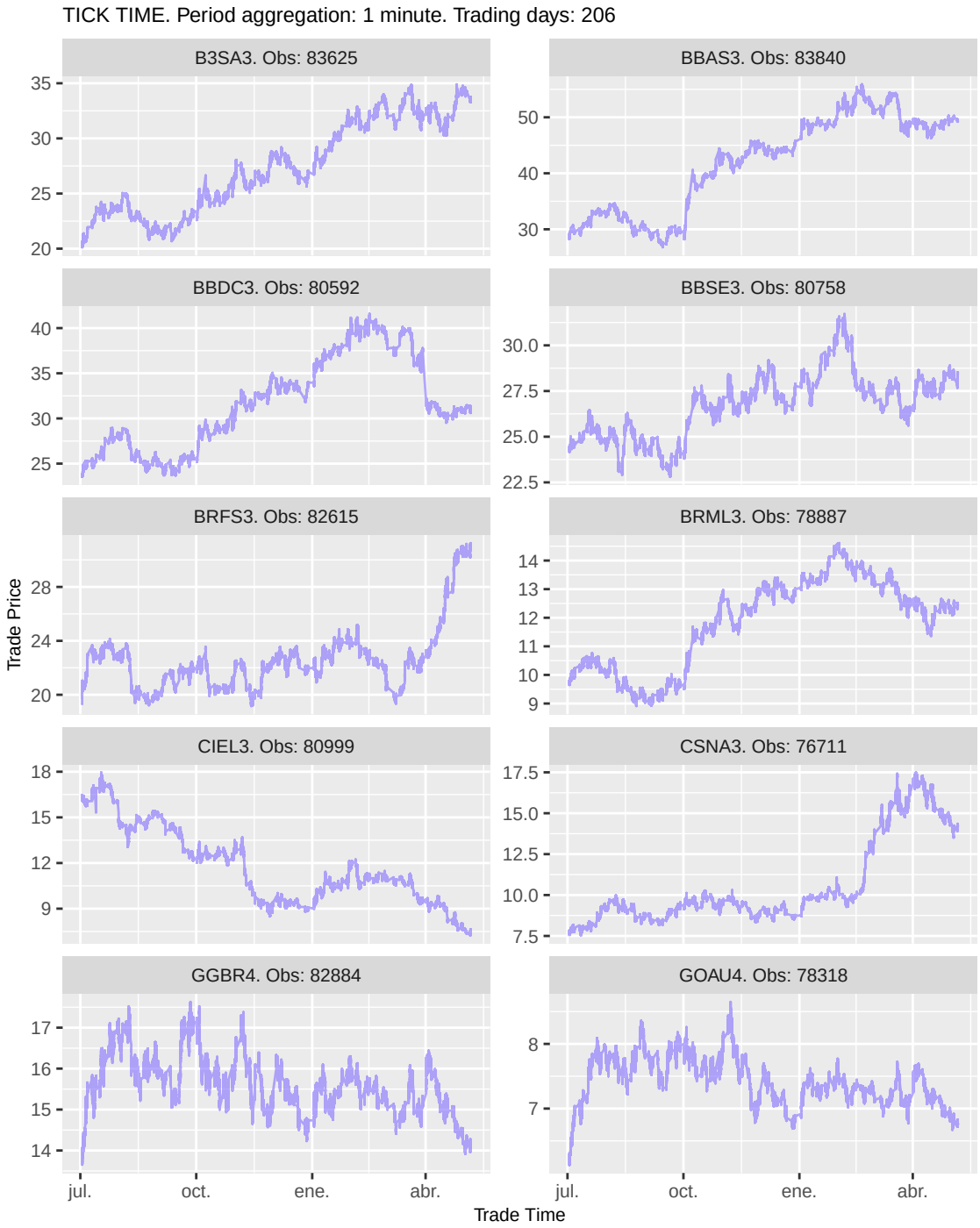


Figura 6.3: Series temporales de operaciones negociadas I. Tick time

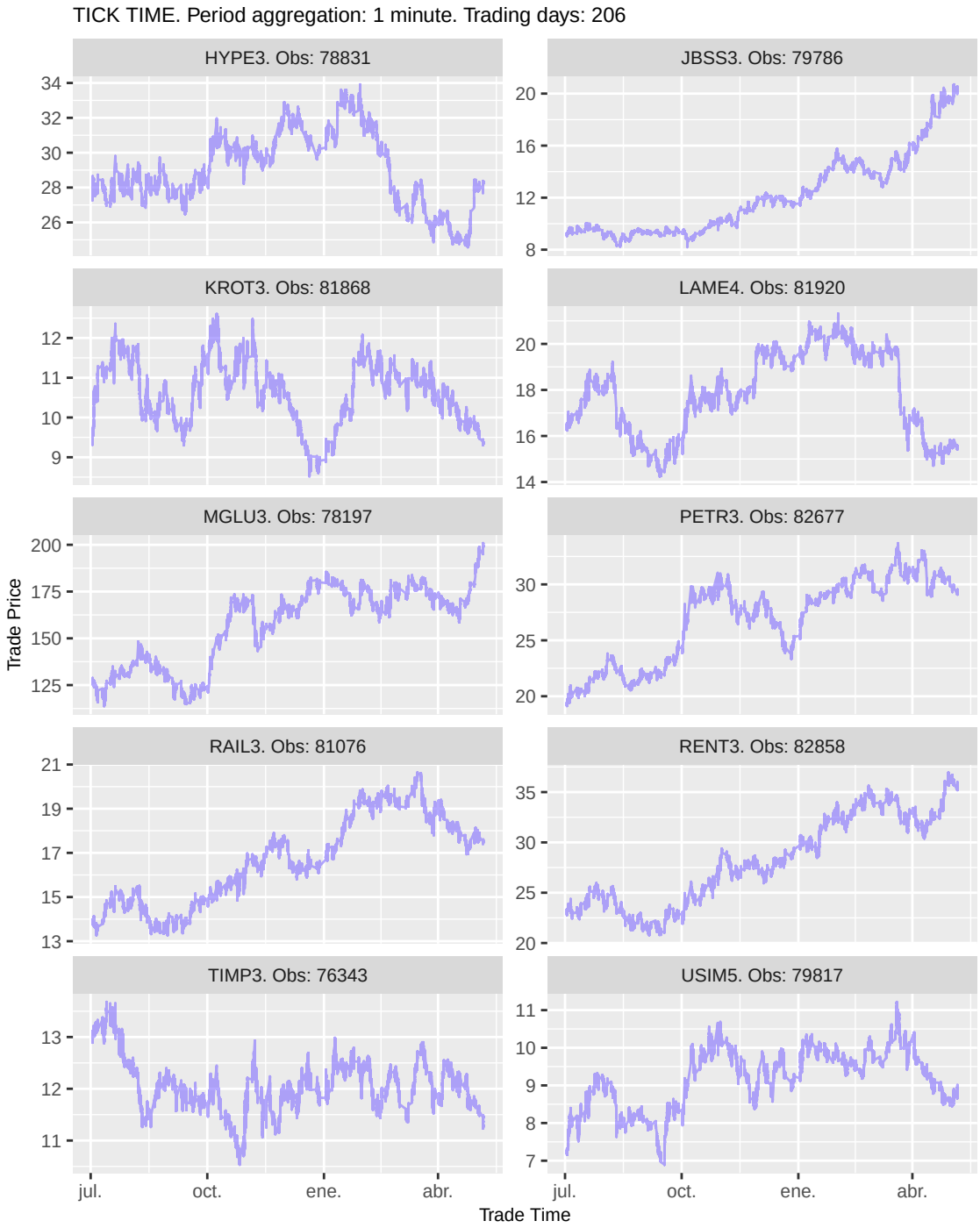


Figura 6.4: Series temporales de operaciones negociadas II. Tick time

Con respecto a la variable *número de observaciones diarias*, en la tabla 6.4 se muestra un resumen estadístico del número de observaciones diarias de los 6 activos utilizados en la experimentación relativa a la segmentación, además del correspondiente a los otros 20 valores empleados en la experimentación con modelos de inteligencia artificial.

Tabla 6.4: Operaciones negociadas diarias

Ticker	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Transaction Time						
ANIM3	170	621.75	767.5	868.56	970.25	2965
BRPR3	133	525.75	834.5	1008.52	1261.25	4149
LEVE3	122	522.50	796.0	848.88	1072.25	3839
PARD3	148	497.50	718.5	840.21	974.75	4520
SAPR4	125	515.50	790.5	991.25	1209.00	5191
VULC3	135	590.75	819.5	956.60	1262.50	2923
Tick Time						
B3SA3	2516	4309.25	5420.0	6043.06	6980.50	25390
BBAS3	1959	4703.00	5796.0	6513.69	7442.00	23324
BBDC3	1108	2012.50	2504.5	2829.25	3152.25	14618
BBSE3	1120	2011.00	2806.5	3016.51	3576.25	9704
BRFS3	1476	2819.25	3686.5	4245.11	5122.00	13638
BRML3	837	1960.00	2679.0	3023.71	3808.00	8512
CIEL3	1189	2539.75	3298.5	3811.05	4660.75	9530
CSNA3	561	1337.00	1988.0	2484.25	3176.50	12072
GGBR4	1438	3039.25	3855.0	4298.47	5113.00	13937
GOAU4	702	1665.50	2488.0	2791.07	3396.75	12936
HYPE3	735	1819.25	2259.5	2574.40	3194.25	7441
JBSS3	928	2279.75	3019.5	3716.94	4568.00	14586
KROT3	1198	2749.00	3664.0	3981.73	4805.50	10383
LAME4	966	2392.75	3099.0	3700.68	4512.00	11932
MGLU3	830	1635.75	2126.5	2249.72	2689.00	7913
PETR3	1473	2774.75	3674.0	4170.52	4822.00	14984
RAIL3	1015	2457.50	3154.0	3549.36	4239.50	10047
RENT3	1317	3128.50	3844.0	4058.91	4802.25	11003
TIMP3	667	1476.75	2012.5	2292.28	2866.75	9774
USIM5	786	1882.25	2605.5	3075.41	3718.75	11204

El máximo de observaciones diarias de los primeros 6 *tickers* se encuentra entre valores próximos a 3000 hasta cifras superiores a 5000, en alguno de los casos. En cuanto a los 20 últimos *tickers*, el máximo de uno de los activos supera ligeramente las 25000 observaciones en un día, con medias

aproximadamente seis veces superiores que las de los primeros 6 *tickers*, en algunos casos.

6.2 Reconstrucción del libro de órdenes límite

Para extraer características de las órdenes de compra y venta, primero se debe reconstruir el libro de órdenes y obtener su estado en determinados momentos. En Gould et al. (2013), se realizó una revisión de la literatura sobre diversos aspectos relativos al análisis y modelado de libros de órdenes límite, además de describir los denominados *stylized facts* de los mercados financieros, definidos como regularidades estadísticas no triviales que están presentes en los datos procedentes de dichos mercados. En Christensen & Woodmansey (2013), se definió la reconstrucción del LOB como el proceso por el cual se toman los datos emitidos y se regeneran para obtener el LOB multidimensional.

El LOB se reconstruye por medio de la combinación y procesado de los datos relativos a los archivos de órdenes de compra y de venta correspondientes al mercado principal. Las variables de estos datos utilizadas en la investigación se enumeran en la tabla 6.5, las cuales se describen en los correspondientes archivos de texto anteriormente mencionados.

Tabla 6.5: Variables de los datos de órdenes

Session Date	Order Price
Instrument Symbol	Total Quantity Order
Order Side	Traded Quantity Order
Sequential Order Number	Secondary Order ID
Order Datetime Entry	Order Status

Algunas de estas variables también se encuentran en los datos de operaciones negociadas, y otras son exclusivas de los datos de órdenes, las

cuales se definen a continuación:

- Order Price: precio al que se envía una orden al mercado.
- Total Quantity Order: cantidad del activo financiero en la orden.
- Order Side: lado de negociación de la orden, de compra o venta.
- Traded Quantity Order: cantidad del activo negociado de la orden.
- Sequential Order Number: número secuencial de registro de la orden.
- Secondary Order ID: número adicional que identifica la orden.
- Order Datetime Entry: instante temporal de registro de la orden.
- Order Status: estado de la orden dentro de su ciclo de vida.

La figura 6.5 contiene gráficos de caja de las órdenes diarias por cada *ticker* analizado. La media se representa por un cuadrado de color violeta.

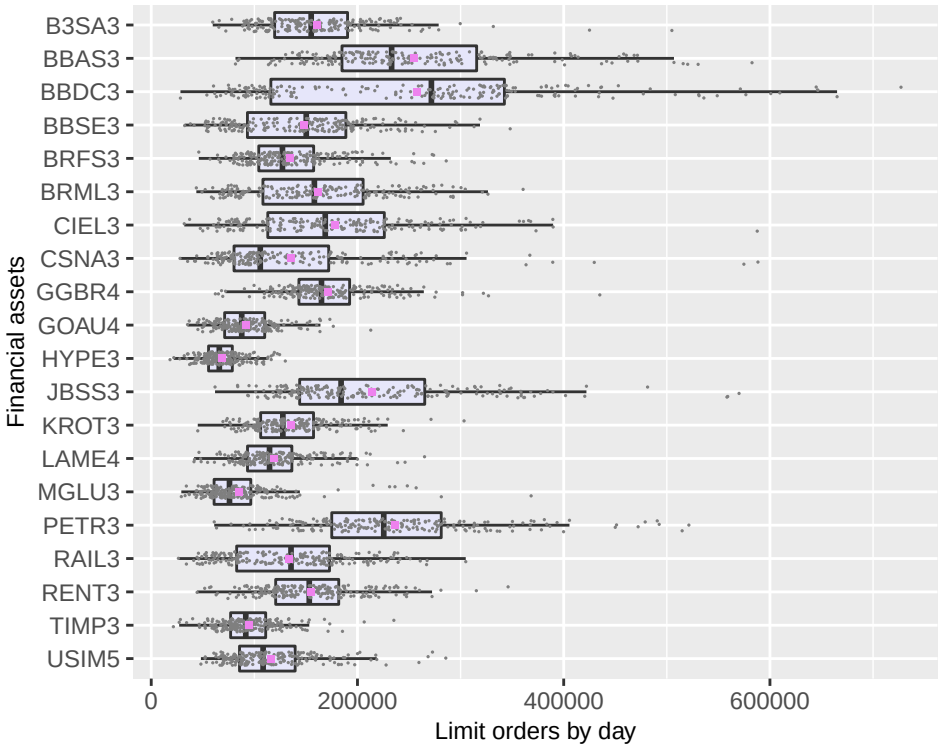


Figura 6.5: Órdenes límite por día. Datos brutos

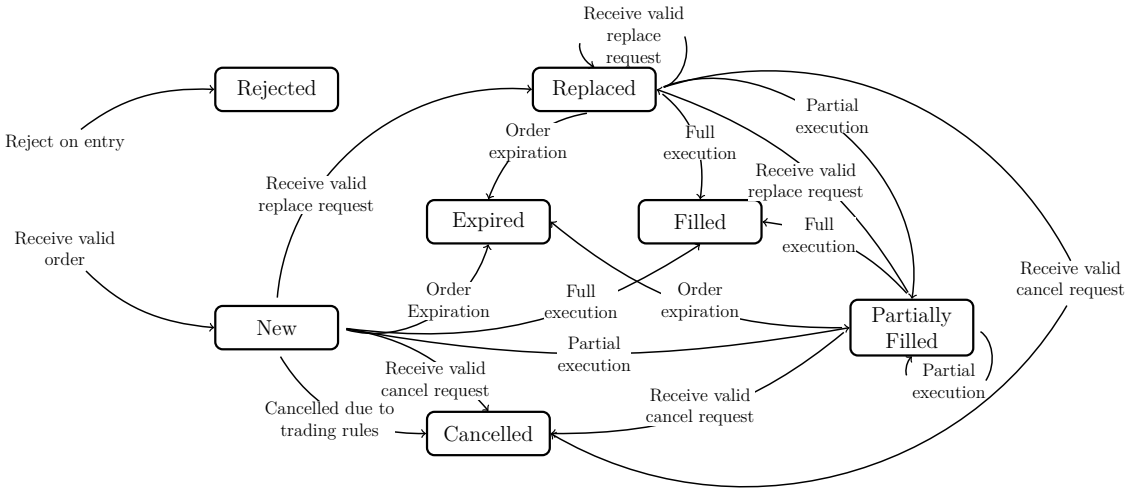
Las órdenes generadas en un día se añaden a aquellas que proceden de períodos previos y permanecen activas. El activo con *ticker* BBDC3 tiene un máximo de órdenes diarias que excede las 700000 órdenes, con un promedio de órdenes diarias ligeramente superior a 250000, lo cual refleja el esfuerzo computacional requerido para obtener los estados del LOB para los tiempos x_i de 582892 operaciones negociadas que tiene el activo con *ticker* BBDC3, como se indica en la tabla 6.3. El resto de *tickers* presentan menores cifras, pero el esfuerzo computacional para procesar los datos de estos activos, aunque menor que para el activo con *ticker* BBDC3, todavía es considerablemente elevado.

Estos datos de órdenes de compra y de venta también se procesaron previamente. Se eliminaron aquellas observaciones con errores, tales como valores de *Total Quantity Order* y *Order Price* negativos o nulos, además de aquellas para las cuales el *spread* era negativo, como se planteó en Barndorff-Nielsen et al. (2009).

Todas las órdenes tienen un ciclo de vida que puede pasar por diferentes estados. A efectos de lo contemplado en esta investigación, se ha representado el ciclo de vida de una orden en la figura 6.6, tomando como referencia el ciclo mostrado en el documento B3 (2018) del mercado B3. El gráfico representa las combinaciones posibles entre los diferentes tipos de estados. Hay 7 *Order Status*: *new*, *partially filled*, *filled*, *cancelled*, *replaced*, *rejected* y *expired*. Los posibles estados del ciclo de vida de una orden se definen a continuación.

- New: orden nueva que llega al mercado.
- Partially filled: orden parcialmente ejecutada.
- Filled: orden totalmente ejecutada.

- Cancelled: orden cancelada.
- Replaced: orden modificada.
- Rejected: orden rechazada cuando llega al mercado.
- Expired: orden que expiró.



Fuente: (B3, 2018)

Figura 6.6: Ciclo de vida de las órdenes

Los estados finales pueden ser: *filled*, *cancelled*, *rejected* y *expired*. En la figura, los estados finales son receptores de flechas, nunca emisores. El origen de una orden se registra con el *Order Status new*. Si dicha orden es rechazada en origen, pasaría al estado final *rejected*. En caso contrario, la orden original podría ser modificada, cancelada, cruzada total o parcialmente o expirar. La orden modificada podría pasar a los mismos estados que una orden original, y también podría modificarse nuevamente. Cuando una orden es parcialmente ejecutada, podría repetirse dicha ejecución, pero también podría pasar a los mismos estados que una orden *new* o *replaced*.

Cada orden con estado *new* se vincula con otros estados a través del *Sequential Order Number*, el cual permite conocer si la orden inicial se ha movido a estados diferentes. Si estuviésemos analizando el LOB en un instante determinado y existiesen varias órdenes con el mismo *Sequential Order Number*, vinculados a una orden con estado *new*, la orden activa en dicho instante sería aquella con el *timestamp* más reciente, y si hubiese varias simultáneas, aquella con mayor *Secondary Order ID*. El estado del LOB en un determinado instante incluye las órdenes activas en ese momento. Estas órdenes serían aquellas con el estado *new*, *replaced* o *partially filled*, las cuales no se cruzaron previamente y no fueron canceladas, rechazadas o expiraron.

Se desarrolló el algoritmo 2 para extraer características del LOB. La secuencia de estados del libro de órdenes se reconstruye internamente, pero no forma parte del output, ya que la finalidad era obtener las características del LOB en cada *Trade Time* de la serie de operaciones negociadas, no obtener el estado del LOB en un instante determinado. Este algoritmo extrae las características de los estados del LOB en cada uno de los días de análisis, para la secuencia de *timestamps* de cada *ticker* segmentado.

El algoritmo está formado por una función principal y cuatro funciones internas. Las órdenes se dividen en subconjuntos determinados por la secuencia de instantes temporales de las series de operaciones negociadas, de forma que los instantes posteriores al inicial actualizan el primer subconjunto. Sobre dichos subconjuntos de órdenes, se aplica la función *order_book*, con la finalidad de obtener los estados del LOB en cada instante de la secuencia. Esta función actúa de forma equivalente con las órdenes de compra y de venta, separando en dos subconjuntos los dos

Algoritmo 2: LOB features extraction

Input : trades, orders and session dates
Output: LOB features for each Trade Time

```
1  function features_extract(trades, orders, sessionDates):
2      filter(sessionDates) in trades → trades
3      get (timestamps sequence) from trades → seq
4      filter(sessionDates) in orders → orders
5      function order_book(orders):
6          group (SequentialOrderNumber)
7          arrange (desc(timestamp), desc(SecondaryOrderID))
8          filter (first group row)
9          filter (not final order states)
10         filter (OrderPrice > 0 & TotalQuantityOrder > 0)
11         remove (orders generating spread < 0)
12         return OrderBookState
13     end
14
15     function lob_features(OrderBookState):
16         function active_BuyOrders(OrderBookState):
17             filter(buy orders)
18             arrange(desc(OrderPrice), timestamp, SequentialOrderNumber)
19             return ActiveBuyOrders
20         end
21
22         function active_SellOrders(OrderBookState):
23             filter(sell orders)
24             arrange(asc(OrderPrice), timestamp, SequentialOrderNumber)
25             return ActiveSellOrders
26         end
27
28         features extraction → lobFeatures
29     end
30
31     n=length(seq)
32     timestamp ⩽ seq[1] → orders_1
33     for i=2 to n do
34         seq[i-1] < timestamp ⩽ seq[i] → orders_2, ..., orders_n
35         return ordersList
36     end
37     order_book(ordersList[1]) → obs_1
38     lob_features(obs_1)→lobFeatures_1
39     for i=2 to n do
40         order_book(bindRows(obs, ordersList[i])) → obs
41         lob_features(obs)→lobFeatures_2, ..., lobFeatures_n
42         return lobFeaturesList
43     end
44 end
```

tipos de órdenes y uniendo los resultados al final. Sobre la lista de estados del LOB obtenidos con la función *order_book*, se ejecuta la función *lob_features*. Esta función tiene dos funciones internas que devuelven las órdenes activas de compra y de venta. A partir de las mismas, se obtienen las características del LOB en cada instante de la secuencia de *timestamps* de la serie de operaciones negociadas.

6.3 Segmentación

Los datos utilizados de operaciones negociadas se registran por días de forma independiente, por lo que se dispone de una primera partición de las series. Esta particularidad permite segmentar cada uno de los días por separado. A priori, se desconocía qué modalidad de agregación sería la más adecuada, por lo que se seleccionaron tres períodos de agregación diferentes. Se realizó una comparativa de cuatro métodos de segmentación, en términos de tiempo de ejecución de trabajos de segmentación, de RSS y de BIC resultantes de cada segmentación. El método óptimo o exacto, aquí denominado segmentación directa o *single (SI) segmentation*, se comparó con el método alternativo, compuesto por tres modalidades de agregación de períodos diferentes: *5-minutes (5M)*, *1-minute (OM)* y *1-second (OS)*. La tabla 6.6 resume estadísticamente la variable *número de observaciones diarias* de estas agregaciones para cada *ticker*.

En estos casos, los máximos se reducen de forma significativa, especialmente en las agregaciones a 5 minutos y a 1 minuto. La reducción del número de observaciones diarias permite ejecutar segmentaciones más rápidas.

Tabla 6.6: Observaciones diarias por períodos de agregación

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
5M						
ANIM3	38	82.00	83.0	81.54	83.00	83
BRPR3	44	72.25	79.0	75.92	81.00	83
LEVE3	47	75.00	80.0	77.10	82.00	83
PARD3	37	72.00	78.0	74.71	81.00	83
SAPR4	47	79.00	82.0	79.68	83.00	83
VULC3	45	79.00	82.0	79.10	83.00	83
OM						
ANIM3	103	248.25	274.0	281.64	314.50	409
BRPR3	64	177.25	241.5	245.40	307.50	406
LEVE3	69	197.25	246.5	243.38	292.00	414
PARD3	66	178.25	232.0	235.82	287.25	415
SAPR4	83	215.25	264.0	263.78	319.75	414
VULC3	85	230.25	282.0	284.72	341.25	415
OS						
ANIM3	137	439.50	529.0	613.29	697.00	2413
BRPR3	86	322.50	518.5	663.13	873.50	2520
LEVE3	92	331.25	493.0	537.89	667.75	2851
PARD3	90	326.50	508.0	574.33	687.75	2836
SAPR4	104	365.50	542.5	668.33	781.50	3061
VULC3	113	440.00	596.0	706.76	897.25	2426

La experimentación consistió en ejecutar los trabajos de segmentación con los cuatro métodos sobre los 1236 conjuntos de datos, correspondientes a 206 días de negociación por cada uno de los 6 *tickers*. En total, se ejecutaron 4944 trabajos. El tiempo de ejecución se registró en horas sobre cada uno de los trabajos de segmentación enviados al supercomputador, contemplando la lectura del input, transformaciones de las variables, la segmentación y el almacenamiento del output. Se utilizó un núcleo del procesador con las mismas características para cada uno de los trabajos. También se computaron por días los valores de RSS y BIC para cada método de segmentación. Para ello, se utilizaron las expresiones anteriormente indicadas, los datos de los ajustes lineales de los intervalos de la segmentación directa y de los segmentos finales obtenidos en la doble segmentación.

En la experimentación realizada, se seleccionó un número mínimo de observaciones h en cada segmento igual a 6 y 3 para la primera y la segunda segmentación, respectivamente. Habitualmente, las series temporales de alta frecuencia experimentan saltos significativos entre observaciones, por lo que en la segunda segmentación se seleccionó el valor mínimo que h puede tomar, para reducir la influencia de los saltos sobre los ajustes en el caso de que los segmentos tuviesen un número de observaciones mínimo más alto. Con respecto a la primera segmentación, se seleccionó el mínimo valor de h para permitir la partición de los segmentos iniciales en al menos dos segmentos.

Se utilizó el test no paramétrico de Friedman (Friedman, 1937) para determinar si los cuatro métodos de segmentación sobre los 1236 conjuntos de datos, obtenidos por los 206 días de negociación de cada uno de los 6 *tickers*, tenían diferencias en cuanto a rendimiento evaluado por el tiempo de ejecución, el RSS y el BIC. Se determinó si uno o más métodos tenía rendimiento significativamente diferente contrastando la hipótesis nula *Ho: todos los métodos tienen el mismo rendimiento*. El resultado del test se muestra en la tabla 6.7. Se rechaza la hipótesis nula, por lo que no todos los métodos son iguales, en relación a su rendimiento.

Tabla 6.7: Test de Friedman

	Friedman's chi-squared	df	p-value
Execution time	3114.194	3	< 2.220446e-16
RSS	2579.457	3	< 2.220446e-16
BIC	3297.071	3	< 2.220446e-16

Se computó la diferencia crítica y todas las diferencias de promedio de posiciones entre métodos por pares, para un nivel de significancia de 0.05, obteniéndose el resultado de la tabla 6.8.

Tabla 6.8: Test de Nemenyi

Critical difference	k	df
0.1335	4	4940

Debido a que se comparó el mismo número de métodos en igual número de conjuntos de datos, la diferencia crítica obtenida es la misma para las métricas: tiempo de ejecución, RSS y BIC.

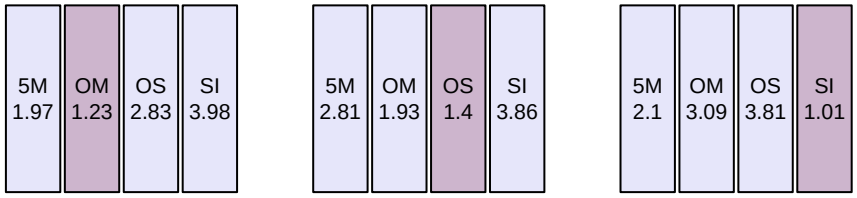
La matriz con todas las diferencias por pares constituye la matrix de Nemenyi, representada en la tabla 6.9. Si el valor absoluto de la diferencia entre el promedio de posiciones de dos métodos es superior a la diferencia crítica, se considera que el rendimiento de los métodos es diferente, por lo que la hipótesis nula de que dichos métodos tienen el mismo rendimiento se rechaza. En base a los resultados de la tabla 6.9, se rechazó la hipótesis nula citada para todas las métricas consideradas.

Tabla 6.9: Matrices de Nemenyi

	5M	OM	OS	SI
Execution Time				
5M	0.0000	0.7411	-0.8592	-2.0113
OM	0.7411	0.0000	-1.6003	-2.7524
OS	-0.8592	-1.6003	0.0000	-1.1521
SI	-2.0113	-2.7524	-1.1521	0.0000
RSS				
5M	0.0000	0.8803	1.4171	-1.0433
OM	0.8803	0.0000	0.5368	-1.9235
OS	1.4171	0.5368	0.0000	-2.4604
SI	-1.0433	-1.9235	-2.4604	0.0000
BIC				
5M	0.0000	-0.9879	-1.7067	1.0943
OM	-0.9879	0.0000	-0.7189	2.0821
OS	-1.7067	-0.7189	0.0000	2.8010
SI	1.0943	2.0821	2.8010	0.0000

En la figura 6.7, se realiza una representación gráfica similar a la uti-

lizada en Calvo & Santafé (2016) para otros *post hoc tests*, en la que los métodos de segmentación son los nodos y dos nodos se vinculan si la hipótesis nula de tener el mismo rendimiento no puede ser rechazada. En este caso, no hay vinculación entre nodos, ya que todos los métodos tienen rendimientos diferentes entre ellos. Hay 3 bloques de 4 nodos cada uno, donde los valores de cada nodo indican el promedio de posiciones, y los nodos con color oscuro tienen el mejor resultado para la métrica correspondiente. Empezando por la izquierda, cada bloque se refiere a las métricas *tiempo de ejecución*, *RSS* y *BIC*, respectivamente. El nivel de significancia es 0.05.



(a) Tpo. ejecución **(b)** RSS **(c)** BIC

Figura 6.7: Promedio de posiciones de métodos de segmentación

Atendiendo al criterio de menor tiempo de ejecución, se seleccionó la modalidad basada en agregación primaria a un minuto, por lo que ésta se utilizó para realizar la segmentación de los *tickers* que están entre los más negociados del mercado. Por otra parte, la modalidad seleccionada presenta un lugar intermedio en la evaluación de RSS y BIC. En la tabla 6.10, se recoge el resumen estadístico descriptivo de la variable *segmentos por día* obtenidos con la doble segmentación con agregación de períodos a un minuto para cada *ticker*.

Tabla 6.10: Segmentos por día

Ticker	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
B3SA3	111	168.00	195.5	206.72	236.00	408
BBAS3	107	210.50	250.5	271.83	313.75	681
BBDC3	69	129.00	157.5	169.72	205.00	365
BBSE3	56	113.25	140.0	144.68	167.75	305
BRFS3	77	123.50	146.5	158.50	179.00	368
BRML3	40	71.00	89.0	96.03	117.75	219
CIEL3	45	88.25	105.0	112.68	134.00	244
CSNA3	33	60.25	77.0	94.79	112.50	303
GGBR4	42	108.00	130.5	134.26	154.75	241
GOAU4	21	44.00	62.0	62.61	76.00	143
HYPE3	60	99.00	121.0	125.00	144.50	254
JBSS3	42	80.00	104.5	113.91	135.00	340
KROT3	50	87.25	112.0	115.93	136.75	224
LAME4	51	102.25	124.5	131.36	150.00	297
MGLU3	53	99.25	134.5	135.78	160.00	355
PETR3	80	129.00	154.5	170.76	204.75	452
RAIL3	50	100.00	119.5	127.15	144.00	291
RENT3	90	162.25	187.5	192.41	219.00	353
TIMP3	27	58.00	70.0	72.79	84.75	172
USIM5	23	65.25	82.5	87.24	101.00	216

6.4 Selección de variables

Las características extraídas de los intervalos procedentes de la segmentación de las series temporales de alta frecuencia se utilizaron para predecir el comportamiento futuro de tres variables respuesta. Una de éstas es la duración de los intervalos temporales obtenidos, que se midió directamente a partir de la diferencia del *timestamp* final e inicial de cada intervalo. Con respecto a las otras dos variables respuesta, se emplearon proxies, con la finalidad de predecir la volatilidad y la direccionalidad asociadas a los movimientos tendenciales del precio en futuros intervalos temporales. Estas variables forman parte del conjunto de características extraídas de cada segmento, a las que se añadieron algunas de las que se recogen en los trabajos revisados, además de otras que se consideró que podrían ser

explicativas de la varianza de las variables respuesta.

La literatura científica de alta frecuencia recoge evidencias de la influencia de determinadas variables en el comportamiento de los precios. Muestra de ello es lo expuesto en Cont et al. (2013), donde se manifestó que los cambios en los precios, durante períodos cortos de tiempo, son impulsados principalmente por el *order flow imbalance*, definido como el desequilibrio que se produce entre la oferta y la demanda de los precios de las mejores órdenes de compra y de venta. Asimismo, en Xu et al. (2018) se estudió el *multi-level order flow imbalance*, referido a múltiples niveles del libro de órdenes. En Cao et al. (2009), se contempló el desequilibrio en la longitud del libro de órdenes en función de la diferencia de cantidades agregadas de acciones en los lados de compra y de venta, dividido entre la suma de éstas. De forma similar, en la experimentación realizada se extrajeron características que reflejan un desequilibrio en el libro de órdenes por agrupaciones de niveles.

La tabla 6.11 contiene la selección de características obtenidas en cada uno de los intervalos procedentes de la segmentación realizada. En función de la procedencia de las características, éstas se dividen en dos bloques: características de las series de operaciones negociadas y características del libro de órdenes límite.

Las características extraídas de los intervalos temporales de alta frecuencia funcionan como regresores para predecir la volatilidad, la duración y la direccionalidad de los movimientos presentes en futuros intervalos. La descripción de cada una de las características se realiza en los siguientes apartados.

Tabla 6.11: Selección de características

Trades	
Average price	Interval transactions
Price variance	Interval transaction variance
Fitted price variance	Volume per second
Residual variance	Return per second
Mean absolute error	Volatility per unit of time
Interval duration	Squared log-return per second
Average trade duration	Total squared log-returns per second
Trade duration variance	Squared log-return per second variance
Interval observations	
LOB	
Average buy market	Average five levels OBI
Average sell market	Average ten levels OBI
Average buy volume	Average total levels OBI
Average sell volume	

Average price

Se refiere a la media aritmética del precio y_i en cada uno de los segmentos. Para un segmento j con observaciones i,

$$\bar{y}_j = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} y_i \tag{6.3}$$

Price variance

Relativa a la varianza del precio y_i en cada segmento. Sea un segmento j con observaciones i,

$$\sigma^2_{y_j} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j} (y_i - \bar{y}_j)^2 \tag{6.4}$$

Fitted price variance

Se trata de la varianza de los valores ajustados del precio $\hat{y}_i$ con la recta de regresión en cada segmento j,

$$\sigma^2_{\hat{y}_j} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j} (\hat{y}_i - \bar{y}_j)^2 \tag{6.5}$$

Residual variance

Es la varianza de los residuos ϵ_i obtenidos del ajuste lineal de cada segmento j,

$$\sigma^2_{\epsilon_j} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j} {\epsilon_i}^2 \tag{6.6}$$

Mean absolute error

Se refiere a la media aritmética del valor absoluto de los residuos ϵ_i de cada segmento j,

$$mae_j = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} |\epsilon_i| \tag{6.7}$$

Interval duration

Duración total en segundos de cada intervalo asociado al segmento j,

$$D_j = x_{i_j} - x_{i_{j-1}+1}, \tag{6.8}$$

donde x_{i_j} y $x_{i_{j-1}+1}$ son los *timestamps* final e inicial del intervalo, respectivamente. Un ejemplo de la representación gráfica de la evolución de un día es la que se observa en la figura 6.8.

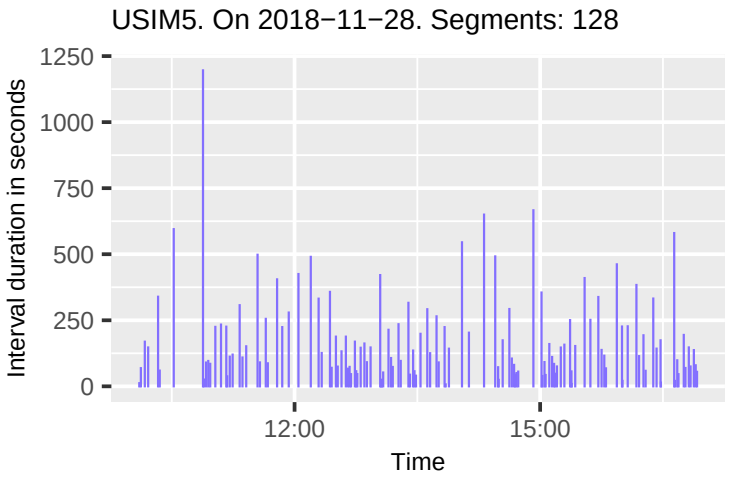


Figura 6.8: Evolución de la duración

Average trade duration

Es la media aritmética de la duración d_i en segundos de los precios y_i de cada segmento j,

$$\bar{d_j} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j - 1} d_i \tag{6.9}$$

$$d_i = x_{i+1} - x_i \tag{6.10}$$

Trade duration variance

Varianza de la duración de los precios d_i de cada segmento j,

$$\sigma_{d_j}^2 = \frac{1}{e_j - 2} \sum_{i_{j-1}+1}^{i_j - 1} \left(d_i - \bar{d_j}\right)^2 \tag{6.11}$$

Interval observations

Número de observaciones e_j de cada segmento j,

$$e_j = i_j - i_{j-1} \tag{6.12}$$

Interval transactions

Las observaciones simultáneas de algunos precios se registraron como una nueva variable cuando se ejecutó el procedimiento de limpieza de datos, por lo que el número de transacciones en cada segmento es mayor o igual al número de observaciones de éste, ya que pueden existir segmentos en los que no se produjeron observaciones simultáneas. Esta variable corresponde a la suma de todas las transacciones realizadas en el intervalo temporal de cada uno de los segmentos j,

$$T_j = \sum_{i_{j-1}+1}^{i_j} T_i \tag{6.13}$$

Interval transaction variance

Podemos tener una o más transacciones por cada observación de un segmento. La varianza del número de transacciones en cada segmento j se calcula como

$$\sigma^2_{T_j} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j} \left(T_i - \bar{T_j}\right)^2 \tag{6.14}$$

Value per second

Valor total $v_i y_i$ de las operaciones negociadas de cada segmento j por segundo,

$$v_j = \frac{1}{x_{i_j} - x_{i_{j-1}+1}} \sum_{i_{j-1}+1}^{i_j} v_i y_i \,, \tag{6.15}$$

donde el denominador representa la *interval duration* D_j.

Return per second

El criterio para determinar la direccionalidad en cada uno de los segmentos obtenidos consistió en clasificar el retorno, calculado entre los extremos de dichos segmentos, en una de las tres regiones que separan los cuantiles con probabilidades 1/3 y 2/3. Se considera que un segmento es alcista, estacionario o bajista dependiendo de que el retorno se clasifique como mayor que el segundo cuantil citado, entre dichos cuantiles o menor que el primero de estos cuantiles, respectivamente. Los segmentos tienen duración irregular, y podría ser que dos segmentos tuviesen el mismo re-

torno y las duraciones fuesen diferentes. Para diferenciar este aspecto, se definió una métrica que considera la información en X y en Y. Se trata del retorno del intervalo ajustado por la duración de éste, calculado como el retorno aritmético por segundo de cada segmento j,

$$R_j = \frac{\frac{y_{i_j} - y_{i_{j-1}+1}}{y_{i_{j-1}+1}}}{x_{i_j} - x_{i_{j-1}+1}}, \tag{6.16}$$

cuyo denominador corresponde a la *interval duration* D_j y el numerador es el retorno aritmético, donde y_{i_j} e $y_{i_{j-1}+1}$ equivalen al precio final e inicial de cada intervalo, respectivamente. En la figura 6.9, se muestra un ejemplo de una serie de retornos por segundo de todos los intervalos obtenidos por la partición de un día específico.

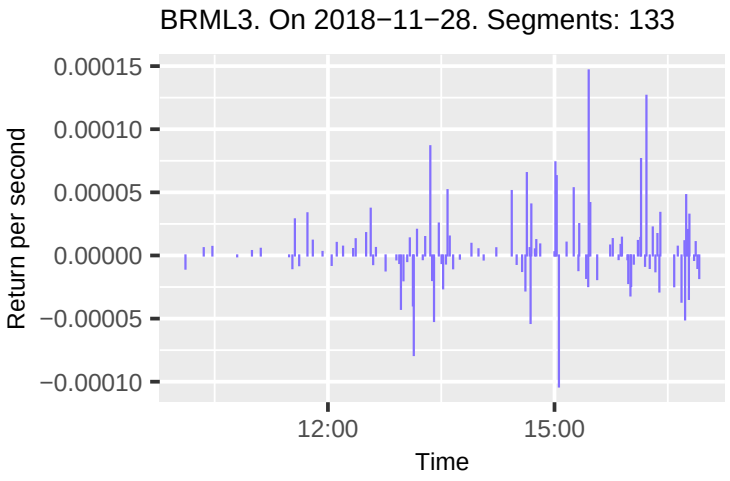


Figura 6.9: Evolución del retorno por segundo

Volatility per unit of time

La volatilidad es una variable no observada de los precios, por lo que para su estudio se emplean proxies. Tomando como referencia la volatilidad

ajustada por la duración entre transacciones presente en Engle (2000), se formuló la siguiente variable proxy para predecir la volatilidad asociada a cada movimiento contenido en el segmento j para las observaciones i,

$$\sigma_j^2 = \frac{1}{e_j - 2} \sum_{i_{j-1}+1}^{i_j - 1} \left(\frac{r_i}{\sqrt{d_i}} - \mu_{r_x} \right)^2, \tag{6.17}$$

donde r_i son retornos logarítmicos entre dos operaciones negociadas consecutivas y μ_{r_x} se refiere a la media de los *log-returns per square root duration* de cada segmento j para las observaciones i,

$$r_i = log(y_{i+1}) - log(y_i) \tag{6.18}$$

$$\mu_{r_x} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j - 1} \frac{r_i}{\sqrt{d_i}} \tag{6.19}$$

En la figura 6.10, se muestra un ejemplo de la evolución de la volatilidad por unidad de tiempo a través de todos los intervalos en los que se segmentó la serie de cotizaciones de un día determinado.

Squared log-return per second

Se trata del retorno del segmento j respecto a la raíz cuadrada de su duración al cuadrado,

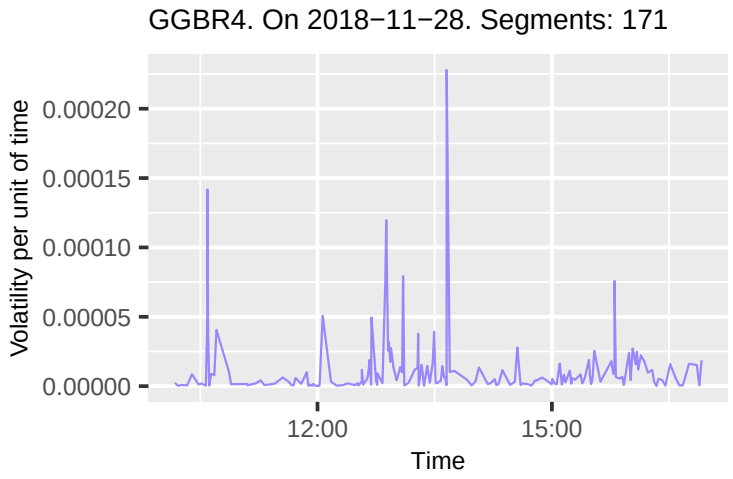


Figura 6.10: Evolución de la volatilidad por unidad de tiempo

$$r_{D_j}^2 = \left(\frac{r_j}{\sqrt{D_j}}\right)^2 = \frac{r_j^2}{D_j}, \tag{6.20}$$

donde D_j es el *interval duration* y r_j es el retorno logarítmico del segmento j,

$$r_j = log\, y_{i_j} - log\, y_{i_{j-1}+1} \tag{6.21}$$

Total squared log-returns per second

Suma de los *squared log-returns per second* de cada segmento.

$$S_{r_{j_i}} = \sum_{i_{j-1}+1}^{i_j-1} \left(\frac{r_i}{\sqrt{d_i}}\right)^2 = \sum_{i_{j-1}+1}^{i_j-1} \frac{r_i^2}{d_i}\,, \tag{6.22}$$

Squared log-returns per second variance

Varianza de los retornos logarítmicos al cuadrado por segundo en cada segmento j para las observaciones i,

$$\sigma_j^2 = \frac{1}{e_j - 2} \sum_{i_{j-1}+1}^{i_j - 1} \left(\left(\frac{r_i}{\sqrt{d_i}} \right)^2 - \mu_{r_x^2} \right)^2 , \tag{6.23}$$

donde $\mu_{r_x^2}$ es

$$\mu_{r_x^2} = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j - 1} \left(\frac{r_i}{\sqrt{d_i}} \right)^2 = \frac{1}{e_j - 1} \sum_{i_{j-1}+1}^{i_j - 1} \frac{r_i^2}{d_i} \tag{6.24}$$

Average buy market

Para cada instante x_i, tenemos precios y volúmenes de compra por cada uno de los niveles del libro de órdenes. Multiplicamos cada uno de los precios por su correspondiente volumen y sumamos los productos obtenidos de todos los niveles. Finalmente, calculamos la media aritmética de las sumas resultantes en cada segmento j,

$$\bar{V}_{b_j} = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} (P_b V_b)_i , \tag{6.25}$$

donde $P_b V_b$ es la suma de los productos precio-volumen de todos los niveles del libro de órdenes en cada instante x_i.

Average sell market

Se calcula de la misma forma que la característica *average buy market*, pero para las órdenes de venta del libro de órdenes,

$$\bar{V}_{s_j} = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} (P_s V_s)_i \tag{6.26}$$

Average buy volume

Es la media aritmética de la suma de los volúmenes de compra V_b de todos los niveles del libro de órdenes para todos los instantes x_i del segmento j,

$$\bar{V}_{b_j} = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} (V_b)_i \,, \tag{6.27}$$

Average sell volume

Calculado como la característica *average buy volume*, pero con volúmenes de venta,

$$\bar{V}_{s_j} = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} (V_s)_i \tag{6.28}$$

Average five levels OBI

Es un tipo de desequilibrio del libro de órdenes, denominado *order book imbalance (OBI)*. En primer lugar, se suman los volúmenes de los prime-

ros 5 niveles de cada lado del libro de órdenes (V_{5b}, V_{5s}) en cada instante x_i de cada segmento j. A continuación, se divide la diferencia entre los volúmenes obtenidos, compra menos venta, entre la suma de estos volúmenes. Finalmente, se calcula la media aritmética de las fracciones obtenidas para los instantes temporales de cada segmento,

$$\overline{OBI}_{5_j} = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} \frac{V_{5b_i} - V_{5s_i}}{V_{5b_i} + V_{5s_i}} \tag{6.29}$$

Average ten levels OBI

Se calcula de forma similar que la característica *average five levels OBI*, pero considerando 10 niveles del libro de órdenes,

$$\overline{OBI}_{10_j} = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} \frac{V_{10b_i} - V_{10s_i}}{V_{10b_i} + V_{10s_i}} \tag{6.30}$$

Average total levels OBI

De la misma forma que las dos anteriores características, se obtiene el promedio del desequilibrio para la totalidad de niveles del LOB en cada segmento j,

$$\overline{OBI}_j = \frac{1}{e_j} \sum_{i_{j-1}+1}^{i_j} \frac{V_{b_i} - V_{s_i}}{V_{b_i} + V_{s_i}} \tag{6.31}$$

6.5 Aprendizaje automático

Las series de cotizaciones segmentadas están formadas por las variables relativas a las operaciones negociadas, a las que se añaden las variables del libro de órdenes límite obtenidas en cada *timestamp* de la serie de operaciones negociadas. Estas variables forman el conjunto de datos de cada *ticker* sobre el que se extraen las características de cada segmento. Los conjuntos citados se separan en los subconjuntos de entrenamiento y validación en la proporción 70/30, de forma que el subconjunto de entrenamiento está formado por los segmentos relativos a los primeros 144 días, mientras que al subconjunto de validación le corresponden los segmentos presentes en los 62 últimos días de las series temporales de cada *ticker*. La división de los conjuntos iniciales en los subconjuntos de entrenamiento y validación se efectuó respetando la secuencia temporal, de forma que las observaciones del subconjunto de validación ocurren temporalmente después de las observaciones del subconjunto entrenamiento. El siguiente paso consistió en extraer las características de cada intervalo de los subconjuntos citados y calcular los cuantiles con probabilidades 1/3 y 2/3 de las variables respuesta, obtenidos de los datos de dichas variables en los subconjuntos de entrenamiento. Los valores resultantes dividen las tres regiones del etiquetado de las variables respuesta. A continuación, se preparó la incrustación de los métodos de aprendizaje automático mediante la construcción de los casos, compuestos por la sucesión ordenada de las características de k segmentos, donde cada bloque es un *timestep*. En la experimentación, se seleccionó un valor para k igual a 8, que aportó una ligera mejoría en los resultados de las pruebas iniciales realizadas, en relación a los obtenidos con valores de 3, 5, 10, 20, 50 y 100.

El último término de cada caso corresponde a la etiqueta de la variable respuesta correspondiente, de tal forma que cada uno de los casos tiene parte en X y parte en Y, formando las divisiones: *train*X, *train*Y, *test*X y *test*Y. Los casos se construyeron por días, de forma consecutiva. En la tabla 6.12, se muestra el número de segmentos y casos de cada uno de los subconjuntos citados por cada *ticker*. Se destaca que el método de segmentación desarrollado proporciona un número suficiente de casos para entrenar de forma fiable los modelos de inteligencia artificial, con los datos utilizados. Además, el número de casos en el subconjunto de validación tiene un tamaño suficiente para conseguir una validación fiable de los modelos.

Tabla 6.12: Segmentos y casos

	Segments		Samples	
Ticker	Train	Test	Train	Test
B3SA3	29081	13503	27929	7426
BBAS3	37140	18858	35988	12263
BBDC3	22960	12003	21808	6000
BBSE3	20473	9332	19321	8536
BRFS3	21450	11202	20298	10229
BRML3	13693	6089	12541	3176
CIEL3	16636	6576	15484	5036
CSNA3	10439	9088	9287	5084
GGBR4	20429	7229	19277	5120
GOAU4	9969	2929	8817	2003
HYPE3	17521	8230	16369	5521
JBSS3	13837	9629	12685	7363
KROT3	16604	7278	15452	5217
LAME4	19108	7953	17956	4668
MGLU3	20521	7450	19369	5059
PETR3	24845	10332	23693	4533
RAIL3	17437	8756	16285	4408
RENT3	26343	13293	25191	9744
TIMP3	10912	4083	9760	2659
USIM5	12948	5023	11796	3602

Las divisiones *train*X y *train*Y de los casos se emplearon para entrenar

los modelos de inteligencia artificial construidos. Las características de diseño de estos modelos y sus respectivos hiperparámetros se exponen a continuación.

XGBoost

Los parámetros seleccionados se clasifican en tres bloques: parámetros generales, parámetros *booster* y parámetros de aprendizaje. El primer bloque se refiere al tipo de *boosting* empleado, que fue de tipo árbol. En cuanto al segundo grupo, se establecieron valores para los parámetros de la tasa de aprendizaje η igual a 0.001, la reducción mínima de pérdidas γ para realizar una partición adicional en un nodo del árbol igual a 3, la profundidad máxima de un árbol igual a 5 y el ratio de submuestreo del entrenamiento igual a 0.75. Los parámetros de aprendizaje seleccionados se refieren al objetivo de aprendizaje con la función para multiclase *softprob* y a la métrica de evaluación para la validación de los datos, denominada tasa de error para clasificaciones multiclase, que es igual a la proporción de número de casos fallidos entre el total de casos. Finalmente, se fijaron los valores de dos parámetros relativos a la ejecución computacional: el máximo número de iteraciones *boosting* igual a 1000 y el número de rondas para detener el entrenamiento, si el rendimiento no mejoraba, igual a 50.

El input de los modelos se compone de las características de la tabla 6.11 descritas anteriormente, repetidas un número de *timesteps* específico. La importancia de características del modelo se calcula considerando cada una de las variables de la secuencia de *timesteps*, razón por la que se agruparon para determinar la contribución total de cada una de las características seleccionadas.

El procedimiento habitual para realizar predicciones con un modelo de clasificación de aprendizaje automático consiste en entrenar el modelo sobre un subconjunto de datos de entrenamiento, hacer predicciones con un nuevo input *testX* y realizar una comparación del output con la variable respuesta *testY*, obteniendo la matriz de confusión y sus métricas asociadas. El resultado proporciona la capacidad predictiva del modelo.

6.6 Recursos computacionales

Los algoritmos que se muestran en la presente tesis doctoral, así como el resto de *scripts* relativos a esta investigación, se desarrollaron en lenguaje *R* (R Core Team, 2020). El paquete *strucchange* (Zeileis et al., 2002) se utilizó para la segmentación directa de series temporales. En relación a la experimentación con los modelos de inteligencia artificial citados en este documento, se emplearon principalmente los paquetes *xgboost* (Chen et al., 2020) y *caret* (Kuhn, 2020). Con respecto a la explotación de los datos, fundamentalmente se utilizó la colección de paquetes *tidyverse* (Wickham et al., 2019), los paquetes *plyr* (Wickham, 2011) y *data.table* (Dowle & Srinivasan, 2020), así como los paquetes *quantmod* (Ryan & Ulrich, 2020), *zoo* (Zeileis & Grothendieck, 2005), *tidyquant* (Dancho & Vaughan, 2020a), *timetk* (Dancho & Vaughan, 2020b) y *tibbletime* (Vaughan & Dancho, 2020). Los cinco últimos, relativos a la modelización y análisis cuantitativo financiero y al tratamiento específico de series temporales. El paquete *PerformanceAnalytics* (Peterson & Carl, 2020) se usó para cuestiones vinculadas al análisis de riesgo y rendimiento. Para aspectos relacionados con formatos de fecha y tiempo, se utilizaron los paquetes *lubridate* (Grolemund & Wickham, 2011) y *hms* (Müller, 2020). Para la construcción, conexión y gestión de bases de datos, se emplearon

los paquetes *DBI* (R Special Interest Group on Databases [R-SIG-DB] et al., 2019) y *dbplyr* (Wickham & Ruiz, 2020). Los tests estadísticos se efectuaron con el paquete *scmamp* (Calvo & Santafé, 2016).

La experimentación cuyos resultados se presentan en esta tesis doctoral se llevó a cabo utilizando los recursos del *Centro de Supercomputación y Visualización de Madrid (CeSViMa)*, perteneciente a la *Universidad Politécnica de Madrid*. La ejecución de los trabajos se realizó en el *Magerit-3*, un supercomputador compuesto por 68 nodos *Lenovo ThinkSystem SD530*, cada uno de los cuales dispone de 2 procesadores *Intel Xeon Gold 6230* de 20 cores @2.10 GHz (1.344 GFLOPS), 192 GB de RAM y 480 GB de SSD local. Los nodos están interconectados por dos redes 25 Gbps de baja latencia, una de ellas con arquitectura *flat-tree* destinada en exclusiva al paso de mensajes.

7

Resultados y discusión

Los resultados de la investigación realizada están referidos a la ingeniería de características desarrollada, como solución al problema planteado. Según lo expuesto anteriormente, esta metodología se basa en el método de doble segmentación de series temporales con agregación de períodos, cuyos resultados fueron utilizados para decidir qué método se emplearía para segmentar las series con mayor número de observaciones del mercado bursátil B3. Estos resultados están formados por los valores obtenidos con cada método de segmentación, en relación a las métricas: tiempo de ejecución, RSS y BIC.

Adicionalmente, se mostró la aplicación de la ingeniería de características diseñada a tres problemas predictivos concretos, obteniéndose resultados relativos a las capacidades predictivas de los modelos construidos,

las cuales fueron medidas con las métricas *kappa* y *F1-score*. Esta última referida a cada una de las clases: *high* (*F1-high*), *medium* (*F1-medium*) y *low* (*F1-low*).

Resultados de segmentación

Los valores de las métricas *tiempo de ejecución*, *RSS* y *BIC* obtenidas con el método de segmentación desarrollado se evaluaron estadísticamente considerando todas las segmentaciones diarias, comparando el método óptimo con tres modalidades de agregación de períodos, en términos de dichas medidas. Los resultados están resumidos en la figura 6.7 y también en la tabla 7.1, que contiene el promedio de posiciones de las tres modalidades propuestas y el método óptimo exacto para cada una de las medidas citadas. Según se expuso en la subsección 6.3, los cuatro métodos son significativamente diferentes para dichas métricas.

Como se observa en la tabla 7.1, los tres tipos de doble segmentación consiguieron mayor rapidez de ejecución que el método óptimo aplicado directamente, además de menor error total. La modalidad con agregación a un minuto fue la más rápida y la segmentación con agregación a un segundo fue la que obtuvo menor error, es decir, mayor precisión. En cuanto al BIC, y en relación a las otras dos modalidades propuestas, la agregación a un minuto obtuvo resultados intermedios. Por estas razones, y debido a que se priorizó el menor tiempo de ejecución, se seleccionó la agregación de período a un minuto para segmentar los activos más negociados del mercado utilizado.

Los tiempos de ejecución, así como el RSS y el BIC, se obtuvieron para

Tabla 7.1: Resultados segmentación

	Execution time	RSS	BIC
OM	1.23	1.93	3.09
5M	1.97	2.81	2.10
OS	2.83	1.40	3.81
SI	3.98	3.86	1.01

cada segmentación diaria, y cada *ticker* tiene 206 días de segmentación, por lo que el tiempo de ejecución para segmentar cada *ticker* es igual a la suma de los tiempos de ejecución correspondientes a cada segmentación diaria. En la figura 7.1, se muestran los tiempos de ejecución totales de los trabajos de segmentación con cada *ticker*, para cada uno de los cuatro métodos de segmentación. La representación del tiempo de ejecución total de cada *ticker* indica que la segmentación directa tiene valores muy superiores a las tres alternativas propuestas, que presentan tiempos de ejecución próximos, siendo la doble segmentación con agregación a un minuto la que menores tiempos de ejecución totales emplea, como se ha observado en el test realizado sobre las segmentaciones diarias.

Por definición, el método de segmentación óptimo o exacto es el de segmentación directa. Esta afirmación se corrobora con los resultados obtenidos, ya que la segmentación óptima aplicada directamente encabeza el promedio de posiciones con menor BIC. No obstante, las tres modalidades de doble segmentación son métodos de segmentación óptima o exacta sobre cada uno de los segmentos resultantes de la segmentación primaria con agregación temporal. El resultado final de cada una de estas tres alternativas es la concatenación de segmentaciones óptimas o exactas de cada uno de los segmentos procedentes de la segmentación primaria. El interés de los métodos de segmentación propuestos está en la viabilidad de seg-

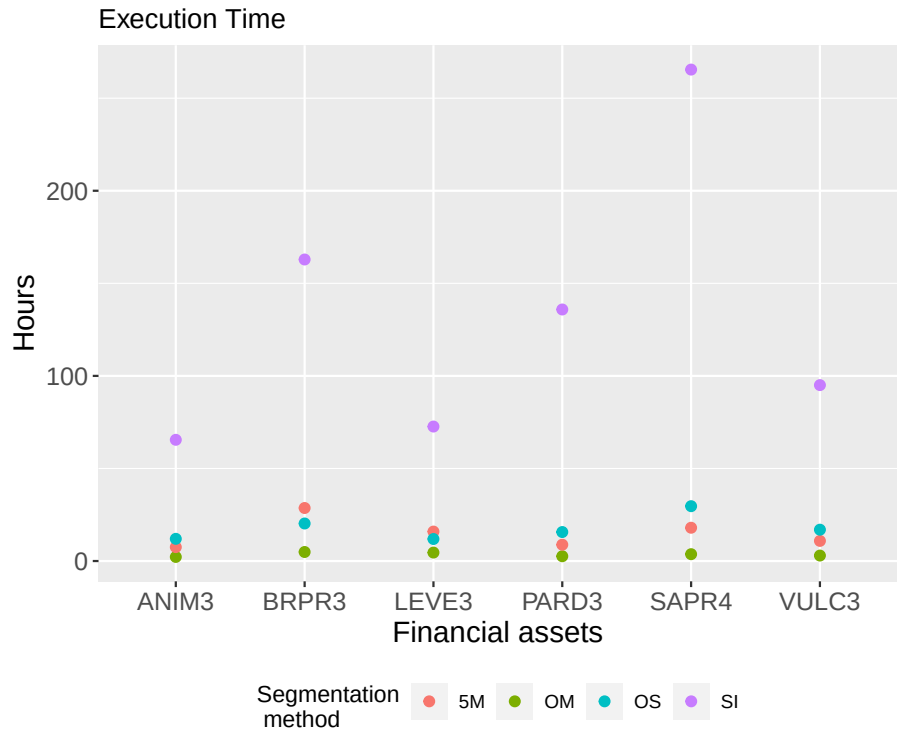


Figura 7.1: Tiempo de ejecución total de trabajos de segmentación

mentar series temporales de alta frecuencia mediante un método óptimo o exacto, ya que los tiempos de ejecución se reducen considerablemente y hacen posible que se pueda utilizar el método óptimo con los valores del mercado bursátil con mayor número de operaciones negociadas.

En la figura 7.2, se observan los valores del BIC de los cuatro métodos de segmentación para los 6 *tickers* seleccionados, donde el promedio de BIC en cada uno de los gráficos de caja se ha representado mediante un cuadrado de color amarillo. Esta figura es una referencia del orden de magnitud del BIC para estos métodos sobre los 1236 conjuntos de datos, además de proporcionar una visión general de las cifras de BIC para cada segmentación diaria. De acuerdo a lo observado en este gráfico, no hay grandes diferencias entre el método óptimo y el resto de modalidades,

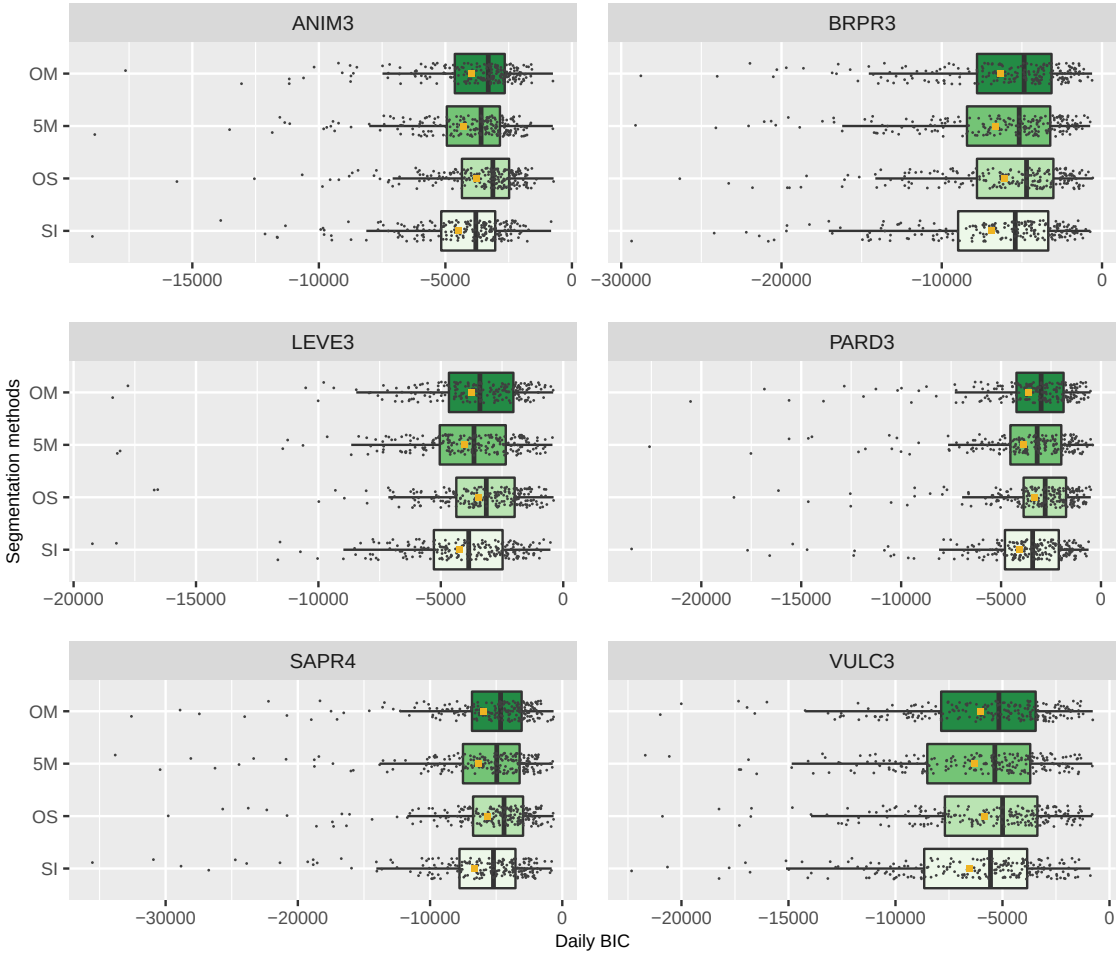


Figura 7.2: BIC diario por activo financiero y método

en términos de BIC. La media del BIC del método óptimo es ligeramente inferior a la del resto de métodos. Este aspecto es positivo, ya que indica que el método propuesto, en cualquiera de sus modalidades, está próximo al método exacto, a efectos del BIC. La gran ventaja del método propuesto sobre el óptimo radica en que el tiempo de computación es considerablemente inferior.

En base a los resultados obtenidos, no se considera viable la utilización del método de segmentación SI para segmentar las series temporales de alta frecuencia con mayor número de observaciones del mercado bursátil analizado, razón por la cual se desarrolló la alternativa propuesta, ya que con una complejidad algorítmica de orden cuadrático, las diferencias en los tiempos de ejecución serían considerablemente mayores para las series con un número de observaciones superior al de las empleadas en la experimentación realizada.

A la vista de los resultados y teniendo en cuenta los criterios de rapidez y precisión, se considera viable emplear cualquiera de las modalidades propuestas. No obstante, la que tiene menores tiempos de ejecución sobre los datos utilizados es la modalidad con agregación a un minuto.

Resultados de modelos predictivos

Los valores de las métricas de rendimiento obtenidas de las predicciones efectuadas con cada uno de los modelos de aprendizaje automático construidos para predecir las variables volatilidad, duración y dirección, se presentan en las tablas 7.2, 7.3 y 7.4, respectivamente. Los resultados están clasificados por *tickers* y métrica de rendimiento: *kappa* y *F1-score*

de cada una de las tres clases. Adicionalmente, se muestran los valores que delimitan la región de cada clase en las columnas denominadas $Q_{1/3}$ y $Q_{2/3}$. Los mejores resultados se destacan en negrita para cada una de las métricas.

Los mejores resultados se obtuvieron en la predicción de la volatilidad, seguida por la duración y la direccionalidad, de acuerdo a los valores de *kappa* de las tablas 7.2, 7.3 y 7.4. En relación a las dos primeras variables, los mejores rendimientos se consiguieron en la predicción de las clases *high* y *low*. Con respecto a la direccionalidad, los mejores rendimientos se lograron para la clase *medium*, según indican las cifras de *F1-score* obtenidas.

En relación a los valores de los cuantiles con probabilidades 1/3 y 2/3 ($Q_{1/3}$ y $Q_{2/3}$), que delimitan las regiones de las tres clases, están referidos a la volatilidad por unidad de tiempo, a la duración de la tendencia intradiaria en segundos y al retorno por segundo, en función de la variable respuesta correspondiente.

En relación a la volatilidad, y según se muestra en la tabla 7.2, el mejor resultado para *kappa* se consiguió para el *ticker* PETR3, con un valor de 0.33. En lo que respecta a la métrica *F1-score*, el mejor valor se obtuvo para la clase *low*, igual a 0.68, conseguido también para el mismo *ticker*. Para la clase *high*, la mejor cifra de *F1-score* se alcanzó con el *ticker* CIEL3, igual a 0.65, mientras que para la clase *medium* la mejor cifra fue igual a 0.5, alcanzada con el *ticker* JBSS3.

En cuanto a la duración, los resultados están reflejados en la tabla 7.3,

Tabla 7.2: Resultados predicción volatilidad

	Performance metrics					
		F1-score			Delimiters	
Ticker	kappa	high	med	low	$Q_{1/3}$	$Q_{2/3}$
B3SA3	0.25	0.50	0.28	0.67	1.45e-06	6.850e-06
BBAS3	0.24	0.52	0.33	0.62	6.30e-07	2.980e-06
BBDC3	0.19	0.35	0.45	0.55	2.20e-07	3.010e-06
BBSE3	0.18	0.46	0.29	0.56	8.60e-07	6.060e-06
BRFS3	0.25	0.56	0.47	0.49	1.87e-06	1.070e-05
BRML3	0.25	0.52	0.47	0.54	1.94e-06	1.645e-05
CIEL3	0.22	**0.65**	0.39	0.37	3.42e-06	2.185e-05
CSNA3	0.16	0.56	0.35	0.38	9.90e-07	8.090e-06
GGBR4	0.24	0.47	0.38	0.60	1.66e-06	1.018e-05
GOAU4	0.22	0.41	0.48	0.57	2.98e-06	2.610e-05
HYPE3	0.21	0.47	0.36	0.58	7.40e-07	5.630e-06
JBSS3	0.24	0.51	**0.50**	0.54	2.39e-06	1.677e-05
KROT3	0.30	0.61	0.47	0.52	2.35e-06	1.814e-05
LAME4	0.24	0.48	0.44	0.58	1.23e-06	9.640e-06
MGLU3	0.19	0.48	0.22	0.61	1.76e-06	8.180e-06
PETR3	**0.33**	0.61	0.30	**0.68**	5.30e-07	2.950e-06
RAIL3	0.24	0.54	0.44	0.53	2.00e-06	1.343e-05
RENT3	0.17	0.40	0.36	0.57	1.90e-06	8.280e-06
TIMP3	0.28	0.54	0.47	0.55	1.30e-06	1.028e-05
USIM5	0.25	0.47	**0.50**	0.54	1.74e-06	1.746e-05

donde el mejor valor de *kappa* se obtuvo con el *ticker* USIM5, igual a 0.26. En relación al *F1-score*, se alcanzaron valores de 0.69 y 0.77 para las clases *high* y *low* y *tickers* GOAU4 y CSNA3, respectivamente. Para la clase *medium*, las mejores cifras fueron iguales a 0.37.

En términos generales, las cifras que presenta la tabla 7.3 para la variable respuesta duración de la tendencia intradiaria son ligeramente inferiores que los conseguidos en la predicción de la volatilidad de la tendencia intradiaria.

Finalmente, los resultados obtenidos para la variable direccionalidad, presentados en la tabla 7.4, fueron significativamente inferiores a los conseguidos para las variables volatilidad y duración. El máximo valor al-

canzado para la métrica *kappa* fue igual a 0.17, conseguida con el *ticker* PETR3. En el caso de esta variable, se consiguió el mejor resultado de la *F1-score* para la clase *medium*, alcanzándose el valor de 0.63 con el *ticker* GOAU4. Con respecto a las otras dos clases, se obtuvieron valores de 0.47 y 044 para las clases *high* y *low* y los *tickers* JBSS3 y CSNA3, respectivamente.

Tabla 7.3: Resultados predicción duración

	Performance metrics					
		F1-score			Delimiters	
Ticker	kappa	high	med	low	$Q_{1/3}$	$Q_{2/3}$
B3SA3	0.18	0.52	0.33	0.48	50.857	121.824
BBAS3	0.20	0.49	0.33	0.56	40.395	95.004
BBDC3	0.17	0.44	0.33	0.56	61.709	143.653
BBSE3	0.17	0.52	0.34	0.46	63.140	164.148
BRFS3	0.23	0.49	0.32	0.61	63.015	164.962
BRML3	0.16	0.50	**0.37**	0.44	107.544	245.928
CIEL3	0.18	0.59	0.36	0.38	73.197	201.058
CSNA3	0.25	0.43	0.27	**0.77**	139.053	323.325
GGBR4	0.19	0.63	0.28	0.41	76.467	177.293
GOAU4	0.22	**0.69**	0.29	0.39	163.862	356.536
HYPE3	0.17	0.50	0.28	0.52	65.058	184.884
JBSS3	0.19	0.37	0.26	0.69	103.490	248.070
KROT3	0.17	0.56	0.26	0.46	86.393	206.133
LAME4	0.21	0.55	0.26	0.55	77.553	182.603
MGLU3	0.20	0.60	0.30	0.45	51.846	160.148
PETR3	0.21	0.60	0.31	0.47	60.671	138.369
RAIL3	0.17	0.44	**0.37**	0.53	82.613	198.650
RENT3	0.18	0.47	0.33	0.55	53.666	131.865
TIMP3	0.19	0.61	0.30	0.41	133.463	327.919
USIM5	**0.26**	0.63	0.32	0.52	116.463	267.837

Tabla 7.4: Resultados predicción dirección

	Performance metrics					
		F1-score			Delimiters	
Ticker	kappa	high	med	low	$Q_{1/3}$	$Q_{2/3}$
B3SA3	0.14	0.34	0.55	0.38	-5.90e-06	6.01e-06
BBAS3	0.16	0.40	0.49	0.42	-6.72e-06	6.80e-06
BBDC3	0.15	0.42	0.47	0.41	-6.21e-06	6.11e-06
BBSE3	0.14	0.34	0.54	0.38	-4.47e-06	4.50e-06
BRFS3	0.15	0.38	0.52	0.38	-5.49e-06	5.34e-06
BRML3	0.11	0.28	0.58	0.31	-3.91e-06	4.19e-06
CIEL3	0.13	0.35	0.55	0.30	-5.16e-06	5.01e-06
CSNA3	0.12	0.44	0.38	**0.44**	-4.46e-06	4.61e-06
GGBR4	0.14	0.33	0.59	0.30	-5.22e-06	4.95e-06
GOAU4	0.16	0.32	**0.63**	0.30	-3.88e-06	3.36e-06
HYPE3	0.13	0.37	0.53	0.35	-4.87e-06	5.10e-06
JBSS3	0.10	**0.47**	0.36	0.37	-4.30e-06	4.66e-06
KROT3	0.14	0.34	0.57	0.28	-5.36e-06	5.62e-06
LAME4	0.14	0.36	0.54	0.34	-5.25e-06	5.25e-06
MGLU3	0.14	0.33	0.58	0.31	-6.78e-06	6.43e-06
PETR3	**0.17**	0.31	0.62	0.36	-6.61e-06	6.66e-06
RAIL3	0.14	0.38	0.52	0.36	-4.94e-06	4.87e-06
RENT3	0.13	0.40	0.47	0.39	-6.41e-06	6.58e-06
TIMP3	0.15	0.29	0.57	0.38	-3.00e-06	2.91e-06
USIM5	0.16	0.32	0.58	0.38	-4.70e-06	4.68e-06

Con la finalidad de explicar los modelos construidos y determinar la influencia que tienen las características extraídas en cada una de las 3 variables respuesta, se ha analizado la importancia relativa de cada una de las características en los modelos relativos a cada uno de los 20 *tickers*. Si se hubiese analizado la importancia de variables en el modelo correspondiente a un único *ticker*, ésta se podría representar mediante un gráfico de barras, donde cada una de éstas sería la importancia de una característica. En este caso, dado que los valores obtenidos se han escalado entre 0 y 1, la suma total de la importancia para un único *ticker* sería igual a 1. Sin embargo, queremos conocer cómo se han comportado las variables globalmente en los modelos de los 20 *tickers* utilizados en la experimentación para cada una de las tres variables respuesta. El resultado obtenido

se muestra en las figuras 7.3, 7.4 y 7.5, las cuales contienen los gráficos de caja que permiten visualizar y comparar las distribuciones estadísticas de la contribución de cada una de las características en los modelos de todos los *tickers* para cada una de las 3 variables respuesta. En las figuras 7.3, 7.4 y 7.5, donde se destacan en color violeta los valores de la media en cada gráfico de caja, podemos ver que existe cierta uniformidad respecto a la importancia de las variables para cada uno de los *tickers*, dentro de cada una de las variables respuesta. Sin embargo, sí existen diferencias significativas dependiendo de la variable respuesta.

En la figura 7.3, referida a la importancia de las variables utilizadas como regresores para predecir la volatilidad vinculada a futuras tendencias intradiarias, las variables más importantes son la suma de retornos logarítmicos al cuadrado por segundo y la volatilidad por unidad de tiempo, seguidas de la duración media entre las operaciones negociadas y la varianza de los retornos al cuadrado por segundo en cada intervalo. Es un resultado lógico, ya que las dos primeras y la última son variables proxies de la volatilidad, mientras que la duración está directamente relacionada con la volatilidad.

En cuanto a la duración, en la figura 7.4 se recogen los valores de importancia de los regresores. Los mayores valores se alcanzan con las distintas métricas de duración: la duración de intervalo, el promedio y la varianza de duraciones entre operaciones negociadas de cada intervalo. Además, también destacan como variables importantes el valor por segundo y los retornos al cuadrado por segundo. El resultado obtenido también resulta lógico, ya que la variable duración del intervalo es la que mayor promedio de importancia presenta y es la variable respuesta, y las duraciones entre

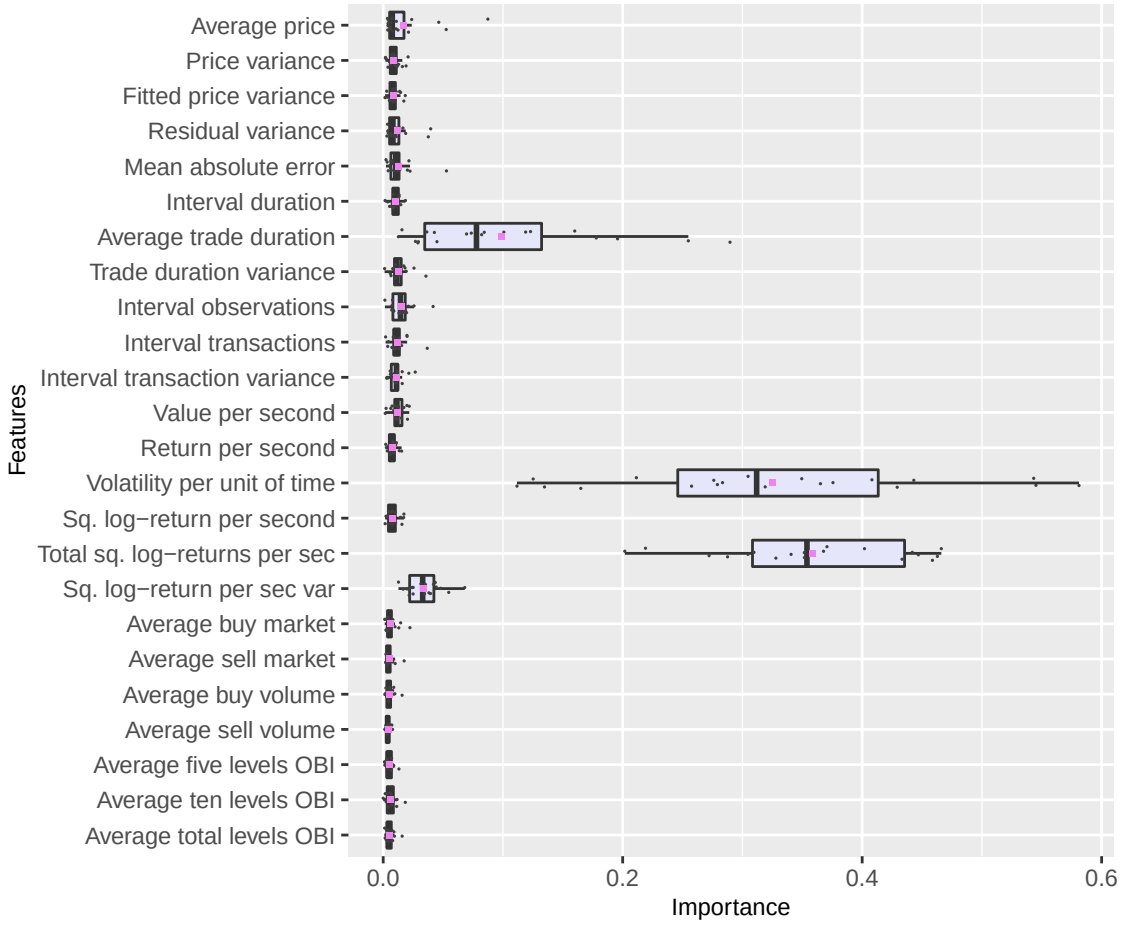


Figura 7.3: Volatilidad. Importancia de variables

operaciones negociadas están relacionadas con la variable respuesta.

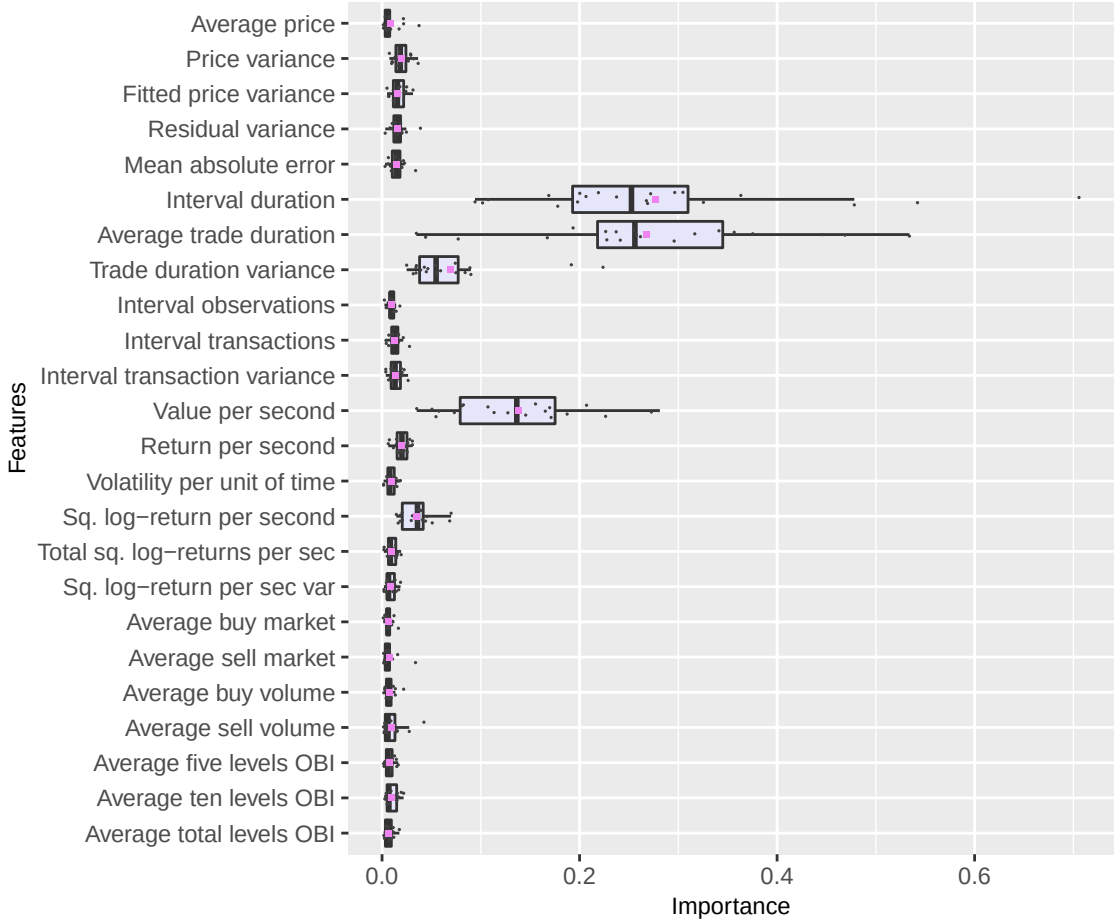


Figura 7.4: Duración. Importancia de variables

En relación a la direccionalidad, la figura 7.5 muestra, como variables con mayor importancia, la duración del intervalo, el retorno por segundo, el retorno al cuadrado por segundo y el valor por segundo. En este caso, se destaca que la variable respuesta no es la que mayor importancia tiene, ya que este lugar lo ocupa la duración del intervalo. No obstante, hay que tener en cuenta que el retorno por segundo es función de la duración del intervalo, ya que el retorno del intervalo se divide por dicha variable. También se destaca que en el caso de la direccionalidad gran parte de las

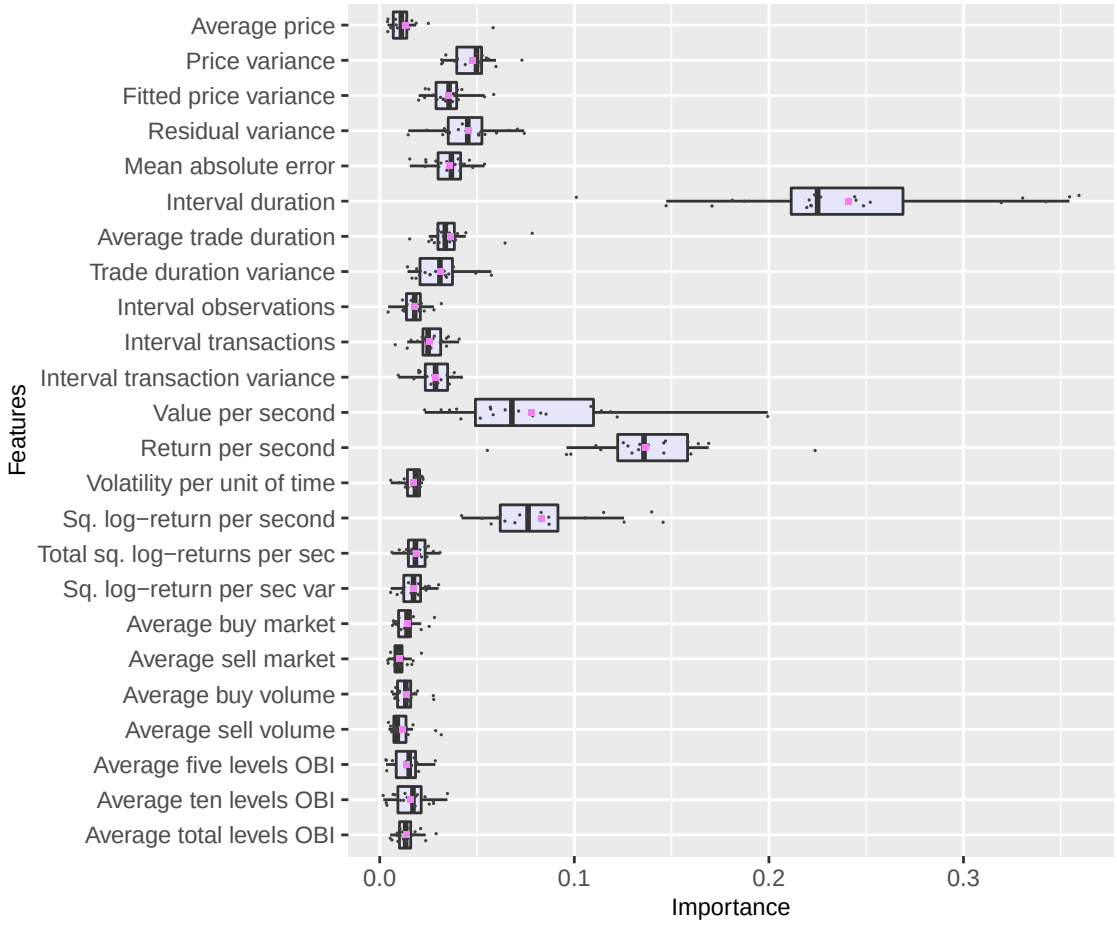


Figura 7.5: Dirección. Importancia de variables

características tienen relativa importancia, lo que no ocurría con las otras dos variables respuesta.

En términos generales, se destaca la mayor importancia de las variables extraídas de las series de operaciones negociadas, para todos los *tickers* y variables respuesta, con respecto a las características extraídas de los estados de los libros de órdenes límite. Únicamente, en el caso de la direccionalidad, las variables relativas al libro de órdenes presentan una ligera importancia, destacando entre éstas los promedios del desequilibrio del libro de órdenes de los primeros 5 y 10 niveles.

8

Conclusiones y líneas futuras

Se planteó el problema de extraer características de tendencias intradiarias en datos financieros de alta frecuencia, cuya resolución consistió en una ingeniería de características para este tipo de datos basada en segmentación de series temporales, cuya principal diferencia con respecto a la ingeniería de características habitual en inteligencia artificial es que la extracción de características se realiza en intervalos irregulares, determinados por la duración de dichas tendencias.

El número de observaciones de este tipo de datos es considerablemente elevado, y se requirió que el método para fragmentar las series en tendencias fuese suficientemente preciso, al menos tanto como el método recogido en la literatura científica denominado óptimo o exacto. Pero la complejidad algorítmica cuadrática de este método excluía su aplicación directa

sobre las series con mayor número de observaciones de los datos disponibles, por lo que fue necesario desarrollar un método de segmentación específico para datos de alta frecuencia. El enfoque fue diferente al planteamiento de reducir la complejidad algorítmica que recoge la literatura revisada. Se trataba de reducir los datos originales y aplicar el método exacto para obtener una alta precisión. Este planteamiento suponía un gran desafío, ya que la complejidad algorítmica cuadrática del método utilizado como base condicionaba el éxito de la investigación doctoral.

Finalmente, el método de doble segmentación con agregación de períodos fue la solución encontrada al problema de segmentar series temporales financieras de alta frecuencia. Sin embargo, la resolución del problema no finalizaba con el diseño y automatización de este método, había que evaluarlo y realizar una comparación tomando como *benchmark* el método exacto. Para ello, se buscaron series con un número de observaciones que permitiese la segmentación con el método exacto en un tiempo razonable. Localizadas dichas series, se diseñó una metodología de evaluación de métodos de series temporales. La prioridad era obtener tiempos de ejecución más bajos sin perder precisión, pero también considerando como referencia el criterio que determina que el método exacto es óptimo.

Se evaluó estadísticamente el tiempo de ejecución de trabajos de segmentación de series temporales de alta frecuencia con el método óptimo y se verificó que la complejidad algorítmica cuadrática de este tipo de segmentación hace que no sea viable su utilización para series temporales como las citadas, debido a su elevado tiempo de ejecución. Se propuso una alternativa con tres modalidades diferentes significativamente más rápidas, comprobándose que el error total es inferior al obtenido con el

método original, y que la modalidad que emplea una agregación inicial a un minuto tiene los tiempos de ejecución más bajos, presentando un equilibrio en términos de BIC, con respecto a las otras dos modalidades.

Utilizando la segmentación citada, se desarrolló una ingeniería de características que permite la extracción de las mismas en intervalos de alta frecuencia en los que se producen tendencias intradiarias con número de observaciones y duración variables. Los intervalos citados se extendieron al LOB, con la finalidad de extraer características de los estados del LOB sobre los subconjuntos de observaciones que componen cada intervalo. La metodología diseñada facilita el análisis de las variables extraídas y su predicción en intervalos futuros por medio de métodos de inteligencia artificial.

La metodología desarrollada se aplicó a la predicción de tres variables respuesta. Para ello, una parte de las características se seleccionó a partir de las destacadas por la literatura relativa a datos financieros de alta frecuencia, a las que se añadieron otras que se consideraron de interés para explicar la varianza de cada una de las variables respuesta. Para poder extraer características de las órdenes límite, a partir de los estados del LOB en cada instante de la serie de cotizaciones correspondiente, se reconstruyó el LOB en cada uno de estos instantes.

Las características extraídas de cada intervalo se utilizaron para alimentar modelos de inteligencia artificial, con la finalidad de predecir la volatilidad, la duración y la direccionalidad asociadas a los movimientos tendenciales de futuros intervalos. Se alcanzó una precisión superior en la predicción de la volatilidad y la duración que en la direccionalidad. La

importancia de las variables seleccionadas se obtuvo a partir del método XGBoost, determinándose que las variables vinculadas a las series de operaciones negociadas son las que mejor explican la varianza de las variables respuesta.

El método de segmentación DSPA consigue la segmentación de series temporales de alta frecuencia de forma precisa y en tiempo razonable, y permite reducir la dimensionalidad de los datos y extraer características de las tendencias intradiarias para su análisis. Dependiendo del criterio seleccionado, se podría escoger una modalidad específica de agregación de períodos temporales. Si el único criterio fuese el menor tiempo de ejecución, la opción sería la agregación de períodos a 1 minuto. En el caso de que el criterio fuese la mayor precisión, la modalidad a seleccionar sería la agregación de períodos a 1 segundo. Finalmente, si el criterio fuese una modalidad que estuviese más próxima al método exacto, en términos de BIC, la opción sería la agregación de períodos a 5 minutos.

La ingeniería de características basada en el método DSPA puede emplearse para predecir variables respuesta vinculadas a tendencias intradiarias de alta frecuencia mediante la utilización de métodos de inteligencia artificial. Esta ingeniería de características se aplicó a la extracción de unas características y predicción de variables respuesta concretas, con un método de inteligencia artificial determinado y utilizando unos datos específicos. No obstante, dicha metodología es de propósito general para series temporales de alta frecuencia y podría aplicarse en un ámbito más amplio, en función del problema que se desee resolver y dentro de las propias especificaciones del método.

Se ha proporcionado la primera solución a un nuevo problema en inteligencia artificial utilizando datos financieros de alta frecuencia. Como ha sido habitual con otros problemas en inteligencia artificial, a la primera solución le han sucedido muchas otras soluciones que, de alguna forma, mejoran la primera propuesta. Como futuros trabajos de investigación, se plantean soluciones que mejoren los resultados obtenidos en esta investigación, abordando los obstáculos de la realidad del momento.

Aunque los activos financieros empleados en la experimentación tienen un volumen representativo, existen otros mercados financieros en los cuales los volúmenes de negociación son mayores que el mercado estudiado. Por otra parte, debemos considerar que en el futuro los volúmenes de negociación podrían incrementarse significativamente o los sistemas electrónicos podrían registrar órdenes a frecuencias mayores que los milisegundos. En estos casos, podríamos tener que tratar con activos financieros con un número de observaciones mucho mayor que los que se han analizado en esta investigación, para los cuales la metodología desarrollada podría tener limitaciones. Sin embargo, esta metodología es escalable, por lo que si la aplicásemos a activos con un número de observaciones considerablemente mayor que los utilizados en esta investigación, se podrían considerar segmentaciones sucesivas. Para abordar este aspecto, sería necesario analizar el problema particular, ya que al reducir la dimensionalidad podríamos perder información, por lo que debería alcanzarse un equilibrio en la solución adoptada. Pero también podrían considerarse otros enfoques para manejar datos con un volumen superior a los presentes en esta investigación, que propongan una mejora de la metodología desarrollada o una alternativa.

Finalmente, también se podría establecer el objetivo de mejorar la precisión en la predicción de las variables respuesta seleccionadas para mostrar la aplicación de la metodología desarrollada, para lo cual se podría utilizar una ingeniería de características alternativa, otras transformaciones de las variables de los datos que permitan la incorporación de otros regresores que expliquen mejor la varianza de las variables respuesta, una optimización de los hiperparámetros del algoritmo XGBoost u otros métodos de aprendizaje automático diferentes al aquí utilizado.

Bibliografía

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723. https://doi.org/10.1109/TAC.1974.1100705

Andersen, T. G., Bollerslev, T. (1997). Intraday periodicity and volatility persistence in financial markets. *Journal of Empirical Finance*, *4*(2), 115–158. https://doi.org/10.1016/S0927-5398(97)00004-2

Arévalo, A., Nino, J., León, D., Hernandez, G., Sandoval, J. (2018). Deep learning and wavelets for high-frequency price forecasting. *International Conference on Computational Science*, *10861*, 385–399. Springer, Cham. https://doi.org/10.1007/978-3-319-93701-4_29

B3 Brasil, Bolsa, Balcão. (2018). *Entry point messaging guidelines. Version 2.9.4.* http://www.b3.com.br/data/files/BD/67/C3/FF/0DBB3610DF40D936790D8AA8/EntryPointMessagingGuidelines2.9.4.

pdf.

Bai, J., Perron, P. (2003). Computation and analysis of multiple structural change models. *Journal of Applied Econometrics*, *18*(1), 1–22. https://doi.org/10.1002/jae.659

Barndorff-Nielsen, O. E., Hansen, P. R., Lunde, A., Shephard, N. (2009). Realized kernels in practice: Trades and quotes. *The Econometrics Journal*, *12*(3), C1–C32. Retrieved from http://www.jstor.org/stable/23116045

Bauwens, L., Giot, P. (2000). The logarithmic ACD model: An application to the bid-ask quote process of three NYSE stocks. *Annales d'Économie et de Statistique*, (60), 117–149. Retrieved from http://www.jstor.org/stable/20076257

Bellman, R., Roth, R. (1969). Curve fitting by segmented straight lines. *Journal of the American Statistical Association*, *64*(327), 1079–1084. Retrieved from http://www.jstor.org/stable/2283487

Bojer, C. S., Meldgaard, J. P. (2021). Kaggle forecasting competitions: An overlooked learning opportunity. *International Journal of Forecasting*, *37*, 587-603. https://doi.org/10.1016/j.ijforecast.2020.07.007.

Brownlees, C., Gallo, G. (2006). Financial econometric analysis at ultra-high frequency: Data handling concerns. *Computational Statistics & Data Analysis*, *51*(4), 2232–2245. https://doi.org/10.1016/j.csda.2006.09.030

Calvo, B., Santafé, G. (2016). scmamp: Statistical comparison of multiple algorithms in multiple problems. *The R Journal*, *8*(1), 248–256. https://doi.org/10.32614/RJ-2016-017

Cao, C., Hansch, O., Wang, X. (2009). The information content of an open limit-order book. *Journal of Futures Markets*, *29*(1), 16–41. https://doi.org/doi.org/10.1002/fut.20334

Chen, T., Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y. (2020). *Xgboost: Extreme gradient boosting*. Retrieved from https://CRAN.R-project.org/package=xgboost

Christensen, H., Woodmansey, R. (2013). Prediction of hidden liquidity in the limit order book of GLOBEX futures. *The Journal of Trading*, *8(3)*, 68–95. https://doi.org/10.3905/jot.2013.8.3.068

Chundi, P., Rosenkrantz, D. J. (2009). Segmentation of time series data. In *Encyclopedia of data warehousing and mining, second edition* (pp. 1753–1758). IGI Global. https://doi.org/10.4018/978-1-60566-010-3.ch267

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*(1), 37–46. https://doi.org/

10.1177/001316446002000104

Cont, R., Kukanov, A., Stoikov, S. (2013). The price impact of order book events. *Journal of Financial Econometrics*, *12*(1), 47–88. https://doi.org/10.1093/jjfinec/nbt003

Dacorogna, M. M., Gençay, R., Müller, U. A., Olsen, R. B., Pictet, O. V. (2001). *An introduction to high-frequency finance* (p. 383). Academic Press. https://doi.org/10.1016/B978-0-12-279671-5.X5000-X

Dancho, M., Vaughan, D. (2020a). *Tidyquant: Tidy quantitative financial analysis*. Retrieved from https://CRAN.R-project.org/package=tidyquant

Dancho, M., Vaughan, D. (2020b). *Timetk: A tool kit for working with time series in r*. Retrieved from https://CRAN.R-project.org/package=timetk

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, *7*, 1–30. Retrieved from http://jmlr.org/papers/v7/demsar06a.html

Dixon, M. F., Polson, N. G., Sokolov, V. O. (2019). Deep learning for spatio-temporal modeling: Dynamic traffic flows and high frequency trading. *Applied Stochastic Models in Business and Industry*, *35*(3), 788–807. https://doi.org/10.1002/asmb.2399

Dixon, M., Klabjan, D., Bang, J. (2017). Classification-based financial markets prediction using deep neural networks. *Algorithmic Finance*, *6*(3-

4), 67–77. https://doi.org/10.3233/AF-170176

Doering, J., Fairbank, M., Markose, S. (2017). Convolutional neural networks applied to high-frequency market microstructure forecasting. *2017 9th Computer Science and Electronic Engineering (CEEC)*, 31–36. https://doi.org/10.1109/CEEC.2017.8101595

Dowle, M., Srinivasan, A. (2020). *Data.table: Extension of data.frame*. Retrieved from https://CRAN.R-project.org/package=data.table

Ederington, L. H., Lee, J. H. (1993). How markets process information: News releases and volatility. *The Journal of Finance*, *48*(4), 1161–1191. Retrieved from http://www.jstor.org/stable/2329034

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, *50*(4), 987–1007. Retrieved from http://www.jstor.org/stable/1912773

Engle, R. F. (2000). The econometrics of ultra-high-frequency data. *Econometrica*, *68*(1), 1–22. Retrieved from http://www.jstor.org/stable/2999473

Engle, R. F., Russell, J. R. (1997). Forecasting the frequency of changes in quoted foreign exchange prices with the autoregressive conditional duration model. *Journal of Empirical Finance*, *4*(2), 187–212. https://doi.org/10.1016/S0927-5398(97)00006-6

Engle, R. F., Russell, J. R. (1998). Autoregressive conditional duration: A new model for irregularly spaced transaction data. *Econometrica*,

66(5), 1127–1162. Retrieved from http://www.jstor.org/stable/2999632

Engle, R., Patton, A. (2001). What good is a volatility model? *Quantitative Finance*, *1*(2), 237–245. https://doi.org/10.1088/1469-7688/1/2/305

Falkenberry, T. N. (2002). High frequency data filtering. *Technical Report. Tick Data, Inc.* Retrieved from https://s3-us-west-2.amazonaws.com/tick-data-s3/pdf/Tick_Data_Filtering_White_Paper.pdf

Felker, T., Mazalov, V., Watt, S. M. (2014). Distance-based high-frequency trading. *Procedia Computer Science*, *29*, 2055–2064. https://doi.org/10.1016/j.procs.2014.05.189

Fletcher, T., Shawe-Taylor, J. (2013). Multiple kernel learning with fisher kernels for high frequency currency prediction. *Computational Economics*, *42*(2), 217–240. https://doi.org/10.1007/s10614-012-9317-z

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189–1232. Retrieved from http://www.jstor.org/stable/2699986

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, *32*(200), 675–701. Retrieved from http://www.jstor.org/stable/2279372

Giot, P. (2001). Time transformations, intraday data, and volatility models. *Journal of Computational Finance*, *4*(2), 31–62. https://doi.

org/10.21314/JCF.2001.071

Gould, M. D., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J., Howison, S. D. (2013). Limit order books. *Quantitative Finance*, *13*(11), 1709–1742. https://doi.org/10.1080/14697688.2013.803148

Griffin, J. E., Oomen, R. C. A. (2008). Sampling returns for realized variance calculations: Tick time or transaction time? *Econometric Reviews*, *27*(1-3), 230–253. https://doi.org/10.1080/07474930701873341

Grolemund, G., Wickham, H. (2011). Dates and times made easy with lubridate. *Journal of Statistical Software*, *40*(3), 1–25. Retrieved from http://www.jstatsoft.org/v40/i03/

Guo, T., Bifet, A., Antulov-Fantulin, N. (2018). Bitcoin volatility forecasting with a glimpse into buy and sell orders. *2018 IEEE International Conference on Data Mining (ICDM)*, 989–994. https://doi.org/10.1109/ICDM.2018.00123

Gwilym, O. ap, Buckle, M., Thomas, S. H. (1997). The intraday behavior of bid-ask spreads, returns, and volatility for FTSE-100 Stock Index Options. *The Journal of Derivatives*, *4*(4), 20–32. https://doi.org/10.3905/jod.1997.407980

Hautsch, N. (2012). *Econometrics of financial high-frequency data* (pp. XIV, 374). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21925-2

Keogh, E., Chu, S., Hart, D., Pazzani, M. (2004). Segmenting time

series: A survey and novel approach. In *Data mining in time series databases* (pp. 1–21). https://doi.org/10.1142/9789812565402_0001

Kercheval, A. N., Zhang, Y. (2015). Modelling high-frequency limit order book dynamics with support vector machines. *Quantitative Finance*, *15*(8), 1315–1329. https://doi.org/10.1080/14697688.2015.1032546

Kuhn, M. (2020). *Caret: Classification and regression training*. Retrieved from https://CRAN.R-project.org/package=caret

Kuhn, M., Johnson, K., others. (2013). *Applied predictive modeling* (Vol. 26). Springer, New York, NY. https://doi.org/10.1007/978-1-4614-6849-3

Lemire, D. (2007). A better alternative to piecewise linear time series segmentation. In *Proceedings of the 2007 SIAM International Conference on Data Mining* (pp. 545–550). https://doi.org/10.1137/1.9781611972771.59

Liu, Y. (2019). Novel volatility forecasting using deep learning–Long short term memory recurrent neural networks. *Expert Systems with Applications*, *132*, 99–109. https://doi.org/10.1016/j.eswa.2019.04.038

Lovrić, M., Milanović, M., Stamenković, M. (2014). Algorithmic methods for segmentation of time series: An overview. *Journal of Contemporary Economic and Business Issues*, *1*(1), 31–53. Retrieved from https://journals.ukim.mk/index.php/jeccf/article/view/124

Müller, K. (2020). *Hms: Pretty time of day*. Retrieved from https:

//CRAN.R-project.org/package=hms

Nemenyi, P. (1963). *Distribution-free multiple comparisons* (PhD thesis). Princeton University.

Nousi, P., Tsantekidis, A., Passalis, N., Ntakaris, A., Kanniainen, J., Tefas, A., Gabbouj, M., Iosifidis, A.(2019). Machine learning for forecasting mid-price movements using limit order book data. *IEEE Access*, *7*, 64722–64736. https://doi.org/10.1109/ACCESS.2019.2916793

Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. *Journal of Forecasting*, *37*(8), 852–866. https://doi.org/10.1002/for.2543

Ntakaris, A., Mirone, G., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2019). Feature engineering for mid-price prediction with deep learning. *IEEE Access*, *7*, 82390–82412. https://doi.org/10.1109/ACCESS.2019.2924353

Pacurar, M. (2008). Autoregressive conditional duration models in finance: A survey of the theoretical and empirical literature. *Journal of Economic Surveys*, *22*(4), 711–751. https://doi.org/10.1111/j.1467-6419.2007.00547.x

Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2019). Deep temporal logistic bag-of-features for forecasting high frequency limit order book time series. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*,

7545–7549. https://doi.org/10.1109/ICASSP.2019.8682297

Peng, Y., Albuquerque, P. H. M., Camboim de Sá, J. M., Padula, A. J. A., Montenegro, M. R. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with support vector regression. *Expert Systems with Applications*, *97*, 177–192. https://doi.org/10.1016/j.eswa.2017.12.004

Perlin, M., Ramos, H. (2016). GetHFData: A R package for downloading and aggregating high frequency trading data from bovespa. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.2824058

Peterson, B. G., Carl, P. (2020). *PerformanceAnalytics: Econometric tools for performance and risk analysis*. Retrieved from https://CRAN.R-project.org/package=PerformanceAnalytics

Poon, S.-H., Granger, C. W. J. (2003). Forecasting volatility in financial markets: A review. *Journal of Economic Literature*, *41*(2), 478–539. Retrieved from http://www.jstor.org/stable/3216966

R Core Team. (2020). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

R Special Interest Group on Databases (R-SIG-DB), Wickham, H., Müller, K. (2019). *DBI: R database interface*. Retrieved from https://CRAN.R-project.org/package=DBI

Racicot, F.-É., Théoret, R., Coën, A. (2008). Forecasting irregularly

spaced UHF financial data: Realized volatility vs UHF-GARCH models. *International Advances in Economic Research*, *14*(1), 112–124. https://doi.org/10.1007/s11294-008-9134-2

Ramos-Pérez, E., Alonso-González, P. J., Núñez-Velázquez, J. J. (2019). Forecasting volatility with a stacked model based on a hybridized artificial neural network. *Expert Systems with Applications*, *129*, 1–9. https://doi.org/10.1016/j.eswa.2019.03.046

Russell, J. R., Engle, R. F. (2010). Chapter 7 - Analysis of high-frequency data. In *Handbook of financial econometrics: Tools and techniques* (Vol. 1, pp. 383–426). North-Holland. https://doi.org/10.1016/B978-0-444-50897-3.50010-9

Ryan, J. A., Ulrich, J. M. (2020). *Quantmod: Quantitative financial modelling framework*. Retrieved from https://CRAN.R-project.org/package=quantmod

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464. Retrieved from http://www.jstor.org/stable/2958889

Sheskin, David J (2000). Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC

Sirignano, J., Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, *19*(9), 1449–1459. https://doi.org/10.1080/14697688.2019.1622295

Terzi, E., Tsaparas, P. (2006). Efficient algorithms for sequence segmentation. In *Proceedings of the 2006 SIAM International Conference on Data Mining* (pp. 316–327). https://doi.org/10.1137/1.9781611972764.28

Tran, D. T., Magris, M., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2017). Tensor representation in high-frequency financial data for price change prediction. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1–7. https://doi.org/10.1109/SSCI.2017.8280812

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2017a). Forecasting stock prices from the limit order book using convolutional neural networks. *2017 IEEE 19th Conference on Business Informatics (CBI)*, *01*, 7–12. https://doi.org/10.1109/CBI.2017.23

Tsantekidis, A., Passalis, N., Tefas, A., Kanniainen, J., Gabbouj, M., Iosifidis, A. (2017b). Using deep learning to detect price change indications in financial markets. *2017 25th European Signal Processing Conference (EUSIPCO)*, 2511–2515. https://doi.org/10.23919/EUSIPCO.2017.8081663

Vaughan, D., Dancho, M. (2020). *Tibbletime: Time aware tibbles*. Retrieved from https://CRAN.R-project.org/package=tibbletime

Verousis, T., Gwilym, O. ap. (2010). An improved algorithm for cleaning ultra high-frequency data. *Journal of Derivatives & Hedge Funds*, *15*, 323–340. https://doi.org/10.1057/jdhf.2009.16

Violante, F., Laurent, S. (2012). Chapter 19 - Volatility forecasts eva-

luation and comparison. In *Handbook of volatility models and their applications* (pp. 465–486). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781118272039.ch19

West, K. D. (2006). Chapter 3 - Forecast evaluation. In *Handbook of economic forecasting* (Vol. 1, pp. 99–134). Elsevier. https://doi.org/10.1016/S1574-0706(05)01003-7

Wickham, H. (2011). The split-apply-combine strategy for data analysis. *Journal of Statistical Software*, *40*(1), 1–29. Retrieved from http://www.jstatsoft.org/v40/i01/

Wickham, H., Ruiz, E. (2020). *Dbplyr: A dplyr back end for databases*. Retrieved from https://CRAN.R-project.org/package=dbplyr

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T.L., Miller, E., Bache, S.M., Müller, K., Ooms, J., Robinson, D., Seidel, D.P., Spinu, V., Takahashi, K., Vaughan, D., Wilke, C., Woo, K., Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Xu, K., Gould, M. D., Howison, S. D. (2018). Multi-level order-flow imbalance in a limit order book. *Market Microstructure and Liquidity*, *04*(03n04), 1950011. https://doi.org/10.1142/S2382626619500114

Zar, Jerrold H (1999). Biostatistical Analysis. Pearson Education India

Zeileis, A., Grothendieck, G. (2005). Zoo: S3 infrastructure for regular

and irregular time series. *Journal of Statistical Software*, *14*(6), 1–27. https://doi.org/10.18637/jss.v014.i06

Zeileis, A., Kleiber, C., Krämer, W., Hornik, K. (2003). Testing and dating of structural changes in practice. *Computational Statistics & Data Analysis*, *44*(1), 109–123. https://doi.org/10.1016/S0167-9473(03)00030-6

Zeileis, A., Leisch, F., Hornik, K., Kleiber, C. (2002). Strucchange: An R package for testing for structural change in linear regression models. *Journal of Statistical Software*, *7*(2), 1–38. Retrieved from http://www.jstatsoft.org/v07/i02/

Zhang, Z., Zohren, S., Roberts, S. (2019). DeepLOB: Deep convolutional neural networks for limit order books. *IEEE Transactions on Signal Processing*, *67*(11), 3001–3012. https://doi.org/10.1109/TSP.2019.2907260