

Modelos híbridos para el pronostico de volatilidad y su uso en medidas de riesgo

Sebastian Velez Hernandez

Tesis de maestría

Trabajo de Grado presentado como requisito para optar al título de Magíster en Finanzas Cuantitativas

Universidad del Rosario

Supervisor: Carlos Castro Supervisor: Jonny Uribe

2023

Resumen

En el presente trabajo se implementan dos modelos híbridos los cuales combinan algoritmos de machine learning con los modelos econométricos clásicos para pronosticar la volatilidad del Ethereum. El primero utiliza estructuras de redes neuronales recurrentes junto con pronósticos de un GARCH(1,1), el segundo modelo aplica SVM para la estimación de los modelos GARCH. Se encuentra que ambos modelos se desempeñan mejor en el pronostico de la volatilidad que sus modelos base, adicionalmente se evalúa la eficiencia en la medición del riesgo mediante el *Value at Risk* en el cual solo el segundo modelo tuvo una mayor efectividad en la gestión del riesgo frente al modelo base.

Clasificación JEL: C40, C53, G17

Palabras clave: Ethereum; Pronóstico de volatilidad; Aprendizaje de maquina; Redes neuronales recurrentes; Máquina de vectores de soporte; Medición de riesgo

Abstract

In the present study, two hybrid models are implemented, which combine machine learning algorithms with classical econometric models to forecast Ethereum volatility. The first model utilizes recurrent neural network structures in conjunction with forecasts from a GARCH(1,1) model, while the second model applies SVM for the estimation of GARCH models. It is found that both models outperform their respective base models in volatility forecasting. Additionally, the efficiency in risk measurement is evaluated using *Value at Risk*, in which only the second model demonstrates greater effectiveness in risk measurement compared to the base model.

JEL Classification: C40, C53, G17

Keywords: Ethereum; Volatility forecast; Machine learning; Recurrent neural networks; Support vector machine; Risk measurement

Tabla de Contenidos

| Re | sumen | Ι |
|----|--|----|
| 1. | Introducción | 1 |
| 2. | Revisión de literatura | 3 |
| 3. | Metodología | 6 |
| | 3.1. Volatilidad Realizada | 6 |
| | 3.2. Modelos Econométricos | 6 |
| | 3.2.1. Modelo HAR | 6 |
| | 3.2.2. Modelos GARCH | 7 |
| | 3.3. Support Vector Machine | 7 |
| | 3.3.1. Support Vector Classifier | 7 |
| | 3.3.2. Support Vector Regression | 11 |
| | 3.4. Redes neuronales | 12 |
| | 3.4.1. Funciones de activación | 12 |
| | 3.4.2. Redes neuronales <i>feedforward</i> | 13 |
| | 3.4.3. Redes neuronales recurrentes | 15 |
| | 3.5. Value at Risk | 17 |
| 4. | Análisis empírico | 18 |
| | 4.1. Datos | 18 |
| | 4.2. Modelo HARV | 20 |
| | 4.3. Modelo GARCH-RNN | 22 |
| | 4.4. Modelo GARCH-SVM | 24 |
| | 4.5. Medida de riesgo | 25 |
| 5. | Resultados | 27 |
| | 5.1. GARCH-RNN | 27 |
| | 5.2. GARCH-SVR | 30 |
| 6. | Conclusión | 36 |
| Re | ferencias | 37 |

1. Introducción

La volatilidad es una medida de suma relevancia en los mercados financieros, ya que proporciona información sobre la incertidumbre y el riesgo asociados con un activo o un mercado en particular. Esta es importante porque puede afectar significativamente las decisiones de inversión y gestión de activos de los agentes del mercado, además de alterar la valoración de los instrumentos financieros e impactar la administración de riesgos financieros de todas las entidades del mercado. En consecuencia, la capacidad de pronosticar la volatilidad en un horizonte de tiempo futuro es crucial para los agentes de los mercados financieros, ya que pueden utilizar esta información para tomar decisiones informadas sobre sus inversiones.

Ante la importancia del pronóstico de la volatilidad, diversos autores se han dedicado a investigar cuáles son los métodos más efectivos para esta tarea. Desde el desarrollo e implementación del modelo ARCH Engle, 1982 y el modelo GARCH Bollerslev, 1986 de los años 80, los cuales han sido los métodos econométricos más usados para modelar y pronosticar la volatilidad de diferentes activos financieros, se han desarrollado nuevos modelos tomando los anteriores como base. Por ejemplo, se tienen los modelos GJR-GARCH, EGARCH, APARCH, TGARCH, entre otros, que incluyen cada vez diferentes hechos estilizados de las series.

Si bien estos modelos econométricos clásicos han sido bien comportados y bastante utilizados, dejan de lado algunos elementos que pueden incidir en el pronóstico, los cuales los nuevos y populares algoritmos de machine learning toman en cuenta y aplican. En el artículo de Bucci, 2020, se emplean diferentes estructuras de redes neuronales recurrentes, ya que estas son capaces de captar una relación no lineal en la volatilidad, además de poder incluir nueva información a través de variables de mercado. El autor encuentra que las diferentes arquitecturas superan a los modelos clásicos en el pronóstico. Por otro lado, la estimación clásica de los modelos de la familia GARCH se realiza por máxima verosimilitud, la cual necesita asumir una distribución de los datos. Chung y Zhang, 2017, por su parte, utilizan el algoritmo de support vector machine y, sin necesidad de asumir una distribución en particular, estiman los coeficientes de la ecuación de varianza del modelo GARCH, obteniendo mejores pronósticos que los obtenidos con la estimación tradicional.

Dados los buenos resultados de los algoritmos de machine learning en el pronóstico de la volatilidad de diversos activos, surgió un nuevo tipo de modelos que combinan los modelos econométricos clásicos de la familia GARCH con diferentes algoritmos de machine learning, como las redes neuronales, árboles de decisión y support vector machine, entre otros. La literatura relacionada con las aplicaciones de estos modelos empleando redes neuronales es amplia. Autores como Hajizadeh et al, 2012 pronostican la volatilidad del SP 500 empleando pronósticos de un EGARCH junto con una ANN, donde los nuevos pronósticos se desempeñan mejor que los generados por el EGARCH estimado. De manera similar, Kristjanpoller et al, 2014 utilizan una ANN junto con un GARCH(1,1) y algunas variables de mercado adicionales para pronosticar la volatilidad de los índices bursátiles de Brasil, Chile y México (IBOVESPA, SP IPSA y BMV IPC), encontrando que los modelos para los tres índices superaron a sus respectivos modelos base. Otro ejemplo es el trabajo de Kim y Won, 2018, quienes aplican este tipo de modelos al índice accionario KOSPI 200, utilizando la información de tres modelos econométricos como entrada de una nueva estructura de red LSTM, que, a diferencia de las ANN usadas anteriormente, captura la dependencia temporal de la serie de tiempo.

Otro de los algoritmos empleados en los modelos híbridos que se encuentra en la literatura es el support vector machine. Con este se estiman los coeficientes de la ecuación de varianza y se utilizan para el pronóstico. En los trabajos de Pérez-Cruz et al, 2003 y Chung y Zhang, 2017, se evalúa el desempeño de este nuevo modelo aplicado a índices accionarios y diversos pares de monedas. Ambos trabajos encuentran que la volatilidad pronosticada haciendo uso de SVM y diferentes kernels tiene un menor error de pronóstico que la volatilidad pronosticada mediante máxima verosimilitud en el modelo GARCH tradicional.

En vista del buen desempeño de los modelos híbridos en índices accionarios y monedas con alta liquidez, en el presente trabajo se van a entrenar y evaluar dos tipos de modelos híbridos para el pronóstico de la volatilidad de los retornos diarios del Ethereum. El primero utiliza un modelo GARCH y GJR-GARCH, variables exógenas del mercado de criptomonedas como entradas para dos tipos de redes neuronales recurrentes: una de tipo simple y otra con celular GRU. El segundo tipo de modelo híbrido emplea el support vector machine en la estimación de los coeficientes de la ecuación de varianza para un GARCH(1,1) y un GJR-GARCH(1,1,1). Ambos modelos tienen asociado un modelo econométrico de referencia con el cual comparar el desempeño del pronóstico del GARCH(1,1) y GJR-GARCH(1,1,1) usando máxima verosimilitud. Adicionalmente, se prueba la efectividad de estos en la medición del riesgo a través del VaR al 1%.

Se encuentra que la inclusión de pronósticos de los modelos clásicos en las estructuras de redes neuronales recurrentes tiene un menor error de pronóstico que su modelo de referencia HARV. De igual manera, la volatilidad generada por los coeficientes estimados con SVM tiene un mejor desempeño que sus modelos de referencia. En cuanto a la efectividad en la medición del riesgo empleando la medida del VaR, los modelos con redes neuronales y el modelo HARV se comportaron de la misma manera. Por otro lado, el modelo con SVM tiene una mejor estimación del riesgo de mercado del Ethereum.

2. Revisión de literatura

Las redes neuronales artificiales han sido empleadas últimamente para pronósticos de series de tiempo financieras, dado que estudios empíricos indican que son adecuadas para predecir variables volátiles y no lineales como lo son los retornos de las acciones o la volatilidad de las mismas (Yan y Ouyang, 2018, Sezer et al, 2020, Freeborough y van Zyl, 2022). Aprovechando estas propiedades Bucci, 2020 utiliza diferentes tipos de redes neuronales para pronosticar la volatilidad realizada del S&P 500; su propósito es capturar la relación no lineal entre la volatilidad y otras variables como los rendimientos de los tesoros de Estados Unidos, factores de riesgo y variables macroeconómicas a través de las redes FNN, JNN y LSTM y contrastar estas redes con modelos econométricos clásicos para la volatilidad realizada como los HARV. Mostró que las redes neuronales recurrentes como la JNN y LSTM superan a los modelos econométricos, adicionalmente con la LSTM se capturan dependencias de largo plazo permitiendo mejores pronósticos de la volatilidad realizada.

Aunque estudios como el de Bucci, 2020 muestran que los nuevos modelos de aprendizaje de maquina superan a los modelos clásicos, estos se siguen utilizando en la industria y la academia, gracias a que capturan el hecho de una volatilidad de los retornos cambiante en el tiempo y la existencia de clusters de volatilidad en estas series (Engle, 1982 & Bollerslev, 1986). Por lo que en vez de dejarlos a un lado se ha propuesto mezclar ambas metodologías para mejorar la calidad de los pronósticos de la volatilidad ya que la familia de modelos GARCH tiene buen desempeño en series relativamente estables.

Hajizadeh et al, 2012 toma esta idea y propone dos modelos híbridos basados en el modelo EGARCH y una ANN para el pronostico de la volatilidad del S&P 500, ambos modelos toman variables exógenas de mercado y variables endógenas como el nivel del índice y los retornos. El primer modelo enlaza ambas metodologías mediante el pronostico del EGARCH como entrada de la red neuronal y el segundo modelo usa simulaciones del EGARCH mas el pronostico de la volatilidad como entradas de la red, con el fin de capturar la estructura de auto correlación de los datos. Ambos modelos híbridos superan al EGARCH.

Estos modelos también se han realizado con otro tipo de activos, tal como lo hicieron (Kristjanpoller et al, 2014 & Kristjanpoller y Minutolo, 2016) pronosticando la volatilidad de tres índices bursátiles latinoamericanos y del petroleo, tomando como salida de entrenamiento la volatilidad realizada de 22 días, proponen una metodología similar a la anteriormente descrita, usando el pronostico de volatilidad de un GARCH(1,1) a 22 días, junto a otras variables explicativas respectivas referentes a cada activo como entradas a una red neuronal, además se adiciono un filtro para asegurar la positividad del pronostico. Las variables adicionales se agregaban a la red en función de su correlación con la volatilidad y el desempeño que tenia la red. Estos resultados se desempeñaron mejor que los modelos base y fueron robustos ante cambios en numero de neuronas y capas de la red.

Hasta el momento solo se ha incluido un solo modelo GARCH con las diferentes redes neuronales,

donde cada modelo de esta familia GARCH recoge características importantes de las series de tiempo financieras, como los efectos asimétricos, clusters de volatilidad, exceso de kurtosis entre otros. Ya que cada modelo aporta información de diferentes características Kim y Won, 2018 decidieron no usar solo uno sino usar varios modelos GARCH para lograr capturar dichas características y pronosticar la volatilidad del índice KOSPI 200. Aparte de este nuevo enfoque emplearon una red neuronal recurrente como la LSTM para tener en cuenta la dependencia temporal de la volatilidad, así pues crearon diferentes modelos combinando de uno a tres modelos GARCH con la red LSTM además de los modelos econométricos base y los modelos híbridos con un red neuronal profunda.

Siendo las entradas de las redes los parámetros estimados de los modelos GARCH adicionando un conjunto de variables exógenas como el precio del oro, la tasa de los títulos de Corea del Sur, el precio del petroleo y variables endógenas en relación al índice de 22 días de negociación anteriores, pronostican la volatilidad para el siguiente periodo. Concluyeron que el modelo con menores errores de pronostico medidos por el MAE, MSE, HMSE y HMAE era el que incluía la información de un GARCH(1,1), EGARCH y un EWMA pues aportaban la magnitud de los choques, la asimetría y los efectos de corto plazo.

Por otro lado Seo et al, 2019 propusieron un modelo híbrido para el pronostico de la volatilidad del S&P 500 similar a los anteriores, sin embargo, ellos incluyeron las Google domestic trends(GDT's) como entradas a su red neuronal, estas GDT's son el volumen de búsquedas en Google en temas económicos y financieros a nivel mundial, Seo, Lee y Kim las consideraron como información del interés del publico en factores macroeconómicos, concentrándose en temas de bancarrota, tarjetas de crédito, hipotecas e inversiones.

Su objetivo era predecir la volatilidad realizada en 5 y 21 días, esto con el fin de estudiar volatilidades semanales y mensuales asumiendo semanas de 5 días bursátiles y un mes de 21 días bursátiles, el pronostico se hacia mediante una ANN, teniendo como entradas las tendencias, y los pronósticos a 5 o 21 días de un modelo GARCH,EGARCH o GJR-GARCH. Finalmente para mejorar la precisión de los modelos los autores decidieron incluir el rezago de la volatilidad realizada entre $t-\tau, t$ calculada con los retornos diarios del índice, con $\tau = 5, 21$ y concluyeron que los mejores modelos para los dos horizontes de pronostico respectivamente eran GJR-GARCH(2,2)-GDT-LRV y GARCH(3,3)-GDT-LRV.

Otra de las técnicas de aprendizaje de maquinas empleada para mejorar los pronósticos de volatilidad es support vector machine, esta técnica fue empleada por Pérez-Cruz et al, 2003 con el fin de mejorar los pronósticos de un GARCH(1,1) para el S&P 100, FTSE100, IBEX35, NIKKEI, GM Y HP, la mejora se da en la estimación de la ecuación auxiliar de la varianza condicional, cuando generalmente se estima por máxima verosimilitud, en este caso se estima mediante SVM, empleando un kernel lineal. Concluyen que cuando las series de retornos no siguen una distribución gaussiana, la estimación por SVM es mejor puesto que no supone ninguna distribución sobre los datos, mientras que si efectivamente se sigue una distribución gaussina la estimación clásica de máxima verosimilitud es mejor.

Chung y Zhang, 2017 estudiaron la predicción de la volatilidad de los pares de monedas mas importantes del mundo, comparando el desempeño de SVM contra el GARCH(1,1), GJR-GARCH(1,1) y EGARCH(1,1) usando tres tipos de kernels, los modelos GARCH-SVM mostraron mejor predicción en cuatro de seis monedas con el kernel polinómico, cinco de seis monedas con el kernel lineal y seis de seis con el kernel radial.

Adicional al estudio de los pronósticos a través de modelos clásicos y modelos de aprendizaje de maquina, se puede estudiar la efectividad de los mismos en la gestión de riesgo, y compararlo con los retornos efectivos del activo, este análisis lo desarrollan Shen et al, 2021 sobre el Bitcoin usando la medida del *Value at Risk*, con tres modelos, un EWMA, un ARMA-GARCH y una red neuronal recurrente con células GRU.

En la precisión del pronostico de volatilidad para 1, 5 y 10 días adelante la RNN supera a los otros dos modelos haciendo uso da la información de 30 días anteriores.Por otro lado la efectividad de la estimación de la medida de riesgo, realizando las pruebas de cobertura incondicional y condicional de Kupiec y Christoffersen la RNN era superada por el modelo EWMA y GARCH, concluyendo que la red subestima las fluctuaciones en un periodo de precios mas volátil.

3. Metodología

En esta sección se encuentran los conceptos y modelos que se emplearan a lo largo del presente trabajo. Primero se encuentran los modelos econométricos, seguido del algoritmo support vector machine y las redes neuronales finalizando con la medida de riesgo *Value at Risk*

3.1. Volatilidad Realizada

Al ser la volatilidad un dato no observable, varios autores desarrollaron una metodología para medir la varianza de un activo financiero. Esta metodología es la varianza realizada, la cual es no paramétrica y usa datos de alta frecuencia de los precios de cotización de los activos para calcular la varianza, siendo desde hace un tiempo considerada como la medida expost adecuada para la evaluación de pronósticos.

El día de negociación se divide en m
 puntos, y se muestrean los precios en estos puntos, si $p_{i,t}$ la varianza realizada de m
 muestras es calculada de la siguiente manera

$$RV_t^{(m)} = \sum_{i=1}^m (p_{i,t} - p_{i-1,t}) = \sum_{i=1}^m r_{i,t}^2$$
(1)

Sheppard, 2010 muestra que $(RV_t^{(m)})$ es un estimador consistente para la varianza integrada del proceso del logaritmo de los precios. El muestreo de los precios se expresa cada n instante del tiempo, así pues se toman muestras cada 5 segundos, 10 segundos, 5 minutos, 15 minutos lo que conlleva a un numero de puntos m diferentes.

3.2. Modelos Econométricos

3.2.1. Modelo HAR

Al ser la varianza realizada observable, se puede modelar con herramientas de series de tiempo, Corsi, 2009 propuso un modelo simple conocido como *heterogeneus autoregression* el cual depende de los componentes diarios, semanales y mensuales de la varianza. La varianza realizada va a estar dada por

$$RV_t = \beta_0 + \beta_1 RV_{t-1}^d + \beta_2 RV_{t-1}^w + \beta_3 RV_{t-1}^m + u_t$$
(2)

con $RV_{t-1}^d = RV_{t-1}, RV_{t-1}^w = \frac{1}{5}\sum_{i=1}^5 RV_{t-i}, RV_{t-1}^m = \frac{1}{22}\sum_{i=1}^{22} RV_{t-i}$. Este modelo captura el alto grado de persistencia de la volatilidad y la dinámica de corto plazo Sheppard, 2010

3.2.2. Modelos GARCH

De forma general se puede escribir el proceso de retornos como

$$r_t = \mu_t + \varepsilon_t \tag{3}$$

donde μ_t es un proceso de media condicional y ε_t es un proceso de varianza condicional definido como $\varepsilon_t = \sigma_t \epsilon_t$. La ecuación auxiliar σ_t^2 es la que va a dar la caracterización del modelo.

GARCH

Bollerslev, 1986 propuso por primera vez el modelo GARCH, este es capaz de capturar los clusters de volatilidad que son comunes en las series de tiempo financieras. El GARCH(P,Q) es descrito mediante la ecuación auxiliar

$$\sigma_t^2 = \omega + \sum_{p=1}^P \alpha_p \varepsilon_{t-p}^2 + \sum_{q=1}^Q \beta_q \sigma_{t-q}^2$$

$$\epsilon_t \sim i.i.d(0,1)$$
(4)

Esta representación se puede comprender como un ARMA donde se toman Q rezagos de la varianza condicional y P rezagos del residuo.

GJR-GARCH

Glosten et al, 1993 propusieron este modelo como mejora al GARCH, porque si bien el GARCH tiene un buen desempeño, no toma en consideración los efectos asimétricos de choques negativos y positivos sobre la volatilidad de los activos financieros. La ecuación auxiliar es definida como

$$\sigma_t^2 = \omega + \sum_{p=1}^P \alpha_p \varepsilon_{t-p} + \sum_{o=1}^O \gamma_o \varepsilon_{t-o}^2 \mathbb{1}_{\varepsilon_{t-o<0}} + \sum_{q=1}^Q \beta_q \sigma_{t-q}^2$$

$$\epsilon_t \sim i.i.d(0,1)$$
(5)

teniendo los mismos términos del GARCH pero adicionando un coeficiente que captura la asimetría mencionada, siendo $\mathbb{1}_{\{\cdot\}}$ la función indicadora, al existir un choque negativo la volatilidad va a ser mas alta en γ frente a un choque positivo.

3.3. Support Vector Machine

3.3.1. Support Vector Classifier

Support Vector Classifier es una técnica de aprendizaje supervisado, desarrollada por Vapnik, 1995 junto a otros colaboradores cuyo objetivo es clasificar los datos, esta técnica se basa en

la premisa de que existe un hiperplano capaz de separar los datos en sus diferentes categorías. Supongamos que se tienen N duplas de la forma $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots (\boldsymbol{x}_N, y_N)\}$, donde $\boldsymbol{x}_i \in \mathbb{R}^p, y_i$ es la clasificación del dato, si se piensa de una forma binaria se tiene que $y_i \in \{-1, 1\}$. Van a existir dos casos, el primero cuando los datos son perfectamente separable y el otro es cuando esto no es posible.

Caso separable perfectamente

Este caso ocurre cuando existe un hiperplano que separa perfectamente los datos como se observa en la figura 3.1, como existen infinitos hiperplanos es necesario encontrar aquel que separa de la mejor manera la muestra en sus categorías. Se define el hiperplano

$$\left\{ \boldsymbol{x} : f(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\beta} + \beta_0 = 0 \right\}$$
(6)

de tal manera que la regla de clasificación viene dada por el signo de $x^T\beta + \beta_0$. Lo que se quiere es apartar las dos clases lo máximo posible de manera que use aquellos puntos frontera que son los mas cercanos al hiperplano y a su vez de los mas alejados de sus grupos, dichos puntos son llamados vectores de soporte.



Figura 3.1: Perfectamente separables

Luego el problema que se debe resolver es maximizar esta distancia (margen) entre el hiperplano y los vectores de soporte, de tal manera que los datos estén mas allá del margen:

$$\begin{array}{l} \max_{\substack{\beta_0,\beta,\|\beta\|=1}} & M \\ \text{s.t.} & y_i(\boldsymbol{x}_i^T\beta + \beta_0) \ge M \end{array} \tag{7}$$

El problema de maximización anterior se puede reescribir como



Figura 3.2: No perfectamente separables, fuente: Hastie et al, 2009

$$\begin{array}{l} \min_{\beta,\beta_0} & \|\beta\| \\ \text{s.t.} & y_i(\boldsymbol{x}_i^T\beta + \beta_0) \ge 1 \end{array}$$
(8)

notando que $M = \frac{1}{\|\beta\|}$, esto es un problema de optimización convexa.

Caso no separable perfectamente

Este caso se da cuando algunos puntos están del lado equivocado del hiperplano por lo que estarían mal clasificados, estos puntos tienen una distancia a la frontera de donde se supone deberían estar como se observa en la figura 3.2. La idea es la misma, maximizar esa distancia pero ahora teniendo en cuenta estos puntos que están del lado contrario de la frontera, el nuevo problema de optimización es

$$\begin{aligned}
& \underset{\beta,\beta_{0}}{\max} & \|\beta\| \\
& \text{s.t.} & y_{i}(\boldsymbol{x}_{i}^{T}\beta + \beta_{0}) \geq 1 - \xi_{i} \\
& \quad \xi_{i} > 0, \sum \xi_{i} < C
\end{aligned} \tag{9}$$

Las variables ξ_i es la proporción en términos de distancia por la cual la predicción esta mal clasificada, y la restricción hace referencia a cuantos puntos mal clasificados se van a permitir en el entrenamiento, el problema 9 suele reescribirse para que sea mejor computacionalmente

$$\begin{aligned}
&\min_{\boldsymbol{\beta},\boldsymbol{\beta}_{0}} \quad \frac{1}{2} \|\boldsymbol{\beta}\|^{2} + C \sum_{i=1}^{N} \xi_{i} \\
&\text{s.t.} \quad y_{i}(\boldsymbol{x}_{i}^{T}\boldsymbol{\beta} + \boldsymbol{\beta}_{0}) \geq 1 - \xi_{i} \\
& \quad \xi_{i} \geq 0
\end{aligned} \tag{10}$$

Este problema se soluciona mediante programación cuadrática, usando multiplicadores de lagrange para escribir el problema primario para posteriormente obtener el problema dual y maximizando el mismo incluyendo las condiciones de Karush–Kuhn–Tucker.

Uso del kernel



Figura 3.3: Ilustración uso del kernel para separar los datos, fuente: Bandgar, 2021

En algunos casos no es posible realizar una separación lineal óptima, por ende se emplea un kernel que aumenta la dimensión del espacio de características de tal manera que en esta nueva dimensión exista un hiperplano que separa los datos de forma óptima, generando una mejor separación de clases, estas nuevas fronteras lineales en el nuevo espacio de características, son fronteras no lineales en el espacio original de características, gráficamente se evidencia en la figura 3.3.

Se tiene un vector de nuevas características $h(\boldsymbol{x}) = (h(x_1), h(x_2), \dots, h(x_N))$, donde $h(\cdot)$ es una transformación no lineal a un espacio dimensional mayor tal que $\boldsymbol{x} \in \mathbb{R}^p, h(\boldsymbol{x}) \in \mathbb{R}^q, p < q$, este nuevo vector esta relacionado en el problema de optimización mediante un producto interior $\langle h(\boldsymbol{x}), h(\boldsymbol{x}') \rangle$, solo se necesita la función del kernel $K(\boldsymbol{x}, \boldsymbol{x}') = \langle h(\boldsymbol{x}), h(\boldsymbol{x}') \rangle$ mas no la transformación $h(\cdot)$. Los kernels mas populares para SVM son:



Figura 3.4

polinomio grado d :
$$K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \langle \boldsymbol{x}, \boldsymbol{x}' \rangle)^d$$

Función de base radial : $K(\boldsymbol{x}, \boldsymbol{x}') = exp(-\gamma ||\boldsymbol{x} - \boldsymbol{x}'||^2)$
Sigmoidal : $K(\boldsymbol{x}, \boldsymbol{x}') = tanh(\kappa_1 \langle \boldsymbol{x}, \boldsymbol{x}' \rangle + \kappa_2)$

3.3.2. Support Vector Regression

En el SVR suponemos que se tienen N duplas de la forma $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_N, y_N)\},$ donde $\boldsymbol{x}_i \in \mathbb{R}^p, y_i \in \mathbb{R}$, ahora la variable objetivo no es una clasificación sino un valor real que es generado por la función

$$y_i = f(\boldsymbol{x}_i) + e_i \tag{11}$$

donde $f(\boldsymbol{x})$ se puede aproximar con

$$f(\boldsymbol{x_i}) = h(\boldsymbol{x_i})^T \beta + \beta_0 \tag{12}$$

Se han propuesto dos medidas de error para aproximar esta función, la primera fue propuesta por Vapnik, 1995 y es una función de perdida lineal ϵ intensiva

$$L_{\epsilon} = \begin{cases} |y_i - f(\boldsymbol{x}_i)| - \epsilon, & |y_i - f(\boldsymbol{x}_i)| > \epsilon \\ 0, & e.o.c \end{cases}$$
(13)

esta función ignora errores inferiores a ϵ . Haciendo un símil con el problema de clasificación donde los puntos alejados de la frontera no pesaban en el proceso de optimización y contaban aquellos vectores soporte, para el problema de regresión excluimos los puntos con residuales bajos, solo los puntos ubicados por fuera o sobre el ϵ – tube (figura 3.4)

sirven como vectores de soporte para aproximar la función. Se introducen dos variables de holgura ξ_i, ξ_i^* para capturar la función de perdida, dejando el siguiente problema de optimización

$$\begin{aligned} \min_{\boldsymbol{\beta},\boldsymbol{\beta}_{0}} \quad \frac{1}{2} \|\boldsymbol{\beta}\|^{2} + C \sum_{i=1}^{N} \xi_{i} + \xi_{i}^{*} \\ \text{s.t.} \quad y_{i} - h(\boldsymbol{x}_{i})^{T} \boldsymbol{\beta} - \boldsymbol{\beta}_{0} \leq \epsilon + \xi_{i} \\ \quad h(\boldsymbol{x}_{i})^{T} \boldsymbol{\beta} + \boldsymbol{\beta}_{0} - y_{i} \leq \epsilon + \xi_{i}^{*} \\ \quad \xi_{i}, \xi_{i}^{*} \geq 0 \end{aligned} \tag{14}$$

Este problema se resuelve de forma similar a los anteriores, partiendo del problema primal, empleando las condiciones de Karush-Kunh-Tucker para llegar al problema dual y maximizar este ultimo obteniendo los pesos óptimos que aproximan $f(\mathbf{x})$

3.4. Redes neuronales

Las redes neuronales son modelos de machine learning implementados desde la década de los 80's, las cuales buscan replicar el funcionamiento del cerebro humano mediante neuronas que procesan información y generan una decisión. Estos modelos son altamente empleados, principalmente por tres características: la primera es su capacidad de capturar relaciones no lineales en los datos; no se necesita asumir ninguna distribución de probabilidad en el proceso; por último es un aproximador universal, lo cual quiere decir que mediante una red neuronal se puede aproximar cualquier función. Estas redes trabajan con neuronas que reciben y envían información, estas son organizadas en capas ocultas, partiendo de una información de entrada, transformada mediante funciones de activación hasta generar una salida. Existen diferentes tipos de arquitecturas de redes neuronales, con diferentes funciones de activación, numero de capas, neuronas y funcionalidades. A continuación se enseñaran las redes neuronales *feedforward* y las redes neuronales recurrentes y las funciones de activación que se pueden emplear.

3.4.1. Funciones de activación

Las funciones de activación neuronal, son las que, en cierta medida permiten capturar esa no linealidad tan deseada, puesto que si no se consideran, se estaría en un mundo completamente lineal. Las funciones de activación más comunes son:

• Función de activación sigmoide

$$f(x) = \frac{1}{1 + e^{-x}} \tag{15}$$



Figura 3.5: Red neuronal *feedforward* con 2 capas, fuente: Dixon et al, 2020

• Función de activación tangente hiperbólica

$$f(x) = tanh(x) \tag{16}$$

• Función de activación ReLU

$$f(x) = max\{0, x\} \tag{17}$$

3.4.2. Redes neuronales feedforward

Este tipo de redes, se caracterizan por tener un flujo de información hacia adelante, es decir, las características de entrada se pasan a la primera capa de neuronas, luego esta con una transformación es transmitida a la segunda capa de neuronas, y así sucesivamente hasta la ultima capa de neuronas, la cual pasa su resultado a la capa de salida. Matemáticamente esta estructura se puede expresar de la siguiente manera siguiendo a Dixon et al, 2020

$$Y = F_{W,b}(\boldsymbol{x}) + \epsilon \tag{18}$$

 \cos

$$F_{W,b}(\boldsymbol{x}) = f_{w^{(L)},b^{(L)}}^{(L)} \circ \cdots \circ f_{w^{(1)},b^{(1)}}^{(1)}(\boldsymbol{x})$$
(19)

donde se tienen L capas de neuronas en la red. Cada neurona esta conectada con otras neuronas en la capa siguiente y anterior a través de unos pesos y unas funciones $f^{(i)}$ las cuales son llamadas funciones de activación, que induce la transformación anteriormente mencionada transmitiendo la información. Cada capa cuenta con una matriz de pesos W y con un vector de sesgos b. Se puede evidenciar un ejemplo de esta estructura en la figura

3.5, siendo una red con 2 capas ocultas de neuronas, 6 elementos de entrada y 2 de salida. Detallando un poco el ejemplo de la primera capa (capa de entradas) a la segunda capa (primera capa de neuronas) se tiene

$$z_i^{(2)} = w_{i0}^{(1)} + \sum_{j=1}^p w_{ij}^{(1)} x_j$$
(20)

$$a_i^{(2)} = f^{(2)}(z_i^{(2)}) \tag{21}$$

La ecuación 20 representa el dato de entrada a la neurona i de la primera capa oculta de neuronas, que no es mas que una suma ponderada entre los pesos y los datos de entrada y la ecuación 21 representa el dato de salida de la neurona i, donde toma esta suma ponderada y le aplica una función de activación, ahora este dato es el que va a ser la entrada para las neuronas de la siguiente capa de modo que

$$z_i^{(3)} = w_{i0}^{(2)} + \sum_{j=1}^{p_2} w_{ij}^{(2)} a_j$$
(22)

$$a_i^{(3)} = f^{(3)}(z_i^{(3)}) \tag{23}$$

la ecuación 22 es la entrada de la neurona i en la segunda capa oculta, tomando en cuenta la salida de las neuronas de la capa anterior, luego aplicando otra función de activación se tiene el dato de salida para la neurona de esta capa (ecuación 23). De manera mas sencilla se puede expresar de forma matricial como

$$z^{(l)} = \boldsymbol{W}^{(l-1)} a^{(l-1)} \tag{24}$$

$$a^{(l)} = f^{(l)}(z^{(l)}) \tag{25}$$

Como otros modelos de machine learning, las redes neuronales también son dirigidas por datos, y se deben entrenar con los mismos. Este entrenamiento busca aprender las matrices de pesos \boldsymbol{W} y vectores de sesgo óptimos \boldsymbol{b} usando una función de perdida $L(y_i, F_{W,b}(\boldsymbol{x}))$ y un tipo de regularización, se resuelve el problema de optimización

$$\min_{\mathcal{W}} \quad \frac{1}{n} \sum_{i=1}^{n} L(y_i, F_{\mathcal{W}, b}(\boldsymbol{x}_i)) + \lambda J(\mathcal{W})$$
(26)

El proceso de optimización empieza con un conjunto de entrenamiento (\boldsymbol{x}, y) pasando a través de la red con unos pesos aleatorios iniciales entre capas, la red va generando los $a^{(l)}$

por capa hasta generar una salida y calcular la perdida. Con las perdidas se retrocede en la red para ir calibrando los pesos, mediante los gradientes, puesto que los pesos aportan información importante a la perdida de la red, de tal manera que se va reduciendo la perdida, a este algoritmo se le conoce como *back propagation*.

3.4.3. Redes neuronales recurrentes



Figura 3.6: Red neuronal recurrente con 3 rezagos, fuente: Dixon et al, 2020

Esta estructura de red se usa en un contexto de interdependencia temporal de los datos, esto además de las características mencionadas de las redes neuronales, logra capturar la dependencia temporal de las series, funcionando como una memoria interna, conservando la información importante en el tiempo. Estas redes, a demás de las entradas, aprovechan el conocimiento que queda en su memoria, conocido como estados ocultos, para generar una salida. Gráficamente esta estructura es como la figura 3.6 donde se tienen un numero de rezagos según la dependencia temporal y estos rezagos son entradas en la red y su salida alimenta la próxima capa con el siguiente rezago, hasta generar la salida.

Si se tiene un conjunto de datos $\{x_t, y_t\}_{t=1}^N$ que están autocorrelacionados, x_t es una serie temporal que va a predecir y_t en una red neuronal recurrente, los pesos de la red van a ser compartidos entre los elementos de entrada debido a esa relación temporal, de lo contrario, si fuesen diferentes se tiene una red neuronal normal. Esta estructura se conoce como red neuronal recurrente simple y sufre de un problema de memoria a corto plazo, debido a esto la información de momentos de tiempo lejano se va perdiendo durante la implementación de *back propagation*.

Cho et al, 2014 propusieron un nuevo tipo de red, la estructura Gated Recurrent Unit (GRU) que soluciona el problema de la perdida de información, esta estructura funciona mediante unidades que contienen 2 puertas, updated gate y reset gate, esta red puede conservar información de largo tiempo atrás, además de poder olvidar información irrelevante. La update gate recibe los datos de entrada y el estado oculto del momento de tiempo anterior, con esta información se decide que información va a pasar al siguiente instante del tiempo, matemáticamente esta relación se expresa

$$z_t = \sigma \left(W^{(z)} x_t + U^{(z)} h_{t-1} \right)$$
(27)

donde W, U son las matrices de pesos respectivas, x son las entradas de la unidad y h es el estado oculto o memoria que traía la red de la unidad anterior. La reset gate recibe los mismos datos pero decide que información va a olvidar, esto se logra usando diferentes pesos para la entrada y el estado oculto.



$$r_t = \sigma \left(W^{(r)} x_t + U^{(r)} h_{t-1} \right)$$
(28)

Luego de filtrar la información con las puertas, se tiene una primera instancia de memoria actual, tomando la información relevante obtenida de reset gate (r_t) , junto la entrada y el estado oculto.

$$h'_t = tanh\left(Wx_t + r_t \odot Uh_{t-1}\right) \tag{29}$$

Finalmente utilizando el aporte de update gate (z_t) y la primera instancia de memoria (h'_t) se obtiene la salida de la unidad GRU que es el siguiente estado oculto de entrada a la próxima unidad

$$h_t = z_t \odot h_{t-1} + (1 - z_t)_t' \tag{30}$$

La red GRU puede almacenar y filtrar información gracias a sus puertas y transmitirla gracias a sus estados ocultos a los siguientes periodos.

3.5. Value at Risk

El Value at Risk es una medida de riesgo bastante usada en la medición y gestión del riesgo de mercado, la cual determina la perdida máxima que se puede tener en un horizonte de tiempo determinado con un nivel de significancia α determinada, este se define como

$$VaR_{\alpha} = \inf \left\{ x \in : F_r \ge \alpha \right\}$$
(31)

De modo que el Value at Risk se puede ver como el α -quantil de la distribución de los retornos F_r , este cuantil se puede obtener de forma parametrica o no parametrica.

Al asumir una distribución puntual de los retornos se tiene la definición del VaR paramétrico, que en el caso de una distribución normal $F_r \sim \mathcal{N}(\mu, \sigma^2)$ el VaR viene dado por:

$$VaR_{\alpha} = \mu + \sigma \Phi^{-1}(\alpha) \tag{32}$$

donde Φ^{-1} es el quantil de una distribución normal estándar, en el caso que $F_r \sim t_{\nu}(\mu, \sigma^2)$ el VaR viene dado por:

$$VaR_{\alpha} = \mu + \sigma t_{\nu}^{-1}(\alpha) \tag{33}$$

donde t_{ν}^{-1} es el quantil de una distribución t-student con ν grados de libertad.

4. Análisis empírico

En el presente trabajo se van a aplicar dos modelos híbridos sobre la criptomoneda Ethereum para pronosticar la volatilidad de este activo financiero, evaluando el desempeño de los mismos y realizando un backtesting sobre el Value at Risk generado a partir de estos pronósticos.

4.1. Datos

Se toma la serie de precios de Ethereum desde el 10 de agosto 2015 hasta el 21 de febrero de 2022 en periodicidad diaria, también se construyo la serie de volatilidad realizada usando precios intradia con periodicidad de 15 minutos, empleando la metodología descrita en 3.1, estas series fueron extraídas y construidas en la plataforma para la investigación en blockchain Dune Analytics. Adicionalmente las variables exógenas Bitcoin y la tasa de cambio se extrageron de Yahoo Finance y la pagina del Banco Central Europeo respectivamente para el mismo periodo con la misma frecuencia.

En la figura 4.1 se tiene el precio y los log-retornos diario del Ethereum, se evidencia un cambio significativo en el 2021 donde el Ethereum entra en una tendencia alsista fuerte, seguida de una caída de gran magnitud, pasando de niveles superiores a los 4 mil dolares a precios al rededor de 2 mil dolares. Se evidencia una amplitud en los retornos diarios, con varios clusters y valores altos para retornos de esta periodicidad. Es por las características y comportamientos de las criptomonedas que este activo se hace llamativo para un análisis de riesgo y volatilidad.



Figura 4.1: Precio y retornos diarios Ethereum

El log-retorno promedio es de un 0.344% con una asimetría negativa y una kurtosis bastante elevada, características comunes para datos de alta frecuencia de las criptomonedas, también se presentan valores extremos con un log-retorno máximo de 41,01% y un mínimo de -58,69%. Además analizando la dependencia temporal de la serie de retorno y retornos al cuadrado, se observa que para los retornos al cuadrado tanto para 5 como 20 rezagos existe una correlación, adicionalmente el coeficiente de un modelo AR(1) es significativo al 1\%, mientras que para la serie de retornos no es significativo, en 5 rezagos no existe correlación y se tiene un ACF bajo, estas estadísticas descriptivas se detallan en la tabla 4.1.

Retornos diarios ETH

| Media | 0.34% | | r_t | r_t^2 |
|-----------|---------|--------|----------|----------|
| Desv. | 6.05% | ACF(1) | 0.01465 | 0.15231 |
| Asimetira | -0.048 | LB(5) | 9.112982 | 189.9303 |
| Kurtosis | 11.536 | LB(20) | 48.9713 | 241.0803 |
| Min | 41.01% | AR(1) | 0.0178 | 0.2264 |
| Max | -58.69% | | | |

Cuadro 4.1: Estadísticas preliminares

Con estas estadísticas preliminares se puede intuir que los retornos diarios del Ethereum no se distribuyen normales, puesto que una kurtosis alta y una asimetría negativa, esto se confirma empleando una prueba Jarque-Bera y una prueba Shapiro-Wilk donde ambas concluyen que existe evidencia estadística para rechazar una distribución normal de los datos, adicionando la evidencia del QQ plot 4.3 sobre una distribución normal, donde en los cuantiles extremos se puede observar como se alejan de los normales teóricos afirmando el hecho de colas pesadas que poseen los retornos financieros, y en especial el de los cripto activos. Por último se evidencia también la dependencia temporal de los retornos cuadrados como proxy de la volatilidad de los retornos con un coeficiente AR(1) significativo al igual que la correlación del ACF(1).

La serie calculada de volatilidad realizada anualizada, tomando intervalos de 15 minutos, se muestra en la figura 4.2, evidenciando una volatilidad alta durante el intervalo de tiempo estudiado, con picos elevados a excepción del periodo 2019-2020.



Figura 4.2: RV frecuencia de muestra 15 minutos



Figura 4.3: Q-Q plot retornos diarios ethereum

4.2. Modelo HARV

Como modelo de referencia se tomara el HARV mencionado en la sección 3.2.1, ya que es un modelo sencillo y ha demostrado una buena modelación de la volatilidad realizada. Este se estimo con datos hasta el 10 de agosto de 2021 mediante mínimos cuadrados ordinarios, los coeficientes obtenidos se muestran en la tabla 4.2, los cuatro coeficientes son estadísticamente significativos.

| | Coeficiente | Error estándar | \mathbf{t} | P-valor |
|--------------|-------------|----------------|--------------|---------|
| β_0 | 0.0050 | 0.001 | 5.148 | 0.000 |
| RV_{t-1}^d | 0.4757 | 0.024 | 19.482 | 0.000 |
| RV_{t-1}^w | 0.2547 | 0.037 | 6.956 | 0.000 |
| RV_{t-1}^m | 0.1428 | 0.032 | 4.428 | 0.000 |

Cuadro 4.2: Coeficientes estimados modelo HAR

Una vez estimado se pronostico los datos del 11 de agosto de 2021 al 21 de febrero de 2022, los resultados se pueden observar en la figura 4.4, donde se evidencia una buena predicción del modelo HARV estimado frente a la volatilidad realizada. Para evaluar el desempeño de los modelos se van a emplear dos medidas de error:

$$MSE = \frac{1}{T - T_1} \sum_{t=T_1}^{T} \left(\sigma_t^2 - \hat{\sigma}_{t|t-1}^2 \right)^2$$
(34)

$$QL = \frac{1}{T - T_1} \sum_{t=T_1}^{T} \frac{\sigma_t^2}{\hat{\sigma}_{t|t-1}^2} - \ln\left(\frac{\sigma_t^2}{\hat{\sigma}_{t|t-1}^2}\right) - 1$$
(35)

Adicional al error cuadrático medio (MSE) que es una medida de error bastante usada, se calcula la función QL de Patton, 2011 la cual es una función de perdida robusta ante el pronostico de la volatilidad. Para los pronósticos realizados estos son los valores de cada función de perdida, MSE = 0,0005, QL = 0,0778, estos serán los valores de referencia a la hora de evaluar el desempeño de los modelos híbridos a continuación.



Figura 4.4: Pronostico HAR comparado con la volatilidad realizada

4.3. Modelo GARCH-RNN

El primer modelo híbrido propuesto es el GARCH-RNN, como se mencionó en la sección 2 varios papers destacan el desempeño de ANN en el pronostico de las series financieras por la capacidad de capturar las relaciones no lineales y no tener que asumir una distribución de probabilidad puntual, pero estas no tienen en cuenta la dependencia intertemporal de la serie, por lo que se usara una red neuronal recurrente que toma esta relación, en primera instancia se puede estimar una red neuronal únicamente con la serie de volatilidad realizada del activo, sin embargo dado el propósito de la investigación de combinar las ventajas de los nuevos modelos de aprendizaje de maquina y los modelos econométricos clásicos para el pronostico de $\hat{\sigma}_{t+1}$, se van a tomar los pronósticos de volatilidad $\hat{\sigma}_{t+1}$ de los modelo GARCH(1,1) y/o GJR-GARCH(1,1) como entradas adicionales a la red.

Es así como se llega al primer grupo de modelos híbridos, los cuales se van a denominar; GARCH VOL, GJR VOL y GJR GARCH VOL, estos modelos son una red neuronal recurrente que tiene por entradas la serie de volatilidad realizada, y el o los pronósticos de los modelos econométricos GARCH(1,1) y GJR-GARCH(1,1). Por otra parte, siguiendo las ideas de Kim y Won, 2018,Bucci, 2020 y Hajizadeh et al, 2012 en donde se usaron otras variables como entradas de la red, se añadirán variables exógenas que inciden en la volatilidad del Ethereum, se han realizado estudios sobre cuales son las variables que impactan el comportamiento de las criptomonedas, así como su magnitud, uno de estos es el articulo de Angela y Sun, 2020 quienes analizan la incidencia de la tasa de cambio EUR/USD, el precio del oro y otras criptomonedas en el precio del Ethereum, encuentran que la tasa de cambio y el precio del Bitcoin tienen una relación con los niveles de Ethereum, mientras que el precio del oro no tiene relación alguna.

Adicionalmente, Dempere, 2019 estudia el poder predicativo sobre el retorno y la volatilidad del Bitcoin, Ethereum y Ripple que tiene algunas variables financieras, a través de modelos GARCH, concluye que para el Ethereum se tiene una relación positiva con los log-retornos de las otras dos criptomonedas mientras que otras variables como el oro, petroleo y las tasas de cambio DEG no tienen ninguna relación. Se seleccionan como variables exógenas la tasa de cambio euro dolar EUR_t , el precio del bitcoin BTC, sus log-retornos $r_{b,t}$ y sus log-retornos al cuadrado $r_{b,t}^2$ como entradas adicionales a los modelos anteriormente mencionados, lo cual genera otro grupo de modelo que contienen los pronosticos de los GARCH, la volatilidad realizada y el conjunto de variables exogenas seleccionado, esos se van a denominar; GARCH EXO, GJR EXO y GJR GARCH EXO.

De igual manera se puede adicionar a la red variables endogenas como lo hacen Hajizadeh et al, 2012, en este caso se consideran los log-retornos $r_{e,t}$ y los log-retornos al cuadrado $r_{e,t}^2$ del Ethereum, que junto a la serie de volatilidad realizada permite establecer un tercer grupo de modelos; GARCH ENDO, GJR ENDO y GJR GARCH ENDO. Por ultimo juntando todas las variables a la red que se han mencionado, la serie de volatilidad, las



Figura 4.5: Estructura variables de entrada

variables exógenas y endógenas y los pronósticos de los modelos GARCH se tiene el ultimo grupo de modelos a evaluar, denominados; GARCH FULL, GJR FULL y GJR GARCH FULL.

Se utilizaran dos tipos de estructuras para la red neuronal, la primera, una red neuronal recurrente simple, y la segunda una Gated Recurrent Unit que selecciona la información pertinente para almacenar y seguir, ambas tendrán como entradas las variables anteriormente mencionadas según el grupo de modelos correspondiente. Estos modelos se entrenaran con un conjunto de datos desde el 10 de agosto de 2016 al 10 de agosto de 2021, siendo 5 años de datos diarios para cada una de las variables, los modelos GARCH(1,1) y GJR-GARCH(1,1,1) se estiman mediante una ventana móvil, siendo la primera ventada del 10 de agosto de 2015 a el 10 de agosto de 2016, donde esta va aumentando en un día, una vez estimado el modelo en la ventana se pronostica la volatilidad del Ethereum a uno, dos, tres y cuatro días con el ultimo dato de la misma, de tal manera que para cada día en las fechas de entrenamiento de los modelos GARCH se tienen 4 pronósticos. Si la red por ejemplo va a pronosticar la volatilidad para el primero de agosto de 2021, se deben usar datos del 27, 28, 29 y 30 de septiembre, el pronostico de los modelos econométricos que debe entrar en la serie del día 28, es el generado dada la información hasta ese día con un horizonte de 3 días, al pasar a pronosticar el 2 de agosto se toma el pronostico de los modelos econométricos generados con la información hasta el 28 pero ahora con un horizonte de 4 días, de tal manera que para cada tiempo t se tienen los pronósticos $\hat{\sigma}_{t+1|t}, \hat{\sigma}_{t+1|t-1}, \hat{\sigma}_{t+1|t-2}, \hat{\sigma}_{t+1|t-3}$, la estructura de variables de entrada se puede observar en la figura 4.5.

Posteriormente se realiza la estandarización de las variables de entrada y se define la estructura de la red con una capa de entrada, una capa oculta GRU o RNN simple y una

capa de salida, para afinar el numero de neuronas y el parámetro de regularización a usar en las redes se emplea un 5-fold validation en un grid search de tal manera que se escoge la mejor mejor combinación de los parámetros

4.4. Modelo GARCH-SVM

El segundo tipo de modelos híbridos propuesto es el GARCH-SVM, este consiste en estimar la ecuación auxiliar de un modelo GARCH(1,1) y GJR-GARCH(1,1) con proceso de media $\mu_t = 0$ mediante el algoritmo support vector regresión explicado previamente en 3.3.2. Para esto se tiene un modelo GARCH(1,1) de la forma

$$r_t = \varepsilon_t = \sigma_t e_t$$

$$\sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2$$

$$e_t \sim i.i.d(0, 1)$$

donde los coeficientes de la ecuación de la varianza ω, α, β son generalmente estimados mediante el método de máxima verosimilitud, bajo el supuesto de una distribución dada de los retornos. Cuando el componente aleatorio e_t se distribuye de manera normal estandar, los estimadores resultantes son estimadores de máxima verosimilitud, pero cuando este no sigue una distribución normal, son estimadores de quasi-maxima verosimilitud como lo muestran Bollerslev y Wooldridge, 1992. Como es usual en las series de tiempo financieras, estas presentan un valor alto de kurtosis y una asimetría, mas aun no se ajustan a una distribución normal. Lo que propone el modelo híbrido, como se menciono es emplear SVR para estimar la ecuación de la varianza del GARCH, teniendo en cuenta que esta técnica no necesita asumir una distribución particular sino ajustar la ecuación con los datos, lo cual puede generar una mejor estimación de los coeficientes. Se asume como estimador de la volatilidad, la volatilidad realizada calculada anteriormente, la ecuación a estimar para el modelo GARCH(1,1) mediante SVR es:

$$RV_t^2 = \omega + \alpha r_t^2 + \beta R V_{t-1}^2 + u_t \tag{36}$$

adicionando el termino $u_t \sim (0, \sigma_u^2)$ al ser una ecuación a estimar mas no una identidad como lo es directamente en el modelo, para el modelo GJR-GARCH(1,1)

$$r_t = \varepsilon_t = \sigma_t e_t$$

$$\sigma_t^2 = \omega + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2 + \gamma \varepsilon_{t-1}^2 \mathbb{1}_{\varepsilon_{t-1} < 0}$$

$$e_t \sim i.i.d(0, 1)$$

se estiman los parámetros $(\hat{\omega}, \hat{\alpha}, \hat{\beta}, \hat{\gamma})$ a partir de la ecuación

$$RV_t^2 = \omega + \alpha r_t^2 + \beta RV_{t-1}^2 + \gamma r_t^2 \mathbb{1}_{r_{t-1} < 0} + u_t$$
(37)

Los parámetros de las ecuación auxiliares, se estiman con un conjunto de entrenamiento, que comprende los retornos y la volatilidad realizada entre el 7 de agosto de 2015 y el 10 de agosto de 2021. Para la selección de los hiperparámetros ϵ y C en el proceso de estimación se utiliza un 5-fold validation en un grid search de manera que se escogen los hiperparametros que generan un menor error, con estos y haciendo uso de los kernel lineal y kernel RBF se realiza la estimación usando SVR y NuSVR. Una vez se tienen los parámetros ($\hat{\omega}, \hat{\alpha}, \hat{\beta}$) para el GARCH(1,1) y ($\hat{\omega}, \hat{\alpha}, \hat{\beta}, \hat{\gamma}$) para el GJR-GARCH(1,1), se procede a realizar el pronostico de un periodo adelante de la volatilidad de los retornos diarios del Ethereum, para el GARCH(1,1) este pronostico $\hat{\sigma}_{t|t-1}^2$ es de la forma

$$\mathbb{E}[\sigma_t^2 | \mathcal{F}_{t-1}] = \hat{\omega} + \hat{\alpha} \varepsilon_{t-1}^2 + \hat{\beta} \sigma_{t-1}^2$$
(38)

y para el modelo GJR-GARCH(1,1) es

$$\mathbb{E}[\sigma_t^2 | \mathcal{F}_{t-1}] = \hat{\omega} + \hat{\alpha} \varepsilon_{t-1}^2 + \hat{\gamma} \varepsilon_{t-1}^2 \mathbb{1}_{\varepsilon_{t-1} < 0} + \hat{\beta} \sigma_{t-1}^2$$
(39)

estos pronósticos se generan a través de una rolling window dejando los parámetros estimados por SVR constantes pero utilizando el ultimo retorno de la ventana, de aquí se obtiene una serie $\{\hat{\sigma}_t^2\}$ que va desde el 11 de agosto de 2021 al 20 de febrero de 2022

4.5. Medida de riesgo

La estimación paramétrica del VaR en el momento t es definida de la siguiente forma

$$\mathbb{P}\left(r_t \le VaR_t(\alpha) | \mathcal{F}_{t-1}\right) = \alpha \tag{40}$$

donde r_t es el retorno diario del Ethereum, \mathcal{F}_{t-1} es la σ -algebra del dia anterior y α es el nivel de significancia dado que sera 1%, 2.5% o 5%. Para realizar la estimación paramétrica es necesario asumir una distribución de probabilidad de los retornos diarios del activo, esta generalmente es asumida como normal pero como se evidencio en la sección

4.1 se rechaza la hipótesis de normalidad para los retornos diarios del Ethereum, por lo cual se escoge una distribución t-student como función de distribución de los datos, pues se adapta a los retornos, sobre todo por sus colas pesadas observadas en el exceso de kurtosis tabla 4.1. Determinada la distribución de los retorno como un t-student el VaR al α nivel de significancia esta dado por la siguiente ecuación:

$$VaR_t(\alpha) = \mathbb{E}\left(r_t | \mathcal{F}_{t-1}\right) + t_{\alpha}^{\nu} \sigma_t \sqrt{\frac{\nu}{\nu - 2}}$$
(41)

siendo σ_t el pronostico de la volatilidad para el periodo t dado el conjunto de información \mathcal{F}_{t-1} para el día anterior y t^{ν}_{α} es el valor critico de una t-student con ν grados de libertad a un nivel de significancia α . Dado que en los modelos GARCH-RNN, GARCH-SVR y HAR se esta asumiendo una media condicional $\mu_t = 0$ la ecuación del VaR a utilizar es:

$$VaR_t(\alpha) = t^{\nu}_{\alpha}\sigma_t \sqrt{\frac{\nu}{\nu - 2}}$$
(42)

Los grados de libertad se estiman siguiendo a Andreev y Kanto, 2004 en donde se permite que el parámetro no se entero y mediante el método de momentos se tiene que un estimador consistente es

$$\hat{\nu} = 4 + \frac{6}{\hat{k}} \tag{43}$$

Los retornos diarios del Ethere
um tienen un exceso de kurtosis $\hat{k} = 8,53$, luego los grados de libertad estimados so
n $\hat{\nu} = 4,70$. Los valores críticos de una t-student con 4.7 grados de libertad son -3.46,-2.62 y -2.04 a nivel 1 %, 2.5 % o 5 % respectivamente. Aplicando la ecuación 42 a los pronósticos fuera de la muestra de la volatilidad generados por los modelos, se tiene una estimación del VaR para el periodo prueba. Para realizar la evaluación de los modelos en la estimación del VaR se va a tomar en cuenta la siguiente medida

$$\hat{\alpha} = \frac{\#(r_t < VaR_t(\alpha))}{\# \text{ observaciones}}$$
(44)

la cual cuantifica el numero de fallos de la medida de riesgo ocurridos durante el periodo de pronostico, para un VaR al 1% se espera que la medida falle un 1% de las veces, con un VaR al 5% se espera que los retornos en la ventana de prueba sobrepasen la medida un 5% de las veces, entre más cerca esta $\hat{\alpha}$ de α más preciso es el VaR calculado.

5. Resultados

5.1. GARCH-RNN

Una vez se estimaron los diferentes grupos de modelos con los datos de entrenamiento, se generaron pronósticos fuera de la muestra de la volatilidad realizada del Ethereum para el periodo del 14 de agosto de 2021 al 10 de febrero de 2022, se pueden comparar estos pronósticos con la verdadera volatilidad realizada en la gráfica 5.1.



Pronosticos mejores modelos

Figura 5.1: Pronósticos mejores modelos comparados con la volatilidad realizada

En función de la figura anterior se observa que de forma general los pronósticos de los tres modelos siguen de buena manera a la volatilidad realizada, en ciertos puntos en el tiempo donde se presentan grandes saltos en la volatilidad realizada, los modelos nos alcanzan a responder a este cambio, si bien se ve un buen comportamiento es necesario cuantificar el error de pronostico para evaluar los modelos. Por esto a los pronósticos se les calculo la medida de error QL, sus evaluaciones se encuentran en la tabla 5.1 donde se observa en negrilla los modelos con menor perdida, donde el modelo con menor error es el GJR VOL, que tiene como variables de entrada los pronósticos del GJR-GARCH(1,1,1) y la volatilidad realizada, seguido por el GARCH VOL y el GJR GARCH VOL todos en su versión GRU con la función de activación tanh.

| | RNN | | GF | RU |
|----------------|---------|---------|---------|---------|
| | anh | sigmoid | \tanh | sigmoid |
| GARCH ENDO | 0.02737 | 0.02587 | 0.03183 | 0.02582 |
| GARCH EXO | 0.03428 | 0.03952 | 0.02772 | 0.04047 |
| GARCH FULL | 0.03973 | 0.04696 | 0.03025 | 0.05753 |
| GARCH VOL | 0.02142 | 0.02133 | 0.01905 | 0.02034 |
| GJR ENDO | 0.03261 | 0.02587 | 0.03182 | 0.02582 |
| GJR EXO | 0.0487 | 0.04584 | 0.04988 | 0.04835 |
| GJR FULL | 0.05794 | 0.04705 | 0.04135 | 0.05752 |
| GJR GARCH ENDO | 0.03583 | 0.02374 | 0.03169 | 0.02604 |
| GJR GARCH EXO | 0.03315 | 0.03418 | 0.03298 | 0.04925 |
| GJR GARCH FULL | 0.03695 | 0.06411 | 0.03789 | 0.03508 |
| GJR GARCH VOL | 0.01939 | 0.02197 | 0.01938 | 0.01992 |
| GJR VOL | 0.02141 | 0.02052 | 0.01904 | 0.02034 |

Cuadro 5.1: Medida de error QL de los pronósticos fuera de la muestra de la volatilidad realizada

Mientras el modelo con mayor error es el GJR GARCH FULL, el cual toma los pronósticos de los dos modelos econométricos y todas las variables exógenas y endógenas mencionadas en la sección 4.3. Además se cuantifica la medida del error cuadrático medio de los 48 modelos, estos errores se encuentran en la tabla 5.2, de igual manera bajo esta otra medida de error se tiene que los tres mejores modelos son el GJR VOL, GARCH VOL y GJR GARCH VOL con la estructura de red GRU y la función de activación tanh, mientras el peor modelo es el GJR GARCH FULL con la red neuronal simple, seguido del GJR EXO con la estructura GRU.

| | RNN | | GR | U |
|----------------|-------------|-------------|-------------|-------------|
| | tanh | sigmoid | tanh | sigmoid |
| GARCH ENDO | 0.000400701 | 0.000387746 | 0.000431219 | 0.000364724 |
| GARCH EXO | 0.000387114 | 0.000413767 | 0.000329695 | 0.000418323 |
| GARCH FULL | 0.000421969 | 0.000441051 | 0.000301979 | 0.000529245 |
| GARCH VOL | 0.00033566 | 0.000312666 | 0.000271987 | 0.000281921 |
| GJR ENDO | 0.000389966 | 0.000387729 | 0.000431131 | 0.00036471 |
| GJR EXO | 0.000497516 | 0.000392963 | 0.000570571 | 0.0004097 |
| GJR FULL | 0.000446482 | 0.000400654 | 0.000413561 | 0.000529228 |
| GJR GARCH ENDO | 0.000525747 | 0.000373305 | 0.000402408 | 0.000384529 |
| GJR GARCH EXO | 0.000389189 | 0.000368618 | 0.000360749 | 0.0004328 |
| GJR GARCH FULL | 0.000519252 | 0.000584213 | 0.000374529 | 0.000347336 |
| GJR GARCH VOL | 0.000307474 | 0.000318044 | 0.000272447 | 0.000278665 |
| GJR VOL | 0.000335625 | 0.000307999 | 0.000271904 | 0.000281916 |

Cuadro 5.2: Mean Square Error de los pronósticos fuera de la muestra de la volatilidad realizada

Ambas medidas de error coinciden en el orden de los tres mejores modelos, de manera que se centrara el análisis en estos tres. Comparando las resultados de las medidas de error del modelo HAR estimado anteriormente QL: 0,0778~MSE: 0,0005 con las obtenidas por los tres modelos (GJR VOL, GARCH VOL y GJR GARCH VOL), se evidencia una mejora en el pronostico frente al modelo de referencia de mas del 70 %.

Ahora para identificar estadísticamente si existe diferencia alguna entre los pronósticos generados por los modelos, se emplea la prueba de Diebold-Mariano la cual tiene por hipótesis nula que los dos modelos probados tienen la misma capacidad predicativa, esta prueba se empleo para las diferentes combinaciones entre el modelo HAR y los tres modelos de la metodología GARCH-RNN haciendo uso de la función QL como medida de error de pronostico, los resultados obtenidos se muestran en la tabla 5.3

| | GJR GARCH VOL | GARCH VOL | HAR |
|---------------|---------------|-----------|----------|
| GJR VOL | 0.181914 | 0.014204 | 0.056566 |
| GJR GARCH VOL | | 0.185384 | 0.10293 |
| GARCH VOL | | | 0.057023 |

Cuadro 5.3: p-valores pruebas Diebold-Mariano

El test empleado muestra que a un 90 % de confianza existe una diferencia en el poder predictivo entre los modelos GJR VOL, GARCH VOL frente al modelo HAR, al igual que para los modelos GJR VOL y GARCH VOL existe una diferencia en el poder predictivo con un 95 % de confianza. Por otro lado no hay una diferencia estadística en el poder predictivo del GJR GARCH VOL con los modelos GARCH VOL, GJR VOL y HAR, el agregar el pronóstico del modelo GARCH al GJR VOL y el caso contrario no aporta un diferencial en el pronostico de la volatilidad.

Después de evaluar el desempeño de los modelos GARCH-RNN en el pronostico de la volatilidad del Ethereum se va a evaluar el comportamiento de los modelos híbridos en la medición del riesgo a través del *Value at Risk*, con nivel α de 1 %, 2.5 % y 5 %. En la tabla 5.4 se muestra el nivel de cobertura ($\hat{\alpha}$) para los tres modelos utilizados, en donde se puede observar que estos tuvieron el mismo desempeño para los tres niveles de confianza 1 %, 2.5 % y 5 %. La medida estimada al 1 % no tuvo ningún fallo en los 190 días de observación, pero se esperaba que fallara 2 de los 190 días, al 2.5 % se esperaba fallara 5 días pero solo se supero el *VaR* 2 días, por último al 5 % se supero el nivel del *VaR* 3 días, donde se esperaba fallara en 10 de los 190 días del periodo.





Figura 5.2: Estimación VaR al 1 % a un día en el periodo 2021-08-14 y 2022-02-20 para los modelos GARCH VOL, GJR VOL y HAR VOL

| | GJR_VOL | GARCH_VOL | HAR |
|-------|---------|-----------|-------|
| 1% | 0% | 0% | 0% |
| 2.50% | 1.05% | 1.05% | 1.05% |
| 5% | 1.58% | 1.58% | 1.58% |

Cuadro 5.4: Cobertura α de los modelos GARCH VOL, GJR VOL y HAR VOL a distintos niveles de confianza

5.2. GARCH-SVR

Como modelos de referencia se tiene el GARCH y GJR-GARCH estimados con el mismo conjunto de entrenamiento para los modelos híbridos, mediante máxima verosimilitud asumiendo una distribución t-student. Los coeficientes obtenidos se ilustran en las tablas 5.5, 5.6

| | Coefficientes | Error Estándar | \mathbf{t} | P-valor |
|----------|---------------|----------------|--------------|-----------|
| ω | 0.00026082 | 0.808 | 3.227 | 1.251e-03 |
| α | 0.1675 | 3.578e-02 | 4.682 | 2.844e-06 |
| β | 0.7726 | 4.123e-02 | 18.737 | 2.495e-78 |

Cuadro 5.5: Coeficientes modelo GARCH(1,1) estimado mediante máxima verosimilitud

| | Coeficientes | Error Estándar | \mathbf{t} | P-valor |
|----------|--------------|----------------|--------------|-----------|
| ω | 0.00026788 | 0.826 | 3.245 | 1.175e-03 |
| α | 0.1611 | 3.158e-02 | 5.103 | 3.348e-07 |
| γ | 0.0224 | 6.147 e- 02 | 0.364 | 0.716 |
| β | 0.7676 | 4.403e-02 | 17.434 | 4.532e-68 |

Cuadro 5.6: Coeficientes modelo GJR-GARCH(1,1,1) estimado mediante máxima verosimilitud

Para ambos modelos se tiene que $(\hat{\omega}, \hat{\alpha}, \hat{\beta})$ son significativos mientras que el coeficiente $\hat{\gamma}$ no es significativo en el GJR-GARCH(1,1), según esta estimación no se tendría un efecto de apalancamiento de la volatilidad ante retornos negativos. Ya con el modelo de referencia, se procede a la estimación de los modelos GARCH-SVR, GARCH-NuSVR, GJR-SVR y GJR-NuSVR, haciendo uso de dos kernel, el lineal y el RBF. Los coeficientes obtenidos mediante el modelo híbrido para el GARCH se encuentran en la tabla 5.7, en donde se puede evidenciar un menor valor en el coeficiente asociado a la volatilidad del día anterior, lo cual implica que una volatilidad elevada no afectaría las volatilidades futuras por un periodo de tiempo largo en comparación con el GARCH, de igual manera que un choque en la volatilidad no repercutirá tanto en las volatilidades futuras. Se observa un comportamiento similar en los coeficientes sobre el modelo hibrido para el GJR-GARCH mostrados en la tabla 5.8 para el kernel lineal, mientras con el kernel RBF los parametros α y β se acercan a los estimados por máxima verosimilitud. La estimación de γ mediante los modelos hibridos sugiere un efecto de apalancamiento simetrico en vez de asimétrico como sucede generalmente.

| | Linear | | | F | RBF |
|---|----------|-----------------------|------------|------------|-------------|
| | | GARCH SVR GARCH NuSVR | | GARCH SVR | GARCH NuSVR |
| _ | ω | 0.00034635 | 0.00081512 | 0.00055945 | 0.00070314 |
| | α | 0.05600269 | 0.053716 | 0.05358868 | 0.06442503 |
| | β | 0.44416473 | 0.39780168 | 0.47991627 | 0.41647662 |

Cuadro 5.7: Coeficientes modelo GARCH(1,1) estimado mediante los modelos híbridos

| | Li | inear | RBF | | |
|----------|---------------|-----------------|---------------|-----------------|--|
| | GJR-GARCH SVR | GJR-GARCH NuSVR | GJR-GARCH SVR | GJR-GARCH NuSVR | |
| ω | 0.0002105 | 0.00076809 | 0.0005433 | 0.00073656 | |
| α | 0.1116196 | 0.10235908 | 0.1588684 | 0.08531413 | |
| γ | -0.0255613 | -0.01764898 | -0.0708286 | -0.04464588 | |
| β | 0.4471667 | 0.41640055 | 0.7055297 | 0.62450426 | |
| a | | | | 1 11 14 11 | |

Cuadro 5.8: Coeficientes modelo GJR-GARCH(1,1) estimado mediante los modelos híbridos

Con los coeficientes estimados por cada uno de los ocho modelos se pronostica la volatilidad diaria de los retornos, estos pronosticos junto con el dato de volatilidad realizada y el pronostico de volatilidad de los modelos de referencia se pueden observar en la figura 5.3



Figura 5.3: Pronósticos volatilidad a un día generados a partir de los coeficientes estimados por SVR y NuSVR desde el 2021-08-11 al 2022-02-20

Se evidencia en las gráficas que los pronósticos de los modelos híbridos son menores a los pronósticos de los modelos econométricos de referencia, además que los pronósticos generados por el algoritmo NuSVR son menos fluctuantes que los demás, sumado a esto, los pronósticos de los modelos no se acercan al dato realizado cuando se presentan grandes picos en la volatilidad.

Los resultados de las funciones de perdida asociadas a cada serie de pronósticos de los modelos híbridos se tienen en la tabla 5.9, y los pronósticos del GARCH(1,1) y GJR-GARCH(1,1,1) cuentan con un error QL de 0.15974 y 0,15976 respectivamente y un error cuadrático medio de 0.00115 para ambos. En negrilla se tiene el mejor modelo según cada función de perdida, donde ambas coinciden en que el modelo híbrido con el menor error es el GARCH-NuSVR con un kernel lineal, mientras que el de mayor error es el GJR-SVR con kernel RBF. El modelos GARCH-NuSVR con un kernel lineal bajo el MSE genera mejores pronósticos que los modelos de referencia, a diferencia de la postura bajo la función QL donde los modelos de referencia tienen mejor desempeño predictivo.

Aplicando el test de Diebold y Mariano entre las 10 series de pronostico cuyos p-valores en términos porcentuales se encuentran en la tabla 5.11, se puede observar que al 5 % de significaría no existe diferencia en el poder predictivo de los modelos base. Los modelos GARCH linear NuSVR, GARCH RBF SVR, GARCH RBF NuSVR y GJR linear NuSVR no se rechaza que estos cuatro modelos frente a los modelos base tengan igual precisión predictiva. Los modelos híbridos GARCH presentan una diferencia en el poder predictivo entre si a excepción de la comparación de pronósticos para los modelos GARCH linear NuSVR y GARCH RBF NuSVR.

| | | QL | | QL | | MS | SE |
|------------|-------|---------------|---------|---------|---------|----|----|
| | | Linear | RBF | Linear | RBF | | |
| САРСИ | SVR | 0.23291 | 0.21147 | 0.00131 | 0.00127 | | |
| GANCH | NuSVR | 0.1603 | 0.16097 | 0.00114 | 0.00115 | | |
| | SVR | 0.26816 | 0.28611 | 0.00134 | 0.00136 | | |
| GJN-GAROII | NuSVR | 0.21607 | 0.23562 | 0.00127 | 0.00131 | | |

Cuadro 5.9: Funciones de perdida de los pronósticos fuera de la muestra de la volatilidad realizada

Con el fin de evaluar el comportamiento del segundo grupo de modelos híbridos se toman los modelo GARCH linear NuSVR y GARCH RBF SVR además del GARCH(1,1) calculando el Value at Risk, con nivel α de 1 %, 2.5 % y 5 %. En la tabla 5.10 se muestra el nivel de cobertura ($\hat{\alpha}$) en el cual se evidencia que al 1 % bajo el GARCH RBF SVR solo en un día se supero el VaR mientras en los otros dos en los 194 días del periodo nunca se supero, con este nivel se esperaba que superara la medida 2 días, al 2.5 % y 5 % cada uno de los 3 modelos tuvo una medida de cobertura diferente, con un 2.5 % de significancia se supero la medida de riesgo 1 día, 3 días y 2 días para el modelo GARCH, GARCH RBF SVR y GARCH linear NuSVR respectivamente cuando se esperaba fuese superada en 5 ocasiones, por ultimo a un 5 % se supero en 2 días, 5 días y 4 días cuando se esperaba que fueran 10 días. En la figura 5.4 se encuentra el Value at Risk al 1 % proveniente de cada



((c)) VaR 1% GARCH LINEAR NuSVR

Figura 5.4: Estimación VaR al 1 % a un día en el periodo 2021-08-14 y 2022-02-20 para los modelos GARCH, GARCH RBF SVR y GARCH LINEAR NuSVR

modelo comparado con los retornos diarios del Ethereum, observando que únicamente se supera la medida dada por el GARCH RBF SVR la cual fue de -0.148422 mientras el retorno para ese dia fue de -0.161438

| | GARCH | SVR | NuSVR |
|-------|---------|---------|---------|
| 1% | 0% | 0.5155% | 0% |
| 2.50% | 0.5155% | 1.5464% | 1.0309% |
| 5% | 1.0309% | 2.5773% | 2.0619% |

Cuadro 5.10: Cobertura α de los modelos GARCH, GARCH RBF SVR y GARCH LINEAR NuSVR a distintos niveles de confianza

| | GARCH linear NuSVR | ${\rm GARCH}\; {\rm rbf}\; {\rm SVR}$ | GARCH rbf NuSVR | GJR linear SVR | GJR linear NuSVR | $GJR \ rbf SVR$ | GJR rbf $NuSVR$ | GARCH | GJR |
|--------------------|--------------------|---------------------------------------|-----------------|----------------|------------------|-----------------|-----------------|--------|--------|
| GARCH linear SVR | 0.038 | 2.313 | 0.011 | 0 | 1.298 | 0.14 | 65.411 | 3.815 | 3.932 |
| GARCH linear NuSVR | | 0.284 | 72.882 | 0.001 | 0.223 | 0.006 | 0.062 | 97.098 | 97.261 |
| GARCH rbf SVR | | | 0.137 | 0.06 | 68.35 | 0.143 | 4.214 | 10.739 | 10.996 |
| GARCH rbf NuSVR | | | | 0 | 0.083 | 0.003 | 0.023 | 94.281 | 94.461 |
| GJR linear SVR | | | | | 0 | 23.574 | 0.106 | 0.399 | 0.419 |
| GJR linear NuSVR | | | | | | 0 | 0 | 8.573 | 8.755 |
| GJR rbf SVR | | | | | | | 0.002 | 0.461 | 0.475 |
| GJR rbf NuSVR | | | | | | | | 3.867 | 3.97 |
| GARCH | | | | | | | | | 95.739 |
| | | | | | | | | | |

Cuadro 5.11: P-valores en términos porcentuales para la prueba de Diebold-Mariano

6. Conclusión

En el presente trabajo se implementaron dos clases de modelos híbridos con el objetivo de evaluar su desempeño en el pronostico de la volatilidad del Ethereum y su efectividad en la gestión del riesgo frente a modelos clásicos. El primer tipo de modelo emplea estructuras de redes neuronales recurrentes, de tipo simple y GRU, teniendo como entradas los pronósticos del GARCH y GJR-GARCH en adición de otras variables. El segundo tipo de modelo utiliza los algoritmos de SVM y NuSVM en el contexto de regresión para la estimación de la ecuación de varianza de los modelos GARCH(1,1) y GJR-GARCH(1,1,1) haciendo uso de diferentes kernels.

Dentro de la primera clase de modelos, el de mejor desempeño es el GJR VOL que tiene como entrada los pronósticos del GJR y la volatilidad realizada seguido del GARCH VOL que tiene como entrada los pronósticos del GARCH y la volatilidad realizada con un error QL de 0.01094 y 0.01095 respectivamente frente a un 0.0738 del HARV, estos modelos generaron pronósticos estadísticamente diferentes de acuerdo con la prueba de Diebold-Mariano. En cuanto a la eficiencia en la medición del riesgo los diferentes modelos no presentaron diferencia en la medida de cobertura siendo esta alejada del nivel de significancia establecido.

Por ultimo en la segunda clase de modelos, el mejor modelo hibrido acorde a las dos funciones de perdida es el NuSVR-GARCH utilizando el kernel lineal, sin embargo este solo supera al GARCH y GJR-GARCH clásicos bajo el error cuadrático medio mientras tomando como medida de comparación el error QL los modelos clásicos superan al mejor modelo híbrido de esta clase, por otro lado en la medición del riesgo el modelo mas eficiente fue el GARCH SVR con el kernel RBF obteniendo una medida de cobertura mas cercana al nivel de significancia dado en comparación con el GARCH clásico y el GARCH NuSVR.

Referencias

- Andreev, A. & Kanto, A. (2004). Conditional value-at-risk estimation using non-integer values of degrees of freedom in Student's t-distribution. The Journal of Risk, 7(2), 1.
- Angela, O. & Sun, Y. (2020). Factors affecting cryptocurrency prices: Evidence from ethereum. 2020 International Conference on Information Management and Technology (ICIMTech), 318-323.
- Arnerić, J., Poklepović, T. & Aljinović, Z. (2014). GARCH based artificial neural networks in forecasting conditional variance of stock returns. *Croatian Operational Research Review*, 329-343.
- Bandgar, S. (2021). Support Vector Machine. https://medium.com/nerd-for-tech/support-vector-machine-92fa3c57d33b
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of econometrics, 31(3), 307-327.
- Bollerslev, T. & Wooldridge, J. M. (1992). Quasi-maximum likelihood estimation and inference in dynamic models with time-varying covariances. *Econometric reviews*, 11(2), 143-172.
- Bucci, A. (2020). Realized volatility forecasting with neural networks. Journal of Financial Econometrics, 18(3), 502-531.
- Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.
- Chung, S. S. & Zhang, S. (2017). Volatility estimation using support vector machine: Applications to major foreign exchange rates. *Electronic Journal of Applied Statistical Analysis*, 10(2), 499-511.
- Corsi, F. (2009). A simple approximate long-memory model of realized volatility. Journal of Financial Econometrics, 7(2), 174-196.
- Dempere, J. M. (2019). Factors Affecting the Return and Volatility of Major Cryptocurrencies. 2019 Sixth HCT Information Technology Trends (ITT), 104-109.
- Dixon, M. F., Halperin, I. & Bilokon, P. (2020). Machine learning in Finance (Vol. 1170). Springer.
- Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica: Journal of the econometric society*, 987-1007.
- Freeborough, W. & van Zyl, T. (2022). Investigating Explainability Methods in Recurrent Neural Network Architectures for Financial Time Series Data. Applied Sciences, 12(3), 1427.
- Glosten, L. R., Jagannathan, R. & Runkle, D. E. (1993). On the relation between the expected value and the volatility of the nominal excess return on stocks. *The journal of finance*, 48(5), 1779-1801.
- Hajizadeh, E., Seifi, A., Zarandi, M. F. & Turksen, I. (2012). A hybrid modeling approach for forecasting the volatility of S&P 500 index return. *Expert Systems with Applications*, 39(1), 431-436.

- Hastie, T., Tibshirani, R., Friedman, J. H. & Friedman, J. H. (2009). The elements of statistical learning: data mining, inference, and prediction (Vol. 2). Springer.
- Kim, H. Y. & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 103, 25-37.
- Kostadinov, S. (2017). Understanding GRU Networks. https://towardsdatascience.com/understandinggru-networks-2ef37df6c9be
- Kristjanpoller, W., Fadic, A. & Minutolo, M. C. (2014). Volatility forecast using hybrid neural network models. *Expert Systems with Applications*, 41(5), 2437-2442.
- Kristjanpoller, W. & Minutolo, M. C. (2016). Forecasting volatility of oil price using an artificial neural network-GARCH model. *Expert Systems with Applications*, 65, 233-241.
- Mademlis, D. K. & Dritsakis, N. (2021). Volatility Forecasting using Hybrid GARCH Neural Network Models: The Case of the Italian Stock Market. International Journal of Economics and Financial Issues, 11(1), 49.
- Patton, A. J. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1), 246-256.
- Pérez-Cruz, F., Afonso-Rodriguez, J. A. & Giner, J. (2003). Estimating GARCH models using support vector machines. *Quantitative Finance*, 3(3), 163.
- Seo, M., Lee, S. & Kim, G. (2019). Forecasting the volatility of stock market index using the hybrid models with google domestic trends. *Fluctuation and Noise Letters*, 18(01), 1950006.
- Sezer, O. B., Gudelek, M. U. & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, 90, 106181.
- Shen, Z., Wan, Q. & Leatham, D. J. (2021). Bitcoin Return Volatility Forecasting: A Comparative Study between GARCH and RNN. Journal of Risk and Financial Management, 14(7), 337.
- Sheppard, K. (2010). Financial econometrics notes. University of Oxford, 333-426.
- Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
- Xiong, R., Nichols, E. P. & Shen, Y. (2015). Deep learning stock volatility with google domestic trends. arXiv preprint arXiv:1512.04916.
- Yan, H. & Ouyang, H. (2018). Financial time series prediction based on deep learning. Wireless Personal Communications, 102(2), 683-700.